Grant Agreement No.: 611001

UCN

Instrument: Collaborative Project
Call Identifier: FP7-ICT-2013-10
Objective 1.6: Connected and Social Media

D2.3: Final Report on Data Collection

Due date of deliverable: 30.06.2016
Actual submission date: 30.06.2016

Start date of project: October 1st 2013 Duration: 36 months
Project Manager: Henk Heijnen, Technicolor SA
Author(s): Inria, Intamac, NICTA, Altice Labs (former PTIN), Technicolor, University of Cambridge, University of Nottingham
Editor: Inria
Revision: v.1.0 – Final version

Abstract

WP2 focuses on methods to obtain user context (i.e., knowledge of users’ environment, activities, preferences). All this information will then feed the Personal Information Hub (PIH). This deliverable describes the final metrics, methodologies and tools for user data collection. It will also report on results from the small-scale user studies performed in France and in the UK.

<table>
<thead>
<tr>
<th>Dissemination Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
</tr>
<tr>
<td>PP</td>
</tr>
<tr>
<td>RE</td>
</tr>
<tr>
<td>CO</td>
</tr>
</tbody>
</table>

Project co-funded by the European Commission in the 7th Framework Programme (2007-2013)
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>1</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>2 USER CONTEXT METRICS</td>
<td>3</td>
</tr>
<tr>
<td>2.1 End-User Devices</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Sensors</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Home Gateways</td>
<td>6</td>
</tr>
<tr>
<td>2.4 Service Back Ends</td>
<td>7</td>
</tr>
<tr>
<td>3 FINAL DATA COLLECTION METHODS</td>
<td>9</td>
</tr>
<tr>
<td>3.1 End-User Devices</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Sensors</td>
<td>16</td>
</tr>
<tr>
<td>3.3 Home Gateways</td>
<td>19</td>
</tr>
<tr>
<td>3.4 Service Back Ends</td>
<td>22</td>
</tr>
<tr>
<td>4 RESULTS OF THE USER STUDY</td>
<td>44</td>
</tr>
<tr>
<td>4.1 Study Setup</td>
<td>44</td>
</tr>
<tr>
<td>4.2 Technical Data Sets Summary</td>
<td>48</td>
</tr>
<tr>
<td>4.3 Ethnographic Results</td>
<td>51</td>
</tr>
<tr>
<td>4.4 Conclusions</td>
<td>82</td>
</tr>
<tr>
<td>5 CONCLUSION</td>
<td>84</td>
</tr>
<tr>
<td>6 REFERENCES</td>
<td>85</td>
</tr>
<tr>
<td>7 APPENDIX I: TOOLS</td>
<td>86</td>
</tr>
<tr>
<td>8 APPENDIX II: NICTA’S MSPEED PAPER</td>
<td>87</td>
</tr>
<tr>
<td>9 APPENDIX III: HOA PAM 2016</td>
<td>94</td>
</tr>
<tr>
<td>10 APPENDIX IV: TMA 2016</td>
<td>107</td>
</tr>
<tr>
<td>11 APPENDIX V: NEW HOA IMC SUBMISSION</td>
<td>116</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

The UCN project leverages rich knowledge about users to help them find relevant content, identify nearby network resources, and plan how to deliver the actual content to the appropriate device at the desired time. A general challenge for such methods is how to obtain this broader understanding of the user. This includes better knowledge about user’s activities, needs, and interests as well as knowledge about the user local environment. We call this information collectively user context. WP2 has designed and implemented a set of data collectors to automatically obtain user context from various sources. It has also designed mechanisms to collect direct feedback and annotations from the user, combining this data with automatically collected user context. Furthermore, it is important to relate the user context metrics and data collection to real world situations and real world practices in order to assess their effectiveness and the degree to which they may require refinement or augmentation if they are to support real use. For this reason, WP2 undertook two small-scale studies in France and in the UK with real users to observe and gather data about their existing online practices and media consumption and the ways in which concerns about privacy might come into play.

In deliverable D2.1 we described the preliminary set of metrics UCN uses to capture user context. In deliverable D2.2 we reviewed and updated these metrics, and presented the design and implementation of the initial data collection methods and tools. The developed methods cover four different vantage points that together can provide a comprehensive view of the user: end-hosts, both PCs and mobile devices; home sensors; home gateways; and service back-ends. In this deliverable we present the final metrics (Section 2) and detailed descriptions of the final methods and tools (Section 3) for user data collection. We report on updated (with respect to the previous deliverables) and new methodologies for each of the four vantage points including a new mobile network performance measurement tool, called mSpeed, and updates to the sensor, gateway and back end data collection tools and methodologies. mSpeed adds to the set of tools that UCN has built to assist end-users in understanding and diagnosing their Internet performance. This deliverable also reports on the diagnosis tools that run directly on the home gateway to assist users in localizing performance bottlenecks in their home networks.

In addition to describing the data collection methods, we discuss in Section 4 results of the two user studies. In deliverable D2.2, we described in detail the technical and ethnographic data collection process, the ways in which data would be visualized to users so that we could obtain feedback about their activities and the participant recruitment strategies. In this deliverable we will focus on presenting the findings of the studies and will only briefly overview the data collection process and methodology.

Appendix I provides a summary of all the tools and data sets that we have developed/collected in WP2 during the project with links to download source code, binaries or other artefacts when publicly available.
2 USER CONTEXT METRICS

In this section we summarize the user context metrics including new/updated metrics with respect to the initial discussion provided in deliverables D2.1 and D2.2.

2.1 End-User Devices

We have implemented data collectors both for PCs and mobile devices. Table 1 summarizes the end-user device metrics available via these collectors. We collect data on user activity using available sensors, I/O devices and a list of running applications. We record user location at varying granularity based on the sensors and APIs available at a particular vantage point. We collect data on the end system to identify the device type (OS, version, brand), its available and currently used resources for performance characterisation. The network data collection consists of configuration tracking, and passive and active performance measurements. Finally, we measure various performance metrics at a per application level to obtain a more detailed view on user experience with networked applications.

Updates to D2.1: None of the current use cases/applications require mobile devices’ acceleration and rotation state or the exact GPS coordinates, so we have marked them as removed for now. Inria may add these to a mobile version of the Hostview tool if necessary. In addition, we have discontinued the Android based home network monitoring device development (more details in 3.3.4), and we are not currently working on active network probes.

Updates to D2.2: Added back network performance metrics collected by a new measurement tool to evaluate CDN performance in mobile networks (more details in Section 3.1).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Vantage Point</th>
<th>Tool/Partner</th>
</tr>
</thead>
<tbody>
<tr>
<td>User activity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device acceleration and rotation</td>
<td>Mobile</td>
<td>Hostview/Inria, UCN Study/Inria,Nott</td>
</tr>
<tr>
<td>(gyroscope, accelerometer)</td>
<td>Mobile, PC</td>
<td></td>
</tr>
<tr>
<td>Screen state (on/off, full-screen)</td>
<td>Mobile, PC</td>
<td>Hostview/Inria</td>
</tr>
<tr>
<td>Microphone and speaker usage</td>
<td>Mobile, PC</td>
<td></td>
</tr>
<tr>
<td>(on/off)</td>
<td></td>
<td>Hostview/Inria</td>
</tr>
<tr>
<td>Mouse/keyboard/touch I/O events</td>
<td>Mobile, PC</td>
<td>Hostview/Inria</td>
</tr>
<tr>
<td>Foreground application</td>
<td>Mobile, PC</td>
<td>Hostview/Inria, UCN Study/Inria,Nott</td>
</tr>
<tr>
<td>List of running applications</td>
<td>Mobile, PC</td>
<td>Hostview/Inria, UCN Study/Inria,Nott</td>
</tr>
<tr>
<td>Location</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordinates (GPS)</td>
<td>Mobile</td>
<td>Hostview/Inria</td>
</tr>
<tr>
<td>Connected and neighbouring cell ids</td>
<td>Mobile</td>
<td>Hostview/Inria</td>
</tr>
<tr>
<td>System Components</td>
<td>Source</td>
<td>Notes</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Public IP based geo-location (ISP, city, country)</td>
<td>Mobile, PC</td>
<td>Hostview/Inria</td>
</tr>
<tr>
<td>User-identified location</td>
<td>Mobile, PC</td>
<td>Hostview/Inria, UCN Study/Inria,Nott</td>
</tr>
<tr>
<td><strong>System</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS, version, brand</td>
<td>Mobile, PC</td>
<td>Hostview/Inria, UCN Study/Inria,Nott</td>
</tr>
<tr>
<td>Available CPUs, memory, battery</td>
<td>Mobile, PC</td>
<td>Hostview/Inria, UCN Study/Inria,Nott</td>
</tr>
<tr>
<td>CPU, memory, battery usage statistics</td>
<td>Mobile, PC</td>
<td>Hostview/Inria, UCN Study/Inria,Nott</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Associated cell (apn, cid, lac, operator, country)</td>
<td>Mobile</td>
<td>Hostview/Inria, UCN Study/Inria,Nott</td>
</tr>
<tr>
<td>Associated WiFi AP (bssid, ssid, channel, freq, phy rate, security, IP, mode, RSSI)</td>
<td>Mobile, PC</td>
<td>Hostview/Inria, UCN Study/Inria,Nott</td>
</tr>
<tr>
<td>Other active network(s) (Ethernet, Bluetooth, tethering, VPN)</td>
<td>Mobile, PC</td>
<td>Hostview/Inria UCN Study/Inria,Nott</td>
</tr>
<tr>
<td>Network interface statistics</td>
<td>Mobile, PC</td>
<td>Hostview/Inria UCN Study/Inria,Nott</td>
</tr>
<tr>
<td>Latency to home network devices</td>
<td>Mobile, PC</td>
<td>-</td>
</tr>
<tr>
<td>Latency (ICMP ping) to landmarks [new]</td>
<td>Mobile</td>
<td>mSpeed/Nicta</td>
</tr>
<tr>
<td>Latency to DNS server [new]</td>
<td>Mobile</td>
<td>mSpeed/Nicta</td>
</tr>
<tr>
<td>DNS lookup time (curl) [new]</td>
<td>Mobile</td>
<td>mSpeed/Nicta</td>
</tr>
<tr>
<td>TCP connection setup time (curl) [new]</td>
<td>Mobile</td>
<td>mSpeed/Nicta</td>
</tr>
<tr>
<td>Traceroute to landmarks [update]</td>
<td>Mobile, PC</td>
<td>mSpeed/Nicta</td>
</tr>
<tr>
<td>Web performance; bulk transfer (curl) to landmarks [update]</td>
<td>Mobile, PC</td>
<td>mSpeed/Nicta</td>
</tr>
<tr>
<td>Access and home network bandwidth</td>
<td>Mobile, PC</td>
<td>-</td>
</tr>
<tr>
<td><strong>Application performance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU, memory and network usage per application</td>
<td>Mobile, PC</td>
<td>Hostview/Inria</td>
</tr>
<tr>
<td>TCP flows (RTT, jitter, rx/tx bytes, packets, retransmissions, resets, start and end times, source and destination IPs and ports)</td>
<td>Mobile, PC</td>
<td>Hostview/Inria</td>
</tr>
</tbody>
</table>
Non-TCP traffic (rx/tx bytes and packets, “connection” start time and end time, source and destination IPs and ports) | Mobile, PC | Hostview/Inria

*Table 1: End-user device data and metrics*

### 2.2 Sensors

We leverage a multitude of sensors at home to gather contextual data about household routine(s), person presence and household environment. Table 2 lists the metrics collected from various sensors at home.

**Updates to D2.1:** Added new sensor metrics collected by the PT SmartSense platform.

**Updates to D2.2:** The focus of Intamac is directed more heavily at sensors, which contribute to an understanding of presence or lack thereof of individuals within a property. Sensors that do not support this focus are therefore removed from the list of ENSO/Intamac sensors.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Vantage Point</th>
<th>Tool/Partner</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User activity</strong></td>
<td>Movement detection (person, location)</td>
<td>PIR and Camera</td>
</tr>
<tr>
<td></td>
<td>Arrival and Departure</td>
<td>Key fob, Panic Pendant</td>
</tr>
<tr>
<td></td>
<td>Door/window state (open/closed)</td>
<td>Door / Window Contact</td>
</tr>
<tr>
<td><strong>System Status</strong></td>
<td>Lighting choice</td>
<td>Light Control</td>
</tr>
<tr>
<td></td>
<td>Heating/Cooling Mode</td>
<td>HVAC System</td>
</tr>
<tr>
<td></td>
<td>Alarm State – Home/Away/Night</td>
<td>Alarm System</td>
</tr>
<tr>
<td></td>
<td>Doors locked/unlocked</td>
<td>Door Locks</td>
</tr>
<tr>
<td><strong>Power consumption</strong></td>
<td>Smart plug</td>
<td>ENSO/Intamac</td>
</tr>
<tr>
<td><strong>Biometrics</strong></td>
<td>Heart rate (BPM)</td>
<td>Personal Biometrics</td>
</tr>
<tr>
<td></td>
<td>Weight</td>
<td>Connected Scales</td>
</tr>
<tr>
<td></td>
<td>Activity (Sleeping, Active, Resting)</td>
<td>Personal Movement Sensor / Pendant</td>
</tr>
<tr>
<td></td>
<td>Fall Alert</td>
<td>Fall Sensor</td>
</tr>
<tr>
<td></td>
<td>Temperature</td>
<td>Personal Biometrics</td>
</tr>
<tr>
<td></td>
<td>Blood Pressure (Systolic and Diastolic)</td>
<td>Blood Monitor</td>
</tr>
<tr>
<td></td>
<td>Glucose</td>
<td>Glucose Monitor</td>
</tr>
<tr>
<td></td>
<td>Oxygen saturation</td>
<td>Oximeter</td>
</tr>
<tr>
<td></td>
<td>EMG</td>
<td>Health Monitor Device</td>
</tr>
<tr>
<td></td>
<td>ECG</td>
<td>Health Monitor Device</td>
</tr>
<tr>
<td></td>
<td>Foot Temperature</td>
<td>Health Monitor Device</td>
</tr>
</tbody>
</table>
### 2.3 Home Gateways

Table 3 summarizes the gateway metrics. We are collecting data on network configuration and performance, user activity (connections and content consumption) and detailed application level performance via network traffic monitoring.

**Updates to D2.1:** Updated metrics of the Technicolor PT gateway.

**Updates to D2.2:** Added new wireless network and network performance metrics required by the improved home gateway based bottleneck detection algorithm (details in Section 3.3).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Tool/Partner</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong></td>
<td></td>
</tr>
<tr>
<td>Gateway hardware (model, type, serial no)</td>
<td>PT Gateway/Technicolor</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td></td>
</tr>
<tr>
<td>Connected devices (IP, MAC address, DNS name, WiFi capability and configuration, available services via UPnP and mDNS)</td>
<td>PT Gateway/Technicolor, HomeWork/Nott</td>
</tr>
<tr>
<td>Wireless network (RSSI, physical rate, throughput of connected hosts, nearby APs, channel busy time [new], frame retransmission rate [new])</td>
<td>PT Gateway/Technicolor, HomeWork/Nott, OpenWRT/Inria</td>
</tr>
<tr>
<td>Internet connection (PPP connection and DSL line statistics, e.g. status, SNR for DSL)</td>
<td>PT Gateway/Technicolor</td>
</tr>
<tr>
<td>Network interfaces configuration and statistics (e.g. packets/bytes recv/sent, errors)</td>
<td>PT Gateway/Technicolor</td>
</tr>
<tr>
<td>IP level configuration and statistics (e.g. packets/bytes recv/sent, errors)</td>
<td>PT Gateway/Technicolor</td>
</tr>
<tr>
<td>DNS statistics</td>
<td>PT Gateway/Technicolor</td>
</tr>
<tr>
<td>Access link delay (ICMP ping to the first IP hop) [new]</td>
<td>OpenWRT/Inria</td>
</tr>
<tr>
<td><strong>User activity</strong></td>
<td></td>
</tr>
<tr>
<td>Device (dis)connect (DHCP reqs, acks, offers, state)</td>
<td>PT Gateway/Technicolor, HomeWork/Nott, OpenWRT/Inria</td>
</tr>
</tbody>
</table>
Table 3: Home gateway data and metrics

<table>
<thead>
<tr>
<th><strong>Metric</strong></th>
<th><strong>Vantage Point</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Activity</strong></td>
<td>Channel, Duration, Program StartTime, StartTime, Station ShortName, Title, Episode Title, PT IPTV service backend</td>
</tr>
<tr>
<td>Description, ProgramId, IsHd, Program Metadata Long Description, Program Metadata Title, Program Schedule Duration, Program Schedule Start Time, Program Schedule Station Name, Zapping</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>LiveTV_Channel_EPG_Tuned, LiveTV_Channel_Search_Tuned, LiveTV_Channel_Views_Duration, Facebook_Shared, Twitter_Shared, LiveTV_WatchOnTV_Selected, VOD_Movie_AddedToFavorites, VOD_RentMovie, VOD_Movie_Played, EPG_Alert_Program, EPG_Recording_Alert_Program, Recording_Record_Program, Recording_EPG_Program, GA_Play_Episode, GA_Program_Views_Duration</td>
<td></td>
</tr>
<tr>
<td>PT IPTV mobile companion app (MeoGo)</td>
<td></td>
</tr>
<tr>
<td>TV program data: program title, description, duration, channel name, watch_start_time, watch_stop_time, rating (out of 5 stars), comments; Facebook posts related to this program; Tweets related to this program; multi-user chat room history</td>
<td></td>
</tr>
<tr>
<td>NICTA social TV service backend server</td>
<td></td>
</tr>
<tr>
<td>Friends, likes, geo-location, wall posts, email address</td>
<td></td>
</tr>
<tr>
<td>Facebook data of PT IPTV users</td>
<td></td>
</tr>
<tr>
<td>Following/followers, tweets (ID, timestamp, user ID, contents), geo-location</td>
<td></td>
</tr>
<tr>
<td>Twitter data for PT IPTV users and Stanford Twitter dataset (snap.standford.edu/data)</td>
<td></td>
</tr>
</tbody>
</table>

*Table 4: Service backend data and metrics*
3 FINAL DATA COLLECTION METHODS

In this section we report on the final data collection methods and tools designed and implemented by the consortium. Completely new or updated content (with respect to deliverables D2.1 and D2.2) is indicated in the subsection headers. Appendix I provides a summary of all the developed tools with links to more information/downloads when available.

3.1 End-User Devices
3.1.1 End-host Monitoring with HostView

HostView is a tool that runs on end-hosts to collect network performance data annotated with the user perception of network performance. It adopts a hybrid measurement methodology that combines network measurement techniques to infer application performance with techniques from HCI to measure user perception. In our previous work, we developed an initial version of HostView [1] that ran on Linux and OS X to collect network packet traces and application process executable names, which we can use to infer network performance per application, as well as a number of end-host performance metrics such as WiFi signal strength/noise and machine CPU load and memory utilization. HostView also recorded user feedback on application performance with two mechanisms: a system-triggered questionnaire based upon the Experience Sampling Methodology (ESM), and a user-triggered mechanism called an “I’m annoyed!” button. The ESM mechanism prompted the user no more than three times per day with a questionnaire about their experience with online applications. The “I’m annoyed!” button was always displayed at the corner of the screen and we asked users to click on it when they are dissatisfied with their application performance.

Our experience using the data collected with HostView highlighted a number of shortcomings with the first version of HostView that we have addressed in a new version developed in the context of the UCN project. It should be noted that the initial version of the tool was a grad student project, so we have re-implemented HostView from scratch with a number of improvements to better serve the needs of UCN. The new version still collects network packet traces and the mapping of network flows to application process names information on network environment Here we describe the main changes from the original implementation of HostView.

**Improved identification of services running on web browser.** HostView identifies online activity by mapping network flows to the corresponding application process name. One issue is that many applications today run on top of a web browser and this method cannot distinguish flows from each of the services running within the browser. Thus, we needed to improve HostView to identify network traffic for the individual services running on top of the browser. The new version of HostView includes plugins for most popular browsers (Firefox, Chrome, and Internet Explorer) to address this issue.

**Support for missing OSes.** HostView supported only Linux and OS X laptops and desktops, but a large fraction of users access online services through handheld devices (e.g., smartphones or tablets) or via Windows systems. The new version of HostView is targeted initially to these missing platforms and the prototype is developed for Windows.
Improved identification of network environment. HostView captures the network environment (i.e., which network the user is connected to) with the BSSID/ESSID if the interface is wireless, the MAC address of the first network device if the interface is wired. It also recorded the country, city, and ISP to which the user was connected. This information, however, was obtained at the collection server by mapping the public IP of the host at the upload time. If the upload was done from a different environment than the environment where the traces were collected, then HostView would mislabel the traces. The new version of HostView gets the labelling of the environment at collection time.

Improved user activity tracking. HostView captures the list of running applications and the application in the foreground, which is a limited notion of user activity. The new version of HostView tracks I/O activity more closely (mouse, keyboard, microphone, camera) as well as whether the screen is in the screensaver or full screen mode. This allows us to track user presence and activity more closely.

Improved user feedback. We have re-designed the questionnaires based on the results of the UCN user study discussed in Section 4. The questionnaire now has three simple questions (as illustrated in Figures 1-3). The first asks about which activities the user is conducting with each application. The second asks the user to rate the overall performance of the computer (using a sliding button to select from poor to excellent). The third asks users to list problems experienced with any of the applications they are using. We have made the ESM less intrusive. Instead of popping up the questionnaire in the middle of the screen, HostView presents a notification in the taskbar to signal that the tool would like feedback.

![Figure 1: HostView first question.](image-url)
Figure 2: HostView second question.

Figure 3: HostView third question.

The new HostView was deployed in the two small-scale user studies (details in Section 4). Inria plans to deploy HostView in a larger-scale user study in the last months of UCN and exploit the tools and the collected data for future research on Internet quality of experience. For the larger-scale user study we have continued to develop the end-user data visualizations that we
prototyped and used as a part of the small-scale study to help the ethnographic interviews and data collection. The participants of the larger-scale study will have access to a website where they can browse their own HostView data to learn more about their computing activities, daily patterns and applications’ performance.

### 3.1.2 Mobile Device Monitoring

In addition to HostView, Inria and the University of Nottingham have together experimented with mobile device monitoring and data collection in the context of the UCN user study (see Section 4 for details). In the study, we collected data from mobile devices using an existing user activity tracking application called *Moves*[^1] and a custom activity logger developed in the UCN project.

*Moves* is a popular “user activity diary” application for iPhone and Android. It runs constantly in the background and automatically records walking, cycling, and running activities associated with time of day and user location. It lets users view the distance, duration, steps, and calories burned for each activity. *Moves* has an open API for 3rd party connected applications that can access the user activity data subject to authorization by the user. We leverage the *Moves* API in the UCN study to access data about user locations over time (i.e. places visited by the user instead of fine-grained geo-location data). The places are either automatically labelled (e.g. by address or based on known landmarks) or manually named by the user (e.g. home or work).

We have also developed a lightweight user activity logging application for Android[^2] to get more information about the mobile device’s system configuration, resource use and user activities. Similar to *Moves*, the application runs in the background, and periodically records the current network configuration; battery, screen, and audio state; system wide CPU and memory use; network traffic counters (per interface and application); a list of running applications and the foreground application; and the mapping of sockets to applications for traffic identification. In this initial design for the UCN study, the activity logger uploads JSON formatted data periodically to our secure servers where the objects are stored in a key-value database.

### 3.1.3 CDN Performances in Mobile Networks [New]

Content Distribution Networks (CDNs) play an integral role in the Internet, helping service providers improve the quality of experience to end-users regardless of traffic levels. The main idea of CDNs is to offload origin servers by caching content closer to the requesting client. In this context, closer can mean lower network latency, higher bandwidth, or both. We have built and deployed a measurement tool to characterise CDNs from the point of view of cellular network users. The tool, mSpeed, is available on iTunes app-store, and we have used this platform to experimentally evaluate the performance of two commercial CDN services, namely Akamai and Limelight. mSpeed adds to the set of tools that UCN has built to assist end-users in understanding and diagnosing their Internet performance. Below we discuss the

[^1]: https://www.moves-app.com/
[^2]: We prototyped a similar software for iOS but due to the limited APIs for data collection and complicated installation procedure for non-AppStore applications, we decided to stop working on the iOS tool.
measurement tool design and some initial results from the first deployment. Full details can be found in the technical report in Appendix II.

3.1.3.1 Measurement Platform: mSpeed

mSpeed is a crowdsourcing-based network measurement platform targeting both cellular and WiFi network measurements (http://www.mspeedapp.com/). Currently, an mSpeed application is available for iOS devices. The application offers end-user instantaneous upload/download speed and latency measurements to a set of closest landmark servers. The application also includes an operator comparison feature, where users can visualise and compare cellular network operators’ performance in their country and at specific locations. In addition to these set of pre-defined measurements, mSpeed can perform a range of scriptable network measurements, by downloading an experiment script every time it is run. Results are collected at a central server for offline analysis.

Figure 4 illustrates interactions between an mSpeed client and the server. The client first connects to the server and fetches a measurement script (which we call mScript) using HTTP (Step 1). The client parses the mScript to obtain a set of measurements, the order in which the measurements are to be performed, and the input parameters for each measurement. Note that this first step also ensures that the device's radio is in an active state before measurements are initiated. The mSpeed client performs each measurement in sequence, collecting results in JSON format (Step 2), and displaying progress and instantaneous readings to the end-user. The client then uploads aggregated results to the server (Step 3), including the GPS coordinates and a timestamp. The test results are then shown to the user, including network upload/download speed, and latency.

![Figure 4: The mSpeed platform interaction.](image-url)
3.1.3.2 Measurement Scripts: mScripts

The mSpeed measurement scripts (mScripts) are JSON text files stored on the backend, describing the measurements to be performed by the client (i.e., primitives with input parameters). Figure 5 illustrates an example mScript. display_name is a string displayed in the app while the measurement is executed, and command is a key to refer to the measurement primitive code in the client. parameters provides a set of input parameters depending on the measurement. The experimenter can design mScripts by combining a set of measurement primitives in a specific order to conduct a measurement campaign of interest.

The first three measurements shown in this example, are Download Speed, Upload Speed, and Latency, and must always be present in any mScript, as they are the user-facing measurements, used to show users their network performance and for the operator comparison feature. In this example, the additional measurement "IPv6TCPConnect" is for research purpose, and allows to test IPV6 connectivity. Currently available measurement primitives are described below.

```
    "measurements": {
      "downlink_bw": {  
        "display_name": "Download Speed",  
        "command": "bwget",  
        "params": {  
          "bwget_socket_count": 8,  
          "bwget_request_per_thread": 10,  
          "bwget_url": "http://d3rehuc764a80.cloudfront.net/1k.dat",  
          "bwget_max_duration": 7 } },  
      "uplink_bw": {  
        "display_name": "Upload Speed",  
        "command": "upload",  
        "params": {  
          "max_duration": 7,  
          "bytecount": 5000000,  
          "url": ["http://closest.mspeedapp.com/upload.php"] } },  
      "rtt": {  
        "display_name": "Latency",  
        "command": "ping",  
        "params": {  
          "pkt_count": 10,  
          "pkt_gap": 0.1,  
          "pkt_size": 64,  
          "hosts": ["www.mspeedapp.com"] } },  
      "IPv6TCPConnect": {  
        "display_name": "IPv6 connectivity",  
        "command": "IPv6Check",  
        "params": [{  
          "target": "ipv6.google.com",  
          "connect_timeout": 4,  
          "port": 80 } ] }  
    }   
```

Figure 5: Example mScript.

The benefit of our scriptable measurement platform is we can control parameters for a specific set of experiment easily without upgrading the application on the client device. Different scripts can be served for clients based on the cellular operator, country, and network type. For instance, we can have different parameters for conducting measurement on wifi and cellular or foreground and background. On the iOS platform, applications cannot perform scheduled tasks in the background. The iOS mSpeed application, therefore, relies on the "significant location change" API, which allows applications to be woken up when the user changes location (this is typically occurring when the cellular radio hands over to a different base station). The mSpeed application is programmed to only allow one measurement per block of 6 hours, to limit battery and network bandwidth usage. The background measurement feature is opt-in only and can be disabled by the end-user.
3.1.3.3 Measurement Primitives

**Basic Primitives.** *Ping* and *pingDNS* measure network latency to an arbitrary target and to the client's local DNS server. Both measurements can be parameterised with the target host (ping only), packet count, packet size, inter-packet time and timeout values. Currently, we performed ten pings to the target but selected the minimum value to minimise the effect of network traffic. *Traceroute* implements the traditional traceroute tool. It returns round-trip times, hop count and IP address of each intermediate router along the path from the client to the server. *curl* is a measurement based on the curl library, and provides a flexible multi-protocol data transfer library, returning a wide range of measurements such as DNS lookup time, TCP connect time, effective target IP address, etc.

**Bulk transfer.** This primitive emulates a data transfer of arbitrary size from any third party web server, to our client. The method mimics a large object download by issuing several back-to-back HTTP GET requests for a small file, in one TCP connection, requesting both the HTTP keep-alive and HTTP pipelining feature from the server. Web servers reply to such requests with multiple HTTP responses back-to-back in the socket, including the file each time. The end result is a tool, which can transfer an arbitrary amount of data from any web server, which supports HTTP 1.1 pipelining and keep-alive. We also capture instantaneous speed for every second to see the progression of window size and performance when downloading small and big objects over the same connection. We use this primitive to estimate the available bandwidth from a third party web server to our client. In addition, the process is repeated over several parallel TCP connections, in order to maximise the chance of 'filling the pipe'.

**CDN evaluation.** This primitive is used to perform experiments to a chosen CDN service. This primitive builds upon the other primitives discussed above. First, the mSpeed client downloads a single object (which is specified as input) from the server selected by the CDN; the client forces the server to ignore any cached copy of the object cache and obtains the object from the origin server. (Note that although HTTP Cache-Control is not honoured by many CDNs, including Akamai and Limelight, a CDN cache server can be forced to obtain a new copy of an object by sending a HTTP GET request with a random search string). This step serves as a proxy for the cache-miss scenario. Second, the client sends a HTTP GET request for the same object with the same random string as in the first step to the same CDN server to obtain the object. This step emulates the cache-hit scenario. In cases where CDN use different cache key approach, we use time different between the first and second download as another indicator. With this technique, we can verify that the second download is not a subject of transparent caching deployed by mobile operators. Finally, the client performs ping, traceroute, and bulk transfer measurements against the selected server to measure latency, route and per hop latency, and the available bandwidth along the path to the server.

3.1.3.4 First Results

The mSpeed application was launched in December 2013 and we collected measurement results from an experiment campaign that focused on the performance of Akamai and Limelight CDN services. This campaign was run from 1 Jan 2014 through 28 April 2014. During this period, mSpeed was used from 900 unique iOS devices, resulting in over 5600 mScript experiments. These experiments were roughly equally split between cellular and WiFi networks. While
mSpeed was used from 70 different countries, we note that about 60% of the mSpeed users are from Australia, Thailand, and the US.

