EMULATION AND VALIDATION OF A 4G SYSTEM PROPOSAL

Stefan Alfredsson[1], *Anna Brunstrom*[1] *and Mikael Sternad*[2]

[1] Department of Computer Science, Karlstad University, SE-651 88 Karlstad, Sweden
[2] Signals and Systems, Uppsala University, SE-751 20 Uppsala, Sweden

ABSTRACT

This paper presents a wireless link and network emulator, along with experiments and validation against 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. The emulator has been used to experimentally investigate the resulting interaction between the transport layer and the link layer. The paper gives an overview of the emulator design, and presents experimental results with three different TCP variants in combination with various link layer characteristics.

1. 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 promising technology for 4G 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 [19] 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 [18], [20].

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 important to investigate how such a radically new design would interact with higher-layer protocols.

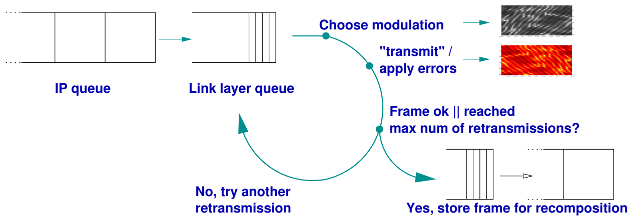

Figure 1. The WIPEMU core functionality

To this end, a network emulator of the Wireless IP system was constructed, with the purpose of investigating the impact from different parameter settings on upper layers. The remainder of the paper describes the design and construction of the emulator, followed by an overview of investigated transport protocols and the experiment environment. The paper continues with a presentation and discussion of the obtained results from experiments with three TCP variants, with different modulation schemes. The paper then ends with the conclusions.

2. WIPEMU

There exists a number of network emulator projects. A few examples are NIST Net [7], End-to-end Network Delay Emulator (ENDE) [22], 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 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 enable the investigation of the issues mentioned above, the "WIPEMU" network emulator was constructed. It 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.

After packets have been collected 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 packet 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. The recomposed IP packet is then forwarded to its destination. Allowing the release of erroneous IP packets (as opposed to discarding packets with erroneous frames) 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.

Regarding the consulted channel data, it is an array of sample values, 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.

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.

3. TRANSPORT PROTOCOLS

For the experiments presented in this paper we have focused on examining the performance of three different TCP variants. These 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, and is used as the baseline protocol. 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 provide better resiliency to non-congestion related packet loss.

The TCP version used is 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 that 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. The utilization of radio resources will be better since it is used for retransmissions to a lesser 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 the transmission rate is increased in response to incoming acknowledgements, and reduced in response to packet loss. In TCP Westwood+, the rate of the incoming acknowledgements is used to calculate a bandwidth estimation. This estimate is then used to provide a more adaptive

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

window reduction in case of packet loss or timeouts.

Relating the protocols to the number of link layer retransmissions, a low retransmission limit will cause many packets to 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 retransmission timer. If TCP experiences a timeout, it assumes that the network is congested, and tries to probe for the available bandwidth from zero. 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 error rate is high.

4. EXPERIMENT OVERVIEW

The emulated scenario considered in the experiments consists of a mobile user downloading content from a server. The 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 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). The experiments compare the three different TCP variants (as described in the previous section) against a varying number of link layer retransmission limits and modulation schemes, in presence and absence of channel prediction errors.

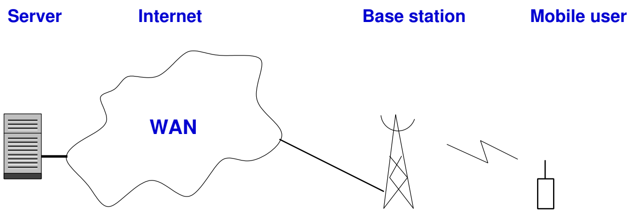

Figure 2. Logical experiment setup.

The experiment setup to implement this scenario consists of three networked computers, 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.

As mentioned, the experiment consists of transmitting bulk data from the sender to the receiver. Meanwhile, all transmitted and received packets are collected at both endpoints for later analysis. Between each transmission, parameters of the system can be changed.

For the wireless channel, the receiver and transmitter are assumed to have single antennas. Data was obtained from

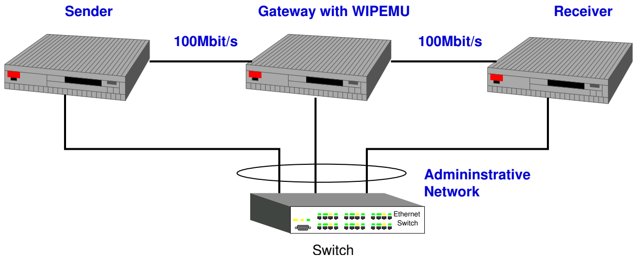

Figure 3. Physical experiment setup.

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, σ, 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[2] for each frame (in the case of adaptive modulation; for fixed modulation the same modulation scheme is always used). When the frame is later “transmitted” in WIPEMU, the “real” channel is used when determining if a symbol is received in error or not. The predictor operates at a 100 km/h target velocity, to see the impact of high prediction errors.

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 because the delay in the link layer depends on both 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 Wireless IP 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.

Table 1 contains a compilation of the relevant parameters for the wireless network, the fixed network and the protocol settings.

5. RESULTS

The main performance metric extracted from the experiments is the throughput, however the number of transport layer packet retransmissions are also shown to explain the obtained results. 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 4-7 show the obtained TCP[3] throughput/retransmissions versus the maximum number of allowed link layer retransmissions, for the cases of fixed and

[2] The switching levels for the adaptive modulation are optimized to provide maximum throughput in each link layer frame [17].

