# A 4G Link Level Emulator for Transport Protocol Evaluation

Stefan Alfredsson, Anna Brunstrom
Department of Computer Science
Karlstad University
SE-651 88 Karlstad, Sweden.
Email: {Stefan.Alfredsson, Anna.Brunstrom}@kau.se

Mikael Sternad
Signals and Systems
Uppsala University
SE-751 20 Uppsala, Sweden.
Email: Mikael.Sternad@signal.uu.se

*Abstract*— **This paper presents a wireless link and network emulator, based upon the "Wireless IP" 4G system proposal from Uppsala University and partners. In wireless fading downlinks (base to terminals) link-level frames are scheduled and the transmission is adapted on a fast time scale. With fast link adaptation and fast link level retransmission, the fading properties of wireless links can to a large extent be counteracted at the physical and link layers. A purpose of the emulator is to investigate the resulting interaction with transport layer protocols. The emulator is built on Internet technologies, and is installed as a gateway between communicating hosts. The paper gives an overview of the emulator design, and presents preliminary experiments with three different TCP variants. The results illustrate the functionality of the emulator by showing the effect of changing link layer parameters on the different TCP variants.**

## I. Introduction

The demand for higher capacity and wider coverage of wireless network access is increasing. As the third generation mobile systems is becoming commercialized, research on the next generation systems, 4G, is becoming more intense.

One view in the industry is that 4G will not be a homogeneous technology as in earlier mobile systems, but rather a mixed composition of different systems and technologies. Perhaps a combination of 3G, WLAN, GPRS, and others, with the common theme that they will provide a seamless service to the user. The user should be connected to the most suitable service, depending of current network coverage.

Even so, existing technologies are not expected to meet the future demands in bandwidth capacity and coverage. New systems need to be developed. As technology progresses, new ways of transmission is being invented, or combined with older technologies. For example, GSM, a second generation system, is using time multiplexing (TDMA) to allow transmission to multiple users. In UMTS, a third generation system, code multiplexing (CDMA) is used, allowing for simultaneous transmission by the users without the need for time multiplexing. Another evolving technology is orthogonal frequency division multiplexing (OFDM), which uses multiple carrier frequencies dedicated to a single data source. OFDM is used for example in the recently released IEEE 802.11a WiFi standard, which delivers up to 54 Mbit/s in the 5 Ghz band.

One 4G system proposal based on OFDM is developed within the "Wireless IP" project [18] at Uppsala University, in cooperation with Chalmers University of Technology and Karlstad University. Our main focus is to cover wide areas to service vehicular users, in excess of speeds of 100 km/h with a 30-fold bandwidth increase compared to UMTS/3G. To realize this goal, adaptive OFDM is used in combination with channel prediction. Transmissions can then be scheduled to maximize the total satisfaction of the users, depending on their current channel quality. This is combined with channel coding, increased cross layer interaction, link level ARQ, and other mechanisms [17], [19].

Since the Wireless IP system is intended to carry Internet traffic, it should be designed to provide a good service to the network and transport layers. An important feature of this system is that it is based on a fast feedback loop for adapting the transmission and scheduling policy. Due to this fast loop, fast link-level retransmissions, on a timescale of a few milliseconds, become feasible. This is in marked contrast to the large latencies of present 2G and 3G networks. It is interesting to investigate how such a radically new design would interact with higher-layer protocols.

There exists a number of network emulator projects. A few examples are NIST Net [7], End-to-end Network Delay Emulator (ENDE) [21], Ohio Network Emulator (ONE) [2], Delayline [8], Dummynet [15], Seawind [9], and various trace-based approaches [14], [13], [12]. Common for most of the emulators is that they model the network with probabilities and distributions for packet loss and delays. For our purposes, this is abstraction is too coarse. For example, we want to see the interaction of fast link layer retransmissions, in combination with adaptive modulation (which gives a varying throughput on a short time scale), and user scheduling in both time and frequency. As we are also investigating issues with transport protocols that do not require full reliability, or protocols that can distinguish between error loss and congestion loss, we want to be able to deliver bit errors on the link to the transport layer.

To this end, a network emulator of the Wireless IP system was constructed from scratch, with the purpose of investigating the impact from different parameter settings on upper layers. For example, how can maximum capacity be attained depending on switching levels for adaptive modulation, what target bit error rate should be used, or how many link level retransmissions can be used before it negatively interacts with the TCP retransmission timer?