Overall, our measurement campaign uncovers a number of interesting facets about CDN performance in wireline (using WiFi as proxy) and cellular data networks. First, we find that the available bandwidth to both Akamai and Limelight cache servers are similar, regardless of the network distance between the cache node and the client. This is explained as the cellular network tail is most likely the bottleneck link. Second, we find that the latency advantage of Akamai when serving fixed network clients, with its cache nodes typically located in ISPs’ networks, is reduced for most clients connecting through cellular networks. Finally, our measurements uncovers some instances of sub-optimal server selection by CDN servers, wherein clients are being served by cache servers that are located at distant locations, resulting in poor round-trip latencies between the clients and the servers.

3.2 Sensors

3.2.1 Intamac IoT Platform [Update to D2.2]

Intamac provides Internet of Everything solutions to business and consumers around the world for applications in security, energy management, vulnerable person care, home automation and lighting, etc.

The Intamac ENSO platform consists of a cloud service; a multitude of different types of sensors; and sensor hubs that connect the sensors to the cloud services. A home can have multiple sensor hubs, but usually a single hub is used.

3.2.1.1 Sensors Types

There are many options for sensors that may be used. The table below is a subset of sensors available. In particular, the focus of the UCN project is directed more heavily at sensors, which contribute to an understanding of presence or lack thereof of individuals within a property.

<table>
<thead>
<tr>
<th>Sensor Type</th>
<th>Sensor ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Door Contact</td>
<td>DC</td>
<td>A simple magnetic switch that messages when the sensor detects if the door or windowed opens or closes. A house typically has Door Contacts installed at each entrance to the dwelling or at vulnerable entry points such as patio doors.</td>
</tr>
<tr>
<td>Passive Infrared Movement Detector</td>
<td>PIR</td>
<td>This detector detects movement. Most are pet-friendly, and some have cameras that can take photos when movement is detected.</td>
</tr>
<tr>
<td>Key Fob</td>
<td></td>
<td>Similar to a car key fob, the user can arm or disarm the hub. Often has panic and night mode buttons.</td>
</tr>
</tbody>
</table>
3.2.1.2 Sensor Communications

Sensors generally communicate only with a hub. In turn, the hub communicates with the outside world. Some sensors can communicate directly between each other – for example Sprue Aegis WiSafe2 sensors.

Almost all sensors use wireless technology to communicate. There are several options for the communication, including ZigBee, Z-Wave and various proprietary 860Mhz protocols. Less common are Bluetooth, DECT and WiFi.

Sensors communicate their activity immediately that it occurs to the hub. The hub translates the sensor message into a standardised message that the Intamac ENSO platform can process.

3.2.1.3 Sleepy Sensors

Most sensors run on battery power. Most battery-powered sensors are “sleepy”. To save battery consumption, a sensor will stop communicating with the hub until it has something to report. Most sleepy sensors will send a regular “heartbeat” message to the hub, usually once per hour. PIR type sensors will go to sleep immediately that they detect and report movement and will not wake up until a period of time has passed with no movement detected. This strategy is necessary to achieve a 1-2 year battery life. A “busy” sensor that regularly communicates with the hub will have a drastically shorter battery life.

Sensors that are permanently powered, such as smart plugs, are not sleepy. Sensors that can be commanded to perform an action, such as sirens, smart plugs or light switches are usually permanently powered and not sleepy.

3.2.1.4 Sensor Messages

A sensor reports activity only and not the interpretation of the activity. For example, a Door Contact sensor could report “Door Open”, but not “Intrusion Detected”.

The messages a sensor can send dependent upon the type of sensor.

Sensors can also report telemetrics such as energy consumption or temperature. Telemetrics are usually requested rather than published.

Sensors can also report whether they are tampered with, such as if the case is opened. Sensors also report battery status, connectivity status, etc.

3.2.1.5 Hub Automation and Modes

The hub responds to the sensors messages and relays messages to the cloud platform. A hub can be in different modes – usually Armed, Alarming, Disarmed and Night. The different modes influence the responses to the events that may occur.

Rules applied to the hub provide the definition of the behaviour. For example:
1. If the Hub is in *Armed* mode and a Door Contact of type Perimeter reports status *Opened*, then change state to *Alarming*.

2. If states changes to *Alarming*, sound all Sirens; start Recording state on all video cameras.

3. If Keypad Disarm button is pressed then change hub state to *Disarmed*.

4. If hub state changes to *Disarmed*, silence all Sirens.

5. If hub state changes, inform cloud of new state.

Likewise, the cloud also includes a rules engine to process cloud based responses to events. Actions performed in the cloud include informing the system users of alarm conditions, sending commands to the hub, etc.

### 3.2.1.6 Presence Detection

A sensor will not, in itself, provide a presence detection solution. Presence detection includes the following:

- Who is present?
- Where in the dwelling is each person present?
- Is the person moving between locations or leaving a location?
- If a person is not in the dwelling, are they nearby or heading to or away from the dwelling?
- What is each person’s usual routine?

The PIH recommender will take the relevant sensor data to calculate presence. For example, if a key fob or fall sensor is present, it is likely that the device owner is present. If video face detection detects more people in the house than there are residents defined, it is fair to assume that visitors are present. If a smart phone is present in the house (e.g., is connected to the WiFi), it is likely that the owner is present. This data can be cross-referenced to corroborate the likelihood.

### 3.2.1.7 Video Cameras

Video cameras can detect movement and sound and report these events to the cloud. The cameras can be configured to record a video clip when it detects activity.

Video cameras can also be used to stream live video to a user while away from the house. Streaming is limited by the domestic network bandwidth available for upload.

Video cameras work independently of the hub and communicate directly with the cloud.

### 3.2.1.8 Cloud communications

Devices such as hubs and video cameras talk with the cloud via secure XMPP and REST protocols.
Video is streamed from the camera as either live video or clips stored in the camera. Streaming is secured via cloud services. For example, an authorisation token obtained by the viewing app is valid for 1 minute and needs to be renewed regularly during playback.

3.2.2 PT SmartSense [Update to D2.2]

Moved to section 3.4.2.

3.3 Home Gateways

Users today often access Internet from home. A collector running on the home gateway can observe all the network traffic in the home network and to/from the Internet. Hence, it can track all users’ networked activities when users are at home. It can also observe which devices are connected to the home network and when as well as the performance of the home network. We have developed data collection tools for the Technicolor commercial gateways (Section 3.3.1), and we also leverage data collection tools developed for gateways running open platforms (Section 3.3.2) where we can obtain more detailed measurements. In Section 3.3.3 we provide an update on our works in home network diagnosis based on gateway data.

3.3.1 Data Collection on Technicolor Gateways

Technicolor worked together with Portugal Telecom on a home gateway-monitoring framework (gateway probes and scalable backend). Below we describe the gateway platform, the data collection methodology. D2.2 described the pilot deployment that involved a number of real PT customers in the first two years of the project. The success of this deployment caused Technicolor to move this gateway monitoring initiative to the business units. These results influenced the design of two new Technicolor products: WiFi Dr and DSL Dr.

**Technicolor Gateway Description:** The hardware platform (i) supports both ADSL2+ and Fiber ONT connectivity, (ii) 4x100 Mbps full duplex Ethernet ports, (iii) a WiFi access point enabled by a Broadcom 802.11b/g/n 2x2 radio with MIMO support, operating at 2.4Ghz with support for both 20Ghz and 40Mhz channels. The gateway firmware supports the Open Gateway Services Initiative (OSGI) specification - a java middleware framework that enables drop-in modules that can extend the functionality and services offered by the gateway in a modular manner.

**Collection Methodology:** We have implemented a lightweight OSGI module that periodically queries a set of metrics that change over time. The OSGI module starts with a default set of metrics and an associated frequency. The metrics we collect are restricted by the list of metrics currently exposed by the gateway API. Hence, we can only collect a sub-set of the metrics listed in Section 2.3. In particular, we collect only network metrics and user activity metrics (except HTTP headers). At every period, the module queries the metrics against a proprietary API, formats the results as a JSON object, and sends this using a HTTP POST to a backend server. The server responds with a new set of parameters (and frequency) or a default reply (which maintains the current configuration). Particular design choices in our data collection (frequency, metrics, data format) are configurable, but in practice these are dictated by a set of very stringent
operational constraints: the additional monitoring could not affect the stability of the gateway platforms, or negatively impact any of the services offered. This is on a fairly resource constrained hardware platform. This precludes the monitoring module from maintaining large amounts of state, using excessive bandwidth, polling frequently. After a considerable testing effort, the monitoring module in our test deployment was configured with a polling period of 1 minute. The JSON objects received at the server vary between 15-20kB.

3.3.2 Data Collection on Open Gateway Platforms

Data collection tools for open gateway platforms allow us to get more detailed measurements of user traffic traversing the gateway, because we are no longer limited by the gateway API to collect measurements. WP2 has developed and worked with data collection tools that run on two distinct open gateway platforms: OpenWRT (in collaboration with the BISmark project); and a Linux Eee PC notebook developed for the Homework project. In the previous deliverables we have provided a more detailed discussion of the data collection tools we have developed for these platforms. Below, we describe the final data collectors for the OpenWRT gateways that are used by our home network diagnosis tools (Section 3.3.3).

OpenWRT Gateways: OpenWrt is a Linux distribution (current version is 15.05) for embedded devices that provides a writable filesystem with package management allowing us to fully customize the device through the use of packages. OpenWRT can be run on many off-the-shelf home routers and wireless access points, including TP-Link WDR3600 and Netgear WND3700/3800 that we have used in the development. The TP-Link WDR3600 has a 560 MHz MIPS CPU, 128 MiB of RAM, four-port Ethernet switch, and two 802.11n wireless interfaces (2.4 GHz and 5 GHz). Netgear WNDR3700/3800 routers have a 450 MHz CPU, two wireless interfaces (2.4 GHz and 5 GHz), and either 128 MiB (3800) or 64 MiB (3700) of RAM.

Collection Methodology: We have implemented a set of lightweight data collection tools for OpenWRT to support our home network diagnosis work (see Section 3.3.3). First, to estimate the home wireless network capacity we poll the wireless driver (ath9k in our case) periodically to record the current channel busy time (fraction of time during which the access point sensed that the channel was busy), and per client physical layer (PHY) bitrate of the last frame sent to the client, frame delivery ratio (which captures the fraction of frames successfully delivered to the client), and the total number of bytes sent. We measure the access link performance by periodic pings (ICMP ping) to the first upstream IP hop (inside the ISP network). In our prototype implementation the wireless metrics were polled three times per second, and the access link RTT ten times per second. Our experiments show that these impose only a minor overhead (< 30%) on typical routers.

3.3.3 Home Network Diagnosis [New]

With the availability of cheap broadband connectivity, Internet access from the home is ubiquitous. Modern households host many networked devices, ranging from personal devices such as laptops and smartphones to printers, media centers, and a number other Internet of

---

3 Our dataset also includes a smaller, initial deployment with 30 second polling.
4 http://projectbismark.net/
Things (IoT) devices. These devices often connect with each other and to the Internet via a wireless home network; this connectivity has become an important part of the “Internet experience”. Unfortunately, home users have few means to identify when their home network bottlenecks their Internet performance, and hence often attribute poor performance to the access ISP. Access ISPs are in no better position to determine the cause of performance bottlenecks in the last mile, and yet their hotline must answer numerous calls from unsatisfied customers. Tools for correctly pinpointing whether the home wireless or the access ISP bottlenecks Internet throughput are valuable not only to home users and ISPs, but also to inform the wider debate on regulating the residential broadband market.

UCN in collaboration with the BISmark project have characterized downstream throughput bottlenecks in 2,652 home networks in the United States (in a home gateway deployment sponsored by the Federal Communication Commission). We have developed an algorithm called HoA (for Home or Access) to identify whether downstream bottlenecks occur in the home network or in the access link. Our results (published at PAM 2016 and described in Appendix III) show that as access link capacity exceeds 20 Mbps, the home wireless network often introduces throughput bottlenecks.

Wi-Fi is the preferred way of accessing the Internet for many devices at home, but it is vulnerable to performance problems. Unfortunately, Wi-Fi performance is highly variable. For example, in dense urban neighbourhoods it is typical to see tens of competing Wi-Fi networks and other non-Wi-Fi devices (e.g., microwave ovens), which will cause contention and interference. Suboptimal installation of the Wi-Fi access point (AP) can also degrade performance. For example, the AP may be placed in a location that leaves devices with weak signal. Our discussions with residential Internet Service Providers (ISPs) indicate that often, when customers call to complain about poor performance, the problem is in the home Wi-Fi, not the ISP network. However, diagnosing problems in the home Wi-Fi is challenging for ISPs due to the lack of visibility within the home network.

We believe that ISPs should instrument APs to monitor Wi-Fi performance and assist in diagnosis. To this end, we propose a method to estimate the link capacity of a Wi-Fi link using physical layer metrics passively sampled on commodity access points (wireless physical rate, retransmission rate and the count of packets/bytes transmitted). We build a model that predicts the maximum UDP throughput a device can sustain, which extends previous models to consider IEEE 802.11n optimizations such as frame aggregation. We validate our model through controlled experiments in an anechoic chamber. Over 95% of the link capacity predictions present errors below 5% when using our model with reference data. Link capacity is useful for Wi-Fi diagnosis: a device’s throughput may be limited because of medium access problems (i.e., Wi-Fi or non-Wi-Fi) contention prevents access to the medium) or frame delivery problems affecting the link capacity (i.e., when the channel quality is poor). Full details of this work can be found in our paper published at TMA 2016, attached to this report as Appendix IV.

Our original HoA algorithm presented performance issues when running on commodity home routers. We have developed a new algorithm that leverages the link capacity estimation model. Pinpointing whether the home wireless or the access ISP bottlenecks Internet throughput is valuable for home users who want to better troubleshoot their Internet experience; for access
ISPs that receive numerous calls from frustrated home customers; and for informing the debate on regulating the residential broadband market.

We designed an access bottleneck detector based on lightweight pings of the access link and a wireless bottleneck detector based on the above discussed model of wireless capacity. We evaluated the accuracy of our detectors using controlled experiments, and showed that we can detect both downstream and upstream throughput bottlenecks with more than 93% true positive rate and less than 8% false positive rate. Our performance evaluation shows that our algorithm runs online with less than 30% load average on a standard Netgear home router. The full draft (under submission to IMC 2016) of this work is attached as Appendix V.

3.4 Service Back Ends

In this section, we describe the data collection methodology and deployment at partners’ existing media service back ends: NICTA’s Social TV platform and Portugal Telecom’s IPTV service.

3.4.1 NICTA Next-Generation Content Discovery & Distribution Project

The Next-Generation Content Discovery & Distribution project is a research project to design, develop and trial technologies in the area of social TV, recommendation and content distribution. The project represents a collaborative effort between Australian Centre for Broadband Innovation (ACBI) and National ICT Australia Ltd (NICTA). The project also has a number of partners including the Australian Broadcasting Corporation, Technicolor, AARNET & others.

The project has two overarching objectives:

- **Exploring Usage, Recommendation & Privacy:** as more user-generated and studio-generated content becomes available, the task of finding and positioning content becomes difficult. In this context, this project specifically focuses on maximising content recommendation effectiveness, while taking into account the privacy implication to the end users. Therefore, this aspect of the project has two key objectives:
  - Develop new content recommendation techniques, which overcome the current recommendation systems’ limitations including low accuracy and loss of privacy
  - Develop new privacy-preserving mechanisms to compute statistically sound aggregated usage and behaviour data, while preserving the privacy of data belonging to individual users or service providers.

- **Efficient Content Delivery:** the bandwidth and storage costs associated with distributing large amounts of on demand content are extremely high. This project addresses this problem by developing peer-assisted, scalable & efficient content distribution methods for on-demand video.

**Methodology**

The proposed methodology draws on both theoretical and experimental research, and the use
of trial sites to feed data back into the research. When appropriate, the experimental approach will seek to use live prototypes with real users, in particular for the collection of data for the behavioural-related research (e.g., recommendation engines).

To this end, NICTA established two test beds in the following order:

- A laboratory test-bed, internal to NICTA, serving 6-10 users; this test-bed is primarily used for development, testing and demonstration.
- A university or community residence environment for some hundreds of users. The focus of this trial was the collection of data for the behavioural aspects of the project, in collaboration with the University of New England in Armidale.

We have applied the lessons learned from these environments in the context of UCN.

**Data Collection and Privacy**

As discussed, the trials for the project were established primarily with the purpose of gathering data. Privacy of the trial participants, however, was considered of utmost importance, and trial participants were informed as to exactly what information was being recorded. A range of data from the trials was captured, stored and analysed, though no attempt was made to reconcile this data to any user. No user identification or IP address information was stored along with the usage data and while each participant was identified by a unique key associated with their client software bundle, no login, email or any other form of identification was required.

Usage data was collected for the duration of the trial, and was stored at NICTA, in accordance with NICTA’s data safety policies and practices. The following key user reactions were recorded:

- Title / Genre & Channel Selection
- Like, Play, Stop, Pause, Resume events
- Twitter, Facebook and Recommendation posts

Trial participants were offered free live, catch-up TV and video-on-demand services, for the duration of the trial. The live TV service consisted of free-to-air TV channels (http://www.freeview.com.au/), with a catch-up TV service offered in partnership with the Australian Broadcasting Corporation, with the videos selected from their iView service (http://www.abc.net.au/iview/). For the video-on-demand service, videos were selected from popular TED Talks (http://www.ted.com/). The respective hosts of future trials may also later add additional material. In the case of a university, for example, this may include lectures or other informational material.

During the trial, the participants were able to watch TV through their computer, and their usage history was stored and securely transmitted to a data collection server for later analysis. Specifically, any “Like” “play”, “stop”, “pause” events, and the associated video item (title, genre, etc. or TV channel) were collected. The collected data was used periodically to enable a video recommendation service, where users are presented with videos that are more likely to interest them.
From an ethical standpoint, data was only collected on videos played through our client application, which was restricted to the videos we provided as part of the service. In addition, there was no specific content of sensitive nature in the video on demand system (no adult-only material or other). All videos provided by either the ABC or from TED Talks were reviewed by the project team before presentation within the system.

Service Overview
The service provided by the Social TV platform includes a range of content delivered to either a PC/Mac based client application or a more traditional Set-top Box, all via IP. The service includes:

- Live TV - Captured from a local, conventional Live TV signal and re-broadcast over IP using multicast
- Video on demand/catch up TV using ABC iView content
- Online Content (Such as TED Talks), Local Content (Such as university lectures) & social network integration such as Twitter and Facebook
- Recommendation fusion of online + broadcast + catch-up TV content

The service was provided at no cost to trial hosts or participants and offered participants a modern user interface to select and view content. The experience is identical regardless of the platform, providing a consistent user experience, and access to standard Live TV, ABC iView, Internet and locally added content.

![Figure 6: Service Appearance - Recommendations Screen](image)

In addition to a standard EPG that is expected for a Live TV service, users also experience a recommendation screen, detailed information screens and a capability to provide and view ratings and feedback on programs. Facebook and Twitter integration allows participants to automatically post and view comments on the program being watched.
Figure 7: Appearance - Program On-Screen Twitter Integration

Users are able to comment on programs directly while watching TV, and view other comments on the program being watched.

Figure 8: Service Appearance - Program Detail

Much like iView from the ABC, users are able to get detailed information on each program as well as have immediate access to social network information on the program.
Content Licensing: The right to re-broadcast iView content is provided by the ABC, with explicit approval granted for each trial. The Live TV signal is a re-broadcast local TV signal, and is therefore license free, and TED talks are free for redistribution. Any additional content delivered on the platform by the trial hosts must be owned by the trial host or licensed for redistribution on such a platform.

Recommendation Service
The recommendation service, developed at NICTA, is realised through the use of collaborative filtering techniques, where users and video items are logically clustered in the recommendation engine. This clustering is not directly exposed to participants, who only see a list of suggestions. The quality of the recommendation engine output will then be evaluated and refined in an attempt to maximise the quality of the recommendation. The quality of the recommendation will be measured, e.g. as the ratio of items viewed from the recommended list to the overall number of item viewed.

Video on-demand recommendations have been around for some time, but this study allows us to obtain a live, on-demand and catch-up TV services data set. The live TV events can be annotated with Electronic Program Guides, to obtain information about the show currently being watched.

Technical Overview
The project utilises a primary server in NICTA in Sydney as an aggregation point for content delivered as part of the service offering to trial participants. A replica of this primary server serves as the main delivery server for each trial site. Essentially, this server requires a connection to AARNET to be kept up to date, and a reliable network connection to its clients. In addition, the server requires an aerial connection to enable it to receive and re-transmit free to air TV. This is illustrated at a high level in the diagram below (Figure 9).

The project requires a single on-site 3RU Server with ~2TB or RAID enabled Storage, Video Ingest and streaming capability. This server performs the content ingest and multicast of the Live TV, and also serves the on demand streams. The system requires a secure, cooled environment, as well as a local TV aerial feed to enable it to capture the local TV channels.

Clients for the trial may be in one of 2 forms: an application for a PC/Mac, or a stand-alone set-top box running the same application. In the initial stages of the trials, only the desktop application will be available. The application is a browser-based desktop application bundle that must be downloaded and installed on the user’s PC. The application is supported on Windows, Linux or MacOSX based machines, and requires a reasonably modern computer (last 5 years or so) with at least 2GB RAM to run effectively.
Trials at Armidale

The project had trials outside NICTA, at the University of New England (UNE), Armidale. With its commitment to research, excellent local network infrastructure, large local student population and proximity to early NBN deployments, the University of New England in Armidale was a prefect trial site for the service.

Approval process: Independent of the technical evaluation, implementation of the trial will required an approval process. The UNE required to know which data was to be collected, and NICTA delivered a document describing the exact information collected and the timeframe of the data collection.

The Trial with UNE: the trial in the University of New England leveraged existing NICTA infrastructure, adding a single on-campus server with access via AARNET. It was considered best to install the server in a secure server room that has a simple access to a roof UHF antenna, and was close to the colleges. The server had a uniquely resolved interface for student clients and one for the staff network, + 1 for management.

Participant Population & Network: The primary targets for the trial were the dorm rooms at the colleges. Rooms are simple and have no Aerial, though students could have a TV and their own aerial (estimated <50% have TVs). Foxtel redistribution was tried for a while over IP though the take-up rate was poor – students didn’t want to pay for it.
Housing 2100 students, there is a 10G link to the colleges - 2 x 1G to each of the 8 colleges, with 1GBit/s to each room. The Colleges are all on a L2 Switched VLAN and ideally, Service delivery ports of the server were best in this VLAN as the bulk of the traffic could remain within this VLAN rather than be routed back through the campus network.

The campus wireless network is A//B/G Dual radio “Dumb” Access Points with no 802.11n yet. This network is geared to supporting common areas and was unlikely to be viable for delivery of much (if any) of content from the service. Indeed, usage on the service could dramatically affect other services on the wireless network, so the service was considered unlikely to be made available via the wireless network.

Security: ABC had stringent requirements around the security of iView content, and NICTA confirmed their approval based on the parameters for likely restricted access to the server room on campus. The server room is card swipe access only with revocation, and ~20 IT staff have access. Network access to server was also secured.

Data Rates and Protocols: VOD Video (iView) is 650K with the Multicast being up to 19Mbit/s according to the channel. Total bandwidth of multicast traffic for all channels is estimated at 110Mbit/s. It is expected that the load on the server for VOD traffic could be quite high, given the number of potential participants, and performance testing will be conducted by NICTA. The Following protocols will need to be enabled in order to deliver the service:

1. Client side delivery is HTTP/HTTPS or multicast
2. IGMP V2 required on the server as network does not yet have IGMP V3 support
3. Management and content replication on server require RSYNC and SSH enabled

The 1st Trial Summary
Location: University of New England, student dormitory
Population: 2500 students
Trial period: ~ 7 months (Feb 22, 2013 – Sept 22, 2013)
Stats:
   1286 downloads
   135 monthly active unique users
What We Have Learned

Several approaches for TV show recommendation engines were compared; the best method found to be performing on the iView workload is matrix factorization, a commonly used method for movie recommendation. The matrix factorization algorithm factorizes the users’ input ratings matrix into two latent matrices: a user-factor matrix and an item-factor matrix. The dimension of the latent matrices, k, is an external factorization parameter. One of the advantages of the matrix factorization model in the context of UCN, is that inputs from several information sources (user ratings, social media, etc.) can be incorporated into the same latent factor domain. Unfortunately, the online-social network augmented recommendations could not be designed due to lack of data in the trials, and is left for future work.

In this trial, students mostly watched live TV – this results in a sparse dataset for VoD, and makes it hard to evaluate the recommender output (which only operates on VOD). We need better content for VoD matching the demographic of participants.
We need to interview students to understand user reticence / retention issues. However, we have no direct access to students, and have to relay through student services and head of collages.

During the trial, we had technical issues due to multicast deployment on campus - multicast works on wired network only, where some students prefer Wi-Fi.

Doing user-based trial research in the TV/entertainment area is challenging. The main difficulties are:

- **Content acquisition** - how to find content which appeals to a large portion of the trial population.
- **User acquisition** - how to convince potential users to invest time and effort into using a new system or platform, especially for a temporary trial. User utility needs to be high in this case.
- **Engineering required** - doing user-based trials requires product design and engineering experience, which is difficult to find in a research organization.

### 3.4.2 PT Smart TV and SmartSense [Update to D2.2]

The aim of the Smart TV project is to design and develop new services for the Portuguese IPTV users, i.e. MEO users. It includes suggestions and recommendations, visualization of aggregated data derived from normal TV consumption information as well as social networking integration. Currently, data is being collected from different sources: IPTV Set Top Boxes (STBs), mobile TV apps, Twitter and Facebook. Data can be collected at the GTWs, for simple usage services, at the Set-Top-Box (STB). In this last case, users who want to access a smarter service, have to install an application on their STBs which will feed data into the so called PT SmartData cloud platform. This allows algorithms to process and recommend based on their viewing patterns. It is also possible to collect data from a special mobile app version of the PT IPTV solution called MEOGo.

In addition, in the Smart Sense project data from well-being sensors are being collected and channelled either through the home Internet connection of the GTW or any other Wi-Fi connection using a mobile phone as a GTW.

**Objectives**

There are two main objectives in these projects:

- Recommend valuable TV programs/channels, based on users viewing history of channels and TV programs. In order to accomplish this, is necessary to:
  - **Mine the user’s STB**, or other consumption device like a Smartphone/Tablet, to gather information about what is being watched, namely the channel, the show, the time of the day, how long was watched and how long was the show.
  - **Profile the user**, based on the “mined” information gathered in almost real-time. The more relevant aspects are the top channels being watched, top shows, top genres, trends of TV consumption by day and time of the day and top recordings.
o **Recommend a channel or TV show**, by combining the data from the user history with the EPG of the week before and after.

- Present aggregated and post-processed sensor data over time (e.g. week) to the user to allow him (or any other care taker) to monitor his well-being. In order to perform this task, it is important to:
  - Collect information from sensors which can give information about weight, blood pressure, sugar level, etc.; this information can be collected by the user or any other person (care taker) as long as it is authorized. It can be automatically fed by the sensor or introduced manually.
  - Go through a learning process to understand what is normal and is a deviation and process aggregated values (e.g. body mass)
  - Represent graphically the relevant values to be visualized on a TV or mobile device.

### 3.4.2.1 PT SmartTV

This project in particular concerns collecting data from the usage content consumption and suggest/recommend related content.

**Methodology**

The methodology used serves both research and business purposes. So, it is gradual, iterative and goes through the following phases:

- A laboratory test-bed, internal to PT, serving 4-6 users; this test-bed is used for development, testing and demonstration.
- A community of beta testers (150-200 users). The focus of this trial is to intensively collect feedback in order to improve the proposed solutions in terms of usability and functionality.
- A larger group of users (200-1500 users) in a pre-commercial status.
- Promote and release a commercial version, depending of course on the results collected in all previous phases.

Currently a new interface version of the IPTV service has been launched (meaning, is ready for phase 4) which exposes some of the implemented functionalities (e.g. top 5, suggestions) based on information collected at the backend.

Nevertheless, due to the nature of the new software used on the STB (which gives more detailed and personalised data), namely the proprietary and confidentiality rights, the experiments were limited to a reduced set of users in a controlled environment (meaning, we are still in phase 1). The plans are to expand in the coming months, due to the public release of the software to PT customers.

On the other hand, there is more flexibility for the mobile usage (i.e. MEOGo app) and soon it will be possible to get viewing data from all clients who want to install a simple app (for now is limited to a controlled environment). This will allow them to send data related to their TV consumption to the backend and receive, in return, recommendations accordingly. Some
additional incentives might be offered. It is also possible to relate the MEOGo account with the respective STB, providing cross platform recommendations.

**Data Collection, Privacy and Security**
Currently usage data is collected and stored at PT, in accordance with PT data safety policies and practices.

Nevertheless, in the new and experimental solution the user has to activate the service on his STB, install an app on his mobile phone/tablet or give permission to use his/her social networks data. Therefore, the user will be perfectly aware of the terms and conditions of the collected data and the used means to accomplish it. The process is completely user-centric.

Currently, and according to the UCN, and in particular the concept of a PIH-oriented architecture, the user is able to set up the terms of disclosure. The user can choose which privacy and security mechanisms will be used to transfer his data and who will be able to have access to which data. This work is part of the research and experimental track conducted by PT/Altice Labs in WP4 and documented in more detail in deliverable D4.2.

**User Consumption**
Assuming the user allows data to be sent to PT, every minute an event is built, based on the viewed content, and sent to the backend, namely the current channel, program, start and end time, title, episode, description, program id and time viewed and, if it is a recording (Catch up TV or recordings), also the date and time that the user saw the content. The following table summarizes the collected information from the set-top-box:

---

**From the MEO IPTV Set-Top-Box**

<table>
<thead>
<tr>
<th>Data field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STBoxID</td>
<td>UUID for set-top box.</td>
</tr>
<tr>
<td>Tunetime</td>
<td>Timestamp of last channel shift.</td>
</tr>
<tr>
<td>Channel</td>
<td>Numeric identification of TV channel.</td>
</tr>
<tr>
<td>Duration</td>
<td>Length of TV program.</td>
</tr>
<tr>
<td>ProgramStartTime</td>
<td>Initial timestamp of TV program.</td>
</tr>
<tr>
<td>TimeShiftSeconds</td>
<td>Time in seconds of transmission comparing with ProgramStartTime.</td>
</tr>
<tr>
<td>StationLongname</td>
<td>Full name of channel.</td>
</tr>
<tr>
<td>StationShortname</td>
<td>Short name of channel.</td>
</tr>
<tr>
<td>Streamtime</td>
<td>Timestamp of sent event.</td>
</tr>
<tr>
<td>Title</td>
<td>TV program name.</td>
</tr>
<tr>
<td>EpisodeTitle</td>
<td>Title of TV program episode.</td>
</tr>
<tr>
<td>Description</td>
<td>TV program description.</td>
</tr>
<tr>
<td>ProgramID</td>
<td>Program UUID.</td>
</tr>
<tr>
<td>Program_isAdult</td>
<td>Boolean value of Adult Classification.</td>
</tr>
</tbody>
</table>
---

32
The events generated by the mobile MEOGo app are also recorded. This is a companion app that allows the user to interact with the STB and watch live or recorded TV from a mobile device, wherever he wants. This app gives a lot more information than the STB, namely information about the interface usage and VoD. The following table summarizes some of the most important events that are collected with the app.

