Real-time probing of end-to-end capacity and available bandwidth in heterogeneous local networks

Real-Time Probing of End-to-End Capacity and Available Bandwidth in Heterogeneous Local Networks

Archi Delphinanto, Ton Koonen
Electrical Engineering Department
Eindhoven University of Technology
Eindhoven, The Netherlands
[A.Delphinanto, A.M.J.Koonen]@tue.nl

Shuang Zhang, Frank den Hartog
TNO
Delft, The Netherlands
[Shuang.Zhang, Frank.denHartog]@tno.nl

Abstract— **Current QoS solutions for home networks are inspired by Internet technologies and typically based on traffic classification and prioritization. They are not well adopted because they need to be supported by every device in the end-to-end path to be effective. Besides, their use still does not guarantee that enough bandwidth will be available between a server and a client. It is therefore preferable to diagnose a home network in real time before admitting a new flow. We demonstrate a new method which probes the end-to-end path capacity as well as the available bandwidth in a small-scale IP network. Its distinguishing features are the applicability to networks consisting of legacy devices and any mixture of wired, wireless, and no-new-wires links, in combination with fast convergence, high accuracy, and non-intrusiveness.**

I. INTRODUCTION

A process of convergence is currently taking place in many consumer Local Area Networks (LANs), such as home networks, but also in-car networks and personal area networks. This means that many different devices will be able to intercommunicate over a combination of different networks, in contrast to the traditional single service/device/network stove-pipe systems. The direct consequence is that home networks are becoming ever more heterogeneous and that a single network consists of many different physical- and link-layer technologies and topologies, interconnecting many different devices with each other and the Internet, enabling many different services. The Internet Protocol (IP) suite is the main enabler for the required interoperability. Correspondingly, a growing amount of consumer electronics devices contain an IP stack.

To prioritize the traffic flow of one service over the other, many Quality-of-Service (QoS) solutions are available, but they do not seem well adopted in home networks. They do not provide any guarantee that enough bandwidth will be available between a server and a client. Furthermore, they need to be supported by every device in the end-to-end (e2e) path to be effective, which is relatively expensive for the owners of home networks. It is therefore preferable to diagnose any network in real time before admitting a new flow. We developed and implemented a new method (Allbest) that probes the capacity as well as the available bandwidth between a server and a client in home network, or rather any small-scale IP network. The distinguishing features of this tool compared to other probing methods are its applicability to networks consisting of legacy devices and any mixture of wired, wireless, and no-new-wires links, in combination with its fast convergence time, high accuracy, and non-intrusiveness. This paper describes the demonstration set-up of Allbest, after a short overview of the main characteristics of its probing method.

II. THE ALLBEST PROBING METHOD

Our capacity estimation method is based on the packet-pair dispersion technique, which is usually implemented by sending two packets back-to-back on the network, thus minimizing the chance that crossing traffic will disperse the packets. It is the bottleneck link in the path that will delay the second packet with respect to the first. C can then simply be calculated from the minimum dispersion D and the packet size L as $C = L/D$. Assuming the crossing traffic stochastic, a series of n packet-pair probes is usually performed to maximize the chance of finding a packet pair of which crossing traffic hasn't influenced the bottleneck dispersion further down the path. In contrast to the current literature, we [1] then calculate D from the minimum RTT of the first packet (RTT_1) of a probe pair and the minimum RTT of the second packet (RTT_2) of the pair, yielding for C:

$$C = L/(\min_{i=1...n}[RTT_2(i)] - \min_{i=1...n}[RTT_1(i)]) . \qquad (1)$$

RTTs can be measured without adaptation of the client side by sending MTU-sized UDP packets to a non-activated port. The client then automatically generates ICMP "Destination port unreachable" packets. These reply packets are relatively small, and therefore often experience hardly any delay on the way back to the probing sender/receiver. For symmetric media we may also use ICMP Ping probing packets (which generates equally sized ICMP Echo packets), and assume that the delay and dispersion is the same for both directions of travel. Equation (1) then yields $C/2$.

Existing packet-pair dispersion techniques do not work in wireless media in round trip, because the reply packet of the first probe packet contends with the second probe packet on the air interface [1]. To avoid this contention, we are sending a single packet with size 2xMTU, instead of two packets back-to-back. On the network, this packet will be fragmented (and behave like two individual packets back-to-back), and only after defragmentation a single reply will be sent back by the client. This will give us the correct RTT_2. We can find the correct RTT_1 by sending a separate series of single probe packets (i.e. not back-to-back) with size MTU. The available bandwidth A can then be calculated from [2]:

$$A = L / (\min_{i=1...n}[RTT_2(i)] + \operatorname{avg}_{i=1...n}[RTT_1(i)] - 2\min_{i=1...n}[RTT_1(i)]) \quad (2)$$

This is the first tool that successfully applies probe round-trip-time (RTT) measurement to wireless LAN and does not assume any home network topology *a priori*. It only needs a simple software upgrade of the server side of the e2e path of a flow within the home network, which for many use cases will be the home gateway. The results presented in [1] and [2] show that it is fast, and accurate enough to make informed decisions about the admission of IPTV streams.