The rest of the paper describes the design and construction of the emulator. Furthermore, an overview of an experiment environment is given which displays example experiment output showing the impact of different link layer retransmission limits on different TCP variants.

## II. WIPEMU

Named "WIPEMU", the emulator is intended to be plugged into a real network environment as a gateway. This enables a wide range of operating systems and TCP/IP implementations to be tested, since the common interface is a regular ethernet connection.

WIPEMU is implemented as a software module in FreeBSD, and works by collecting packets on the incoming interface of the gateway. These packets are treated as if they were transmitted over a wireless link, and then the packets are transmitted on the outgoing interface to the destination. Together with the FreeBSD *dummynet* [15] system, the fixed part of the network path can also be emulated. It is abstracted into a packet loss ratio with a possible delay component. With the use of dummynet *pipes*, different loss ratios and delays can be combined to form complex network scenarios.

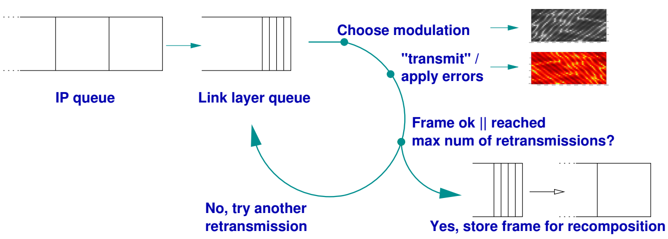


Fig. 1. The WIPEMU core functionality

After packets have been received in the gateway, they are placed in a queue of IP packets, see Figure 1. As packets fill up the queue, WIPEMU dequeues one packet at a time and decomposes it into link layer frames. In the specification of the Wireless IP system, 1500 frames times 25 channels are transmitted every second. These are destined to multiple users, over a downlink with 5 Mhz radio bandwidth partitioned into 200kHz channels. At present the emulator only handles one channel á 1500 frames/s, but multi-channel capability and scheduling between users will be implemented in the future. Each frame consist of 120 symbols (108 symbols useful for data) where each symbol is modulated according to multilevel quadrature amplitude modulation (M-QAM). This modulation can either be fixed (for example, always 4 bits per symbol), or adaptive (transmit more bits per symbol when the channel is good).

For every frame, predicted channel data is consulted to decide the current signal-to-noise ratio (SNR). In the case of adaptive modulation, this ratio controls which modulation level to use. A high SNR enables higher modulation to be used, and the reverse for low values of SNR. When the frame is then "transmitted", the non-predicted (i.e. "real") channel is used when calculating the probability that the frame is received with symbol errors.

If the frame was transmitted without errors, it is stored for later recomposition into its belonging IP packet, and the next frame in the queue is transmitted. In the case of a transmission error, a number of link layer retransmissions are performed. Retransmission takes priority over transmission of new data[1]. To achieve optimal use of the wireless channel, the transmissions are pipelined. This means that frame reordering can occur, which may lead to reordering on the transport layer. If the frame is still in error after the maximum number of allowed retransmissions, the symbol error rate is calculated from the channel data, and bit errors are applied. These errors will then be contained in the re-composed IP packet, and their presence is often detected because the network or transport layer checksum is invalid. This enables experiments with protocols for loss differentiation (for example Checksum-based Loss Differentiation [6] or TCP-HACK [3]), or semi-reliable protocols (for example TCP-L [1] or UDP-Lite [10]), or other protocols that are able to handle packets with bit errors.

The consulted channel data is an array of sample values, at present one per frame, indicating the received channel power. This power experiences variations in strength, or "fading", for vehicular users. This channel can be obtained in a number of ways. One way is to use *channel sounding* to get the measurements from a real environment. Another way is to use ray-tracing models, for example [4]. A third way is to use accepted mathematical models to simulate the channel, such as Rayleigh or Jakes fading models. These models produce the fast (short-term) fading characteristics. There may also be shadow fading (also known as slow fading) involved, which can be modelled by an additive slowly varying contribution to the received power, on the dB-scale. Shadow fading is often modelled as an AR(1) process with prescribed variance. For the experiments in this paper we use a Jakes model with added shadow fading (this is further described in the next section).