**Table 5: Meo STB recorded values**

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiveTV_Channel_EPG_Tuned</td>
<td>EPG opened</td>
</tr>
<tr>
<td>LiveTV_Channel_Search_Tuned</td>
<td>Search opened</td>
</tr>
<tr>
<td>LiveTV_Channel_Views_Duration_XX</td>
<td>Channel viewed for XX minutes (1, 5, 10, 15, 30, 60, 90 and 120)</td>
</tr>
<tr>
<td>XXXXX_Facebook_Shared</td>
<td>Program shared on Facebook (via live TV, VOD, recording...)</td>
</tr>
<tr>
<td>XXXXX_Twitter_Shared</td>
<td>Program shared on Twitter</td>
</tr>
<tr>
<td>LiveTV_WatchOnTV_Selected</td>
<td>Program sent to STB</td>
</tr>
<tr>
<td>VOD_Movie_AddedToFavorites</td>
<td>VOD movie added to favourites</td>
</tr>
<tr>
<td>VOD_RentMovie_XXXXX</td>
<td>VOD program rented</td>
</tr>
<tr>
<td>VOD_Movie_Played</td>
<td>VOD program played</td>
</tr>
<tr>
<td>EPG_Alert_Program, EPG_Recording_Alert_Program</td>
<td>Alert created for a program in EPG</td>
</tr>
<tr>
<td>Recording_Record_Program, Recording_EPG_Program</td>
<td>Program recorded</td>
</tr>
<tr>
<td>GA_Play_Episode</td>
<td>Recording played</td>
</tr>
<tr>
<td>GA_Program_Views_Duration_XX</td>
<td>Recording viewed for XX minutes (1, 5, 10, 15, 30, 60, 90 and 120)</td>
</tr>
</tbody>
</table>

**Table 6: MeoGo recorded events**

From the events collected from TV consumption it is possible to get a number of measures, like the top channels and top shows, which can be used to shape the order in which the recommendations are given to a user.

To recommend something new to the user, something that he hasn’t seen but might like, other users’ information, besides the items already described, are also collected. This is necessary to find users with similar preferences/tastes.
...From the big data platform GOLIAS

Since we had initially only a small group of testers that use our app to collect information, we needed to get more users to have the largest information possible and get better results. That is why we are now using information from real PT clients. Whenever a client requests content from his STB, there is an action that is logged on the server. All these logs are stored in a big data platform called Golias. This is where PT does all kinds of studies about users’ behaviour and gets reports with indicators to give to business departments. To use this data in UCN, we extract anonymized TV consumption logs from Golias from around 500,000 users. The type of information that is being collected has the same attributes that we extract from our app, and are described on Table 5. This extraction is done weekly.

![Figure 10: Top programs for February 2015 on MEOGo (number of views)](image-url)
Manual Input and Social Networks

Besides MEOGo, there is also an application for content discovery named Guider. This mobile app allows the user to search for new content, through filtering by genre, time and classification, and to classify movies and TV shows with the following information, which is also being collected:

...*From the mobile content discovery App Guider*

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star rating</td>
<td>A 0 to 20 star rating</td>
</tr>
<tr>
<td>Target audience</td>
<td>Mood that the program inspires.</td>
</tr>
<tr>
<td>Mood</td>
<td>Age target of the program.</td>
</tr>
</tbody>
</table>

*Table 7: Guider collected information*

The user has also the possibility of giving information from his Facebook account. If permission is given, there is also being collected the following information:

...*From Facebook*

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likes</td>
<td>All the things that the user “liked” on Facebook, movies/TV shows related and otherwise</td>
</tr>
</tbody>
</table>
\\begin{table}
\centering
\begin{tabular}{|l|p{10cm}|}
\hline
\textbf{Watched} & Movies/TV shows defined as watched by the user \\
\textbf{Want to Watch} & Movies/TV shows that the user defined that wants to watch \\
\textbf{Classification} & Numerical classification (0 to 5) given by the user to a movie or TV show \\
\hline
\end{tabular}
\caption{Table 8: Facebook collected events}
\end{table}

Besides Facebook also Twitter has been considered. TVPulse is a service which collects tweets filtered by country (in this case Portugal) and that are related to television (i.e. a selected set of channels and programmes form the EPG to maximise the performance of the search). The developed platform analyses the tweets to associate them with programs that are being transmitted and extracts user tags/classifications from the tweet corpus. The tags associated with the programs are then used to search YouTube for related videos. In UCN we collect this generated data to present users with YouTube videos and to have information about what programs are more popular on social networks. The following table illustrates data being collected for each program:

...From Twitter and the build on-top TVPulse service

\begin{table}
\centering
\begin{tabular}{|l|p{10cm}|}
\hline
\textbf{Data} & \textbf{Description} \\
\textbf{Title} & Title of the TV program \\
\textbf{Channel} & Channel in which was transmitted \\
\textbf{Start Time} & When started the transmission \\
\textbf{End Time} & When it ended \\
\textbf{Tags} & Tags extracted from the tweets that classify/represent the program, with respective weight \\
\textbf{Youtube Videos} & List of related YouTube videos returned from the search with the tags \\
\textbf{Tweets} & List of more relevant tweets \\
\textbf{Hashtags} & List of Hashtags associated with the program \\
\hline
\end{tabular}
\caption{Table 9: TVPulse collected events}
\end{table}

An end-user web application called NowUp uses the tweets to identify the most popular moments in a TV show. Then it segments the shows that have been aired accordingly, by highlighting the most tweeted moments in a video montage. The publically available “hot shortcuts” are then displayed along with the tweets and the related YouTube videos. It also integrates rating and other social relevant information (i.e. tops).

Enhancing and Integrating other Recommendation Systems

Inria together with Technicolor developed a recommendation system named Structured Recommendation System (SRS). This system analyses IMDB user reviews, extracts the
common expressions that are used to classify the movies/shows (similar to genres classification), and relates shows to each other based in these expressions. The main recommendation platform uses this output to complement the data that has already been collected about every programme (described below) and adds a list of all related shows.

The Structured Recommendation System is also capable of producing user-oriented recommendations, but it needs the user manual classification of shows. These manual classifications are already being collected in the Guider app, so this information is being sent to the SRS. When this additional user information is available, the recommendation system uses the SRS tailored related/recommended programs instead of the generic ones.

Complementarily to the Structured Recommendation System Inria/Technicolor developed also a Structured Classification System (SCS). This system also analyses IMDB user classifications but this time to provide generic Tags that classify the shows. The platform will also use these tags to complement the information that classifies each show.

**Context**
To contextualize the data collected from users and to limit recommendations only to the available content, the EPG is collected, the one used in PT’s MEO Electronic Programming Guide service. From this source, all the programs that were transmitted in the last seven days are discovered, as well as the ones being transmitted in the next week. Information is collected about the program that will be aired on a specific channel and for each program the service returns its description, i.e. the air date/time, program's duration and classification (genres). But, because this information is mainly managed manually, it is not always complete and accurate. For that reason, the use of The Movie Database (TMDb) open database is important to correct and complement the EPG data. The following information is collected for each show:

...*From the TMDb*

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Title</td>
<td>Original title of the show search for (necessary because the majority of programs in Portugal have the Portuguese translation)</td>
</tr>
<tr>
<td>Release Date</td>
<td></td>
</tr>
<tr>
<td>Popularity</td>
<td></td>
</tr>
<tr>
<td>IMDB ID</td>
<td></td>
</tr>
<tr>
<td>Genres</td>
<td></td>
</tr>
<tr>
<td>Vote Average</td>
<td></td>
</tr>
</tbody>
</table>

*Table 10: TheMovieDB collected data*

**Service Overview**
The service provided to the end user can be split into many components, two of them inside an ordinary IPTV STB, and another two on a mobile device. The first is an application that runs on the STB and records what the user is watching. This application runs whenever the user
wants and sends the data to the SmartData backend platform, that stores it. The second is the application that runs in the STB menu that shows the recommendations using the retrieved data. The third is the MEOGo mobile IPTV app that both collects the user consumption and shows recommendations. The fourth is the Guider mobile app, that records users’ manual classification of shows.

In the backend there are also two main systems: one that creates a “user profile”, based on what the user saw the last month (retrieved from SmartData), and another where the recommendation algorithm is located. The first system also collects all the context information described in this document, on “Manuel input and Social Networks” and “Context”. All this is mainly used to contextualize the data collected from the user and to enrich the information that we have about the programs that can be recommended.

On the recommendation system, the “user profile” is fed to the algorithm, in parallel with the context data, as described in deliverable D3.2. In the end it provides recommendations for what is being transmitted at that time (live) and for what has already passed (recordings). The user can also get recommendations on related programs (mainly by genre), search for available shows and see highlighted programs, all of them ordered according to the user tastes.

![Live TV recommendations](image)

*Figure 12: Live TV recommendations*
The STBs provided by PT IPTV MEO’s service are based on Ericsson Mediaroom (formerly Microsoft Mediaroom). The application that records the viewings consists of an application layer that runs on top of the Mediaroom software and adds data to the SmartData backend platform. Here, the data is routed to a PostgreSQL database, for lifelong storing, and to a Cassandra big data database, where it is maintained for a short lifetime (e.g. TTL for EPG is one month).

The Smart TV Recommendation can be divided in two modules that sit on top of the data provided by the SmartData platform. In the first module, the Smart TV API, all the data concerning a user is given access through a REST API, including the most watched channels,
programs and the trends. These data are used by the Smart TV Recommender that also provides a REST API with the channel, programs and related programs recommendations for a specific user.

![High level architecture](image)

**Figure 4: High level architecture**

3.4.2.2 SmartSense

The SmartSense project aims to provide valuable information about the well-being of its users, relating historic measurements to live data.

To help users being updated about their health evolution, this projects aims at facilitating and storing vital measures, like weight, blood pressure and cardiac rhythm. Knowing that data is only helpful if it can be used to give relevant information to users, it also has the aim to provide the evolution of users’ well-being and detect abnormal values, by comparing with historic data.

We are also trying to help users having a healthier lifestyle. With a TV application called Smart Training, users have access to a list of exercises that they can do, tailored according to their health status. They can manage training sessions and see their results progress over time.

**Methodology**

This project was developed following three steps:

- A pool of volunteers from PT Inovação/Altice Labs (any employee can use the application);
- A selected tester partner; a Portuguese institution which takes care of elder people (Santa Casa da Misericórdia)
- Real users who are served by the same institution of the last point
Currently we are on the third step where we have a set of approximately 20 real users that use the system, besides PT collaborators.

**Data Collection, Privacy and Security**

All measures can be collected using physical devices (sensors) connected to a GTW by Bluetooth (or some other selected protocols e.g. ZigBee - adaptors have to be developed case by case) or manually by the user or any authorised caretaker. Each user has his own account that he uses to log in and perform the measurement. The available data that has been monitored can be seen on the following table.

<table>
<thead>
<tr>
<th>Well-Being data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
</tr>
<tr>
<td>Blood Pressure (Systolic and Diastolic)</td>
</tr>
<tr>
<td>Weight</td>
</tr>
<tr>
<td>Glucose</td>
</tr>
<tr>
<td>Movement</td>
</tr>
<tr>
<td>Oxygen saturation</td>
</tr>
<tr>
<td>Heart Beat Rate</td>
</tr>
<tr>
<td>EMG</td>
</tr>
<tr>
<td>ECG</td>
</tr>
<tr>
<td>Foot Temperature</td>
</tr>
<tr>
<td>Foot Humidity</td>
</tr>
<tr>
<td>Body Mass Index (calculated)</td>
</tr>
<tr>
<td>Respiratory Rate</td>
</tr>
<tr>
<td>Step Counter</td>
</tr>
</tbody>
</table>

*Table 11: Recorded measures*

We are also trying to integrate fitness bands as a new input. With this complementary data, that is collected more frequently, we have more granularity in the changes of the vital signs. This new source is also important to track users’ activities/exercise, complementing the Smart Training app. For now, we have access to pedometer and are trying to gain access to sleep data.

**Service Overview**

Measurement devices are connected to one Raspberry Pi gateway that sends data to the SmartData backend platform. The devices are used as usual, allowing the evaluation of blood pressure, cardiac rhythm and weight. Through a web portal is possible to access the measurements evolution, calculate the Body Mass Index (BMI) and check median values. The measures can be seen on a large range of devices, from a TV Screen to a Smartphone. There can also be physicians or caretakers that see the evolution and can be notified by abnormalities.
Figure 5: User interface on a Meo STB

Technical Overview
As in the previous use case, data retrieved by devices is sent to the SmartData backend platform where it is routed to a PostgreSQL database, for lifelong storing, and to the Cassandra big data database. On top of that sits the REST API that provides access to Cassandra data. The portal uses those endpoints to provide meaningful information to users.

Figure 6: High level architecture
Currently, the connection between the STB and the backend is secured through a VPN but it is being replaced by a secure encryption algorithm. More details can be found in deliverable D4.2. As in the previous case, in the future privacy policies will be handled by the user.
4 RESULTS OF THE USER STUDY

In deliverable D2.2 we described in some detail the ethnographic work that was being undertaken with groups of users in France and the UK in order to establish ‘ground truth’ for the future collection and profiling of user context. As we pointed out at the time a range of metrics and algorithms will be used across the various components making up the UCN platform, but it is important to relate these to real world situations and real world practices in order to assess their effectiveness and the degree to which they may require refinement or augmentation if they are to support real use. For this reason researchers from Inria and the University of Nottingham undertook a series of small-scale studies with real users to observe and gather data about their existing online practices and media consumption and the ways in which concerns about privacy might come into play.

In the preceding deliverable we described in some detail the ethnographic process that would be undertaken, the kinds of data that would be logged and the ways in which data would be visualized to users so that we could obtain feedback about their logged activities. We also presented the ways in which users were recruited, the ethics process that had to be pursued in the two separate countries, and the ways in which the system was tested prior to deployment. In this deliverable we will therefore be focusing on presenting the findings of the studies and will only describe the data collection process and methodology in terms of giving an overview of what was finally captured and in order to delineate ways in which the process was adapted and augmented to capture additional kinds of data.

4.1 Study Setup
4.1.1 The French Study

As indicated in deliverable D2.2, participants were being drawn from a variety of household configurations in France. In France recruitment was initiated by sending out an invitation to participate to a database of 2,500 potential participants already managed by the Centre Multidisciplinaire des Sciences Comportementales Sorbonne Universités-INSEAD who were research partners of Inria and who could therefore offer access to certain shared resources. 140 responses were received to this initial invitation and subsequent requests for more information enabled us to whittle the list down to 27 households that were distinct from one another in some way. This list was further reduced down to 12 households to make the ethnographic work tractable. Studies in these households began at the end of January 2015 and continued across phased groups until the end of July 2015.

Selection of the households was based upon a variety of factors. These included: the geographical location of the household and consequent access to services (e.g. was the household in a rural, suburban or urban area?); the kind of accommodation being lived in (e.g. was it an apartment, a small terraced house, a large detached house, etc.?); the constitution of the household itself (e.g. was it single occupancy, a couple, a family with small children, a family with older children, a group of friends etc. living together, a retired couple, and so on?); the actual number of people living in the household and the extent to which it was regularly augmented by visitors; the occupation of the prospective participants, their age, their background, and the household’s socio-economic character; the nature of the local access to networks, and the extent to which the prospective participants made use of other networks in other locations; the kinds of computing devices being used and the ones being used specifically
for the consumption of media; the kinds of consumption of media practices prevalent in the household (e.g. just watching dedicated TVs, downloading and watching on laptops and mobile devices, catch-up TV and live streaming, and so on); and, finally, the degree to which the prospective participants were active users of social media (in order to support the project interest in how recommendation by social media impacts consumption).

The final 12 participants studied ranged in age from 20 to 40. 5 of them lived in town centre accommodation, 7 of them lived in suburban areas. 9 lived in apartments (which is commonplace in cities in France) and 3 lived in detached houses. 3 of the participants lived with their parents and other siblings, 2 with just 1 parent, and 1 with her daughter. 1 lived with his brother and 1 lived with his grandmother in an apartment shared with other, unrelated, families. Aside from this 1 shared with a friend, 1 was a lodger, and 2 lived on his or her own. Over half of the participants (7) were students. This demographic was an inevitable consequence of having the study promoted by s-INSEAD. 1 of the students also worked as a clerical assistant. Aside from this there was a solutions manager, a photographer/film-maker, a store assistant, a factory worker, and 1 who, at the time of the study, was unemployed. 10 of the 12 participants joined the study using laptops, 3 of which were running Mac OS and 7 Windows. 7 of those using laptops were also using smartphones, 3 of which were iPhones and the other 4 Android-based. However, in the end only 2 of these phones were actually used to capture data for the study. On top of this 2 participants joined the study with just an iPad and an iPhone. None of the participants made any significant use of desktop machines (some lived in households where there was a family desktop in the living room that they sometimes shared with others in the house for tasks such as printing). All of the participants made some use of OpenVPN for the purposes of logging their network activity. Only 3, however, ended up running the software for application and process logging.

At the conclusion of the study only very partial logged data had been acquired for 4 of the participants. This happened for a variety of reasons. One of the foremost causes was forgetting to re-start OpenVPN after it had closed down because of changes of network or shutdown of the device. The general approach in France was to collect network packet traces from each device. To avoid developing logging software for every platform, we installed a VPN client on each device to route all device traffic via our VPN server where we ran tcpdump to capture the first 300 bytes of every packet. The VPN server forwarded all HTTP (port 80) traffic through a Squid proxy to log all HTTP requests. A number of issues arose with OpenVPN. For a start, the OpenVPN iOS client ceased to be free after the study had already started and the data collection facilities set up. As indicated, OpenVPN often failed to set up the tunnel automatically upon connection to a network for OS X and iOS devices. It also failed to route all traffic via the tunnel for Windows machines. This issue of depending upon users to maintain the logging software and its connectivity resulted in a modification of the logging strategy for the later UK study. Aside from this, data was also lost because of users changing device over the course of the study or, in one case, borrowing a device from a relative whilst their own device was being repaired. For a couple of the participants a lack of data reflected the fact that the chosen device was actually one that they did not often use.

In deliverable D2.2 we outlined an ethnographic approach that would involve visiting each of the study participants twice after they had been registered, once at the mid-point of the deployment and once at the end. This happened with 4 of the participants in the end. 6 others were only interviewed at the conclusion of the study, 1 had to be interviewed over Skype, and
1 participant went overseas for a significant part of the study and was never interviewed at all. Additionally, whilst the ideal length of deployment was 6 weeks, most were active for 2 months, and one participant ran the software for a total of 5 months. These variations were a practical outcome of variability in participant availability for interview and the logistics of managing ethnographies in a remote location and all of the costs thus incurred.

One further element that was subject to variation was the use of Google Calendar to keep diaries for self-reporting during the study. All of the participants were set up with a unique calendar for this when they registered to participate. In the event, however, only 3 participants made any entries and even then, in a very restricted fashion. This feature of the study was therefore left out of the later UK study.

4.1.2 The UK Study

As was planned at the time of the preparation of deliverable D2.2, a similar study to the French study was conducted in the UK that ran from the end of November 2015 until the end of January 2016. In this case nearly all of the participants were recruited through existing contacts as opposed to using an independent recruiting body such as INSEAD in France.

The study gathered data from a total set of 10 households and comprised 14 individuals and their devices. A notable difference from the French study was that several households registered to the study with more than one individual. All of the households were located around Nottingham in the UK. Participants here ranged from families with young children to older couples and included people from both blue and white-collar occupation groups as well as students and retirees. Actual household composition varied from one and two parent families of varying sizes to couples and single-occupants. The study also encompassed a range of housing including both semi-detached and terraced homes to flats and even social housing. 7 of the 10 households joined the study using laptops, with 2 of them signing up with 2 laptops rather than 1, making 9 overall, 8 of them Windows-based and just 1 Mac OS. All but 1 of the households included a smartphone, 3 of them with 2 rather than 1, making 12 phones in total, 6 of them iPhones. 5 households registered tablets, 2 of which were iPads. Another important distinction was the use of desktop computers, which were absent from the French study completely. 3 households signed up with desktops, 1 of them with 2.

As indicated above, several important lessons had been learnt from the studies undertaken in France. This led to the incorporation of some new technical features to underpin the logging of user activity. A key part of this was a revision of how to capture data via VPN and the use of an improved version of HostView. In the UK study we switched to the native VPN client for each platform (OS X, iOS, Android). Both VPN servers were configured to provide a fixed IP address to the devices, assigned by the registration application, for identifying the participants from the network traffic and HTTP logs. On Windows devices we used the new Windows version of HostView to collect raw performance metrics, application information, and explicit user feedback on network application performance. Because HostView collects packet traces with libpcap (capture limited to TCP & IP headers and full DNS packets) and HTTP requests using a custom packet parser no VPN is required. A further distinction to be made between the French studies and the UK studies was the addition of a Technicolor gateway in some of the households in the UK. These were designed to act as an AP behind the user’s gateway and were put in place to gather additional wifi performance metrics. One other noteworthy difference between the deployments in France and the UK was that in the French deployment the
OpenVPN server logged the public IP of each connecting client, whilst in the UK we already had this data for Windows devices with HostView.

Methodologically the actual study process was more or less identical in both the French and the UK studies, with nearly all of the meetings between participants and the ethnographer taking place at the participants’ homes. An initial visit was mostly concerned with explaining the data collection setup, their rights to pause collection or withdraw from the study, and the installation of the monitoring tools on the selected devices. In subsequent visits participants were asked to provide a short tour of the existing technological arrangements in their homes and were invited to talk through their routines and discuss how their technology use was embedded within them. Participants were then asked to review the data gathered so far with the ethnographer using the online visualisation tool discussed in deliverable D2.2. This provided an opportunity for them to identify and explain patterns and exceptions within the data and to construct activity labels for certain kinds of network traffic. The visualisation also presented a timeline of browsing activity, from which individual URLs could be identified. For the UK study the timeline was cross-referenced with a map that showed where the participant was at the time. This information could be viewed on a per device basis, or combined together. A further feature of the UK study was that three participants, chosen on the basis of being thought most likely to do the exercise, were asked to look at the visualisation prior to the interview, and to record their thoughts. This allowed us to gather some data on how, in ‘best case scenario’ (i.e. those most likely to do so), people might engage with representations of their use data in the absence of a mediating party (in this case, the interviewer).

The participants also looked at a rudimentary categorisation of the traffic to assess how closely this reflected their own understanding of their online activity. Here a visualisation presented a categorisation tree of the participant’s browsing. The groupings for this were generated by Alchemy, a natural language processing algorithm. In the UK study this was also used to filter out network-generated activity (largely advertising, tracking and OS related) from the participant’s data. Whilst the visualisation interviews did generate data on the means by which participants made sense of their data, the effectiveness of the visualisation itself was hampered by the limited success of Alchemy in generating categorisations which were recognisable to the participants. Another constraint was the amount of network-generated activity that made it past the filtering. The poor performance of Alchemy was useful in sensitising us to the challenges of conducting this data cleaning with algorithms, but it also meant that the visualisation interviews were constrained by the need to work around these limitations.

In some cases, participants were also asked to look at profiles and to assess the extent to which the profiles matched their circumstances and their patterns of use. In France the profiles discussed were those already held within the user’s advertising settings for their Google accounts. In the UK the profiling exercise was used as vehicle for moving beyond the limitations of the Alchemy-based categorisation tool. Here we decided to run a follow up study in which the participants were presented with hand-made profiles, which used their data to make a series of inferences about their daily activities and their interests. This included detailed manual filtering to try and remove network-generated activity. This was a far more processed cut-through of the data than the cleaned and structured visualisation of the largely (limited filtering excepted) raw data used in the original study. This not only gave us insight into how the task of sense-making changed when the data had already been processed by hand, but also
provided us with a sense of the work required of a third party (the profiler) to try and extract sense from the data. Due to the time available, this study was limited to four of the original participants: one single student; one single working adult; one family with a pre-teen child; and one retired couple. As with the original study, participants were interviewed during a working session as they went through the profile with the interviewer.

Across all of the studies, once the study had concluded the ethnographer assisted each participant to remove the logging software from their devices and return all of their settings to their normal state.

4.2 Technical Data Sets Summary

![Bar chart showing total participation times per participant across the two deployments.]

*Figure 7: Total participation times per participant across the two deployments.*
Figure 8: Total amount of captured network traffic per participant.

Figure 9: Traffic type distribution (as fraction of bytes).
Figure 10: Location tracking coverage.

Figure 11: Unique visited locations.
This section provides a short summary of the collected technical data from the two setups. Figure 16 shows the participation time for all the study participants in France and in UK. We calculate the participation time as the time from the first captured network packet or HTTP request to the last. The participation durations vary as a result of individual schedules, and in some cases, due to technical problems with the data collection as explained above. The amount of observed traffic varies across users (Figure 17) and appears not to depend on the data collection, time indicating some high-level differences in the Internet use among the participants. The large majority of the observed traffic in terms of bytes (Figure 18) and flows (not shown) is HTTP(S) traffic (identified based on a destination port 80 or 443). There are few noticeable exceptions in the French setup are due to VPN routing problems resulting in only DNS traffic being sent to the VPN. Figure 19 shows the proportion of time for which we obtain a location for the participant according to the Moves data, as well as the proportion of those in which the participant’s devices were using the network. Both in France and in the UK, the coverage for active Moves users is between 50–90% of the total participation time; inactive users either did not install the app or ran it for a short time only. However, located network activity is relatively rare, at <20% of total participation time for every participant. Similarly, Figure 20 depicts the number of unique locations visited, and in which the network was active.

4.3 Ethnographic Results

In this section we present a range of findings from the user studies that are articulated around the core interests of UCN and the development of the platform. They are organised around four key concerns: What has been learnt that is of specific relevance to the development of the Personal Information Hub and the associated architecture?; What has been learnt about appropriate ways of collecting data that can populate the PIH and be made use of in the world?; What has been learnt that is of specific relevance to the tailoring of the platform in ways that will support service personalisation?; and What has been learnt that is of specific relevance to the provision of privacy and security mechanisms within the platform?. Taken together these findings underpin and support the directions adopted by the technical work-packages and the key decision to structure the architecture around the provision of a Personal Information Hub.

4.3.1 Collecting Personal Data

Of course, before data can even be managed or shared via the PIH it has to be collected from various sources and assembled in ways that might be used. Generally this could be seen to sit under the rubric of ‘detecting context’, though in practice other matters can also be seen to be relevant here, such as detecting patterns, routines, and recurrent practices of various kinds.

4.3.1.1 Detecting Context

Now it is clearly the case that some aspects of what may be termed ‘context’ are open to being ‘seen’ or ‘inferred’\(^5\) by machines. Of specific relevance here is the fact that current network technologies confer the ability to detect various contextual factors in relation to use of the Internet. A number of candidate phenomena open to detection were exposed over the course of

---

\(^5\) Scare quotes are used here to reflect the fact that terms such as ‘seeing’ and ‘inferring’ are descriptions of human rather than machine-based perception and reasoning and must therefore be treated with caution.
the studies in the UK and France, some of which we were able to exploit, some of which we weren’t. These included:

- The use of specific devices. For our study, if devices were registered to the study, they were identifiable.
- The use of fixed devices, e.g. desktop machines, in already established locations, e.g. the living room. In the French study no users registered with desktop devices, though one laptop was solely used in a single location.
- The use of specific applications. This was possible for some devices if they had installed HostView.
- The use of hardware on the devices, e.g. video, audio, etc. This was also possible for devices using HostView.
- Times of use for devices, applications, hardware, etc. All data coming from HostView and OpenVPN or the native VPN was time-stamped so obtaining this information was trivial.
- The location in which a device is used. Detecting location, like time, is a relatively trivial task, though with lower resolution and reliability. For our purposes location was chiefly obtained for mobile devices through the installation and logging of data fromMoves, as indicated in deliverable D2.2. Moves is a popular mobile activity tracking application where the user location is recorded using GPS (or, if GPS is unavailable, network-based geo-location). Additionally users can label frequently visited places (e.g., Home, Work). Moves is available for Android and iOS, and it provides an API for third-party applications to retrieve user data from the backend. The Moves Daily Places API was polled every 30 minutes for each device. Each response consisted of an array of dated place segments, i.e., the times that a device entered and subsequently left an area defined by latitude and longitude, with an optional location tag. We complemented the Moves location data with public IP based geo-location using RIPE Stats APIs. In the French deployment the OpenVPN server logged the public IP of each connecting client. In the UK we had this data for Windows devices with HostView. The IP based geo-location data was collected offline after the study.
- Transitions within a personal ecology of devices, that is, people moving to using one device rather than another out of a set of devices that are clearly assigned to them, e.g. from a smartphone to an iPad or a laptop. Some device switching was imputable during the study, but involved the comparison of data streams, which, for us, was a manual process.
- Changes of location, both between sites and within sites. This was more straightforward for mobile devices with location services turned on, and was captured for a number of participants (in terms of changing sites) through the use of the Moves application. Movement within sites was not captured, though there are a number of ways in which fine-tuned switching of locations could be sensed, given the right technology.
- The use of devices in specific types of public setting, e.g. cafes, libraries, airports, etc. Again Moves was able to provide for some of this within our study, though the actually labelling of settings required manual input and this was not something that many
participants undertook. Identification of various settings therefore involved manual inspection.

- Network availability. This was available where network performance was actually being monitored, though only in terms of whether a network was being used or not. The range of networks offered to a device in any one location was not being captured.

- Actual network use and network data, including network performance. Network performance was not always the focus of the data collection, but information about network use and the domains accessed via the network formed the core part of the dataset gathered for our studies.

- Light, temperature, humidity, motion, noise-levels, etc., in other words, traditional sensor-based data. Sensor data was not captured for this round of user studies, but is clearly open to capture by a variety of means and forms part of the data collection portfolio offered by other partners within the project. However, the routine capture and use of sensor data is currently often viewed as problematic due to the manner in which such tasks are accomplished: typically by remote third parties with the data stored on their own servers. Approaches such as UCN’s, which places ownership of data in the hands of those who created it, address a range of issues relating to the use of sensed data, including rendering the data intelligible (see [5] and [6]), and the privacy concerns raised by the use of such pervasive sensing.

- Use of (proximity of) IoT-registered devices. Once again this did not happen in the studies reported here but is another feature of the data collection portfolio amongst some of the project partners.

- Times of other events, e.g. changes of location, sensor-based events, etc. Wherever logging of location or sensor output is happening this is equally trivial to capture.

- Activity, and its time, location, etc. Some kinds of activity are open to being inferred from the specific use of certain applications or the visiting of certain sites. The latter was part of what powered both the initial categorisation process and the later profiling exercises that were conducted within the studies, with varying degrees of success. It was also open to ‘detection’ on the basis of initial tagging by participants.

- Routines, as a set of recurrently logged sequences of phenomena. Whilst patterns of activity were visible within the data, the actual labelling of these as routines was subject to manual input on the basis of human reasoning. However, some routines might be open to crude forms of machine inference on the basis of initial input from users, in a similar way to the tagging of activities. Routines are also open to description in machine-based (rather than commonsense) terms, such as patterns of network activity, sites visited, sensor inputs, etc.