III. Description of the Demonstrator

Allbest consists of a home-built configurable UDP packet generator and likewise ICMP Ping packet generator, combined with Wireshark [3] to measure RTTs with high-precision. Any RTT > 2 x min[RTT(i)] is discarded, and we have verified that most of those long RTTs are caused by uncontrollable processing delay in the laptops due to other tasks of the operating system. A measurement takes as many probe pairs as dictated by the desired accuracy, but with a maximum of 10 seconds, and thus is instantly informing the users of the network status.

The demonstrator is shown in Fig. 1. The Allbest server runs on the “Allbest Prober” laptop, and it probes the client via a heterogeneous topology [2]. A computer running a D-ITG traffic generator [4] sends crossing traffic to the client. A network attached storage (NAS) is used as a media server for streaming a movie to the client. XBMC [5] installed on the client is the corresponding media player. The NAS server, traffic generator and Allbest prober are interconnected with 100 Mbit/s Ethernet. The client is connected via the bottleneck link.

During the demonstration we will first configure IEEE 802.11g (Linksys WRT54GL v. 1.1), and later HomePlug (Linksys Powerline AV) as bottleneck links. We switched off 802.11g’s automatic rate adaptation and Clear to Send (CTS) protection mode and run the network on the maximum physical rate of 54 Mbit/s. While Allbest is performing the probing, the graphical user interface (GUI) of Allbest will show the capacity as well as the available bandwidth of the path from the prober to the client.

The demonstrator will be shown from a cold start. The D-ITG generator and the NAS server are turned off, while the prober starts probing the client. The GUI on the prober will show the capacity and the available bandwidth of that path. Despite the inevitable 2.4 GHz interference in the demonstration environment, the estimated path capacity will be around 38 Mbps, which is the expected throughput of IEEE 802.11g on the network layer [1].

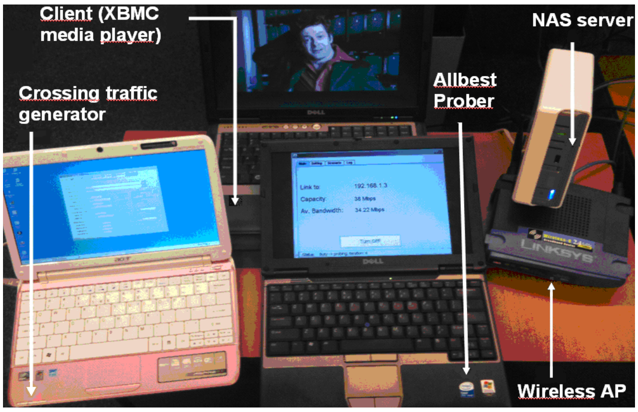

Figure 1. Our Allbest demonstrator. The probe generator runs on the Allbest Prober laptop. The crossing traffic generator is a netbook running D-ITG. Path capacity and available bandwidth estimation are shown on the GUI of the prober, while a video is being streamed from the NAS server to the XBMC media player on the client.

The available bandwidth will be whatever the interference in the demonstration room dictates. It will most probably be enough to stream a 4 Mbit/s MPEG-4 movie from the NAS server to the client, which we will therefore start. As the streaming goes on, the Allbest GUI will display the (unchanged) path capacity and the new, reduced available bandwidth. With D-ITG we then inject crossing traffic to the path being probed. We adjust the amount of crossing traffic until it is depleting the bandwidth resource of the path. This will be indicated by an estimated value of the available bandwidth close to what is minimally needed for the video stream not experiencing packet loss, and obvious degradation of the video quality on the media player.

The whole procedure is then repeated with HomePlug. The bottleneck link is created by two Homeplug adapters plugged into a multiple-socket extension cord. The extension cord is connected to the mains power socket in the demonstration hall via a low-pass filter, to keep external noise out of our system. We will also probe the capacity and available bandwidth of the conference’s Wifi network.

References

[1] A. Delphinanto, T. Koonen, S. Zhang, and F. den Hartog, “Path Capacity Estimation in Heterogeneous, Best-effort, Small-scale IP Networks”, Proc. of the 35th IEEE Conference on Local Computer Networks (LCN 2010), Denver, CO (USA), October 2010.

[2] A. Delphinanto, T. Koonen, and F. den Hartog, “End-to-End Available Bandwidth Probing in Heterogeneous IP Home Networks”, Proc. of the 8th IEEE Conference on Consumer Commuication & Networking Conference(CCNC 2011), Las Vegas, NV (USA), January 2011.

[3] www.wireshark.org

[4] http://www.grid.unina.it/software/ITG/

[5] http://xbmc.org/