## III. Experiment Overview

A scenario was created where a mobile user is downloading content from a server, to illustrate how the emulator is used. The mobile user has a wireless connection to a base station, which in turn is connected to the rest of the Internet, which also connects the server. This scenario is shown in Figure 2. It should be noted that as the emulator currently handles only

[1] In the present implementation, the frame is retransmitted and received separately, without using soft recombining with the previously received incorrect frame.

one channel the scenario is limited to one user in the cell with link adaptation and without scheduling (one channel is allocated all the time to the single user).

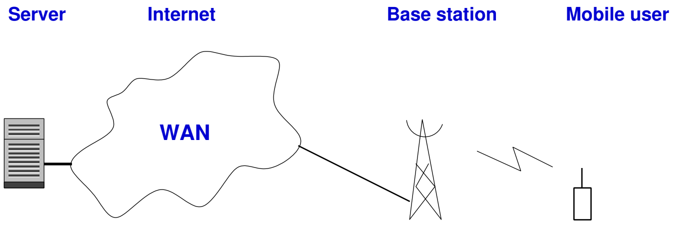


Fig. 2. Logical experiment setup.

The experiment setup to implement this scenario consists of three networked computers [20], as shown in Figure 3. The first is acting as a sender or content provider, the second is running the WIPEMU emulator, and the third is acting as a receiver and consumes data from the sender. All computers are also connected to an administrative network. This is used to control the experiments, so that packet capture in the emulated network is not affected by non experiment related packets.

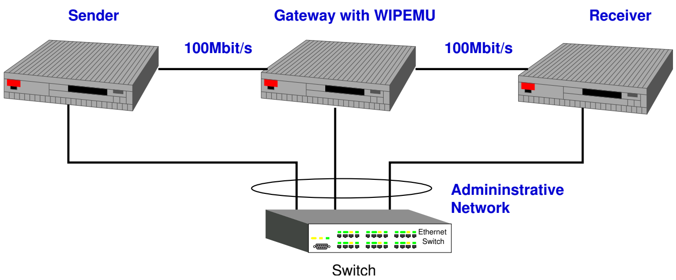


Fig. 3. Physical experiment setup.

As mentioned, the experiment consists of transmitting bulk data from the sender to the receiver (see Table I for a compilation of the relevant parameters). Meanwhile, all transmitted and received packets are collected at both end-points for later analysis. Between each transmission, parameters of the system can be changed. For example, different wireless channel data can be used, different number of link layer retransmissions, different modulation schemes, enabling or disabling certain TCP options, and so on. By choosing an appropriate set of parameters, many aspects of how the Wireless IP system interacts with upper layers can be examined.

For the illustrative experiments presented in this paper we have focused on examining the performance of three different TCP variants, with a varying degree of link reliability. The three variants are regular TCP, TCP-L [1] and TCP Westwood+ [11]. The reasons for these choices are that regular TCP should naturally be studied because it is the dominant transport protocol on the Internet today. TCP-L is an experimental protocol developed at Karlstad university (thus a local interest of including the protocol), and is used to illustrate the effect of allowing bit errors to propagate from the transport layer. TCP Westwood+ is a promising approach to improve upon the bandwidth estimation algorithm in TCP, that should also provide better resiliency to the non-congestion related packet loss in wireless networks.

The end hosts are running the Linux 2.4 kernel in standard configuration, except for the disabling of timestamps to give comparable results with TCP-L (which does not support the timestamp option). This TCP incorporates many of the suggested features in the research community, such as slow start and congestion avoidance, fast retransmit, fast recovery, timestamps, SACK, FACK, D-SACK, fine grained network timers, undoing window adjustments, and rate-halving [16]. As Linux is a widely installed operating system on Internet servers, we believe the standard configuration can be representative for TCPs currently deployed.

TCP-L is a receiver-side modification of TCP that enables the receiving application to make a tradeoff between correct data and performance (mainly higher throughput). Thus, the application must be able to tolerate errors in the data stream to some extent. By accepting and acknowledging packets with bit errors, better performance can be obtained because the TCP congestion control is not invoked as often as if erroneous packets were discarded (as in standard TCP). Retransmissions of erroneous packets is also avoided, presumably leading to less jitter and a more fluent stream of packets, as well as better utilization of radio resources since it is used for retransmissions to a lower extent. As errors may occur both in the header and the payload, TCP-L tries to recover the header. However, when recovery fails, the packet is discarded and treated as a packet loss.