- Assemblages of things and activities, as a set of recurrently logged co-occurrences of phenomena or co-occurrent placements of things. This was not attempted within the studies we conducted outside of the manual profiling process, though some basic detection of things like this is possible (see If This Then That or Samsung’s ARTIK platform). Beyond this, some machine-based inferences might be possible here, using the same kinds of resources as were outlined for the detection of activities and routines.
• Use by specific people. Whilst this is hard to detect with absolute certainty, the use of certain devices and in certain ways, or the use of sensors to detect presence in habitual locations can allow for a certain amount of inference. In our studies it was assumed that the devices being used were being used by the people who had registered those devices when they first started to participate in the study. By and large it was found that this assumption held.

4.3.1.2 Ethnography

**Ethnographic interaction**

One of the key concerns in work package 2 of the UCN project has been to explore the collection of data about people's activities online from two highly distinct perspectives. The first, as outlined above, was concerned with trying to gather as much data as possible that might be somehow indicative of a user’s context by logging information generated by technologies and the networks to which they are connected. However, we also made use of an almost diametrically opposed strategy for acquiring data about context: ethnography. Here the point is to arrive at an understanding of the context in which data has been generated through direct human interaction. This is useful as a means of assessing: a) what is being accurately and usefully logged by machines; b) the extent to which the data logged is adequate in its own right for the purposes of actually understanding what the data means; and c) what has been completely missed by the data logging mechanisms put in place. In order to set the scene for the further discussion of our findings below it is worth briefly emphasizing what happens when data is being collected through interactions between participants and ethnographers, because this is fateful for what comes out of the exercise and the situated character of data collection cannot simply be set aside. In particular, it is worth reminding ourselves here that the point of proceeding in this fashion was to place someone in the position of wanting to make sense of people’s personal data in order to make use of it in some way. In this case it was an ethnographer engaged in a research exercise, but it was argued from the outset that the ethnographer would be standing in the position of being a proxy for a range of possible 3rd parties who might be wanting to make use of the data for a range of possible purposes, including commercial transactions and the tailoring of services. Our aim in that case was to try and understand what it takes to be an agent operating in this fashion and to see how that might help to inform the future constitution of such agents – real or digital – working as intermediaries between a user’s personal data and the parties seeking to use that data in some way. Whilst the PIH has not been formally assigned the role of an agent in this way there are clearly certain aspects of how it will need to function that will be holding this kind of articulating position between a user and the outside world.

Something important to note here, in that case, is that the accounts provided by the participants in interview were part of an unfolding sequence of interaction between a researcher and a study participant. This sequence has an accountable form that cannot be simply set aside or overstepped. For the French studies it was typically organised in the following way:

- Greetings (at the door or via interphone)
- The participant bringing the ethnographer into the house
- Locating where the interview will take place
- Setting up of devices and recording equipment
• The ethnographer providing a preamble regarding what will happen
• Some general chat about technology and the routine
• Looking at some specific data (itself implicated by prior examination of the data and the designation of target features to discuss)
• Pulling out of features occasioned by looking at the data for further discussion
• Looking at the categorisation tool as a new topic with similar occasioning of talk
• Looking at the user’s Google profile as a new topic (this was an instructed process (again occasioning specific aspects of the talk))
• (For the final visit) Uninstalling the software as a final topic
• Leave-taking (of a standard form but with insertion sequences regarding implicated matters such as further contact or the process for reimbursement)

Within each of these interaction-based activities there are existing ways of doing things that have to be managed in the locally unfolding sequence of talk, e.g. changing topic, indicating a specific feature of interest, providing explanations, indicating surprise, questioning and answering, formulating what has been said, and so on.

Capturing online activity through interaction
Obviously the point of conducting the ethnographic work was to encourage the production of an alternative representation of a participant’s online activity to that being acquired through the technical process of logging. Unsurprisingly the outcome of this was a socially grounded representation, the implications of which we shall be pursuing more closely in Section 4.3.2.

Something that was commonplace in all of the interviews was some generalisations on the part of users about the habitual characteristics of their technology use and how that might shape what their online activity looked like. So something that most participants talked about was what kinds of things they do on what things and in what places:

Example 1:

C: In fact most of the time, when I get up I come straight in here [her office] … But when I go to bed I take my laptop with me to bed to look at stuff … The phone I may use in the kitchen, but the laptop-
I: That stays here?
C: Yes.
I: Or in the bedroom
C: Exactly.
I: So, the phone you might use anywhere in the apartment?
C: Yes.
I: Okay. Erm. And are there different things you do in the apartment with the Macbook and the phone?
C: Yes, for example, all the applications like Instagram- … Facebook Messenger, things like that are only on the phone. … Most of the time for messages on Facebook for example

Another thing that was often spoken about was what actions occur in what sequence, e.g.:

* Note that all the interviews in France were conducted in French. The data is therefore presented here in translation.
So in the morning it’s my phone, not the iPad. First of all I look at my texts, and then it’s WhatsApp. After that I look at Instagram, then Facebook, and lastly I look at my mail.

A further thing that was frequently mentioned was what kinds of things typically go together, e.g.:

**Example 2:**

(Discussing how her external routine relates to her online routine – there are some regular dropouts at certain times of the day)

E: … I use internet pretty much constantly (laughing) … And you haven’t even seen the phone! (laughing more)

I: But the pattern, the pattern of times –

E: Yeah, because when I’m eating I don’t like eating with nothing so I’ll stop any internet and watch something while I’m eating instead. That’s why you’re seeing the internet drop out (Note that she downloads things in advance to watch (typically using torrents) rather than streaming because her machine doesn’t handle streaming well. So things to watch are stored on an external hard-drive that is usually connected to her machine).

### 4.3.2 Managing and Making Sense of Personal Data

One of the key innovations in UCN is the creation of a Personal Information Hub (PIH) that can act as an intermediary between users, their data, and the parties who might wish to make use of that data in various ways. In this section we examine the findings coming out of the user studies in terms of how users are oriented to managing their data and what others might make of it.

#### 4.3.2.1 Inferring Practice

A challenge raised by the capture of the kinds of data outlined in the preceding section is developing an understanding of what human actions we might be able to reliably infer from such traces. Even though it was purely grounded in network traces, the profiles we developed of participants would seem to suggest that time and location alone may be enough to identify a considerable quantity of actions. On the basis of the studies and our profiling work we have begun to develop a typology of practices. Our goal here is to aid in the identification of practices that might be targeted for network detection.

The following typology (see Fig. 21) is grounded in social practice theory (henceforth, practice theory). Practice theory locates the site of social action in the doing of the routine and mundane and, in its simplest formulation, separates practices into three elements which combine to give the practice its footprint: skills (how the activity is done); meanings (the purpose of the activity); and materials (the technologies involved). Within this framework a practice is understood to be a network of these elements that is maintained, undermined or reconfigured by alterations within them. It is anticipated that, as technology develops, more and more elements of a practice may become detectable.
Figure 12: A typology of practices.

Figure 21 is focused upon the sense-ability of user practices according to Internet browsing, time and location data. The types are ranked by detectability: the darker the shading the harder the task. Supplementary practices are a special case in that they may be detectable in their own right, but they are strongly tied to a parent practice (indeed one might categorise them as an activity within the parent practice rather than one in their own right), which determines if, when and/or how they happen. A key difficulty lies in detecting the relationship between the practices. As the typology suggests, the nature of habitual practices – reproduced again and again with high consistency – makes them the prime candidate for network detection.

4.3.2.2 Uncovering the Meaningfulness of Data

Understanding the data from a machine perspective

Another important challenge for UCN, that was previously articulated in deliverable 4.2 and that is further discussed in Tolmie et al, 2016, is that, whilst one may be able to capture contextual data in a technical sense, this does not necessarily give one any insight into how the data might or might not be socially meaningful. The following examples illustrate this issue:

Example 3:
I use the phone more than the computer because that can give the impression to other people that I’m not at home… Whilst when I’m on the computer it’s obvious that… I prefer to make my location more obscure (laughing) … Because for example if I want to end a conversation with someone I can say ‘Sorry, I’m about to take the metro, sorry.’… Or ‘I’ve got a meeting’

Example 4:
For example, in the morning when I wake up the laptop often doesn’t have any battery left because I’ve watched a film and fallen asleep and… So the laptop has no battery and when I wake up I look at my mail … on my phone

In example 3 technical sensing may provide for seeing when the user’s phone or laptop are being used, and even where they are being used. However, nowhere in that data would a system be able to uncover the rationale relating to when and where the different devices are being used.

Example 4 is in some ways more complex, even though the rationale is more straightforward. A developed system of technical monitoring would be able to capture battery levels and to build in correlations between available battery-life and device use by a user. Further monitoring might also uncover patterns of application use and the regular opening of mail on the laptop as a first feature of application use on the device after extended non-use between certain hours of the
night. Analysis of exceptions might then uncover the use of the phone for accessing mail at a similar point in time when the battery levels are depleted on the laptop. However, it is one thing to recognise the potential to engage in such correlations, but it is another thing to see that these correlations might be useful and understand how such correlations might be meaningful and might feature in a user’s reasoning such that you would prepare a system to look for them in the first place. This is strongly indicative of a need for certain amounts of user input if one is to identify ways in which to best steer a system towards the capture of meaningful aspects of a user’s context.

Understanding the data from a human perspective
A key feature of the user studies was that the participants were actually shown the data their online use had generated during the interviews and were invited to account for the data through a joint exploration of the data using the various tools provided. As described in deliverable D2.2, this was adopted as a deliberate policy in order to provide participants with a means of displaying how they might work to manage their data when it is exposed to a third party. It was clear that, as a matter of course, participants sought to articulate what could be seen through the data, as someone who might be able to see what the data is saying really. Although some measure of this was about being the person who had generated the data in the first place and being therefore possessed of a specific expertise regarding what the data could say, it was also very much about being a human being, possessed of social competence, and able to engage in common sense reasoning (i.e. what you know of what anyone like this might reasonably do).

As indicated above, the interactional organisation of the sessions meant that arriving collaboratively at an understanding of the data involved passing through several distinct phases. First of all there was what might be termed ‘the work of preparation’. The sheer volume of data generated during a study meant that not all of it could be discussed. Specific features were therefore identified for discussion. Arising candidates here included: any visible pattern, e.g. a routine; and visible anomalies, such as inactivity, exceptional activity, or unusual times of use. After this there was the work of presentation and explanation, which might involve: bringing up the tool as a topic in its own right; explaining what it is intended to do; and demonstrating its workings. The work here typically took certain things for granted that were not taken to be accountable in their own right by the participants, e.g. graphical representations of measures of some phenomena over time. Finally, there was an unfolding working through of the data (discussed in more detail below). Both parties handled this, though the interviewer mostly drove it. It amounted to a working up together of what an appropriate account of the things seen in the data might amount to. This often involved resorting to external resources and understandings (again see below).

Looking at the participant’s accounts of their data actually generated during the interviews, the details demonstrated again and again the process of making the data socially meaningful. The following list shows the predominant resources and things which participants’ drew on when making sense of a particular piece of data.

- Time
  - Visualisation Timeline – when did it happen?
  - Calendars – what was the participant doing at a particular time?
- Location
  - Visualisation Map – where did it happen?
- Hardware
  - What device did it happen on?

- Actions
  - Routine practices of the participant
  - Routine practices of close ties (family members, friends, colleagues)

If the participant was unable to make sense of the data using the above reference points, they would commonly abandon the task, and move on to the next piece of data. On occasion, or if pushed by the interviewer, they would draw on the following to explain away the apparent anomaly.

- Network activity
  - Something happening in the background

- They must have been using Facebook (or other social media)
  - This was seen to be a potential source of non-routine actions and thus a possible source of unfamiliar results in the data

- Misclicks
  - Equally possible as a source of non-routine actions and resulting unfamiliar results in the data

The thread running through these lists is the routine and mundane aspects of participants’ daily experience. Time, location and hardware are all means of establishing means of embedding the logged data in participants’ repeated patterns of actions (practices) and those of the people close to them, who occupy the same social milieu and have a shared orientation to the organisation of their activities. By locating the network data within their nexus of practices, and explicating its links to surrounding practices, participants sought to articulate the meaning of their data as just another part of their ordinary everyday existence. It was only where this proved impossible that participants would pursue a strategy of seeking explanation of things as being about non-routine action. However, it is noticeable that there is, in itself, a set of routine resources for accounting for the apparently inexplicable. Prime amongst these are: Facebook – where people often receive novel webpage recommendations from contacts; individual errors; and very occasionally, when nothing else seemed reasonable, an anomaly generated by the network itself rather than any of their own actions. Two participants used this strategy when explaining activity in the logs whilst they were asleep:

**Example 5:**

I Oh, so yes, that’s one thing I noticed, this [network activity] goes all the way through the night for a couple of days.

P3 Hmm-mm. She doesn’t sleep.

P5 I, I, I have had difficulty sleeping.

I Yes.

P5 But no, I mean, there is no reason. I presume that if I’ve left an app open and my phone is charging on Asleep and the screen has gone to sleep…

I Yes.

P5 The app is still running in the background, isn’t it?

I Yes so it could well be doing stuff.

P5 So the app’s, you know…
It should be noted that there were a variety of parties who were all able to articulate what the data might mean on the basis of common sense reasoning, with varying degrees of confidence. These included not just the person producing the data, but also people who knew them, the researcher, the coder, and so on. As a part of the overall exercise some participants were also invited to see what sense they could make of the data on their own, without the researcher or any other notional ‘expert’ as a resource to assist them. Here we were particularly interested in discovering what sense they were making of how the machine might understand them.

Similarly, computer scientists involved in the project were asked to see what sense they could make of the data without recourse to the actual participants and thereby local understanding or accounts. On the basis of this the computer scientists then constructed user profiles to be played to users so that we could assess their reaction to this external reasoning about their affairs (see the discussion regarding Profiles below). The relevance of common sense reasoning to the explication of logged data, as opposed to what might just be seen in a technical sense within the data itself, was demonstrated forcibly by this latter profiling exercise (see section 2.4). In the interviews themselves the ethnographer was thus effectively being positioned as an agent in all of this, mediating between the coder and the participant.

**Understanding representations of data**

It was noticeable that there were distinct organizational formats to each tool-based sequence that shaped the way the interaction could unfold. These are detailed in turn below.

**The data visualization tool.** Within the data visualization tool different domains were represented as unique vertical bars beneath the time-rate graph. This particular representation meant that the ethnographer and participant would work through what happened retrospectively as a sequence of action, surfacing the frequency of specific site visits/occurrences, with candidate URLs open to being examined in detail.

In the following exchange, we see how the meaning of a particular URL (skype.com) is reached through the collaborative working up of an account, with input from the Interviewer [I], and a wife [P5] and husband [P3]. They are looking at the timeline showing when this address was accessed. Note how the parties draw on both the data and one another’s accounts in working towards a shared understanding. As they close in on an agreed account the confirmations of one another’s claims increases.

**Example 6:**

P3  Skype. Skype.com I don’t use Skype.

P5  No.

I  That [the timeline] was very busy on Christmas Eve.

P3  Christmas Eve, you were at home so Sam [their son] would have been on it quite a lot.

P5  Do you think he’s using Skype?

P3  No.

I  I did have someone [another participant], the same actual pipe.skype [URL], who, um, they also don’t use the Skype so that might be an advertising thing or something.
The other notable feature of this exchange is how the social and moral organisation of the participant’s own everyday life together are drawn on to establish meaning, including in this account the family’s routines over the Christmas holidays, and their ideas about what reasonable Skype usage looks like (i.e. not six hours of solid use). We also see unspoken assumptions regarding how their experience maps on to the data, so the ‘clump’ of Skype URL requests is understood to amount to 6 hours constant use by a member of the household, whilst the number assigned to the Roadblocks URL (977) is remarked on twice, even as an exclamation, but without any clear understanding of what this number is a count of.

The role of the interviewer is vital to how these kinds of sequences of interaction unfold. For a start, in order to aid the participant in engaging with the data, the interviewer adopts a strategy
of translating elements of the data into a syntax recognisable to the participant. URLs become website visits, clicks activities (e.g. shopping). Locations on the map become places. Once the data has some legibility, the work becomes a matter of going through in a somewhat ad hoc basis to establish whether data is of interest, and if so what form that interest should take.

A lot of the time establishing whether data was of interest consisted in large part of discarding URLs which were agreed by interviewer and participant not to be directly related to the activities of the participant. Although some filtering was in place its limited effectiveness meant that a lot of the presented data was generated by network activity, such as advertising and the provision of services (for example, apple.com was commonly the most requested URL on Apple devices). Meaning was developed through the generation of accounts between the parties involved, using a process of satisficing to reach an account which was agreed to be plausible enough to adopt. This process is necessitated by the gap between what the human participant experiences on the one hand, and what the network records on the other. Whilst a participant visits a webpage, the logs might record the network drawing on a dozen URLs to provide the resources to build that page. The problems this gap creates are captured in the exchanges between participant and interviewer, here discussing one of the visualizations:

Example 7:
I How, how did you interpret the… I mean so the X-axis is pretty obvious, but what about the Y?
P2 Um, I did wonder what the units were.
I Yeah.
P2 Um, so I don’t know whether that was… it could have… I figured it couldn’t have been number of websites visited or something like that. Um, but now I look at these numbers and they’re sort of quite high numbers so I don’t know. Maybe individual times I’ve been on… well, the size or the amount of data or?
I Yeah. So, I think it’s the number… so, um, yeah, I’m not even technically… it’s basically the number of connections that are made.

This example shows how the gap problem is managed: by both parties agreeing to elide the issue by using the network logs heuristically, rather than didactically. Here it is agreed to treat the Y-axis as a relative measure of times the site was visited, a solution which is treated as adequate by both of them.

Interestingly this experience gap also provides for toleration of ambiguity where network activity leaves traces in the data. These URLs appeared in the logs of their Internet use, yet they were not recognised by the participants. Only very rarely did a participant suggest an account which dismissed these traces as being the work of background processes rather than their own activities directly. Instead it was reasoned that the participants themselves had in some way visited the site without being aware of the URL. A number of the participants independently arrived at an account in which they had clicked on a link posted to Facebook by a contact, without registering what the URL of the link was.

Example 8:
P2 I linked to that by accident maybe, you know. Um, if I saw something on Facebook and I just clicked on it I’ll say oh that’s a lot of rubbish or, you know… I didn’t actually read it.

In the rare examples where a background process was identified, it was still framed as an action they had themselves undertaken:
Example 9:

P3 And so if you’re searching for something and then you get the Google ads come up at the top you click on one of those ads, that’s what that would be.

Tagging activity. Early on in the deployment of the package for logging user activity in France functionality was included that allowed users to tag their online activities for themselves. This preceded later attempts to include machine-based categorization of sites visited within the visualizations shown to user during interviews. The initial interface for selecting tags proposed a small number of basic tags that could give users an idea of how they might label their activities. These initial labels were: work; video streaming; social; shopping; research; news; hobby; health; gaming; finance; family; and entertainment (see Figure 22 below):

Thus, if users were regular visitors to Amazon to do online shopping, they could tag visits to Amazon as ‘shopping’ and every visit to Amazon from that point onwards would automatically be labelled ‘shopping’ within the data. The labelling of the initial tags was informed by an earlier project\(^7\) that had examined people’s online activities and found that they tended to commonsensically articulate the things they did in these terms. During earlier interviews, in that case, part of the visit was dedicated to opening up the tagging interface, explaining it to the users, and encouraging them to do tagging of some representative parts of the data to seed the future tagging of their activity. Additionally, users were able to create and assign their own tags if they found that the existing tags did not properly express what they were doing when they visited a site. Once data within a set area had been tagged users (and the researchers) were then able to display an overview of activity for a chosen period in terms of what had been tagged.

---

Figure 14: A participant examines the displayed results.

Whilst this procedure clearly allowed for the possibility of users labelling their online activities in ways that made sense to them, it presented a number of issues. These issues were part of the reason for subsequently moving towards machine-based categorization and discussing this in interviews instead. Here is an outline of some of the principal issues uncovered:

1. The work of tagging
   One of the key difficulties encountered with this exercise was that users clearly found it laborious. Even a small selection of data from within the overall online activity that was logged covered a large number of sites. Inevitably working through the list of domains thus generated and applying tags to the domains that were recognized took a long time and users were torn between ‘getting the job done’ and an obvious fatigue arising from the repetitiveness of the task”. In the following example the user has already been tagging domains for some time, which are displayed in alphabetical order by default, but she is still only near the beginning of the alphabet:

   **Example 10:**
   C: And all of this is only for the 16th? Really? Just the 16th of April?
   I: Er, the 5th I think. It was Easter. It was Easter day.
   C: Wow! Just one day and all of that!
   I: Yes. Yes, yes- Ah, No. It says [the visualization] the 6th. It's the day after in fact.
   C: (We'll never be able to finish it. Look at all of that!)
   I: No, it's the day- the day after…
   C: It's too much!

   It is almost certainly the case that any effort the participants were prepared to devote to the task arose from a sense of accountability to the researcher who was sat with them. It is unlikely the same degree of assiduousness would be present without it carrying such a sense of interactional obligation.
2. Network dependency
A further difficulty was that the capacity to tag turned upon there being a stable link over the network to the server hosting the visualization tool. In practice this link proved to be of variable quality. Users were thus often obliged to exit and re-enter the visualization interface, then re-locate the specific period they were interested in tagging, in order to progress with the activity. This made an already laborious process even more onerous to undertake. The following shows one such case, where the process of tagging is interrupted mid-stream, and the participant is obliged to log back in again and find where she’d got to:

**Example 11:**
Clicks on ‘dpstream.cc’ and selects ‘video streaming’ from the drop-down menu and clicks on ‘TAG’
Waiting for it to register
Clicks on the domain name again
The list registers the tag (‘dpstream.cc | video streaming’)  
I: Ah, that’s worked.

C selects next item (‘dpstream.net’), and clicks on ‘TAG’ (video streaming still selected in drop-down menu)
Waiting for it to register

C: Isn’t it doing anything?
Clicks on domain name again and goes to click on TAG but the tag appears in the list just before she does so
Selects ‘dpstream.tv’ from the list and clicks on TAG
Waiting again

…
Tag appears in the list

…
C selects ‘dropbox.com’ from the list and goes to the drop-down menu

C: Dropbox is er …

I: In fact it’s, it’s- (C moves down to work in the dropdown menu) Yes, it’s work, perhaps, it’s-
C selects – waiting for tag to be assigned

C: I’ve got lots more video streaming than other things, I think ((laughing))

I: Yes. Or shopping in fact. Lots of shopping too
Tag appears against dropbox in the list

C: Ah, voila! It’s there.
Scrolling down the list

C: Ur- e-bay
Clicks on ebay.com in the domain list and moves over to the drop-down menu
Scrolls up and down the dropdown menu

C: Oh, shopping (selects) It’s more for looking than buying. It’s almost like a hobby…

I: Yes, it could be. In fact it’s always like that when choosing categories. You can always see loads of possible categories you might use for things
Still waiting for tag to be assigned
Scrolls down the list a little and hovers over the tag button

I: What’s going on? Has it frozen?
Wait a while but still not assigned

C: No, it’s not doing anything

I: No, looks like (C clicks on ebay.com in the domain list again) it hasn’t worked this time.
C clicks on TAG again
C: oh, this is going nowhere
   Login screen reappears instead
C: Ah, I was disconnected.
I: Ah. Okay.
   Keying in username and password
   Being prompted to say whether she wants to save the password
   Clicking on 'no'
   Selecting link to Dataviz
   Admin interface comes up
   Selects link for her machine as displayed against her household
   Displays graph for April 6
   Selects area on top graph so that it's showing the same period as before, then clicks on Tag
   Activity
   Scrolling down list
C: I was at-
I: D I think
C: D
I: Yes
   She scrolls down
I: In fact ey- E I think. Yes, yes. That's right.
   Re-selects ebay.com on the domain list

3. What is a ‘representative’ segment of the data?
   Another prospective issue with the tagging exercise as it was undertaken is that it was premised in
the first instance upon having selected a period on the graph that would encapsulate a set of
domains that might be somehow ‘representative’ of the user’s online activity. For the reasons
already indicated above, making the work of tagging tractable within interviews involved the
selection of periods on the activity graph that exhibited concentrated access to a relatively large
number of domains so that users would not be obliged to range too widely across the data.

   Now one immediate issue here is that there is an assumption within the choices being exercised that high activity periods will surface ‘representative’ activity. The notion of ‘representativeness’ in play here was very much shaped around an interest in uncovering the kinds of things the user was doing online on a regular basis, as a matter of habit and routine, or in relation to clearly identifiable patterns of use of some kind. However, it is clear that there are no good reasons to expect that the domains associated with this kind of use will be any more visible in highly concentrated periods of activity than they would be as a spread of activity across the entire dataset. Thus the domains being uncovered for tagging were centred on the practical requirement of making tagging tractable, rather than what might be more robustly argued to be representative. This being the case, the working up of the domains being uncovered as being somehow representative of use was something that happened collaboratively within the interview, for instance through questions designed to probe whether this was something the user did ‘often’ or that this site was one they were visiting ‘on a regular basis’, etc.

   In some ways the above elicitation technique is not especially problematic and reflects the way that the gathering of data about users is ultimately a practical exercise that has to conform to the nature of the setting and the ordinary constraints of time, access and user patience available. However, in the absence of the research having been done and relevant patterns and routines identified, any part of a user’s data, at first sight, could be potentially representative of
something. Indeed, all of the data might be argued to be representative somehow, even if only of what a user’s activity will look like given the particular set of circumstances in hand. In view of the fact that systematically tagging all of the data was not a reasonable proposition, some selection had to be made. But such selections are always fateful and the ‘representativeness’ of the materials being tagged therefore has to be understood in these terms.

4. Pre-seeding with putative tags
One of the most significant issues uncovered related to the pre-seeding of the interface with some potential tags. Whilst the intention here was to instruct users regarding what kinds of labels might be applied, what actually happened was that the pre-specified tags were oriented to as a set of proposed defaults. This meant that users would most often look to see which proposed tag might most adequately capture the kinds of activity they would engage in when visiting a particular domain and select this as the tag. If the actual activity could be reasoned about in ways that would make it fit within a specific default, this is what was chosen, rather than a new and potentially more accurate label being applied.

It was also the case that, as the scale of the task became apparent and its realization became increasingly repetitive, so it was that users would make use of anything that was halfway reasonable if it was to-hand.

5. Generic tags vs local understanding and situated action
Clearly, if people are making use of pre-seeded tags to describe their activities another potential issue is the degree to which generic tags like this are adequate for capturing local understandings of what their activities really amount to. Alongside of this, a further issue riding upon the nature of the interface provided was that all that the user was able to see was a series of domain labels, which users were only able to recognize a fraction of within the interviews. The tags assigned were then being somehow selected as adequate to capture a sense of what was always being done when the user was visiting that site. However, other parts of our data showed that visiting any one site may be about accomplishing a wide variety of situated and not necessarily commensurate ends. Thus, an important question to be raised is the degree to which one dominant tag can capture the full range of activities that are being accomplished when visiting a particular domain.

6. Re-tagging
Despite the above reservations, it should be noted that a number of occasions did arise when users found that they could not comfortably capture what they were doing with any of the existing tags. Under these circumstances users did then make an effort to create tags of their own. In the following example we can see how users uncovered issues with the adequacy of the presented tags, how users sought to resolve such issues, and how they reasoned about what would amount to an adequate tag under such circumstances.

7. Ongoing refinement of tags
Once users had applied a new tag on one occasion they then found rationales for including other items under that category in the same way as they had with the pre-seeded tags. Thus we can see that, through its recurrent reapplication it is then oriented to as another prospective member of a set of adequate defaults that can be used to describe their online activities.

This would seem to suggest that some kind of bootstrapping of categories might be possible for users over the course of time and systematic reapplication and refinement of tags, with the description of activities encompassed within the tag becoming increasingly adapted to local
understandings of what those activities amounted to. However, it should be noted here that this does not solve the issues regarding the amount of work entailed in applying tags in the first place. More significantly, it does not address the problem of single-tagging of domains that was highlighted in item 5 above. Thus, one important refinement here would be the development of the capacity to apply a cluster of potentially relevant tags rather than a single tag to any specific domain identified.

Categorized activity. Another tool used during the interviews was a tree diagram that presented users with the results of an automated categorization of their online activity (see Figure 24):

![Figure 15: Using the categorization interface.](image)

The categorisation tool provoked a working through of things in an ordinal fashion. The percentages invited a looking at the highest percentages first and the smallest last, with a similar pattern unfolding in the nodes. Very small items with similar numbers were set aside altogether.

It was notable that, when presented with the categorization tree in the absence of supporting interaction with the interviewer, participants struggled to extract meaning from it:

**Example 12:**

I: So what do you think this is showing you, first off?
P: Don’t know.
I: No?
P: Just information, I suppose, really. Just information, I should think.

Something that is particularly invited by the categorization of user activity into overall types of activity is to see within it what they already understand their internet practices to amount to. In other words they find ways of making it comply with rather than challenge how they would
routinely characterise their use. In the two examples below, both participants are looking at the categorisation screen, which, in this instance, does not show any measure of quantity of use.

**Example 13:**
I: What do you think this is showing, overall, firstly?
P4: That I don't use it a lot.

**Example 14:**
P3: Yes, so how much I actually do on my phone was quite amazing. I, I, I think I take it for granted on how much I do search for stuff on my phone.

**Profiles.** In the French study a Google profile was undertaken by the participant under the instruction of the interviewer. Here the results were not open to further manipulation and resulted largely in affirmation of the rather general outline of their age, gender and interests. In the UK study, however, the profiling work was significantly developed. In this case a computer scientist was invited to examine the logged data for specific participants and to use that data to construct a profile of the user. The following figures show a snapshot of this kind of profile:
Go to sleep at around 10pm

Saturday routine:
Go to the gym or jogging (3km) for one hour.
After gym, go home just after 9pm and stay there for approximately 1 hour.
Activity varies after: Will most commonly go to the university campus (Around 1km)
Usually home by 6pm

Sunday routine:
Go to the gym or jogging (3km) for one hour.
After gym, go home just after 9pm
Activity varies after: Will most commonly go shopping in town. May go briefly to the university campus.
Usually home by 6pm

Activities
Going to the gym

Reading articles in Women's magazine

Reading novelty/humor sites

Online rating and monitoring sites

Online shopping
Multiple visits to amazon, carousell.com (3), ebay.co.uk (5), googleĭshop.co.uk (3), googleĭshop.co.uk (3), googleĭshop.co.uk (3), googleĭshop.co.uk (3), googleĭshop.co.uk (3), googleĭshop.co.uk (3)

Friends, Family & Relationships

Single & activity seeking a partner

Close friends: family member with child
Browns family split, school site and courtesette.com's clothes site, but routine (location) suggests no set of one

Several other interest groups
At Newcastle Hall / Raleigh Park / Rhedos Court

Aspirations

Corporate job
Finance / Banking sector, but also interested in consumerist / large retailers / multinatinals
Familiar with (or practiced) psychometric tests

Further Education
MBA / CPD: Many universities (Europe / America / Australia) looked at / considered

High end luxury
Investing in luxury items and destinations / products / home / car

Health
Very interested in fitness / health / prevention of health related problems

Interested in alternative health / natural remedies

Some interest in diabetes

Surgery / Laser eye surgery

Finances

Low to middle income
Use budget stores, op: Ltd, TKMax, Primark
Main method of transport is bus, coach (national expenses) for longer destinations
Use Low cost gym (as cheaper than offered by universities)
Presenting these profiles back to the users whose data had first generated them produced a number of interesting insights. Firstly, it was clear that the work by the profiler was successful in greatly lowering the processing overheads placed on the user, which had limited the effectiveness of the categorization exercise outlined above. The manual filtering of the data removed far more of the activity generated by the network itself than was the case with the users. This removed the need to find plausible accounts for this data. In the original exercise it was notable – as a consequence of their limited familiarity with networks and their technical operation – that users would default to attempting to explain all data as the result of their own activity: who else but they would be active in their browsing logs. As was indicated above, a number of users independently came up with the same account to explain unrecognised URLs – these must have been links posted on Facebook by friends, which they’d clicked to without being aware of the website involved. The manual filtering meant that far less effort was expended on these traces, leaving more time to identify activity that the user was responsible for.