TCP Westwood+ changes the bandwidth estimation algorithm to be more resilient to non-congestion related packet losses. In regular TCP, bandwidth is probed and transmission is increased in response to incoming acknowledgements, and reduced in response to packet loss. In Westwood+, the rate of the incoming acknowledgements is used to calculate a bandwidth estimation. This estimate is then used to provide a more adaptive window reduction in case of packet loss or timeouts.

For the fixed part of the network, there is a set round-trip delay of 20 ms. In addition, delays are introduced by packet queuing in the gateway (limited to 50 packets), and the transmission delay in the wireless link layer. These last two delays will vary, because of the queue building up, and the delay in the link layer depends both on retransmissions and varying modulation levels. Delays due to hand-over have not been introduced. The link level retransmission delay was set to 2 ms, which is due to the tight feedback loop in the system proposal. These delays are interesting to study in relation to the delays introduced by the fixed network and queueing. For example, TCP keeps an estimate of the round trip time to detect packet losses. If the modulation level drops or many retransmissions are needed, this may interfere with the TCP retransmission timer and lead to unnecessary congestion avoidance.

For the wireless channel, the receiver and transmitter are assumed to have single antennas. Data was obtained from a Jakes model with 12 taps set according to typical urban fading. To account for shadow fading, an AR(1) process provided shadow samples at an interval of 2 meter. The standard deviation, $\sigma$, was set to 4 dB and the pole, $a$, at 0.74. A copy of this channel was then processed to include prediction errors, thus resulting in a "real" channel and a predicted channel. The predicted channel is used to choose the modulation scheme for each frame. When the frame is later "transmitted" in WIPEMU, the "real" channel is used when applying errors. The predictor operated with a 100 km/h target velocity, to see the impact of high prediction errors. The normalized power prediction mean square error (NMSE) is 0.1, which is appropriate for predictions of the short term fading 2 ms ahead at 1900 Mhz carriers for terminals moving at 100 km/h [5].

The experiments compare the three different TCP variants against a varying number of link layer retransmission limits. Note that the results may not be directly comparable, as TCP and TCP Westwood+ provide a reliable service, and TCP-L does not provide full reliability as errors are delivered to the application layer. With a low retransmission limit, many packets will be erroneous. For example, if we only allow one link layer frame retransmission, and the channel quality is predicted badly, there is a high probability that the link layer frame will be damaged, and thus the TCP packet will be damaged. If more retransmissions are allowed, the probability increases that the frame will be transmitted correctly. This however adds more delay to the transmission of the TCP packet, which may interfere with the TCP retransmission timer. If TCP experiences a timeout, it assumes that the network is congested, and tries to probe for the available bandwidth from scratch. Presumably, as TCP-L will allow erroneous packets to be delivered, and as TCP Westwood+ keeps a separate estimate of the available bandwidth, these are expected to perform better than regular TCP when packet loss is high.

## IV. Results

Two main performance metrics were extracted from the experiments, the throughput and the number of TCP packet retransmissions. The experiments were repeated 30 times with different fading environments and the figures show the mean values of the repetitions and the 95% confidence intervals. Figure 4 shows the obtained throughput versus the maximum number of allowed link layer retransmissions. The different curves in the figure correspond to the different TCP variants tested, as explained by the legends in the figure. Starting from the right, where many link layer retransmissions are allowed, all three transport protocols perform well. As the retransmissions become more limited, more and more TCP packets will become erroneous. These packets will be discarded by TCP and TCP Westwood+, and treated as packet loss events. This

| | |
|---|---|
| **Fixed network** | |
| Fixed network delay (RTT) | 20 ms |
| Network queue size | 50 packets |
| **Wireless Downlink** | |
| Frame transmission delay | 0.667 ms |
| Channel model | 12-tap Jakes typical urban fading model @ 100 km/h, 16dB SNR + AR(1) shadow fading with variance 4 dB and pole at 0.74 |
| Modulation | Adaptive 4-256 QAM with switching adjusted to a prediction error of NMSE 0.1 |
| Frame size | 108 symbols |
| Coding | Uncoded M-QAM used |
| Scheduling | None, one-user scenario |
| Link ARQ | 10, 9, ..., 3 retransmissions |
| **Wireless uplink** | |
| Channel model | imposed bandwidth limit and delay |
| Packet loss | 0% (lowest modulation level assumed) |
| Capacity | 20 kbit/s |
| Delay | 2 ms |
| **Transport protocols** | |
| Variants | TCP, TCP-L, TCP Westwood+ |
| Transferred data | 4 Mb bulk data |
| TCP settings | Standard, except for disabled timestamp |
| MTU | 576 bytes |
| Retentive TCP caching | Cleared before new connections |