This reduction in wasted effort leads to the second insight, which was that in contrast to the labour involved in working through the automated profiles, working through the manual profiles was genuinely interesting activity to the users. These profiles were a set of immediately legible claims about the user and their world. Partly this was the work of the profiler in removing extraneous (i.e. network-derived) activity, and partly it was the work they had already done in making the data socially meaningful. Where the profiler’s claims were correct they provided confirmation of ideas the user already held about themselves, or alternatively surprising insights into aspects of their life – for example the sheer amount of online activity
they took part in, or their movement patterns that, in their repetition, led one user to joke about how “boring” they were. Where the claims were judged to be wrong they provided humour, sometimes discomfort, and almost always interest in how such a view had been reached.

In constituting an enjoyable activity, this approach also provides an interesting contrast with previous efforts to incorporate user data into profiles. A number of previous attempts to engage in this kind of exercise have reported difficulty in getting users to participate. It is not coincidental that these studies have sought to engage the user in the profiling process through survey instruments. This is labour rather than engagement. These exercises ask for information from the user without necessarily providing anything much in return. Of course the approach used in our study generates its own challenges for implementing at scale, which we discuss below.

A third insight concerns the limitations that any third party profile will suffer from. This is the very thing which makes user involvement in the profiling so productive. As will be discussed further below, the work of the profiler draws heavily upon common sense assumptions they are able to make about the person they are profiling. Thus in one profile the possibility that the participant has a boyfriend becomes a means of explaining interests suggested by their browsing which would not otherwise have seemed appropriate for the kind of person being profiled. This notion is socially embedded, relying on the profiler’s assumptions about the interests held and activities conducted by the type of person the profiled participant is. The type of person the participant was in this example – female, young, in higher education – was not, to the profiler, the type of person interested in cars and cricket.

The data available to a profiler may lead to similar consequences. In one example, a participant’s location data revealed several brief (i.e. a matter of minutes) pauses at points in the streets around a location that was taken, by the time spent there, to be their house. Further investigation showed that several of the pauses corresponded with what Google maps identified as bus stops. This map then became to the profiler an account of public transport use by the participant. The subsequent interview with the participant revealed a very different account – that what these traces actually showed was a map of his dog’s Points of Interest. As he walked the dog around the local area the animal would stop at particular locations. Bus stops, which the participant identified as a haven for discarded chips and interesting smells, were a favourite. The fact that bus stops offered an explanation for the location traces and were made available to the profiler by the tools he had available, directed the profiler towards the account he settled on. As it happens the presence of a dog was inferred, due to web visits to a local kennels. There was, however, no direct link between the dog and the locations to be explained.

The dog example points to a systemic limitation of profiling based on an individual’s devices. Shared activities are difficult, if not impossible, to identify. Had the dog had an iPhone in the study there would have been a much higher possibility of arriving at the participant’s account. Less facetiously, this kind of problem remains for human participants sharing an activity who do have their own devices, unless the profiler is able to access both devices and cross-reference them. Even in situations where this is possible (as it was in the case of some of the families which participated), many coordinated activities have wider boundaries. In one example, the profiler inferred that the participant was planning a trip to India, based on a location trace to a building which handled visas for that country. It transpired that the participant was driving his girlfriend (who was not in the study) there to drop off the necessary paperwork. However, the visas were actually for work colleagues of hers who she was doing a favour for. It was they
who were travelling to India. Short of having the resources of a large state security actor, and
the reason to direct those resources on to a small group of individuals, there seems no way of
arriving at such an account without involving the users directly.

4.3.3 Protecting Privacy and Maintaining Security

As mentioned above, a body of relevant findings regarding how users are oriented to the
sensitivity of their data and the management of their privacy was first presented to the project
in deliverable D4.1. Here we use findings from the studies in France and the UK in order to
more specifically articulate those observations for UCN and the privacy-preserving
characteristics of the PIH. The interviews surfaced sensitivities across several different
dimensions:

4.3.3.1 What things are done where

One aspect of this relates to what might be visible to other people. Thus, amongst the younger
participants in the French study who were still living at home it was clearly the case that they
felt able to use the internet more freely in their bedroom than they would have in the living
room with their family.

Another aspect of this relates to demonstrably showing due ‘sensitivity’ to others, e.g. by taking
video-streaming and music to your room when you are sharing a flat and your flatmate gets
home:

Example 15:
When Z gets home I move to my bedroom from the kitchen. Or from the sofa. I've been working
on the sofa sometimes recently because it's more comfortable. But most of the day I spend in the
kitchen. But when Z gets home in the evening I move to my bedroom. That's about 7 o’clock. With
the sofa there's less of a pattern. I started sitting on it because before that I was sometimes sitting
on my bed and it was hurting my back. There was no particular time when I was doing that. Just
while Z was still moving around. But Z goes to bed very early. So then I'd go to my room so as
not to disturb her.

4.3.3.2 What things are done when

Something visible here was that both the participants and the ethnographer had a sense of at
what times of the day or at what points within their routine people might appropriately be seen
to be engaging in certain kinds of activity and, where this sense of what is ordinarily appropriate
was breached, either party might treat it as a topic in need account:

Example 16:
I: It seems that you go to bed quite late – right?
S: Yes, that's right
I: I noticed that in the data
S: It's about one o’clock usually
I: So you go to bed late but get up early as well. When do you sleep? Aren’t you sleepy in class.
S: Yes. It's disastrous.
I: Do you have trouble sleeping?
S: No, I just get back sleeping and have too much to do
4.3.3.3 What things are revealed to others you know

Something we have discussed elsewhere (Tolmie et al, 2016) is the fact that people generally exhibit the strongest forms of sensitivity about their personal data and its visibility, not to a world of unknown strangers, but rather to the people with whom they have the closest relationships. This proved to be no different in the studies we are reporting here. In the following example the ethnographer and the participant are looking at some periods of exceptional activity in the data visualization tool. They are both sat at a table in the living room and the participant’s parents are sat on a nearby sofa watching television while they talk:

Example 17:
I: Look at the time. This is after 3 in the morning
(Talking her through the pattern)
I: It's completely different to every other day.
C: When was it?
I: This is late at night – from a Saturday to a Sunday. If you look you can see you were using it through to late into the morning
(C casting nervous glances at her parents) …

4.3.3.4 What things are done at all

Some matters are sufficiently widely deemed accountable that user sensitivity is highly predictable, for instance going online to look at pornography. Here the ethnographer and a participant are looking through the different nodes on the categorisation tool together:

Example 18:
I: There’s also a little bit of pornography-
N: That’s not me that!
I: It's very small
N: Well the computer is there and on – So others could use it I suppose, but no! …

4.3.3.5 User articulations of the private and the personal

Privacy, when discussed in the context of the Internet, is commonly framed as a struggle between individuals on the one hand and major institutions, such as state security services or multinational tech companies, on the other. Such a frame does not reflect Internet privacy as it was formulated by the participants of the study, for whom privacy concerns were constituted in the concrete experiences of their day-to-day interactions, rather than in abstract fears of snooping spies or data-harvesting corporations. Indeed, the concerns participants had were often not even recognised as questions of "privacy", which, as a subject of discussion, rarely elicited much of a response. Instead, these concerns emerged in lived examples of managing what information to share and what to keep private, and where to do either. The sole exception to this picture was banking details, which several participants did cite as a concern when asked about data privacy. Again though, some of these participants had experienced fraud first hand so this was not so much a theoretical worry as a lived one.

One participant spoke of adopting an almost mute persona on Facebook. Many of his contacts on the platform were old friends from his days as a left-wing activist. Others were colleagues from his current role which involved liaising with the police. The overlap of friendship groups
from different periods of his life, as well as his personal and professional spheres, left him wary of saying things in one context which might cause offense in another. Facebook posed similar problems for another participant, an African student studying for a PhD in the UK:

**Example 19:**

| I | I think you said last time you’ve stopped posting anything personal on Facebook. |
| P6 | Yeah, I used to do lots of pictures and I used to... If you look on my last posts I just shared an old memory. Apart from that there was a bomb blast in [home country]. I did something for that. There was a lot of stuff about the Paris attack. I just posted anything that sounded just in solidarity with that. |
| I | Yeah, yeah, yeah. |
| P6 | Basically you won’t really see me as a person or my kids. If you go back down, I used to do lots of pictures about myself, my kids, but I don’t do that anymore. There’s my husband when he was [visiting here]. |
| I | Oh okay, so why did you stop? |
| P6 | Because, um, I have a lot of people that I grew up with that are there now. On Facebook, you know Facebook, everyone is on Facebook. People tend to monitor? They want to … Um, I think that I’m announcing what I don’t really want to announce when I show where I am or what I’m doing. It makes some of them think I’m richer than I actually am. And I start hearing about bills that don’t really concern me. |
| I | Okay yeah. |
| P6 | If someone wants to pay rent and they’re calling me because they think I’m in the UK and I’m now rich. So I, I just fffffffff [exasperated sound] … |
| I | So is that they’re judging you or they ask you for money or something? |
| P6 | They could ask you for money, judge you if you don’t give them. |
| I | So, so they’re seeing things about you on Facebook but not like the whole story. |
| P6 | Exactly, so, so that’s why it’s… I just stopped. Most people I follow on Twitter, they don’t know me and I don’t know them. So I’m free on Twitter. [Laughs] I’m free there, I do what I want. But most times it's just to see what’s happening. |
| I | So you feel like you’re watched on Facebook compared to Twitter. |
| P6 | Yes. Twitter, I, I know just maybe 5 to 10% of my followers. Every other person is just random. Like I like what you post, I follow you. You like what I posted, you’re following me. |
| I | So you feel a lot freer to speak on Twitter than you do on Facebook. |
| P6 | Yes, because although I see nobody knows me so I could get away with anything I want to say - of course I won’t say anything bad but I won’t feel like I’m being watched. |

In both examples, the manner in which Internet technologies dissolve offline separations of time and space and between different cohorts creates a hazardous setting in which social interactions with one group create potentially problematic situations with others. In this latter example the participant explained that she could not simply remove the troublesome contacts, because they were members of her extended family, and to do so would cause great offence. Her solution to this problem was to reassert agency over her privacy by moderating what she revealed on Facebook, and using Twitter as a place to express herself more freely. This contrasted strongly with a third participant, for whom Twitter was used in a professional capacity and so carefully moderated, whilst Facebook was treated as a private space in which he expressed himself freely.

These contrasts lead us to an insight which follows from this situated understanding of privacy. Namely, that establishing from the outside what data is subject to privacy concerns is exceedingly difficult. As was forcibly articulated through the research presented in deliverable
D4.1, situated reasoning can lead to very different privacy conceptions even within the same platform. The same can be said of different subject matters. One participant’s profile identified gambling as an interest, based on URLs that had been requested. In the interview, conducted with the participant and his wife, it became apparent that this account was an uncomfortable one for him. Gambling might appear an obvious example of potentially sensitive data, but in this example the man did indeed gamble, with the full knowledge of his wife. What was sensitive was the profile’s suggestion that gambling was taking place on the shared family laptop, contradicting the participant’s account that he only bet £1 per week using the betting app on his phone. The participant found it necessary to develop a detailed account of how this claim in the profile must have been the result of advertising pop-ups on the laptop that were generated by his son’s gaming:

**Example 20:**

P3 Now that [URLs of gambling website], they are adverts,
P5 Right.
P3 they're popup adverts when [son's] on something on YouTube or his roadblocks or something like that.
P5 Riiight .
P3 Because they get popups all over the place.
P5 Yep .
P3 So … I didn’t know… because I don’t go on William Hill on here-
P5 -No-
P3 -I go on my phone.
I Oh, okay.
P3 I’ve got an app-
P5 Yes.
P3 I wouldn’t go on any of those.=
P5 =No, no.
I =So they’re all something.williamhill.
P3 Yes, I wouldn’t go on any of [those
I ] [Yes.
P5 because I’ve got the app and all I do is my pound accumulator. They are popups.
P3 Ah, card games and gam- games...
P5 Right, yes.
P3 Hobbies, interests, games, [card games and gambling.]
P5 [Card games and gambling] - which would make sense because obviously that’s what it is.
P3 Yes, so you can see there [gestures to other URL] they’re adverts.
I “Traffic manager”, yes.
P3 Yes, so they’ll be popup adverts. So … <that’s something for us to keep an eye on when [son]
I is on there>
P5 [Hmmmm
I [Because he's…] Yes. Yes.
P3 Because yes- so that’s interesting.
P5 -That's interesting.
P3 That is really interesting so.. yes, maybe something we (_) keep an eye on.
I Do you have any Ad Block software or anything like that?
P3 We do…. Um, I’m going to have to go through his settings on his account…
P5 Yes.
P3 And just double check everything what everything is.
I Yes.
P3 They shouldn’t be coming up so…
I Yes.

It was not the claim of gambling that was sensitive, but the suggestion that more gambling was taking place than was known about by his wife. This underscores the observation made in section 4.3.3.3 above that it is often what is revealed to those you know most intimately that proves to be the most sensitive. Thus we see in this excerpt the uncomfortable-ness of this claim being managed by the co-generation of an account which identifies the source as advertising (a rare example of this explanation), and which concludes by demonstrating that that participants’ are responsible parents who, far from being the source of the URL requests, will take steps to protect their son from this material.

By contrast, another participant’s profile suggested they gambled and did online dating, but it was the latter which proved highly sensitive. The participant came from a traditional Indian background and would at some point have an arranged marriage. Any dating without the involvement of her parents was strictly forbidden. In her case, the claimed gambling was of no concern. Whilst her account denied any such activity, she engaged in no particular effort to elaborate the denial. Methodologically this indicates another valuable insight: it through elaborated accounts that users manage the revelation of potentially sensitive material and the presence of such elaborations is therefore an indicator that some kind of sensitivity is present.

4.3.4 Personalising Services

One of the core interests in UCN is the development of a constellation of services that are specifically personalised to match a user’s current situation and their preferences.

4.3.4.1 Recommendation

A key way that it is anticipated some measure of personalisation might be accomplished is through recommendation. Some effort was therefore made during the interviews to probe user views upon recommendation as they currently encounter it and the ways in which recommendation currently seems to happen.

One clear finding here was that direct recommendation is widely resisted unless it is something users have initiated themselves, for instance by conducting a search. Recommendations resulting from expressed preferences on social media, such as within Facebook, were particularly likely to be disregarded. It also appears to be the case that even actual ‘friends’ may overstep the mark here and be treated in a similar fashion if they recommend things too often. In the following example discussion was specifically focused on social media and how the user engaged with recommendations:

Example 21:
C: I do it all the time to my friends but they don’t do it much at all. I send something out as soon as I come across it. They’ve stopped even replying because I send them so much …
However, there are relationships to be found between social media and subsequent exploration and take-up:

**Example 22:**

I: Do you get recommendations through social media?
E: Well the biggest thing is Tumblr. People will post a GIF of something, then I'll search it out and watch it. None of it comes through Twitter. But there are people I'm already following on YouTube.
I: Are there people on Tumblr you favour?
E: Not really. I just follow the links as they arise. If it looks pretty or it looks funny, I'll click on it.
I: How about Facebook? Are there people there?
E: Not that I can remember. Oh, there's a page to find feminist friendly TV series. I follow that. … Mostly it's not exactly recommendation. It's what people are talking about. For instance, Steven Universe, loads of people were talking about it and posting GIF sets. So I thought it must be something interesting. Fresh off the Boat it was the same. I was seeing people talking about it. … These are generally people I follow. I follow a lot of artists, and people into comics and the arts. If they start talking about something I figure it must be something interesting …

The strongest form of recommendation remains what happens in the course of ordinary conversation:

**Example 23:**

L, T, (her brothers) my friends. They give me recommendations. It usually happens in the context of conversation. They'll make some reference to a series. Then I'll write it down somewhere and follow up on it later …

Another common resource for sharing recommendations is WhatsApp:

 Mostly it's by WhatsApp, where they'll give me a link. Sometimes also by Facebook. Some things I'll then just carry on watching

4.3.4.2 Enhancing the User Experience

A common justification for personalization is that it will serve to enhance user experience of the online activity. However, data within the user studies indicated that a broad range of social and situated matters can significantly affect the ways and means by which users experience the Internet, and that these are not readily amenable to being captured using traditional service-level methods.

The following examples illustrate how users’ reasoning about the quality of their experience can be hard to capture with network and application level statistics alone. One problem inheres in the nature of systems log data itself: such outputs have been shown to be opaque to human reasoning, requiring accounts generated by the subjects themselves to render it in any way meaningful (Tolmie et al, 2016). It is clear that a number of similar issues are encountered in the studies reported here when inspecting the output from logs of network traffic.

For example, one of the households in the UK study consisted of a single mother with four children. This household had a major problem regarding competition for scarce resources that coloured the whole nature of their online experience. One example of this competition related to access to the family Netflix account, which was limited to two simultaneous streams. The son reported listening out for his sisters’ episodes to finish:
Example 24:
... like you'd hear the episode, their episode, finish and you'd try and click before they could load the next one. To kick them out and you can start again.

It would be tremendously difficult to unravel this competition from the network traffic logs alone. At most the logs might provide a sense of regular alternation between certain devices for streaming over temporally consistent blocks. Further armed with an appreciation that there are limits on how many devices can stream simultaneously over the kind of network connection provided one might be able to see that some kind of turn-taking system is in operation to handle the problem. But to even do this requires an understanding of the constitution of the household across devices and the application of some human reasoning. It does not just fall out of the logs on its own, yet it is critical to understanding the experience of video consumption confronting this particular household’s members.

In fact, discussion with the members of the household revealed something much more profound at play. The mother has a tablet as does each of the children, but the children kept using hers. When probed on this point it turned out that they make use of hers because it is usually charged, unlike their own. Her complaint about this is that they will then return it with the battery completely run down and then make off with her charger to charge their own instead. In fact, they described themselves as being embroiled in what they called ‘charger wars’.

Another case demonstrates how individuals’ past experiences influence the nature of their current experience as well. One woman had recently moved to the UK from Nigeria, where her poor home connection had resulted in a practice of pre-planned media consumption, using her workplace’s superior connection to bulk download for later consumption at home. In the UK, the improved network led to her choosing to stream media as and when required, except when the regular outages of her provider cause her to resort to walking to a friend’s house to download the desired content before returning home to watch it.

The point we wish to stress here is that how people experience their online activity is a judged phenomenon: people’s experience of using technology, and the networks that underlie that use, is completely woven into their lives’ everyday organisation. It is made to fit with the rationales and accountabilities that hold for how they lead those lives: how they arrange their television watching practices and the sharing of household resources for instance, or how resilience to network failures is dependent on previous experience. In this way, even an event as drastic as outright network failure is, in QoE terms, socially conditioned. It is certainly not just about using technology for technology’s sake – see [4] for a deeper exposition of this point. So it is in the very stuff of how life gets done that the experience of using technology, and the troubles that may arise with it, can be seen to reside. At first sight this leaves a rather daunting gap between the metrics and systems logs that can be automatically recovered and what actual lived quality of experience looks like.

4.3.4.3 Building a Profile

One of the things we have been keen to stress along the way here is the importance of bringing the user into the loop and actively promoting the inclusion of the user in the generation of accounts of their data, prior to such accounts being shared with third parties. As outlined above, a way we actively sought to provoke account generation and its refinement was through the creation of user profiles. It is worth spending a little time examining the work of building such
profiles because, in its own right, it makes visible some of the technological challenges confronting the on-going pursuit of this work.

A key point here is that, as much for the third party going through the data as anybody else, the meaning that is generated during the job of working through the data does not reside 'in' the data itself. Instead it draws upon a whole body of common sense reasoning about what the data might be saying ‘really’ and common sense assumptions based upon what demographic details have already been gleaned about the user in other ways.

The playing out of this reasoning can be seen in the following example of a profilee established as young, female, and in higher education. The profilee's location data logged two locations in close proximity being inhabited overnight. The profiler inferred that this was either the same location being mischaracterised as two, or the profilee had a partner living at the second location. When certain interests – cars and cricket - were suggested by the profilee's browsing data which did not fit with the profiler's assumptions about this type of person, the profiler constructed an explanation based around there being a partner, i.e. a boyfriend, occasionally using the profilee's device. We see here how typifications, and assumptions attached to them, influence the selection and interpretation of data.

As the profiler proceeded the participant type became increasingly refined and the profile moved from a handful of demographic markers to a body of specific actions (websites and locations visited) and intentions (the meaning those visits held for the profilee). The profiler's assumptions enabled him to interrogate the profile he was constructing in particular ways. His questions had two basic forms: i) is this data valid and reliable?, and if so, ii) is this data noteworthy and what does it mean?

i) Is the piece of data valid? Is it generated by the user’s own actions, rather than an artefact?

ii) Is the valid data reliable? Is it a trace of an intentional action by the user, rather than an error of their own making?

iii) Is the valid, reliable data noteworthy? Does the revealed activity help to distinguish the individual from the wider population?

iv) What does the valid, reliable, noteworthy data mean? How should this activity be understood in the context of the user’s life?

URL and location data are dealt with differently, partly due to the different nature of the data and tools to explore them, and partly due to the profiler’s inferences about their different relationships with the user activity which generated them.
### Assessing the data

<table>
<thead>
<tr>
<th>Location activity</th>
<th>URL activity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Is this trace valid?</strong></td>
<td><strong>Is it an advert?</strong></td>
</tr>
<tr>
<td>a) Is the location plausible? Using Google Maps’ satellite view, labels, and street view to identify place.</td>
<td>a) Is it an advert?</td>
</tr>
<tr>
<td>b) How close is the trace to other traces? Close proximity could suggest an error.</td>
<td>b) Is it a resource pulled in by the page? (A Content Distribution Network site for example)</td>
</tr>
<tr>
<td>c) What time/how long did it happen? Does it fit our understandings of how such a place could reasonably be used?</td>
<td>c) Is it a website pulled in by the page? This was problematic for the profiler, as such URLs could be browsed to, suggesting they were legitimate activity by the participant. The timing of the request becomes key – if simultaneous with another website suggests one URL is generating the other.</td>
</tr>
<tr>
<td>d) Is the trace a one off? This query can be used to confirm or deny the validity of the trace, depending on other factors.</td>
<td></td>
</tr>
</tbody>
</table>

| **Is the trace reliable (does it show intentional user activity)?** | **Is it a mis-click?** |
| a) How long did the user spend there? Profiler assumes any recorded location demonstrates intent by participant, unless it’s on the way to somewhere else, in which case trace should be brief. | a) Is it a mis-click? |
| b) Is it an unsuccessfully exploration? | b) Is it an unsuccessfully exploration? |
|  | Both a) and b) can only be indirectly inferred from one-off visits which do not fit with the profile being developed. |

### From data to meaning

| Is the activity **noteworthy** (does the activity help to distinguish the individual)? | “Something that’s off the norm”; “rare”; “provocative” |
| a) | a) |
| b) Something that reveals mundane details of daily routine | b) |
| c) Does it support other identified features? | c) |

| What does it **mean** (where does this phenomena sit in the participant’s life world)? | Establish a provisional type for the profilee, based, in the first instance, on the limited demographic information available. It is the profiler’s assumptions about this type which allows the work of developing meaning to begin, essentially in the form of an imagined conversation between the profiler and the profilee. |
| a) | a) |
| b) As the profiling activity proceeds, this type becomes more firmly established. There is a process of exchange between the type and the logged data, with the former shaping the interpretation of the latter, and the latter shaping the development of the former. | b) |

**Figure 16: The work of profiling.**

### 4.3.4.4 Bridging the gap

Overall the ethnographic data revealed that the work of personalisation depends on factors that are hard to infer from technical data, including the dynamics of the household and user expectations. Having said this, analysis of the data does provide some hope for bridging this gap. Although different users have very different online activity profiles, each individual user
has a limited – and often small – number of settings that account for most of their online activities. If we can identify the few settings that matter for each user, then improving the quality of their experience even for just those instances would already have a great impact for those users. These results open new avenues for future work on identifying such settings through the hybrid range of data sources and collection methods we have presented in this deliverable, and for drawing upon user’s own generation of their accounts to refine their profiles. The implication is that a hybrid data collection method involving collection of both technical (location, network) and human (ethnographic) data, would provide a means of validating initially crude profiles, drawing upon users’ responses to them and then developing further profiles through an on-going interaction between the user and the system itself. It was clearly the case that data collected via our hybrid method enabled rich user profiles, unavailable through traditional means, to be built. Furthermore, when represented back to users we found that these profiles were mostly accurate, though it was also the case that one inaccuracy can easily percolate through the entire profile if uncorrected by user input.

The important point here is that small augmentations of technical data with user-supplied context can enable significant improvements in the richness and accuracy of generated profiles. Interaction with the user in this way also makes the process of profiling more legible to them, not to mention rendering less opaque what may be made of the data about them that resides in the machine. This in turn is likely to help mitigate increasing concerns about privacy infringement and to work to support how users manage the visibility of their actions to others in the context of increasing varieties of personal data held about them in the ever more digitally enabled environments they inhabit.

4.4 Conclusions

In deliverable D4.1 we made the following observations:

1. The social organisation of the world has to somehow be made legible and this is not something that technical data alone can do.

2. Even if social action is made visible, it still takes membership of the household to understand what is really going on.

3. You therefore have to somehow provide people with a means to explain what the data means.

All of the above findings have hinged upon two important considerations:

a) It is through a shared-in-common, common sense, understanding of the social world and its orderly characteristics that what people do online is rendered meaningful. This was the case for the participants themselves, for the ethnographer, and for computer scientists attempting to make sense of the data on their own.

b) The articulation of what the data means really has happened under the auspices of situated social interaction. Even the machine-generated and manually-derived profiles had to be ratified in this way.

A further matter of significance is the extent to which the accounting for data in social interaction is shaped by the relationships that hold between the interactants. Not all things are accounted for to all parties in the same kinds of ways. A limitation to the work undertaken here
that needs to be recognised in that case is that the studies undertaken were oriented to by all parties as a research exercise, not a commercial relationship founded upon specific economic transactions or a service-based relationship founded upon the provision of specific services. The presence of this orientation was forcibly demonstrated over the course of the studies conducted in France where users were obliged to run OpenVPN to facilitate the logging of their network activity. Had this been anything other than a research exercise where different accountabilities can be seen to hold it is doubtful that the participants would have been willing to monitor whether the software was running and restart it on occasions where it had stopped. Nonetheless, the studies outlined in this deliverable have moved us a long way towards understanding what kinds of things have to be dealt with in order to adequately capture user context, to make user data legible for all parties concerned, and to make that data manageable, appropriately private, and usable for the provision of enhanced user experience of their online activity across a range of different settings.

To conclude: in the original description of work for UCN we indicated that we would fulfil the following objectives:

- Undertake a range of focused ethnographic studies of the UCN technologies deployed in real world settings. These will combine qualitative studies of use with captured information from the network to seek to analyse and expose the day in the life of these technologies in use.
- Extend our longitudinal study to investigate how network and service management mechanisms can become more user-centric and reflect human interaction subtlety.

The user studies have worked in a variety of ways to accomplish these ends. We have:

- Deployed technologies designed to capture and store user behaviour, which is a central aspect of what the UCN platform will need to do in order to provide effective personalized services.
- Studied these deployments and everyday patterns of online activity over extended periods across a range of households in both the UK and France.
- Interleaved ethnographic studies with direct logging of data captured from user devices and the network.
- Designed visualization tools and exposed the logged data to users in order to capture in turn how they orient to and attempt to manage the disclosure of their everyday use of technology.
- Analysed the materials gathered through both ethnographic and machine-based processes in order to assess their implications for the design and provision of network and service management technologies.

Exposed in particular the myriad ways in which technology use is embedded within the everyday social and moral organisation of household life and indicated how future design will therefore need to be adapted to facilitate its incorporation into these pre-existing forms of life.
5 CONCLUSION

The goal of WP2 was to design and implement a set of collectors that automatically infer user knowledge and store this data into the PIH. It has also designed mechanisms to collect direct feedback from users and collect datasets annotated with explicit user feedback. As the data collectors are planned to rely on the PIH system for data storage as well as for privacy and security, WP2 has helped to derive and validate the design requirements of the PIH system (WP1), and the security and privacy mechanisms developed in WP4. The collected data has also been used as an input for user profiling, user behavior prediction, and recommendation algorithms developed in WP3.

The key objectives of the WP2 were:

- Identify suitable data sources and develop efficient collectors of user knowledge.
- Design mechanisms to collect explicit user feedback to validate user knowledge.
- Collect user knowledge either in controlled lab experiments, small-scale pilot studies or medium-scale deployments.

As we report in this final deliverable of WP2, we have achieved all these key objectives: we have identified data sources across four different vantage points (end-user devices, sensors, home gateway and service back ends) to collect rich user data; we have designed and built tools to collect explicit user feedback; and undertaken several experiments to validate the data collection methods and tools, and to collect user feedback and assessment of the technologies developed in the UCN project.
6 REFERENCES


## APPENDIX I: TOOLS

The table below lists all available artefacts (web sites, tools, source code, data) produced by the consortium in WP2.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hostview</strong></td>
<td>Web site + binary download Hostview end-host performance and QoE monitoring and data collection tool for Windows.</td>
<td><a href="https://muse.inria.fr/ucnstudy">https://muse.inria.fr/ucnstudy</a> [draft version]</td>
</tr>
<tr>
<td><strong>UCN Study</strong></td>
<td>Web site UCN user study information web site including detailed participating information, consent forms and user registration. In addition, contains a user data visualization tools (requires login).</td>
<td><a href="https://muse.inria.fr/ucn">https://muse.inria.fr/ucn</a> <a href="https://horizab4.memset.net/ucn">https://horizab4.memset.net/ucn</a></td>
</tr>
<tr>
<td><strong>UCN Study</strong></td>
<td>Source code UCN user study data collection tools and backend source code.</td>
<td><a href="https://github.com/ucn-eu/">https://github.com/ucn-eu/</a> with the following sub-directories: ucn-react (visualization using react) ucnserver (admin site + VPN backend) android_monitor (context/activity logger for android) datauploadserver (receive data and store on MongoDB) ucnviz (HTML5 visualization of traces)</td>
</tr>
<tr>
<td><strong>mSpeed</strong></td>
<td>Website + iTunes download Cellular speed measurement tool</td>
<td><a href="http://www.mspeedapp.com/">http://www.mspeedapp.com/</a></td>
</tr>
<tr>
<td><strong>HoA</strong></td>
<td>Source code Online implementation of HoA bottleneck detection as a module for collectd.</td>
<td>Source: <a href="https://github.com/apietila/collectd">https://github.com/apietila/collectd</a> OpenWRT package build files: <a href="https://github.com/inria-muse/browserlab">https://github.com/inria-muse/browserlab</a></td>
</tr>
</tbody>
</table>
8 APPENDIX II: NICTA’S MSPEED PAPER

On The Performance of Content Distribution Networks in the Mobile Internet

Tipajin Thaipsutikul, Sipat Triukose, Sebastien Ardon, Mohamed Ali Kaafar, Anirban Mahanti
NICTA, Australia

ABSTRACT
This paper studies the performance of two commercial Content Distribution Networks (CDNs), Akamai and Limelight, from the vantage point of cellular network users. We first introduce mSpeed, a flexible network measurement platform, which uses the users' smartphones as network measurement vantage points. Using the mSpeed platform, we conduct a measurement study to characterize CDN performance to cellular network clients, as compared with wireline (fixed) network clients. Our study uncovers a number of interesting findings. First, we find that the available bandwidth to both Akamai and Limelight cache servers are similar, regardless of the network distance between the server and the client. Second, we find that the latency advantage of Akamai when serving fixed network clients, with its cache servers typically located in ISP’s networks, is reduced for a large fraction of clients connecting through cellular networks. We also uncover cases where Akamai’s server selection algorithms redirect clients towards distant cache servers leading to severe latency penalties. Such redirections occur more frequently in cellular networks than in fixed networks.