TABLE I
EXPERIMENT PARAMETERS

leads to invocation of congestion control, and therefore the transmission rate is reduced. TCP-L, which tries to deliver the erroneous packets anyway, does not experience loss to the same degree as TCP and TCP Westwood+. However, a slight reduction in throughput is seen anyway, since not all damaged packets can be recovered. Comparing TCP and TCP Westwood+, the latter shows a small improvement over TCP when there is a maximum of three link layer retransmissions.

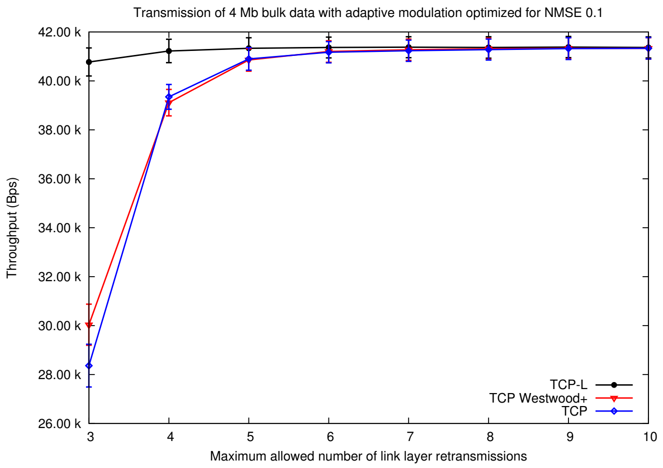


Fig. 4. Throughput as a function of the maximum number of link layer retransmissions

The other metric, the number of TCP retransmissions, is shown in Figure 5. As expected, when many link layer retransmissions are allowed, not many TCP retransmissions are needed. As more errors are introduced at the link layer, the number of transport layer retransmissions increases. TCP and TCP Westwood+ display the same amount of retransmissions, while TCP-L shows a lower number of retransmissions. As explained earlier, damaged packets are accepted instead of being treated as packet loss, leading to fewer transport layer retransmissions.

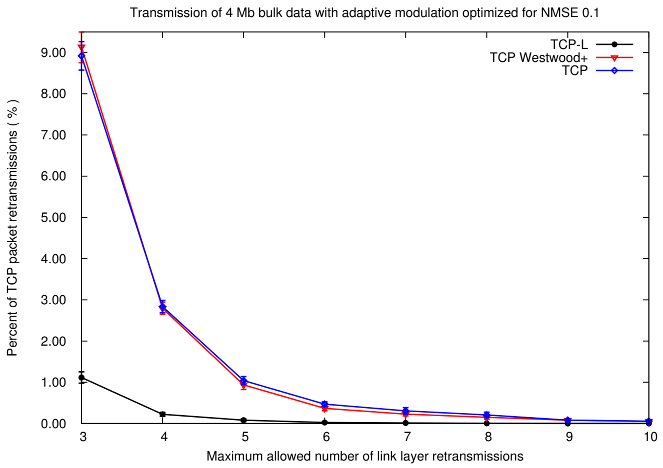


Fig. 5. TCP retransmissions as a function of the maximum number of link layer retransmissions

## V. Conclusions

This paper presented the WIPEMU network emulator. The emulator is intended to be used to evaluate the impact of design decisions of the 4G system proposal from the Wireless IP project at Uppsala University. WIPEMU handles the wireless parts of the network path, while the fixed parts can be emulated with the dummynet functionality in FreeBSD. The emulator is constructed based upon the downlink proposal in [17]. The key components are link layer framing, adaptive modulation, fast link layer retransmissions, and delivery of damaged frames to the network layer. The usability of the emulator has been shown with an illustrative experiment, showing the throughput performance of three different TCP variants over a link layer with variable reliability.