1. INTRODUCTION
Content Distribution Networks (CDNs) play an integral role in the Internet, helping service providers improve the quality of experience to end-users regardless of traffic levels. The main idea of CDNs is to offload origin servers by caching content closer to the requesting client. In this context, closer can mean lower network latency, higher bandwidth, or both.

Over the past decade, the performance of CDNs have received significant attention, but mostly for clients on wireline access networks [3, 6-10]. As the number of mobile devices connecting to the Internet continues to rise, a large portion of content is now consumed over cellular networks and is expected to surpass demand from wireline networks as early as in 2014 [2]. The performance of cellular networks varies greatly depending on network type and generation (e.g., 3G versus 4G), the cell traffic level at specific locations and times, and the cellular backbone network. This raises questions as to whether CDNs are effective in providing performance benefits for cellular network clients. As such, not much is known about the impact of content placement, server selection and load balancing approaches of commercial CDNs in the context of serving cellular network clients.

In this paper, we present, to the best of our knowledge, the first study that characterises CDNs from the point of view of cellular network users. More specifically, we propose and deploy a scriptable crowdsourced measurement platform (mSpeed), and use this platform to experimentally evaluate the performance of two commercial CDN services, namely Akamai and Limelight. Our key findings and contributions are as follows:

1) We propose mSpeed, a scriptable crowdsourced network measurement platform, suitable for both cellular and wireless network measurements.

2) We propose a CDN evaluation methodology, allowing the objective comparison of key CDN performance metrics such as latency and available bandwidth to cache nodes from different CDN providers.

We apply this methodology to study the performance of two major CDN providers, Akamai and Limelight. We chose those two CDN providers as they illustrate the two main CDN architectures: distributed vs datacenter. From this study, we observe that the low-latency advantage of Akamai over Limelight is significantly reduced when accessed over cellular networks, as a high portion of the overall latency is imputed by the cellular network tail.

3) We uncover a number of interesting characteristics pertaining to Akamai’s cache server selection. For example, we find cases where Akamai chooses to serve the client from a non-local cache resulting in degradation of round-trip latency measures.

Our paper is organised as follow. Section 2 dis-
Discusses related work. Section 3 presents our measurement platform, currently available network measurement primitives, and the CDN measurement method. Section 4 analyses the data collected and provides high-level findings regarding the performance of CDNs from mobile clients. We conclude the paper in Section 5.

2. RELATED WORK

Over the past decade, various aspects of CDNs have received attention, although most studies have focused on wireline clients (e.g., see [3,6-10]). Here, we briefly discuss a few representative works. Mao et al. [10] found that clients are often not close in network topology to the name servers they use, and questioned the accuracy of server selection based on IP addresses. Krishnamurthy et al. [8] compared the performance of seven CDNs available at the time of their study. Their study found that a larger CDN footprint may not directly translate into superior performance [8]. Cheng et al. [7] compared perceived latency of two major CDNs that adopt two different approaches. Triukose et al. [13] considered the delivery performance implications of the data center consolidation by the same CDN.

More recently, Xu et al. [14] characterised cellular data network infrastructure and studied the benefit of content placement within cellular data networks. Other studies have considered the benefits of transparent caching inside cellular data networks [4,5]. Our study complements prior work by investigating the performance of two major CDN providers, Akamai and Limelight, on cellular networks. To our best knowledge, this study is the first to analyse throughput and latency performance measures from CDN servers to cellular network clients.

3. MEASUREMENT PLATFORM

We developed mSpeed, a crowdsourcing-based network measurement platform targeting both cellular and WiFi network measurements (http://www.mspeedapp.com/). It is based on a traditional client/server architecture. Currently, an mSpeed application is available for iOS devices. The application offers end-user instantaneous upload/download speed and latency measurements to a set of landmark servers. The application also includes an operator comparison feature, where users can visualise and compare cellular network operators’ performance in their country and at specific locations. In addition to these set of predefined measurements, mSpeed can perform a range of scriptable network measurements, by downloading an experiment script every time it is run. Results are collected at a central server for offline analysis.

3.1 Platform Architecture

Figure 1 illustrates interactions between an mSpeed client and the server. A user starts the mSpeed measurement by tapping a “Start Test” button on the main screen. The client first connects to the server and fetches a measurement script (which we call mScript) using HTTP (Step 1). The client parses the mScript to obtain a set of measurements, the order in which the measurements are to be performed, and the input parameters for each measurement. Note that this first step also ensures that the device’s radio is in an active state before measurements are initiated. The mSpeed client performs each measurement in sequence, collecting results in JSON format (Step 2), and displaying progress and instantaneous readings to the end-user. The client then uploads aggregated results to the server (Step 3), including the GPS coordinates and a timestamp. The test results are then shown to the user, including network upload/download speed, and latency.

![Figure 1: The mSpeed platform interaction.](image)

3.2 Implementation

The mSpeed backend is implemented using a traditional 3-tier web service stack consisting of nginx for the web server, a set of python scripts and a MySQL database. The python scripts are responsible for all dynamically generated content, and implement the mSpeed REST API.

The mSpeed smartphone application is capable of performing a set of predefined network measurements on both WiFi and cellular networks (cf. Section 3.4 for a list of available measurement primitives). Note that the mScript can be modified at any time on the server without the need to upgrade clients. In addition, different scripts can be served for clients based on the cellular operator, country, and network type. The mSpeed application is currently available on iOS device in the Apple app store, an Android version is currently being finalised and
will be available in the second half of 2014.

In addition to foreground measurements triggered by the user tapping the “Start test” button, mSpeed clients can also perform background measurements. This allows mSpeed to collect more data and improve the spatial and temporal sampling of network measurements. On the iOS platform, applications cannot perform scheduled tasks in the background. The iOS mSpeed application, therefore, relies on the “significant location change” API, which allows applications to be woken up when the user changes location (this is typically occurring when the cellular radio handovers to a different base station). The mSpeed application is programmed to only allow one measurement per block of 6 hours, to limit battery and network bandwidth usage. The background measurement feature is opt-in only and can be disabled by the end-user.

3.3 Measurement Scripts (mScript)

The mSpeed measurement script (mScript) are JSON text files stored on the backend, describing the measurements to be performed by the client (i.e., primitives with input parameters). Figure 2 illustrates an example mScript. display_name is a string displayed in the app while the measurement is executed, and command is a key to refer to the measurement primitive code in the client. params provides a set of input parameters depending on the measurement. The experimenter can design mScripts by combining a set of measurement primitives in a specific order to conduct a measurement campaign of interest.

The first three measurements shown in this example, are Download Speed, Upload Speed, and Latency, and must always be present in any mScript, as they are the user-facing measurements, used to show users their network performance and for the operator comparison feature. In this example, the additional measurement "IPv6TCPConnect" is for research purpose, and allows to test IPv6 connectivity. Currently available measurement primitives are described in section 3.4.

3.4 Measurement Primitives

This section discusses the measurement primitives available in mSpeed.

Basic primitives. ping and pingDNS measure network latency to an arbitrary target and to the client’s local DNS server. Both measurements can be parameterised with the target host (ping only), packet count, packet size, inter-packet time and timeout values. Traceroute implements the traditional traceroute tool. It returns round-trip times, hop count and IP address of each intermediate router along the path from the client to the server. curl is a measurement based on the curl library [11], and provides a flexible multi-protocol data transfer library, returning a wide range of measurements such as dns lookup time, tcp connect time, effective target IP address, etc.

Bulk transfer. This primitive emulates a data transfer of arbitrary size from any third party web server, to our client. The method mimics a large object download by issuing several back-to-back HTTP GET requests for a small file, in one TCP connection, requesting both the HTTP keep-alive and HTTP pipelining feature from the server. Web servers reply to such requests with multiple HTTP responses back-to-back in the socket, including the file each time. The end result is a tool which can transfer an arbitrary amount of data from any web server which supports HTTP 1.1. pipelining and keep-alive. We use this primitive to estimate the available bandwidth from a third party web server to our client. In addition, the process is repeated over several parallel TCP connections, in order to maximise the chance of “filling the pipe”.

CDN evaluation. This section discusses the measurement primitives available in mSpeed. This primitive is used to perform experiments to a chosen CDN service. This primitive builds upon the other primitives discussed above. First, the mSpeed client downloads a single object (which is specified as input) from the server selected by the CDN; the client forces the server to ignore any cached copy of the object cache and obtains the object from the origin server. (Note that although HTTP Cache-Control is not honoured by many CDNs, including Akamai and Limelight, a CDN cache server can be forced to obtain a new copy of an object by sending a HTTP GET request with a random search string [11].) This step serves as a proxy for the
cache-miss scenario. Second, the client sends a HTTP GET request for the same object with the same random string as in the first step to the same CDN server to obtain the object. This step emulates the cache-hit scenario. Finally, the client performs ping, traceroute, and bulk transfer measurements against the selected server to measure latency, route and per hop latency, and the available bandwidth along the path to the server.

4. MEASUREMENT RESULTS
The mSpeed application was launched in December 2013 and we collected measurement results from an experiment campaign that focused on the performance of Akamai and Limelight CDN services. This campaign was run from 1 Jan 2014 through 28 April 2014. During this period, mSpeed was used from 900 unique iOS devices, resulting in over 5600 mScript experiments. These experiments were roughly equally split between cellular and WiFi networks. While mSpeed was used from 70 different countries, we note that about 60% of the mSpeed users are from Australia, Thailand, and the US.

In the remainder of this section, we report on CDN performance results as experienced by the mSpeed users. In this study, we focus on the comparison between Akamai and Limelight and examine these CDNs capabilities in both fixed and cellular networks. We use measurements made from WiFi networks as a proxy for fixed network performance.

Overall Latency and available bandwidth characterisation.
Akamai and Limelight have different approaches to server deployment. Akamai deploys its cache servers as close to the network edge as possible, whereas Limelight deploys caches at large datacenters. In prior work, Triukose et al. [13], using measurements to Akamai servers from fixed network clients and a model that aggregated Akamai’s servers into larger data centers, showed that a large number of servers locations does not necessarily guarantee performance advantage over a smaller number of strategically selected server locations. In the following, using measurements from end-users in both fixed and cellular networks, we characterise the performance of these two alternative strategies for CDN deployment.

Figure 3 shows the available bandwidth as experienced by users of cellular and fixed networks along the path to Akamai and Limelight servers at the time of measurement. We observe that in cellular networks, the available bandwidth to both CDNs are quite similar; a little under 50% of the measurements indicate available bandwidth of 3.5 Mbps in cellular networks. In contrast, on the fixed networks, the difference between the two CDNs is more noticeable. About 72% of the fixed network measurements report available bandwidth greater than 5 Mbps to Akamai servers, while 64% of the measurements report equivalent performance with Limelight servers.

One possible explanation is that measuring the end-to-end available bandwidth is actually equivalent to measuring bandwidth available at the bottleneck link. In many cellular networks (e.g., 2G and 3G networks), the wireless tail is potentially the bottleneck, and naturally the available bandwidth in these cases are similar.

In contrast, on fixed networks, due to the high penetration of broadband access, the last mile link is more likely to be the bottleneck for Akamai, which deploys servers at the edges of the Internet. For Limelight, the bottleneck could be any link along the path from the server at the data center to the client. Since in this case, Akamai and Limelight are less likely to share the same bottleneck, the observed available bandwidth distributions are rather different.

Figure 4 shows the cumulative distribution of the round-trip latency to Akamai and Limelight servers. Overall, the latency to Akamai servers are
lower than the latency to Limelight servers. For example, we observe that about 40% of our cellular network clients reported 50 ms or lower latency to Akamai, while only 20% of our cellular network clients reported 50 ms or lower latency to Limelight. Similarly, for the fixed network, we notice that about 80% of the measurements are 50 ms or less for Akamai, while only 40% of the measurements are 50 ms or less for Limelight. In fact, the distribution observed for Limelight in fixed networks is similar to the Akamai cellular network performance.

In the following sections, we investigate further the latency performance of Akamai and Limelight.

A look into the latency performance.

In our experiment campaign, the mspeed client initiates measurements (in foreground or background mode) to both Akamai and Limelight. Next, we look at the pairwise latency measurements, for the entire dataset as well as for the three countries (Australia, Thailand, USA) from where we have most clients.

Figures 5 and 6 show the scatter plot of the latency observed to Akamai and Limelight servers from cellular and fixed networks, respectively. Each point on these graphs corresponds to the latency measured to Akamai and Limelight from a unique client and a unique measurement instance. From Figure 5 (a), we observe that in about 85% of our experiments from cellular clients latency to Akamai is lower than latency to Limelight. However, Figure 5d also shows that about 37% of measurements from Australian cellular networks found Limelight to provide comparatively lower round-trip latency than Akamai.

Turning attention to performance in the fixed networks, shown in Figure 6, we notice that there are about 10% instances where latency to Limelight is lower than latency to Akamai. Further, unlike the cellular network measurements from Australia, in the fixed case only about 11% of the Australian measurements indicate lower latency to Limelight (Figure 6d) as opposed to 36% in the cellular networks case. This suggests that Akamai does perform well in fixed networks compared to Limelight, while in cellular networks the latency performance shrinks, and is quantitatively less noticeable. In the following, we take a closer look at the latency benefits of being served by caches that are located within the client’s ISP and investigate the main reasons behind the increased latency observed for Akamai clients in the case of Australia.

A comparison from the server selection perspective.

We determine the AS number of the client’s ISP based on the client’s IP address (if public) or the first public IP along the path from the client to the cache server based on our traceroute data. Additionally, for cellular network clients, we use the operator name to determine the AS number for the network. If the AS numbers determined for the client and the server match, we designate the server to be local, otherwise it is designated as ‘non-local’.

Figure 7 shows the cumulative distribution of the round-trip latencies when the cache servers are local and non-local. Note that all Limelight servers are non-local. In our measurements, we notice service from a local cache has significant latency advantages compared to service from a non-local cache. For instance, Figure 7 (a) shows that the cellular net-
work with service from a local Akamai cache we observe latencies of 50 ms or less in close to 70% of the instances, whereas with service from a non-local Akamai cache we see latencies of 50 ms or less in only about 40% of the instances. Qualitatively similar results are observed for fixed network clients in Figure 7(b). Further, in general, it appears that Akamai’s non-local caches provide lower latencies compared to Limelight’s caches in both fixed and cellular networks.

Figures 7(c) and (d) show the results for Australian mSpeed users. In this case, we notice that service from a non-local Akamai cache server has a high latency penalty. The latency penalty is more pronounced for cellular network clients compared to fixed network clients. We investigate this latency penalty in some detail.

Figure 8 shows the latency measurements as a function of the estimated physical distance between the mSpeed client and the Akamai selected cache server, for our fixed network and cellular network clients in Australia. The geo-distance between the client and server is computed based on the estimated client and server locations. The location of the client is determined by its GPS coordinates. To determine

the server’s location, we first use a IP to geolocation database. However, these databases are known to have errors [7,12], and thus we augmented this location information using traceroute data collected from our clients. Somewhat surprising is the observation that cellular network clients in Australia are in some instances directed to Akamai servers in Malaysia, Japan, and the United States; we found similar instances of redirection to the United States for clients in Thailand as well. We find a fewer number of such redirections from fixed network clients in Australia, which explains the difference earlier observed when comparing latency experienced by Australian users to Akamai and Limelight in cellular and fixed networks, i.e. Figures 5d and 6d. One possible reason could be a (DNS) misconfiguration in Akamai’s server selection mechanism. The selection of a remote server we observe here severely degrades the latency experience for the Akamai clients in cellular networks. Interestingly this does not happen as often for fixed networks. Future work will investigate the root causes of the observed misconfiguration.

5. CONCLUDING REMARKS

In this paper, we presented mSpeed, a flexible network measurement platform that uses smartphones as network measurement vantage points. A key feature of mSpeed is its flexibility, which allows us to construct custom experiments from a set of available primitives. Then, using mSpeed, we design a measurement campaign to evaluate the performance of two commercial CDNs, namely Akamai and Limelight, from the viewpoint of cellular network and WiFi users.

Our measurement campaign uncovers a number of interesting facets about CDN performance in wireline (using WiFi as proxy) and cellular data networks. First, we find that the available bandwidth to both Akamai and Limelight cache servers are similar, regardless of the network distance between the cache node and the client. Second, we find that the latency advantage of Akamai when serving fixed network clients, with its cache nodes typically located in ISP’s networks, is reduced for most clients connecting through cellular networks. Finally, our measurements uncovers some instances of sub-optimal server selection by CDN servers, wherein clients are being served by cache servers that are located at distant locations, resulting in poor round-trip latencies between the clients and the servers.

6. REFERENCES

APPENDIX III: HOA PAM 2016

Home Network or Access Link? Locating Last-Mile Downstream Throughput Bottlenecks
Srikanth Sundaresan, Nick Feamster, Renata Teixeira

To cite this version:

HAL Id: hal-01294924
https://hal.inria.fr/hal-01294924
Submitted on 30 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Home Network or Access Link? Locating Last-mile Downstream Throughput Bottlenecks

Srikanth Sundaresan¹, Nick Feamster², and Renata Teixeira³

1 ICSI
2 Princeton
3 Inria

Abstract. As home networks see increasingly faster downstream throughput speeds, a natural question is whether users are benefiting from these faster speeds or simply facing performance bottlenecks in their own home networks. In this paper, we ask whether downstream throughput bottlenecks occur more frequently in their home networks or in their access ISPs. We identify lightweight metrics that can accurately identify whether a throughput bottleneck lies inside or outside a user’s home network and develop a detection algorithm that locates these bottlenecks. We validate this algorithm in controlled settings and report on two deployments, one of which included 2,652 homes across the United States. We find that wireless bottlenecks are more common than access-link bottlenecks—particularly for home networks with downstream throughput greater than 20 Mbps, where access-link bottlenecks are relatively rare.

Keywords: Bottleneck location, wireless bottlenecks, last-mile, passive measurements

1 Introduction

Many countries around the world are investing heavily to increase the speeds of access network infrastructure. As the downstream throughput of access links increases, a natural question is whether users are reaping the benefits of these faster speeds. The downstream throughput they are experiencing may be limited by other factors, such as their home wireless networks, which may face performance problems due to a variety of factors (e.g., a poorly placed access link, interference from competing networks or even devices on the same network). In light of these trends, we study a simple question: Do users tend to see downstream throughput bottlenecks more often in their access ISPs or in their home wireless networks? To study this question, we design and implement an algorithm, HoA (Home or Access), that can accurately locate these downstream bottlenecks on commodity home routers. We deploy HoA in 2,652 home networks in the United States and characterize the throughput bottlenecks that we observe across this deployment.

Despite the importance and widespread interest in answering this question, both data and conclusions have proved to be elusive. Although throughput analysis and wireless diagnosis tools exist, each existing tool has some limitation that makes it unsuitable for studying this question—typically, these tools require performing measurements
from multiple vantage points (which are hard to convince users to install in their home networks), performing active measurements (which can affect the performance of the wireless network), or custom hardware (which can hamper widespread deployment). (Section 4 explains how our work relates to previous throughput detection and analysis tools and why existing tools do not apply in our setting.) In contrast, we seek to develop a passive network measurement tool that can run from a low-cost, commodity home network router. This choice necessarily limits the extent of the data that we can collect (and, as a result, the conclusions that we can draw), but it also affords a relatively large-scale deployment. HoA’s simplicity allowed us to implement it on a commodity Netgear router for two in-home deployments: A deployment of BISmark routers across 64 homes and 15 countries; and another deployment that was sponsored by the US Federal Communications Commission (FCC) and included 2,652 homes across the United States. These deployments allowed us to conduct a first-of-its-kind large-scale study of last-mile bottlenecks. Section 2.6 describes the deployments in more detail.

Realizing HoA required tackling several challenges. First, we needed to properly isolate performance problems in the home network versus outside of the home; capturing measurements at the home router offers a convenient solution to this challenge, since it lies between these two parts of the network. Next, we had to identify and validate metrics that were lightweight enough to capture on a low-cost home router, yet sufficient to accurately locate downstream throughput bottlenecks. We also wanted to use performance metrics from passive network traffic capture, to avoid introducing conditions that might either alter the state of the wireless network or disrupt network performance for home network users. Ultimately, we identified two features—the coefficient of variation of packet inter-arrival time and the round-trip time on the wireless LAN—that can be measured passively, are lightweight enough to be deployed on a commodity home gateway, and can identify last-mile bottlenecks in many circumstances. Section 2 incorporates these metrics into a complete identification algorithm.

We offer two important contributions: (1) the design of HoA, a lightweight tool that both accurately detects home access link and wireless network bottlenecks; (2) a detailed characterization of the nature and extent of throughput bottlenecks that commonly arise in many home networks using data from a large-scale prototype deployment of HoA in home routers. We do not determine why a particular bottleneck exists (e.g., it cannot determine whether a wireless problem results from poor device placement, non-WiFi interference, or other causes), but rather only where the problem exists, to the granularity of whether the problem is inside or outside the home. Our study yields the following important findings:

- Access link bottlenecks rarely occur in home networks where downstream access throughput exceeds 20 Mbps. Rather, in these cases, throughput bottlenecks are often introduced by the home wireless network.
- Access link bottlenecks only tend to be common for users whose downstream access throughput is less than 10 Mbps.
- In homes with multiple devices where we detect a wireless bottleneck, it is equally likely that only a single device experiences the wireless bottleneck as it is that all devices in the home experience the bottleneck simultaneously.
Our results suggest that it is worth spending effort to improve home wireless network performance, in addition to the extensive attempts to optimize performance in other parts of the network and end hosts.

2 HoA: Design, Implementation, and Deployments

We describe the design, implementation, and deployment of HoA.

2.1 Design Choices

Our first design choice was to perform measurements from the home access point. Locating bottlenecks at the last mile becomes easier with a vantage point inside the home network. Although vantage points in the access ISP (such as in the DSLAM for a DSL ISP) can see all the home traffic, these locations outside the home obscure metrics that can provide important clues about whether the home wireless network is introducing a bottleneck. Inside the home, we can either instrument end-hosts or the access point itself. Client devices can observe wireless properties from their own traffic but may not be able to observe traffic properties of other clients. A device also cannot determine characteristics of the access link. End-host tools such as T-RAT [22] can monitor TCP properties such as congestion window or duplicate ACKs to identify the causes of throughput bottlenecks but cannot isolate the location of congestion.

Our second design choice was to use passive traffic measurements. Although active probing may yield useful information about the state of the network, it also carries potential drawbacks. It risks introducing extra load on the network, thereby affecting the conditions that we are trying to measure; it may also disrupt the users who are hosting our measurement devices. Thus, we rely on passive measurements of in situ user traffic as the main source of information for detecting performance bottlenecks. We aim to do so without custom wireless drivers or anything that could adversely affect the performance of the networks we are measuring, so we look for features at the IP layer that can indicate performance problems. Possible metrics thus include flow timings and sizes, packet timings and sizes, and information that we can retrieve from TCP headers. We briefly discuss our choices.

2.2 Network Metrics

Packet arrival timings and TCP RTT are promising metrics particularly because our vantage point at the access point allows us to separately compute these metrics for the WAN and LAN portions of the end-to-end path, potentially allowing us to disambiguate problems that occur on either side of the access point.

Packet Interarrival Time. We exploit an observation that is common to many bottleneck links: packets traversing a bottleneck link experience buffering immediately upstream of the link; as a result, they experience smoothed arrival patterns downstream of the bottleneck link. To capture this effect, we use the coefficient of variation of packet interarrival times, $c_v$, which is the standard deviation of packet interarrival time divided
by the mean packet interarrival time. In our example, when the access link is the bottleneck, $c_\text{a} = 0.05$; whereas when the wireless is the bottleneck $c_\text{w} = 0.88$. In Figure 2, the “access link bottleneck” curve presents the distribution of $c_\text{a}$ for 100 experiments where we introduced a bottleneck at the access link; and the “wireless bottleneck” curve for 100 experiments where the bottleneck was on the wireless. There is no overlap between the two curves: $c_\text{a}$ is lower when the access link is the bottleneck versus when it is not.

**Wireless Round-Trip Time.** The second effect is that devices in home networks are only one hop away from the access point, so the baseline latency between the access point and the device should be a few milliseconds (as we measured in our controlled experiments). We observe that the delays caused by buffering in the wireless network (i.e., those caused by throughput bottlenecks) are significantly higher. We measure this effect by capturing the TCP RTT, $\tau$, between the device and the access point. Figure 3 presents the LAN TCP RTT (the RTT over TCP between the access point and a device in the home network) for two downstream throughput bottleneck scenarios: an access network bottleneck and a wireless bottleneck. In both experiments, we established (through repeated experiments) the wireless network capacity to be about 40 Mbps. In the first case, the access link is 30 Mbps, so it is always the bottleneck. In the second case, the access link is 70 Mbps so that the wireless network becomes the bottleneck. When the
access link is the bottleneck, the RTT is about 5 ms. In contrast, when the wireless is the bottleneck, packet buffering at the head of the wireless link (i.e., the access point) increases RTTs to about 25–35 ms.

2.3 Detection Algorithm

For each device, \( d \), we use two independent detectors. One detector uses a decision rule that determines whether an access-link bottleneck event, \( B \), occurs, given a particular observed value of \( c_d \). The other detector uses a decision rule that determines whether a wireless bottleneck event, \( W \), occurs given a particular observed value of \( \tau_d \). We first compute likelihood functions \( f(c_d | B) \) and \( f(c_d | \overline{B}) \) in a controlled setting, where we use our ability to control the throughput of the upstream link to introduce a bottleneck on the access link. We then define our decision rule in terms of the likelihood ratio:

\[
A(c_d = v) = \frac{f(c_d = v | B)}{f(c_d = v | \overline{B})}
\]

where \( v \) is the measured coefficient of variation of packet interarrival time for packets over the observation window. When \( A \) is greater than some threshold \( \gamma \), the detector says that the access link is the bottleneck (i.e., it is more likely than not, given the observation of \( c_d = v \), that the prior is the event \( B \)). We can tune the detector by varying the value of \( \gamma \); higher values will result in higher detection rates, but also higher false positive rates. We use a similar approach for \( W \). The next section presents our choices of threshold.

We can only perform bottleneck detection if the network is sending enough traffic. We set a minimum number of packets per second, \( T_{pps} \), and a minimum number of packets per flow, \( T_{pf} \), for running HoA. Figure 1 shows the distribution of the number of packets per second and packets per flow observed across homes in the FCC deployment. In approximately 40% of measured one-second intervals, we observe packet rates of less than 10 packets per second. We also tested \( T_{pps} \) values of 50, 100, and 150 packets per second, and \( T_{pf} \) values of 25, 50, and 75 packets per flow on real-world deployment data; none of these settings changed our conclusions.

2.4 Calibration

We built a testbed to run controlled experiments to calibrate detection thresholds. The testbed has an access point, its LAN, a network traffic shaper upstream of the access point, a well-provisioned university network, and servers in the university network. The access point is a Netgear WNDR3800 router running OpenWrt. To change the downstream throughput of the emulated access link, we use tc and netem on a second WNDR3800 router running OpenWrt. We run our tests against servers in the same well-provisioned university network to avoid potential wide-area bottlenecks. We run two sets of experiments using the testbed.

We use a traffic shaper to shape the link to different throughput levels while keeping the wireless link constant. In this case, identifying the ground truth is straightforward, as we know the capacities of both the wireless link and the shaped access link. We
use 802.11a and 802.11n for the wireless link with respective capacities of 21 Mbps and 80 Mbps over TCP. We generate 1,356 experiments with 11 different emulated access links, with capacities varying from 3 Mbps to more than 100 Mbps. To introduce wireless bottlenecks, we conduct two sets of experiments. (1) Reduce capacity by degrading channel quality: we do this by positioning the host at different distances from the access point, and with multiple obstructions, and also transient problems by human activity. (2) Reduce the available capacity of the channel by creating contention with an interfering host that sends constant UDP traffic, with the interfering host close to the access point. For each setting, we run a TCP throughput test using iperf. To minimize interference that we do not introduce ourselves, we use the 5 GHz spectrum, which is less congested than the 2.4 GHz range in our testbed. In our repeated controlled experiments, we found that the wireless channel in our testbed delivers a TCP throughput of about 80 Mbps on 802.11n. We performed 1,356 experiments over many operating conditions.

Because there can only be one throughput bottleneck on an end-to-end path, by definition, the detectors should never detect bottlenecks simultaneously. Using the thresholds that we computed for each detector—as we describe for each case below—simultaneous detection occurs only 2% of all time intervals, typically in cases where the throughput values for the home wireless network and the access link were similar.

**Packet Interarrival Time ($T_{ev}$)** We use the results from the controlled experiments described above to compute the likelihood functions $f(c_{ev}|B)$ and $f(c_{ev}|B)$ to determine the detection threshold $T_{ev}$. We first evaluate the detection accuracy of the algorithm for different values of $T_{ev}$. Figure 4 shows the receiver operating characteristic for this detector. When $T_{ev}$ is low (close to zero), the detector will always determine that the access link is not the bottleneck; when $T_{ev}$ is high (close to one), the detector will always identify the access link as the bottleneck. Our results indicate that detection accuracy remains high for a wide range of threshold settings for $T_{ev}$, particularly between 0.7 and 0.9. Detection accuracy is very high in this range, with a true positive rate more than 95% and a false positive rate less than 5%. The range of good thresholds reinforces our
confidence of its robustness as a detection metric. We use a threshold $T_{ee} = 0.8$, which offers the best tradeoff between the true positive and false positive rates, to declare the access link the bottleneck.

**Wireless Round-Trip Time** ($T_T$) We calibrate the thresholds for the likelihood functions $f(\tau_d|W)$ and $f(\tau_d|W)$ using a similar method. We choose a threshold $T_T = 15$ ms, which yields a detection rate of 95% and a low false positive rate of less than 5%. Similar to the $T_{ee}$ parameter, $T_T$ is also robust; we get similarly high true positive rates and low false positive rates for values ranging from 12–17 ms. Higher LAN latencies in the wireless network can result from other wireless problems that may manifest as retransmissions or backoffs. We observe empirically that these wireless issues introduce up to 8–12 ms of delay, whereas delays caused by wireless throughput bottlenecks introduce more than 15 ms of extra delay, thresholds which yield a high detection and low false positive rate in our experiments.