## Acknowledgements

The Wireless IP project is supported by SSF, the Swedish Foundation for Strategic Research. The first author was partly sponsored by the Swedish graduate school PCC++, funded by SSF. Anna Widenius has contributed to the design and implementation of the WIPEMU system.

## References

[1] S. Alfredsson and A. Brunstrom. TCP-L: Allowing bit errors in wireless TCP. *Proceedings of IST Mobile and Wireless Communications Summit 2003*, June 2003.

[2] M. Allman and S. Ostermann. ONE: The ohio network emulator. Technical Report TR-19972, Computer Science, 1997.

[3] R. K. Balan, B. P. Lee, K. R. R. Kumar, L. Jacob, W. K. G. Seah, and A. L. Ananda. TCP HACK: A mechanism to improve performance over lossy links. *Computer Networks*, 39(4):347–361, July 2002.

[4] M. Berlin. Channel modeling and simulation for the mobile internet. Master's thesis, Uppsala University, 2002.

[5] T. Ekman, M. Sternad, and A. Ahlén. Unbiased power prediction on broadband channels. In *IEEE Vehicular Technology Conference VTC2002-Fall*, Vancouver, Canada, Sep 2001.

[6] J. Garcia and A. Brunstrom. Checksum-based loss differentiation. *Proceedings 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002), Stockholm, Sweden*, September 2002.

[7] NIST Internetworking Technology Group. NIST Net network emulation package. *http://snad.ncsl.nist.gov/itg/nistnet/*, November 2004.

[8] D. B. Ingham and G. D. Parrington. Delayline: A wide-area network emulation tool. *Computing Systems*, 7(3):313–332, 1994.

[9] M. Kojo, A. Gurtov, J. Manner, P. Sarolahti, T. Alanko, and K. Raatikainen. Seawind: A wireless network emulator. In *Proceedings of 11th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems*, September 2001.

[10] L-A. Larzon, M. Degermark, S. Pink, L-E. Jonsson, and G. Fairhurst. RFC 3828: The lightweight user datagram protocol (udp-lite), July 2004.

[11] S. Mascolo, L. A. Grieco, R. Ferorelli, P. Camarda, and G. Piscitelli. Performance evaluation of Westwood+ TCP congestion control. *Performance Evaluation*, 55(1-2):93–111, January 2004.

[12] B. Melander and M. Björkman. Trace-driven network path emulation. Technical report 2002-037, Department of Information Technology, Uppsala University, Sweden, 2002.

[13] B. Noble, G. Nguyen, M. Satyanarayanan, and R. Katz. RFC 2041: Mobile network tracing, October 1996.

[14] B. Noble, M. Satyanarayanan, G. Nguyen, and R. Katz. Trace-based mobile network emulation. In *Proceedings of ACM SIGCOMM '97*, September 1997.

[15] L. Rizzo. Dummynet: A simple approach to the evaluation of network protocols. *ACM Computer Communication Review*, 27(1):31–41, January 1997.

[16] Pasi Sarolahti and Alexey Kuznetsov. Congestion control in Linux TCP. In *Proceedings of USENIX 2002*, pages 49–62, June 2002.

[17] M. Sternad, T. Ottoson, A. Ahlén, and A. Svensson. Attaining both coverage and high spectral efficiency with adaptive OFDM downlinks. In *Proceedings of IEEE Vehicular Technology Conference VTC2003-Fall*, Orlando, FLA, Oct 2003.

[18] A. Svensson, A. Ahlén, A. Brunstrom, T. Ottosson, and M. Sternad. An OFDM based system proposal for 4G downlinks. In *Proceedings of Multi-Carrier Spread Spectrum Workshop*, Oberpfaffenhofen, Germany, 2003.

[19] W. Wang, T. Ottosson, M. Sternad, A. Ahlén, and A. Svensson. Impact of multiuser diversity and channel variability on adaptive OFDM. In *Proceedings of IEEE Vehicular Technology Conference VTC2003-Fall, Orlando, FLA*, Orlando, FLA, Oct 2003.

[20] A. Widenius. Simulation of the TCP/IP over the wireless link. In *PCC/Ericsson/TeliaSonera Research Days*, Stockholm, Nov. 18-19 2003.

[21] I. Yeom. Ende: An end-to-end network delay emulator. Master's thesis, Texas A&M University, College Station, TX, 1998.