### 2.5 Limitations

HoA has several limitations. First, because it relies on passive traffic analysis, the link must carry enough traffic to enable analysis. Section 2.3 how we determine minimum thresholds for detection, which are heuristics. Second, constant bit rate traffic could in some cases yield a low $cr$, thus causing HoA to mistakenly detect a throughput bottleneck on the access link; such cases may need to rely on other detection methods. With respect to bottlenecks, HoA cannot identify the root cause of bottlenecks, and it cannot identify bottlenecks far from the last mile, such as peering or server-side bottlenecks. HoA can only locate throughput bottlenecks where the link is work-conserving; because wireless links violate this assumption, HoA cannot detect upstream throughput bottlenecks. Additionally, detection thresholds may be sensitive to certain settings and configurations: $T_T$ may depend on the wireless driver and hardware; in cable access networks, $T_{ee}$ may depend on the channel bonding configuration of the DOCSIS modem. The calibration methods from Section 2.4 may help determine the appropriate thresholds in various settings. Finally, to reduce CPU load, HoA collects data periodically, which does not allow us to capture aspects of the network that vary over small timescales.

### 2.6 Deployments

Table 1 summarizes our two deployments, which we briefly describe below.

**BISmark Deployment.** We deployed HoA on Netgear’s WNDR3700/3800, which has an Atheros chipset with a 450 MHz processor, one 802.11 bgn radio, and one 802.11an radio. The 3800 has 128 Mbytes of RAM, and the 3700 has 64 Mbytes of RAM. The devices run OpenWrt, with the ath9k wireless driver. The driver uses the Minstrel rate adaptation algorithm, with the default setting to a maximum bitrate of 130 Mbps. Every 5 minutes, HoA collects packet traces from the WAN port for 15 seconds and extracts timestamps and per-flow RTTs on either side of the access point, as well as the number
of packets for each connection using tcptrace [21]. tcptrace tracks packets and
the corresponding ACKs to compute the RTTs.

FCC Deployment. We use the FCC’s deployment of Netgear WNR3500L, which has
a Broadcom chipset and a 480 MHz processor, one 802.11bgn radio, and 64 MBytes of
RAM. The devices run a custom Netgear firmware based on OpenWRT. The resource
constraints of the WNR3500L required two changes to our implementation. First, we
imposed a packet limit and a time limit for every trace collection iteration. The collec-
tion runs for 10 seconds or until it has collected 10,000 packets, whichever comes
first. We discard any trace for which the packet filters dropped at least 5% of packets
from our analysis. Additionally, due to resource constraints, we do not perform any pro-
cessing on the device, except for anonymization. Instead, we offload the packet header
traces for offline analysis. To avoid conflicts with FCC’s Measuring Broadband Amer-
ica program, we could only perform our measurements three times per hour.

3 Results

This section explores our findings: (1) In home networks where downstream throughput
exceeds 20 Mbps, the home wireless network is the primary cause of throughput bot-
tlecks. (2) Access link bottlenecks are prevalent in home networks where the down-
stream throughput is less than 10 Mbps. (3) In homes where HoA detects a wireless
throughput bottleneck, it is about equally likely that the wireless throughput bottleneck
is isolated to a single device or observed across all devices.

3.1 Prevalence of Last-mile Bottlenecks

In this section, we explore the prevalence of downstream throughput bottlenecks in ac-
cess links versus home wireless networks using HoA. Specifically, we study the fraction
of tests for which HoA identifies downstream throughput bottlenecks, and to what ex-
tent these bottlenecks are caused by the access link versus the home wireless network.

We perform more than 50,000 tests over a wide range operating conditions in the field.
HoA identifies downstream throughput bottlenecks in 55% of tests in the BIsmark
deployment and 47% of tests in the FCC deployment. When HoA does not detect a
bottleneck, the underlying cause may be low demand or bottlenecks being elsewhere in
the network (e.g., at a peering point). As expected, homes with access-link throughput
Fig. 6: Prevalence of access link and wireless bottlenecks home networks the two deployments deployment. When downstream access-link throughput exceeds about 20 Mbps, only about 20% of last-mile bottlenecks occur on the access link.

less than 10 Mbps experience the largest fraction of throughput bottlenecks; 55% of tests detect a bottleneck. The fraction of tests where HoA detects a bottleneck, however, remains close to 40% even for homes with access-link throughput above 90 Mbps. In the rest of this section, we further characterize the tests where HoA detects a downstream throughput bottleneck.

Figure 6a plots the fraction of downstream throughput bottlenecks in the BiSmark deployment that are located either in the access link or in the home wireless network. We group home networks into bins of 10 Mbps according to the measured downstream throughput of their access links. The results show that many throughput bottlenecks in the BiSmark deployment are due to the wireless network. Our analysis of the bottlenecks per home in the BiSmark deployment shows that the fraction of wireless bottlenecks varies significantly across homes even for homes with similar access-link throughput. For example, homes with access-link throughput less than 20 Mbps had wireless bottlenecks in between 3–58% of downstream throughput bottlenecks, and 11–83% of downstream throughput bottlenecks. By default, we configured these home routers to use 802.11n, which can support significantly higher rates. The default 802.11n configuration supports frame rates of up to 130 Mbps (we observed about 85–90 Mbps over TCP), while 802.11g supports only frame rates up to 54 Mbps. The fact that these networks are experiencing throughput bottlenecks suggests persistent problems with home wireless network deployments in practice.

Figure 6b shows the same results for the FCC deployment. First, access-link bottlenecks only occur frequently for home networks with downstream access throughput less than 20 Mbps. Homes with access throughput less than 10 Mbps experience access-link bottlenecks in about 66% of cases; however this fraction drops rapidly as access throughput increases: for homes with access throughput between 10 and 20 Mbps about 40% of downstream throughput bottlenecks are due to access-link bottlenecks, whereas for homes with access links exceeding 20 Mbps access-link bottlenecks explain only about 20% of downstream throughput bottlenecks. Conversely, wireless throughput bottlenecks become more prevalent in homes with higher access throughput: 33% of downstream throughput bottlenecks for homes with throughput less than 10 Mbps are due to wireless bottlenecks; 40% of the bottlenecks are in the wireless network for homes with
10–20 Mbps access throughput; and, nearly 80% of the bottlenecks are in the wireless network when access throughput exceeds 20 Mbps. That wireless throughput bottlenecks occur even for access links with such low speeds is surprising; the FCC access points support 802.11n, with default frame bitrates of up to 130 Mbps and a maximum frame bitrate of 300 Mbps. Some users had configured their routers to 802.11g, and those users did experience lower throughput. Yet, 802.11g comprised only 10% of all tests, so most of the problems that we observed occurred even with 802.11n.

In about 8% of downstream throughput bottlenecks in homes with access-link throughput less than 10 Mbps, HoA indicates that both the wireless network and the access link are introducing throughput bottlenecks. In principle, this should not occur as, by definition, there can be only one bottleneck. The prevalence of this result for primarily low-throughput access links suggests that in these cases, at least one device in the home network may be experiencing poor wireless conditions in conjunction with an access-link bottleneck.

3.2 Wireless Bottlenecks Within a Home

The previous section demonstrated that wireless bottlenecks are common; in cases where wireless bottlenecks exist, at least one device in the home experiences a wireless throughput bottleneck during the tests. For about 75% of tests when HoA detects a wireless bottleneck, we only observe traffic for one device in the home. For the remaining 25% of tests with a wireless bottleneck, we investigate whether the active devices experience a downstream throughput bottleneck in the wireless network simultaneously. Simultaneous throughput bottlenecks in the wireless network to independent devices might indicate a more systemic problem (e.g., pervasive interference, poor signal from the access point, contention), whereas isolated throughput bottlenecks are more likely to indicate a problem with a particular device. About half of the cases we observed involve throughput bottlenecks that are isolated to a single device; in another 45% of cases, all of the devices in the home simultaneously experience a throughput bottleneck.

4 Related Work

HoA draws inspiration from several previous diagnosis techniques. Zhang et al. developed T-RAT [22] to analyze TCP performance. T-RAT estimates TCP parameters such as maximum segment size, round-trip time, and loss to understand flow behavior. Katabi et al. [11], used entropy in packet interarrival time to estimate shared bottlenecks. Bier et al. [3] used packet interarrival times for distinguishing between different kinds of losses. HoA is similar to some of the approaches used in these papers (e.g., it uses packet interarrival time as input to a detector for access link bottlenecks), but we tailor our approach so that it only relies on data that can be easily collected from a home router. Previous work has studied broadband access performance [4, 8, 9, 20]. In particular, Sundaresan et al. [20] study residential access performance from home routers (also using the FCC Broadband America dataset). There have also been many previous approaches to diagnosing wireless networks. One approach is to deploy passive traffic monitors throughout the network to diagnose wireless pathologies [1, 2, 6, 15, 16].
or to study wireless performance [14]. Kanuparthy et al. [10] developed a tool to detect common wireless pathologies (such as low signal-to-noise ratio, congestion, and hidden terminals) by using both active probes and an additional passive monitor deployed within the network. Kim et al. [12] analyze wireless metrics such as frame bitrates, frame ACKs and retransmission rates to identify root causes of wireless performance problems. Other approaches have monitored wireless networks with custom hardware [5, 13, 16–18]. Unfortunately, it is difficult to deploy multiple monitoring points or custom hardware in many home networks, since it requires deploying equipment beyond what a normal user is typically willing to install or have installed in their home. Other efforts have characterized home networks in terms of connected devices and usage [7, 19]. None of these studies, however, have studied how often the home network constraint downstream throughput.

5 Conclusion

To identify performance bottlenecks in home networks, we developed an algorithm and tool, HoA, that passively observes traffic flows between the home network and the access network to determine the location of last-mile downstream throughput bottlenecks. Our prototype deployment of HoA in 2,652 home networks shed new light on the prevalence of downstream throughput bottlenecks in both home networks and access networks. We find that when the downstream throughput of a user’s access link exceeds about 20 Mbps, a high fraction of throughput bottlenecks are caused by the user’s home wireless network. This finding is significant in light of recent proposed regulations to change the definition of broadband Internet access to increasingly higher speeds. Our study opens several avenues for future work. First, we need methods to identify root causes that explain why various wireless performance problems exist in addition to where they are. Second, a follow-up to HoA could attribute problems that home network users experience to a more complete and more specific set of causes.

Acknowledgments We thank the FCC and SamKnows for helping as develop and deploy HoA in the Measuring Broadband America (MBA) platform. We also acknowledge the participants of the MBA platform. We would like to thank our shepherd, Mahesh K. Marina, and the reviewers for their helpful comments. This work was supported by NSF awards CNS-1535796, CNS-1539906, and CNS-1213157, and the European Communities Seventh Framework Programme (FP7/2007-2013) no. 611001 (User-Centric Networking).

References

APPENDIX IV: TMA 2016

Passive Wi-Fi Link Capacity Estimation on Commodity Access Points
Diego Da Hora, Karel Van Doorselaer, Koen Van Oost, Renata Teixeira, Christophe Diot

To cite this version:
Diego Da Hora, Karel Van Doorselaer, Koen Van Oost, Renata Teixeira, Christophe Diot. Passive Wi-Fi Link Capacity Estimation on Commodity Access Points. Traffic Monitoring and Analysis Workshop (TMA) 2016, Apr 2016, Louvain-la-Neuve, Belgium. 2016. <hal-01292633>

HAL Id: hal-01292633
https://hal.inria.fr/hal-01292633
Submitted on 23 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Passive Wi-Fi Link Capacity Estimation on Commodity Access Points

Diego da Hora¹,², Karel Van Doorselaer¹, Koen Van Oost¹, Renata Teixeira³, Christophe Diot³
¹Technicolor, ²Inria, ³SAFRAN
Email: {First.Lastname}@Technicolor.com¹, renata.teixeira@inria.fr², christophe.diot@safran.fr³

Abstract—Wi-Fi is the preferred way of accessing the internet at home; many devices today connect only via wireless. Unfortunately, Wi-Fi performance is highly variable. For example, in dense urban neighborhoods it is typical to see tens of competing Wi-Fi networks [11] and other non-Wi-Fi devices (e.g., microwave ovens), which will cause contention and interference. Sub-optimal installation of the Wi-Fi access point (AP) can also degrade performance. For example, the AP may be placed in a location that leaves devices with weak signal. Our discussions with residential Internet Service Providers (ISPs) indicate that often, when customers call to complain about poor performance, the problem is in the home Wi-Fi, not the ISP network. Studies of home networks confirm that Wi-Fi can cause poor performance [11], [17]. Diagnosing problems in the home Wi-Fi is challenging for ISPs due to the lack of visibility within the home network. In many cases, however, the ISP provides the home AP.

We believe that ISPs should instrument APs to monitor Wi-Fi performance and assist in diagnosis. Although previous studies have already instrumented APs to monitor wireless performance [8], [10], [11], [7], [18], our aim is to develop a practical solution that ISPs can deploy at scale on commodity APs to diagnose user Wi-Fi problems. This goal brings some restrictions. First, we exclude solutions that rely on APs with multiple Wi-Fi NICs [8], [10]. The use of multiple NICs is appealing because one NIC can implement the usual AP functionality, whereas the other can perform per-packet monitoring without interfering with users’ traffic. We instead directly monitor the NIC that is acting as AP. Second, we only rely on passive measurements. Active measurement solutions [5], [18] can help in on-demand diagnosis, but periodic active measurements would disrupt users’ traffic and drain the battery of mobile devices. Third, we exclude the use of per-packet metrics [17]. Processing metrics per packet introduces overhead in periods of high network load that negatively impact the AP performance. Finally, we want our solution deployed at large scale, so we only rely on standard metrics available on commercial APs, which require no hardware or significant driver modifications.

In this paper, we present a method to estimate the link capacity of Wi-Fi links using passive metrics Wi-Fi drivers commonly expose. We define the link capacity as the maximum UDP throughput between the AP and the device assuming full media availability. Our contribution is to define and validate simple models to estimate link capacity in 802.11n networks. Previous models [4] are for 802.11a/b/g, which did not have frame aggregation. First, we define a model under the assumption of fixed physical data rate (or PHY rate) and no frame losses (§V). Then, we extend this model to the realistic case where the PHY rate varies and frame losses occur (§VI).

Link capacity is useful for Wi-Fi diagnosis. As illustrated in Figure 1, a device’s throughput may be limited by medium access problems (i.e., Wi-Fi or non-Wi-Fi contention prevents access to the medium) or frame delivery problems affecting the link capacity (i.e., when the channel quality is poor) [8]. In §VI, we show with two case studies how to use the link capacity to identify Wi-Fi performance bottlenecks and distinguish between those caused by medium access or frame delivery problems.

We tuned and validated our models experimentally on a commercial AP with a Broadcom NIC, which is a popular NIC in the APs ISPs provide. Our solution is broadly applicable, and we are starting to implement it on other chipsets. The results of our controlled experiments show that we can...
estimate the link capacity with estimation errors similar to the state of the art, but without the need to tune station-specific parameters. Our methods are part of a broader Wi-Fi diagnosis solution that is in trial with two major European ISPs and one major ISP in Asia-Pacific. Each trial deployment is continually monitoring 30–50 APs.

II. BACKGROUND

In this section, we present a brief background on Wi-Fi concepts, in particular of 802.11n.

802.11 MAC protocol. The IEEE 802.11 Medium Access protocol uses carrier sense multiple access with collision avoidance. Enhanced Distributed Channel Access is used to coordinate medium access: nodes only transmit after they sense the medium idle for a duration of Arbitration Inter-frame Space (AIFS) plus a backoff timer. Frames need to be acknowledged by the receiver through an ACK frame, and each transmission is spaced by a Short Inter-frame Space (SIFS). If the ACK is not received, the sender initiates a collision, deferring transmission and exponentially increasing the contention window size if a retransmission is due.

RTS and CTS. The Request to Send (RTS) and Clear to Send (CTS) handshake is an optional mechanism used to reduce frame collisions and to mitigate the hidden node problem. It uses the Network Allocation Vector in RTS/CTS messages to indicate how long the medium will be busy due to impending transmissions.

Medium Sharing and Frame Aggregation. 802.11n introduces frame aggregation, which reduces MAC overhead by allowing delivery of multiple aggregated Mac Protocol Data Units (A-MPDUs) in a single medium access. 802.11n stations are required to support HT-immediate Block ack, which uses Block Ack frames (BA) to acknowledge a set of MPDUs after the reception of an A-MPDU to improve MAC efficiency. The number of MPDUs per medium access depends on the PHY rate used and we represent this number by AGG(P), for a given PHY rate P. The maximum A-MPDU size used by the transmitter (MAXagg) is implementation dependent, but it is mainly limited by the field Maximum A-MPDU Length Exponent advertised by the receiver on control messages [12]. Figure 2 illustrates a typical A-MPDU with RTS/CTS protection. We study the frame aggregation impact on performance of commercial access points in §IV-B.

PHY rate and rate adaptation algorithm. An 802.11n device is able to use any of the PHY rates introduced by 802.11n as well as legacy 802.11g and 802.11b rates. The maximum PHY rate for 802.11n is 65Mbps for a single spatial stream. Devices with two or more antenna chains can increase the PHY rate with Multiple-Input-Multiple-Output (MIMO). Channel conditions determine the best PHY rate to use: if they are good, higher PHY rates maximize performance; otherwise, lower PHY rates increase the probability of reception. The rate adaptation algorithm is responsible for selecting the PHY rate for each frame. Even though rate adaptation algorithms aren’t specified by the standard, popular implementations are Adaptive Multi-Rate (AMR), Once and Sample Rate [19].

III. EXPERIMENTAL SETUP

This section describes the setup of our experiments to validate the link capacity models. We run experiments in two settings: an anechoic chamber, where we know there is no external source of contention and interference; and our lab in Paris, which is a more realistic environment with other Wi-Fi networks and other sources of non-Wi-Fi interference.

Testbed. We study primarily one Technicolor AP with a Dual core Broadcom MIPS 400 MHZ processor, 256 MB DDR RAM and a Broadcom BCM6362 NIC with 802.11n 2x2 technology. This AP chooses the MAXagg based on the maximum Rx A-MPDU length of the station, advertised on control messages. Particularly, we observe that MAXagg is equal to 4, 8, 16 and 32 when the Maximum A-MPDU length is 8, 16, 32 and 64 Kbytes respectively. For brevity, we consider that the AP only uses long guard interval and 20MHz bandwidth, since this is the default configuration for the AP. Although most of our evaluation focuses on one AP, it has the Broadcom driver w1, which is popular among the APs ISPs provide to their customers. We also test the Broadcom bcmwmmac driver and the Atheros ath9k driver in §IV-B. We observe different behaviors among the drivers, but it is possible to tune the parameters of the model to account for these differences.

We perform link capacity tests using two devices: an Android tablet, with an 802.11n 1x1 NIC with MAXagg = 8, and a MacBook pro, with a Broadcom 802.11n 2x2 NIC with MAXagg = 32. We use two applications with the android tablet: iPerf for Android to use perf in server mode, and Wake Lock to prevent the device from entering in sleep mode. We use iperf client on a Lenovo laptop with Ubuntu 12 to generate traffic from the AP to the devices. We use a sniffer, a MacBook pro in monitoring mode, to capture frames the AP sends and receives. We look at the retry bit to infer frame losses at the station and jumps on the MAC sequence number to infer frames not captured by the sniffer. This gives us ground truth on the link capacity and frame delivery ratio.

Fixed PHY rate experiments. In this scenario, we use a driver utility tool on the AP to saturate the link to the device using a chosen fixed PHY rate. We use packets with MAC payload of 1500 bytes, since it is the default Ethernet MTU. We execute one experiment per PHY rate with 5 minutes duration, for both the tablet and the MacBook. We measure the link capacity by calculating the UDP throughput to the

1Considering usage of long-guard interval.
Varying PHY rate experiments. In this scenario, the Lenovo laptop (connected through a gigabit interface to the AP) uses iperf to generate UDP traffic to the station. We perform link capacity tests in the anechoic chamber, using a metal box to generate stable attenuation between the AP and the station. We perform one test without any attenuation and three tests with the device inside the metal box in different positions, to obtain different levels of link quality. Experiment duration is 20 minutes per scenario, and we measure the link capacity by calculating the UDP throughput to the station using packet logs from the sniffer.

AP sampling. Our link capacity solution can be implemented by periodically sampling AP parameters, as shown in §IV-B. Table I describes the metrics we sample from the AP. We use wlc, a Broadcom utility program, to sample the AP. \( BUSY_{\text{agg}} \) and \( BUSY_{\text{convo}} \) are measured by the Wi-Fi driver over a period of 2 seconds. The other metrics can be sampled at any granularity, restricted only by the sampling overhead. It is possible to discover \( MAX_{\text{agg}} \) of connected stations by monitoring the AMPDU-chain size on periods when only one station is transmitting.

IV. LINK CAPACITY UNDER IDEAL CONDITIONS

This section proposes a model to estimate the link capacity assuming that the PHY rate is constant and there are no losses (i.e., 100% frame delivery). We remove these simplifying assumptions in the next section.

A. Model

Our model of link capacity extends the model of Jun et al [4] to work under 802.11n MAC improvements, including frame aggregation. We calculate the link capacity for a given PHY rate, \( P \), by dividing the UDP payload of the A-MPDU by its transmission time:

\[
LC(P) = \frac{AGG(P) \times \text{UDP payload}}{\text{A-MPDU TxDelay}(P)} \times (1 - B_s)
\]

where \( B_s \) is the fraction of time the AP is busy sending beacons.

A-MPDU UDP payload. We calculate the UDP payload of the A-MPDU frame exchange by multiplying the UDP payload per IP packet by the number of MPDUs sent at \( P, AGG(P) \). We show how to obtain \( AGG(P) \) in §IV-B.

A-MPDU transmission delay. To compute the A-MPDU transmission delay, we use 802.11 protocol parameters to model an A-MPDU exchange of \( N \) packets of size \( S \) using PHY rate \( P \). Figure 2 shows our A-MPDU frame-exchange model, which uses Enhanced Distributed Channel Access (EDCA).

We compute the A-MPDU transmission delay as:

\[
\text{TxDelay}(N,S,P) = T_{\text{IFS}} + T_{\text{BO}} + 3 \times T_{\text{SIFS}} + T_{\text{RTS}} + T_{\text{CTS}} + T_{\text{ACK}} + \text{DATA}(N,S,P)
\]

where \( T_{\text{PHY}} \) is the transmission delay of the PHY header, and \( S_{\text{MAC}} \) is the MAC header size. We add 22 trailing bits (16 + 6) to form the OFDM symbols. This is an approximation since we don’t consider padding bits.

For the purposes of calculating the link capacity, we use \( S = 1500 \) bytes (default MTU for Ethernet networks), UDP payload of 1472 bytes and \( N = AGG(P) \). Other model parameters are defined in Table II. We consider frame exchanges using Best Effort Access Category, since it is the default configuration for bulk traffic transfer, and the use of implicit Block Ack Request. We consider control frames with PHY rate \( \in [1, 2, 6, 12, 24] \) Mbps, and assume that the PHY rate to transmit a control frame is lower than that of a data frame. We use the transmission delay of control frames proposed by Jun et al [4] (shown in Table III).

B. Parameter tuning

Our model in §IV-A has two parameters that we must estimate for the particular AP under study: \( B_s \) and \( AGG(P) \). In practice, ISPs work with relatively few models of APs and estimating these parameters is simple. We show how to obtain \( B_s \) by using the beacon interval and the number of SSIDs advertised. Even though \( AGG(P) \) is implementation dependent, we show that the two most used Broadcom 802.11n drivers use the same function \( AGG(P) \).

Beacon overhead (\( B_o \)). We estimate the \( B_o \) as the fraction of time the AP is busy sending beacons. We calculate the delay to transmit a single beacon as its transmission time + \( T_{\text{IFS}} \).

We calculate how many beacons are sent each second as the inverse of the beacon interval, and multiply it by the number of advertised SSIDs.
TABLE I
DESCRIPTION OF METRICS MEASURED ON THE ACCESS POINT.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Granularity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP metrics</td>
<td>2 s</td>
<td>% of time receiving Wi-Fi traffic</td>
</tr>
<tr>
<td></td>
<td>2 s</td>
<td>% of time medium busy due to non-Wi-Fi signal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Size of last transmitted A-MPDU chain</td>
</tr>
<tr>
<td>Metrics per station</td>
<td>any</td>
<td>KB octets sent / received</td>
</tr>
<tr>
<td></td>
<td>any</td>
<td>PHY rate of last non-management frame sent / received</td>
</tr>
<tr>
<td></td>
<td>any</td>
<td>Fraction of frames successfully delivered to station</td>
</tr>
</tbody>
</table>

The AP under study advertises 3 SSIDs using 242-byte beacons at PHY rate 1 Mbps every 100 ms. We calculate that the AP sends 30 beacons each second, each beacon with duration of 1981 µs. Since the AP spends 50,630 ms every second sending beacons, we have $E_b = 5.943%$.

Frame Aggregation per PHY rate ($AGG(P)$). Either the driver or the hardware decide the aggregation of frames into A-MPDUs, with most modern 802.11n devices opting for the latter [1]. Broadcom NICs capable of IEEE 802.11n can use either the closed proprietary Wi driver or the open-source driver bcm80211.

We setup a small controlled experiment to understand how our AP, which uses the Wi driver, selects the number of frames per A-MPDU for different PHY rates. In this experiment, the AP sends data to the android tablet or to the MacBook using fixed PHY rate, as described in §III. The number of frames per A-MPDU used by the tablet and the MacBook is shown in Columns 2 and 5 of Table IV, respectively. We see that, when transmitting packets of the same size, the A-MPDU size varies per PHY rate, with more packets per A-MPDU at higher PHY rates up to $MAX_{AGG}$.

The inspection of brccsmac’s source code helps explain this behavior. The transmitter limits transmission duration to a threshold $t_{zop}$. It estimates how many bits can be sent during $t_{zop}$ and then computes the maximum number of frames per A-MPDU with the equation:

$$AGG(P) = \min \left( \left\lfloor \frac{P \times t_{zop}}{MAC \text{ frame length}} \right\rfloor, MMAX_{AGG} \right)$$

The MAC frame length is given by MAC payload + $S_{MAC}$. Considering a MAC payload of 1500 bytes and the default value of $t_{zop}$ in brccsmac of 5 ms, we were able to correctly estimate $AGG(P)$ for all PHY rates in Table IV. This suggests that both Wi and brccsmac uses the same $t_{zop}$ threshold values.

While analyzing the sniffer logs, we observe an artefact on how the AP handles frame aggregation. Between PHY rates 52 Mbps to 78 Mbps, the A-MPDU size of transmitted frames successively alternates between two values, $AGG(P)$ and $MAX_{AGG} - AGG(P)$. The most extreme case is on the MacBook at PHY rate 78 Mbps, where we observe alternating A-MPDUs of sizes 31 and 1, resulting in reduced throughput in comparison with the usage of back-to-back transmissions of A-MPDUs of size 31. We consistently observed this artefact on all transmissions of the AP. We conjecture that the NIC is internally splitting outbound packets in blocks of size $MAX_{AGG}$ and then attempting to deliver all packets inside this block before moving to the next.

We perform a fixed PHY rate experiment with a Unix machine using an Atheros card with ath9k driver to check whether we see the same behavior. With the Atheros card we observe no instances of A-MPDUs with alternating sizes. This result indicates that this artefact is specific to the Broadcom NIC. Further tests are necessary to confirm whether this behavior happens on other models of Broadcom NICs.

C. Experimental validation

In order to validate the proposed model, we perform controlled experiments in an anechoic chamber using fixed PHY rate capacity tests as described in §III, comparing the measured link capacity with the estimated link capacity. This step should highlight any discrepancies between the link capacity model and the achieved UDP throughput. At PHY rates 52 Mbps to 78 Mbps we consider frame exchanges with A-MPDUs sizes of 16, since the transmitter uses 2 medium accesses to deliver 32 packets.

Figure 3 shows the MAC efficiency, the ratio between the link capacity and the PHY rate, during tests with the MacBook. Under fixed A-MPDU sizes, the AP deliver more frames at higher PHY rates, resulting in more medium accesses and larger MAC overhead. We see this behavior between PHY rates 39 Mbps and 78 Mbps with $AGG(P) = 16$, and between PHY rates 104 Mbps and 130 Mbps, with $AGG(P) = 32$. We are able to accurately estimate the link capacity at all PHY rates. However, the link capacity estimation is slightly positively biased (1.925%, on average), due to the extremely optimistic model assumptions.

Figure 4 shows the MAC efficiency for tests with the Android Tablet. A-MPDU size reaches maximum value when
PHY rate \( \geq 26 \) Mbps, thus the downward MAC efficiency between PHY rates 26 Mbps and 65 Mbps. We observe that estimated LC differs from measured LC when PHY rate \( \geq 26 \) Mbps, showing a positive bias of up to 17% at PHY rate 65 Mbps. Our analysis of sniffer’s packet logs reveals that the AP takes additional time between A-MPDUs of size 8 (around 200\( \mu \)s). We made additional tests with other android devices with \( \text{MAX}_{\text{agg}} \) of 8 and 32, but we only observe this when \( \text{MAX}_{\text{agg}} = 8 \). When including the additional delay in the model, we obtain a smaller positive bias (3.96\%) on the link capacity (adjusted LC). We consider in the next section two link capacity models: one using purely protocol information (LC_original) and a second (LC_adjusted), which includes the observed additional delay for devices with \( \text{MAX}_{\text{agg}} = 8 \). Also, we consider the presence of a positive bias of 4\%, and we deduce it before usage.

V. LINK CAPACITY IN PRACTICE

This section adapts the model from §IV to work in practice. Rate adaptation algorithms frequently change the selected PHY rate [6] and frames may be lost (i.e., frame delivery < 100%). We first adapt the model from §IV to take these issues into account. Then, we discuss how to obtain the inputs for the model. Finally, we validate our model using controlled experiments and comparing it with the state of the art.

A. Model

As Wi-Fi link capacity varies over time, our model estimates the link capacity for a given time interval \([t_0, t_0 + \tau] \). Even though the PHY rate changes over time, the AP uses only one PHY rate for each frame. Thus, we can obtain “instant” link capacity measurements by applying the model from Equation 1. Let \( P(t) \) be the PHY rate used at \( t \) and \( FDR(t) \) be the frame delivery rate at \( t \). We estimate the link capacity for the time interval \([t_0, t_0 + \tau] \) as follows:

\[
\text{LC}(t_0, \tau) = \frac{1}{\tau} \int_{t_0}^{t_0+\tau} FDR(t) \times LC(P(t)) dt.
\]

B. Model inputs

The model in Equation 5 takes four inputs:

1) The initial estimation time, \( t_0 \), is simply the time operators will run link capacity estimation. In our existing trials with ISPs, we report the link capacity estimates periodically, but we can imagine scenarios where operators will request the estimate on demand as well.

2) The estimation interval, \( \tau \), depends on how frequently and fine-grained operators want to estimate link capacity. Values of \( \tau \) that are too large will average link capacity over a long time interval and may miss variations of link capacity that are important to diagnose Wi-Fi performance. On the other hand, if \( \tau \) is too small (for instance, less than a second) the variations of link capacity estimates in short periods of time become harder to interpret and may be user’s Wi-Fi performance.

3) The function of PHY rate over time, \( P(t) \). We obtain \( P(t) \) by periodically polling the Wi-Fi driver. Ideally, we would get the PHY rate per frame as previous work has done for OpenWrt APs [10]. However, per-frame measurements impose high load on the system, particularly during moments of high network load. We show that we can obtain low sampling error by periodically sampling the driver for the PHY rate of last transmitted data frame.

4) The frame delivery ratio over time, \( FDR(t) \). Similar to \( P(t) \), we obtain \( FDR(t) \) by polling the Wi-Fi driver. Unfortunately, \( FDR(t) \) is not available for upstream traffic. We can approximate upstream \( FDR(t) \) based on the downstream values, but this deserves further investigation, which we leave for future work.

We evaluate the sampling error of different \( \tau \) and \( \lambda \) parameters as follows. We emulate PHY rate sampling using the sniffer logs of the UDP iperf capacity tests, as described in §III. We periodically sample the PHY rate of the last data frame, with periodicity \( \lambda \) over an estimation period \( \tau \). We consider PHY rate sampling with periodicity 1 ms as the ground truth. We compare the link capacity sampled with parameters \( \lambda \in \{1 \text{ s}, 0.3 \text{ s}, 0.1 \text{ s}, 0.03 \text{ s} \} \) and \( \tau \in \{1 \text{ s}, 10 \text{ s}, 30 \text{ s}, 60 \text{ s} \} \) with the ground truth, obtaining the sampling error. We execute 100 runs, randomly choosing the starting time and report average sampling error and standard deviation.

We can see in Figure 5 that sampling error decreases for larger \( \tau \) and smaller \( \lambda \). This is expected since estimation error tends to decrease with more PHY rate samples, which is given by \( \tau/\lambda \). Therefore, there is a trade-off between sampling...
overhead, imposed by $\lambda$, and estimation granularity, imposed by $\tau$. We see that sampling error is, on average, below 2% when $\tau/\lambda \geq 30$.

C. Validation

In order to evaluate our algorithm, we performed controlled experiments in the anechoic chamber. We performed UDP iperf capacity tests as described in §III, using $\tau = 10$ s.

Comparison method. We compare our method with the state of the art in Wi-Fi throughput estimation “Wi-Fi based TCP throughput” (Witt) [10]. Witt is calculated by linearly fitting a custom metric, link experience, with TCP throughput ground truth data. Link experience and Witt are given by:

$$\text{link}_\text{exp} = (1 - a) \times (1 - c) \times \sum \frac{P_t}{P_i}$$  \hspace{1cm} (6)

$$Witt = \beta_1 \times \text{link}_\text{exp} + \beta_0$$  \hspace{1cm} (7)

where $a \in [0, 1]$ is a percentage of airtime utilization used by external sources, and $c \in [0, 1]$ accounts for local contention. While originally Witt is used to predict TCP throughput, we fit Witt to predict UDP throughput.

Both LC and Witt have very high correlation coefficients with the throughput, respectively .997 and .996. A key difference between both methods is that Witt finds the ratio between PHY rate and data rates (the MAC efficiency) by fitting $\beta_1$ on a data set and minimizing errors with the intercept $\beta_0$. We calculate the MAC efficiency per PHY rate. We see this when comparing Witt’s original ($\beta_0, \beta_1$) parameters (−.494, .733) with parameters found by fitting only the tablet data (2.502, 0.629) and only the Macbook data (0.875, 0.783). Witt cannot distinguish .11n devices with low and large frame aggregation usage, causing less accurate throughput predictions.

We compare the estimated link capacity using 4 methods: 1) Witt with original $\beta$ values; 2) Witt with $\beta$ values found by fitting this device’s fixed PHY rates tests (Witt: reference); 3) LC with original values (LC: Original) and; 4) LC considering the AP instance inefficiencies at high PHY rates, as seen in §IV (LC: Reference).

Figure 6 shows the distribution of estimation errors of the 4 different approaches. As expected, training the prediction method with reference data from the device under test significantly reduce prediction errors. However, training per station is impractical in operational environments. Over 90% of the predictions had and error below 15% for original LC, and over 95% presented error below 5% when using LC with reference data. We conclude that, in order to obtain more accurate link capacity estimations, the model should be tuned to properly incorporate performance inefficiencies on devices with restricted A-MPDU capabilities (i.e., $MAX_{\text{AGG}} \in \{8, 16\}$) as shown in §IV. Fortunately, this has to be done once per AP model.

VI. DIAGNOSIS OF THROUGHPUT BOTTLENECKS

In this section, we present how we can use the proposed link capacity to diagnose downstream throughput bottlenecks. We show that, by using the estimated link capacity defined in §V in conjunction with other AP metrics, we can estimate the available bandwidth, which helps identifying throughput bottlenecks. Further, we show how we can diagnose instances of reduced available bandwidth.

A. Medium Access and Frame delivery losses

Figure 7 illustrates how the MAC overhead, frame delivery losses (FD), and medium access losses (MA) explain the difference between the nominal physical link capacity and the available bandwidth.

The link capacity defined in §V estimates how much bandwidth the link supports assuming full medium availability, but in reality we share the unlicensed medium with many Wi-Fi and non-Wi-Fi sources. Wi-Fi cards can export, for a given period, the percentage of time the medium was busy due to the reception of nearby Wi-Fi frames ($BUSY_{\text{win}}$) or to the presence of high noise ($BUSY_{\text{nonwin}}$). We use these metrics to account for medium sharing. The AP we study report these counters with a resolution of 1%. We define medium access losses (MA) as the fraction of LC lost due to busy medium, and available bandwidth (AB) as the fraction of LC available for usage.

$$MA = LC \times (BUSY_{\text{win}} + BUSY_{\text{nonwin}})$$  \hspace{1cm} (8)

$$AB = LC \times (1 - BUSY_{\text{win}} - BUSY_{\text{nonwin}})$$  \hspace{1cm} (9)

Consider an ideal Wi-Fi link, with no frame loss nor medium sharing. This link would present the maximum LC
VII. RELATED WORK

There is a large body of work on throughput estimation techniques in Wireless networks. Jun et al. presents an analytical model to calculate the theoretical maximum throughput of IEEE 802.11a, b and g [4]. Skordoulis et al compare the different frame aggregation mechanisms proposed in IEEE 802.11n, giving simulation results for the maximum throughput [15]. These models, however, always consider fixed PHY rate usage. We extend these models by considering IEEE 802.11n parameters and estimating the link capacity when PHY rate varies, while experimentally validating the results.

Active measurement methods have been proposed for estimating the available bandwidth in wireless networks. Lakshminarayanan et al. proposes Probegap, a probing technique to estimate bandwidth in multi-rate scenarios, such as Wireless networks [7]. Mingzhe Li et al. proposes WBTest [9], a tool that uses packet-pairs and packet-trains to determine achievable throughput in IEEE 802.11 networks. Those techniques are unfit for large scale ISP deployments, since the active measurements can disrupt users’ traffic as well as require cooperation between endpoints.

Many works have characterized wireless performance in the wild using the AP point of view [2], [11], [16], often relying on passive measurement collection. Patro et al. proposes Witt [10], a metric that estimates available bandwidth based on passive metrics from APs, using it to characterize the Wi-Fi quality in 30 homes. We obtain estimation errors similar to Witt, but our model is less sensitive to parameter tuning.

There is work on diagnosing wireless performance problems in enterprise and campus networks, where multiple APs are managed by the same entity [2], [3]. These solutions often require combining multiple points of view to diagnose performance problems, while we can only rely on the single home AP. Rayanchu et. al proposes a method to diagnose frame collision from weak signal [13]. This method requires modifications to the Wi-Fi kernel of both the AP and the device. Lakshminarayanan et al. proposes the use of a second NIC to allow users to diagnose the medium usage, allowing for accurate diagnosis of Wi-Fi and Non-Wi-Fi interferences [8]. Syrigos et al. proposes the use of active measurements with fixed PHY rate to diagnose common 802.11 pathologies [18]. Kanaparthi et al. proposes using user-level probing to diagnose common WLAN performance problems [5]. Shrivastava et al. proposes AINShark, a system which is able to detect different non-Wi-Fi interference sources by using energy samples from the Wi-Fi card [14]. These methods are able to diagnose Wireless problems with a great degree of precision, but either require the disruption of the user activity with active tests [5], [18] or a second NIC for dedicated spectrum monitoring, limiting deployment [8], [14]. We instead focus on providing high level diagnosis, distinguishing between medium access and frame delivery throughput losses.

VIII. CONCLUSIONS AND FUTURE WORK

We presented an algorithm to estimate the link capacity based on passive metrics from APs, which is ready to be
deployed at scale. We show that it is possible to estimate the link capacity per PHY rate based on a limited set of parameters related to the particular AP instance. Then, we extend the initial model to estimate the link capacity when the PHY rate varies. We measured the link capacity in different link quality conditions and found that more than 90% of the estimations present error below 15% without prior parameter tuning, and more than 95% present estimation error below 5% with appropriate parameter tuning using fixed PHY rate tests. Also, our method achieves below 2% sampling errors when the ratio $\tau/\lambda \geq 30$.

We would like to use the proposed model to understand the home wireless performance of real users. Some of the monitoring capabilities described are already in trial in two large two major European ISPs and one major ISP in Asia-pacific. We plan to use this proposed model to perform a characterization of the Wireless performance in these home networks, answering questions such as “how often is wireless performance poor?” and “what is the most common root cause for reduced wireless performance?”. Finally, we want to see how available bandwidth estimations correlate with QoE metrics, which more closely reflect the user experience.

ACKNOWLEDGEMENTS

We thank Gabriel Matei and Ali Louizir for helping with the anechoic chamber experiments. Thanks also to our shepherd Pedro Casas and the anonymous reviewers for their helpful comments. This work was supported by the European Community Seventh Framework Programme (FP7/2007-2013) no. 611001 (User-Centric Networking).

REFERENCES


Online Identification of Last-Mile Throughput Bottlenecks on Home Routers

Michele Pittoni, Anna-Kaisa Pietilainen, Renata Teixeira, Srikant Sundaresan, Nick Feamster
1INRIA, 2Samsara Networks/CSI, *Princeton University

Abstract

We develop a system that runs online on commodity home routers to locate last-mile throughput bottlenecks to the home wireless or the access ISP. Pinpointing whether the home wireless or the access ISP bottlenecks Internet throughput is valuable for home users who want to better troubleshoot their Internet experience; for access ISPs that receive numerous calls from frustrated home customers; and for informing the debate on regulating the residential broadband market. Developing such a system is challenging because commodity home routers have limited resources. The main contribution of this paper is to develop a last-mile throughput bottleneck detection algorithm that relies solely on lightweight metrics available in commodity home routers. Our evaluation shows that our system accurately locates last-mile bottlenecks on commodity home routers with little performance degradation.

1 Introduction

With the availability of cheap broadband connectivity, Internet access from the home is ubiquitous. Modern households host many networked devices, ranging from personal devices such as laptops and smartphones to printers, media centers, and a number other Internet of Things (IoT) devices. These devices often connect with each other and to the Internet via a wireless home network; this connectivity has become an important part of the “Internet experience” [22, 30]. Unfortunately, home users have few means to identify when their home network bottlenecks their Internet performance, and hence often attribute poor performance to the access ISP. Access ISPs are in no better position to determine the cause of performance bottlenecks in the last mile, and yet their hotline must answer numerous calls from unsatisfied customers. Tools for correctly pinpointing whether the home wireless or the access ISP bottlenecks Internet throughput are valuable not only to home users and ISPs, but also to inform the wider debate on regulating the residential broadband market.

In this paper, we develop a system to locate last-mile throughput bottlenecks—which we define as throughput bottlenecks either in the access ISP or in the customer’s home. Clearly, throughput bottlenecks may happen elsewhere (e.g., at peering interconnects [18]). We focus on bottlenecks that are close to users as these more likely affect all of a user’s traffic and users can take direct actions to remedy them.

Last-mile throughput bottlenecks are difficult to identify, for two reasons. First, ISPs lack tools that run within the customer premise, which is necessary to accurately identify last-mile bottlenecks that are caused by the home network. Some tech-savvy home users can run a set of tools (as we discuss in more detail in §7) to help identify last-mile throughput bottlenecks in an ad-hoc fashion, but this ad-hoc procedure is too complex for many home users [11]. Second, throughput bottlenecks are often intermittent, because they depend on the interaction among user traffic, wireless quality of the different devices in the home network, and the access link performance. As a result, single tests will likely fail to capture many bottlenecks that affect user experience. A solution that runs continuously is key to identifying last-mile bottlenecks.

To address these challenges, our system runs directly on commodity home routers. The home router is between the home and the access network and hence is ideally placed to locate last-mile bottlenecks. The home router is typically always on, thus permitting continuous monitoring. Our previous work [30] developed a proof-of-concept algorithm, called HoA (for Home or Access), that ran offline on a server to locate last-mile downstream throughput bottlenecks based on the analysis of packet traces collected from home routers. Our attempts to run HoA online on commodity home routers, however, revealed the challenges with performing per-packet analysis on such resource-constrained devices (§2).

The main contribution of this paper is a last-mile throughput bottleneck location algorithm that runs online in commodity home routers. We design an access bottleneck detector based on lightweight pings of the access link (§4), and a wireless bottleneck detector based on a model of wireless capacity using metrics that are easily available in commodity home routers such as the wireless physical rate and the count of packets/bytes transmitted (§5). We evaluate the accuracy of our detectors using controlled experiments (described in §3), and show that we can detect both downstream and upstream throughput bottlenecks with more than 93% true positive rate and less than 8% false positive rate. Our performance evaluation in §6 shows that our algorithm runs online with less than 30% load average on a Netgear router.
2 Design Constraints and Choices

In this work we aim to design a throughput bottleneck detector for the last mile that has the ability to distinguish between access-link and wireless bottlenecks, that operates with minimal network and system overhead, and with minimal disruption to the end user. Our goal narrows down potential design choices and introduces certain constraints, which we list below.

Design choice: The detector must be located inside the home. To identify throughput bottlenecks in the home wireless or the access network, we need a vantage point inside the home network. This is because other vantage points typically do not get a view inside the home network. A vantage point in the access ISP, for instance, may be able to detect that traffic is being bottlenecked somewhere in the end-to-end path (for instance, using a tool such as T-RAT [31]), but it will be unable to identify that the last mile is the cause, let alone localize it to the access link or the home wireless network. Inside the home network, end clients have a view of the wireless network, but not a direct view of the access link, or even a full view of the wireless network; it may not be able to observe traffic between the home router and other clients in the network. The home router, on the other hand, is the ideal vantage point for locating last-mile bottlenecks as it sits in between the home wireless and the access network and hence can directly measure the performance of each network independently. Moreover, the home router is always-on allowing for the identification of intermittent bottlenecks.

Design constraint: The detector must introduce minimal overhead. The system should not disrupt users. This implies that we must avoid introducing too much traffic in either the access link or the home wireless. This rules out conceptually the simplest detector, that would simply run an active throughput test from the router to some well-connected server in the Internet and to devices within the home and compare the two measurements. Such tests introduce high overhead that is intolerable in one-shot tests, but not continuously. Since wireless conditions are highly variable, the detector would need to run continuously. Probing devices in the home wireless is also problematic. Not only might probe traffic interfere with user traffic, or even change the wireless network itself by introducing side effects such as contention, but answering to these probes may drain device battery. Finally, given our system is running on the home router, we must also avoid overloading the router’s CPU, which could lead to drops in user traffic.

Design constraint: Passive packet capture is not feasible. Since active throughput measurements are not viable, another natural approach is to passively observe user’s traffic as it crosses the router, as in HoA [30], which examines per-packet arrival rates and computes per-flow TCP metrics to locate last-mile downstream throughput bottlenecks. The existing implementation of HoA collects packet headers on the router, but offloads actual bottleneck identification to a server. This offline approach is feasible in a small-scale deployment for research purposes, but not as a large-scale operational solution due to the volume of data, and also due to privacy concerns. Unfortunately, our attempts to run HoA online on the router have failed mainly because of the overhead of analyzing each packet online. Our study of the fraction of dropped packets when we run tcpdump on a TP-Link WDR3600 access point as we vary the traffic load illustrate this problem. We run tcpdump in two modes: “save” refers to only storing the packets in memory (e.g. `tcpdump -w`); whereas “pipe” refers to the case where tcpdump parses the headers with no other per-packet computation (when omitting the `-w` flag). We see that using tcpdump to only write packets in HoA’s implementation causes almost no drops, but when we parse packets in the router we start seeing approximately 10% packet drops at 30 Mbps. This rate increases to 80% at 70 Mbps. High rates of packet drops will make the system unreliable at identifying bottlenecks at higher speed links.

Design choice: Lightweight active and passive metrics at the wireless router. These design constraints lead us to two types of lightweight measurements. First, we consider lightweight active measurements, for example, low frequency RTT measurements that should not disrupt users. Second, we consider metrics available by polling the router operating system. Most commodity routers report a number of statistics such as the number of bytes/packets transmitted and received, queuing statistics, and wireless quality metrics (such as PHY rate, RSSI, frame delivery ratio). We will discuss the specific metrics we use for detecting access bottlenecks and wireless bottlenecks in §4 and §5, respectively. In the next section we explain the experimental setup we use to train and test these detectors.

3 Experiment Setup

We develop and test our algorithm using a simple testbed to recreate different access link and wireless quality scenarios. Our testbed consists of two TP-Link WDR3600 access points, a test server, and a Mac laptop. The access points have a 560 MHz MIPS CPU, 128 MB of RAM, and two wireless interfaces (2.4 GHz and 5 GHz). Both routers run version 15.05 of the OpenWRT firmware. We connect one access point downstream of the other over an Ethernet cable; the downstream access point acts as wireless router, while the upstream access point acts as a traffic shaper to emulate various access link scenarios. We connect the test server to the upstream server via a Gigabit Ethernet switch. Finally, we connect the Mac laptop over wireless to the downstream access point.

We conduct all our experiments over 802.11n. Although the access points support both 2.4GHz and 5GHz, we use only the 5 GHz band to limit the amount of interference from external sources. We created a variety of wireless scenarios that capture a range of performance ranging from poor to excellent. We do so by moving the client between different rooms, at various distances from the access point, as well as
by creating interference. To add interference, we add a second client that sends constant bitrate UDP traffic at 100 Mbps to a separate access point on the same channel that the testbed was using. Using these methods we achieved a range of wireless capacities ranging from less than 20 Mbps to almost 100 Mbps. We tested both 20MHz and 40MHz channels; in the latter case, though the maximum (theoretical) capacity is 300 Mbps, the access point itself becomes a bottleneck at about 100 Mbps. We note here that these numbers are approximate, and reveal only a general notion of the quality of the wireless; this is because of the inherent variability associated with wireless network performance.

For the access link, we tested different levels of throughput limitations, ranging from 10 to 90 Mbps, by controlling the kernel queuing disciplines on the shaper with tc.

In total, our experiments span 15 scenarios for a total of more than 7 hours of runtime. For each scenario, we tested the throughput by sending TCP traffic from the server to the client with iperf3. We tested both a single flow and multiple parallel flows, with long (5 minutes) and short (30 s) runs. We labeled each scenario as either access bottleneck or wireless bottleneck based on the prior measurement of the wireless link capacity and on the access throughput limit set in the shaper.

Figure 1: Diagram of the experiment tested

![Diagram of the experiment tested](image)

4 Access Bottleneck Detection

In this section, we identify, tune, and validate metrics to detect upstream and downstream bottlenecks in the access-link. We show using extensive controlled experiments that lightweight metrics suffice to detect such bottlenecks with high accuracy.

4.1 Metrics

In §2, we discussed the need for lightweight metrics, and how that rules out throughput tests. Instead, we identify an intuitive property of bottlenecked links that we can exploit: increased packet queuing at the bottleneck link buffer. An obvious way to exploit this metric would be to poll router for queuing statistics to directly identify the presence of queues. The problem with this is that it only works to detect upstream bottlenecks and in cases where the access point is also the modem. In many setups the modem/router and the AP are physically separate devices. In such cases the queue will not increase in the AP if the access link is the bottleneck.

We instead use the result of queue buildup: increased queuing delay at the bottleneck link. This principle has been used in congestion control protocols, especially those aimed at low priority traffic (e.g., LEDBAT). It has also been confirmed by studies that have observed home network performance from home routers [28]. In our case, we want to isolate the delay of the last mile link. To do so, we identify the first hop inside the ISP’s network and probe it with ping. In this way, the ping packet traverses access link in both direction and pass through the queues at both ends. Assuming FIFO queuing, the round-trip time (RTT) is proportional to the sum of the queue lengths. These measurements, in addition to being lightweight, will also capture both upstream and downstream bottlenecks. A few samples during a bottleneck episode will typically suffice to indicate the presence of a bottleneck.

4.2 Detection Algorithm

The access bottleneck detector keeps an estimate of the minimum and maximum RTT observed on the access link (respectively, $d_{\text{min}}$ and $d_{\text{max}}$) and compares each new sample to those values. We consider samples that are higher than a threshold as positive. The detector periodically decides whether there was an access bottleneck based on the ratio of positive samples in the last period.

Intuitively, $d_{\text{max}}$ corresponds to probe propagation delay plus transmission delay with no queuing delay, whereas $d_{\text{max}}$ captures the probe delay when buffers are full, so maximum queuing. We estimate $d_{\text{min}}$ based on the minimum measured RTT over a moving window, and $d_{\text{max}}$ based on the maximum RTT over the same window. We pick a long enough window to increase our chances of encountering both queues empty and full during the window; yet, if the window is too long underlying conditions might change. In our algorithm, we use a window of one hour.

Given an estimation period, $T$, at time $t$ the detector considers the set of samples collected in the interval $(t - T, t]$, denoted $S_t(t)$. We identify the set of positive samples as follows.

$$ P_t(t) = \{ s \in S_t(t) : s > d_{\text{min}} + (d_{\text{max}} - d_{\text{min}}) \cdot \delta_t \}, $$

where $\delta$ is a measured RTT and $0 < \delta < 1$ is a threshold.

We detect an access bottleneck at time $t$ if the fraction of positive samples is higher than a predefined threshold $\rho_A$:

$$ \frac{|P_t(t)|}{|S_t(t)|} > \rho_A. $$

The access bottleneck detector relies on four parameters: the estimation period, $T$, the sampling rate, $r$, and the thresholds, $\delta_t$ and $\rho_A$. Our goal is to select $T$ short enough to capture traffic bursts that will trigger network congestion. At the same time, we must have enough samples during $T$ to run access bottleneck detection. If $T$ is too small the rate $r_A$ must increase. As we will discuss in $\S 5$, practical limitations with capturing the wireless statistics prevent us from setting intervals shorter than 10 seconds and we must have the same...
representation of wireless state encountered by users’ traffic, as we discuss in §2. We instead poll the wireless driver (ath9k in our case) to obtain metrics that map to link quality. We then feed these metrics to a model that estimates wireless link capacity. We note that the estimated capacity relies on statistics that are obtained from the driver based on user traffic, and are therefore likely to reflect actual performance that users get. Our metrics, which we list below, are simple to obtain and are available in most drivers.

We use our estimate of wireless capacity with the achieved throughput to detect the presence of a bottleneck. Since wireless capacity varies across different clients connected to an access point, we poll the driver for the metrics related to each client. The specific metrics we choose are the physical layer (PHY) bitrate of the last frame sent to the client, frame delivery ratio (which captures the fraction of frames successfully delivered to the client), and the total number of bytes sent. We also collect non-client-specific metrics, such as the fraction of time during which the access point sensed that the channel was busy.

With these metrics, we can estimate the wireless link capacity by using a state of the art model [13]. The model was developed to give an upper bound on capacity for UDP traffic. It estimates the instantaneous link capacity for a PHY rate, $P$, based on number of aggregated frames at $P$ times the size of a maximum-sized UDP packet and then dividing by the delay to transmit the set of aggregated frames at rate $P$. Then, it estimates the link capacity for a period, $T$, as the average of the instantaneous capacities for the sampled PHY rates in the period times the corresponding frame delivery ratio. We make two modifications to this model. First, we adapted it for TCP traffic, factoring in the increased overhead due to TCP acknowledgments. Second, we remove the highest 10% PHY rate samples for each estimation period. The Minnstrael rate adaptation algorithm used in our access point sends 10% of “look around” frames by default to test if the medium improved [19], which were causing the model to overestimate the link capacity.

5.2 Detection Algorithm
The wireless bottleneck detection algorithm consists of a simple comparison between the estimated wireless link capacity and the achieved throughput. We compute the latter from the device byte counters over time.

Similar to the access bottleneck detector, there is an estimation period $T$ and the detector makes a decision based on the samples collected during the previous period. Let $LC(t,c)$ be the estimated wireless link capacity for client $c$, and $ptu(t,c)$ the observed throughput over the time interval $[t - T, t]$ for $c$. The detector flags that $c$ is experiencing a wireless bottleneck at time $t$ if the following test holds.

$$\frac{LC(t,c) - ptu(t,c)}{LC(t,c)} < \delta_w,$$

where $0 < \delta_w < 1$ is a threshold.
per flow on the wireless bottleneck. To make our comparison easier, we apply HoA to detect bottlenecks every 10 seconds as we do with our system.

HoA considers a threshold $T_{\text{wr}} = 0.8$ on the coefficient of variation $cv$ of packet inter-arrival times to detect access-link bottlenecks. In our experiments, we tested various thresholds and achieved true positive rate higher than 95% and false positive rate lower than 5%, in the range between 0.7 and 1.8. The best tradeoff is when set $T_{\text{wr}} = 1.4$, which achieves 99% true positives and 2% false positives. Our detector achieves the same true positive rate with only slightly better false positive rate of 1% (in §4.3).

The wireless bottleneck detector relies on three parameters: the estimation period, $T$, the sampling rate, $r_w$, and the threshold, $\delta_w$. The capacity model we use recommends having at least 30 samples per estimation period, and we found that polling the router more often than three times per second had significant impact on performance. The CPU load increases linearly as we increase the sampling rate, until we reach three samples per second when CPU load increases exponentially. Hence, we set $r_w=3$ samples/second and $T = 10$ seconds. We discuss the setting of $\delta_w$ next.

5.3 Accuracy

We use the data from our controlled experiments to evaluate the accuracy of the wireless bottleneck detector. Figure 3 shows the ROC curve for the wireless bottleneck detector with $\delta_w \in [0.2, 0.49]$. We test relatively large values of $\delta_w$ because the estimated capacity is an upper bound on the available bandwidth. In reality, we found that the achieved throughput is lower, for example due to TCP dynamics. The $\delta_w$ parameter compensates for this overestimation.

As can be seen in Figure 3, the accuracy of this detector is not as good as the access link detector. This can be partly attributed to imprecise capacity estimations and partly to the difficulty of achieving the theoretical available throughput. We selected $\delta_w = 0.36$, which gives a true positive ratio above 93% and a false positive ratio below 8%, as the threshold for this detector.

6 Evaluation

In this section, we first evaluate the accuracy of our system when compared with the state of the art, HoA [30], using the data from our controlled experiments. Then, we deploy our system in BISmark routers to evaluate its performance overhead in practice.

Comparison with HoA. We apply the HoA detectors on the same set of controlled experiments we used to evaluate the accuracy of our system. Originally, HoA made a decision every second on the access link bottlenecks and one decision...
Figure 4: Distribution of the load average on B1Smark routers before and after the deployment

7 Related Work

Measuring and diagnosing network performance issues has a long history that has spanned many types of networks and performance metrics. In this section, we briefly discuss approaches that focus on the last mile and on throughput.

Residential Access Performance Analysis. Tools such as Ookla’s speedtest [21], NDT [5], or Netlyze [16] can help residential users measure the throughput achieved from an end-host connected from within the home network. These tools measure end-to-end throughput to a server “close by,” but they do not localize whether the achieved throughput is bottlenecked in the home or access network. The last years has seen a number of studies of broadband access performance [4,9,10,28]. In particular, Sundaresan et al. [28] study residential access performance from home routers. All these studies, however, focus on inferring the capacity of access links, and not on identifying whether the access link is the throughput bottleneck.

Wireless Performance Analysis. Many approaches to diagnosing wireless networks rely on multiple monitoring points [1,3,7,14,20,23] or custom hardware [6,17,23–25]. These are difficult to deploy in the home network setting since they require deploying equipment beyond what a normal user is typically willing to install or have installed in their home. In contrast, our algorithm runs on commodity access points. WiSlow [15] is a tool that analyzes wireless metrics collected at end-hosts to identify root causes of wireless performance problems. WiSlow is a nice complement to our system as once we identify a wireless bottleneck, we can run WiSlow on one of the hosts in the home network to identify the root cause. A recent study of wireless performance in homes [29] also runs on access points. This study focuses on correlating the achieved TCP throughput and the corresponding metrics at the wireless layer, and not on methods to identify when the wireless bottlenecks throughput. Our wireless bottleneck detector described in §5 relies on the Wi-Fi capacity estimation model from Da Hora et al. [13]. This model works from commodity home access points, but it does not detect when home wireless bottlenecks throughput.

Home Network Analysis. A number of measurement efforts have characterized home networks in terms of connected devices and usage [8,12,26]. None of these studies, however, have identified when the home network bottlenecks throughput. Netprints [2] is a diagnostic tool for home networks that solves problems arising due to misconfigurations of home network devices including routers. The closest to our system is HoA [30], a system that analyzes packets crossing the home router to localize downstream throughput bottlenecks to either the access link or the home wireless network. As we discussed in §2, however, commodity home routers have limited resources and hence cannot sustain per-packet analysis as traffic rates increase. Our system, on the other hand, relies only lightweight metrics and is hence able to detect throughput bottlenecks online on commodity home routers.

8 Conclusion

In this paper, we developed a system that runs on commodity home routers to locate throughput bottlenecks to the home wireless or the access link. The main contribution of our work is to develop a system that runs online under the practical constraints imposed by commodity home routers. Our access bottleneck detector relies on low frequency probing of the access-link RTT to detect packet queuing that occurs when the access link is bottlenecked. This detector achieves a 99% true positive rate for a 1% false positive rate. Our wireless bottleneck detector relies on polling the router for traffic and wireless quality statistics. The accuracy of wireless bottleneck detection (93% true positive with less than 8% false positive ratio) is lower than that of access bottleneck detection, but still good to be useful in practice. Overall, our detectors are as accurate as the state of the art, HoA, which relied on per-packet analysis, and hence could not run online on commodity routers. Our evaluation in the B1Smark platform showed that although the load average on the routers after the deployment of our system increased moderately, the vast majority of measurements show less than 30% load average.

In future work, we are considering ways to improve the accuracy of the wireless bottleneck detector, for example by integrating lightweight active measurements only when we detect traffic. Another possibility is to maintain an estimate of the access link capacity by measuring the throughput when an access bottleneck is detected. This could be used to avoid false detection of a wireless bottleneck when the capacity of the two links is close. We are also aiming for a more extensive experimental study in realistic scenario, for example with different access link technologies. We are aware that active queue management (AQM) algorithms may invalidate some of our assumptions. AQM does not present a problem in current access links, but as standards and technology change, we will continue to evaluate the accuracy of our method.
References