[3] These figures show only the TCP performance, as showing all protocols would make them much harder to read. Figure 8 contains a comparison of all protocols for the best modulation scheme.

Fixed network	
Fixed network delay	20 ms (RTT)
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, 16 dB 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 0.0, 0.1 [17], adjusted to perfect prediction, and fixed to BPSK, 4-,8-, and 16-QAM.
Frame size	108 symbols
Coding	Uncoded M-QAM used
Scheduling	None, one-user scenario
Link ARQ	30, 20, 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	3 Mb bulk data
TCP settings	Standard, except for disabled timestamp
MTU	Standard, 1500 bytes
Retentive TCP caching	Cleared before new connections

Table 1. Experiment parameters

adaptive modulation. The different curves in the figures correspond to the different modulation schemes used, as explained by the legends in the figures. Starting from the right, where many link layer retransmissions are allowed, the transport protocols perform well. As the link layer retransmissions become more limited, more and more packets will become erroneous. These packets will be discarded by TCP and treated as packet loss events. This leads to invocation of congestion control, and therefore the transmission rate is reduced.

The goal for most communications systems is to achieve the highest possible throughput. For the case of fixed modulation in this scenario, this seems to be obtained by using 4-QAM. If a lower modulation is used, it becomes more robust to the channel noise, but does not utilize the available capacity in full, while a higher modulation is more sensitive to noise, leading to an increased error rate. This is however heavily dependent upon the average SNR ratio. In this case, it is 16 dB. If it were to be e.g. 20-25 dB, a higher fixed modulation would probably have provided a higher throughput.

The reason for the drop in throughput in the left part of Figure 4 can be seen in Figure 5. Retransmissions above a few percent is generally considered a bad environment for the current TCP congestion control scheme, which can easily be seen by correlating between the two figures.

An improvement of a fixed modulation scheme is to use adaptive modulation that adapts to the current channel conditions. This can be done on different time scales, depending on the fading characteristics of the channel. For this scenario the adaptation is done per link layer frame. It therefore adapts both to the experienced fast fading, as well as the experienced slow or shadow fading. Figure 6 shows the TCP throughput for three adaptive modulation schemes. The top curve assumes that the channel is perfectly predicted. In the two curves below, the channel prediction is uncertain. Fixed 4-QAM is also shown as reference as the best fixed modulation. To get optimal performance, the adaptive modulation switching levels need to take this uncertainty into account. In the figure, the lowest AM[4] curve uses the same switching levels as with the perfectly predicted channel (the top curve). For the middle AM curve the switching levels are optimized for a normalized mean square error in the prediction of 0.1. This leads to more conservative switching levels, because we know there is uncertainty in the prediction. As seen in the figure, this leads to a performance increase, compared to using switching levels optimized for no prediction error. It is also observed that the adaptive modulation achieves better throughput than any of the fixed modulation schemes. We also note that the amount of transport layer retransmissions are reduced more sharply when adaptive modulation is used compared to fixed modulation, as seen in Figure 7.

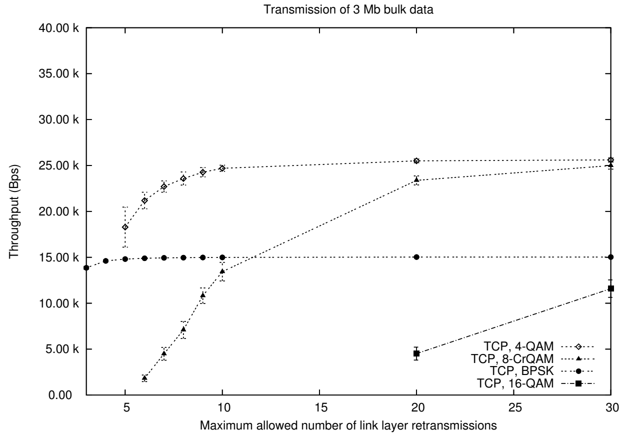

Figure 4. Fixed BPSK, 4-QAM, 8-CrQAM, 16-QAM uncoded modulation: TCP throughput

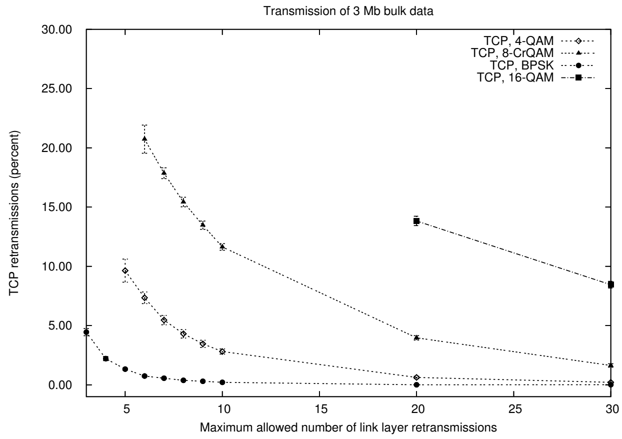

Figure 5. Fixed BPSK, 4-QAM, 8-CrQAM, 16-QAM uncoded modulation: TCP retransmissions

We now correlate the obtained results to the theoretical limits. The observed asymptotic throughput for the fixed BPSK modulation is about 15000 byte/s = 120000 bit/s. Each frame is 0.667 ms long, leading to $120000 * 0.667 * 10^{-3} \approx 80$ bits/frame = $80/108 \approx 0.74$ bits/symbol. According to [17], the frame ("bin") capacity for BPSK is 1 bit/symbol, given an SNR above about 8 dB. For the

[4] Adaptive modulation.

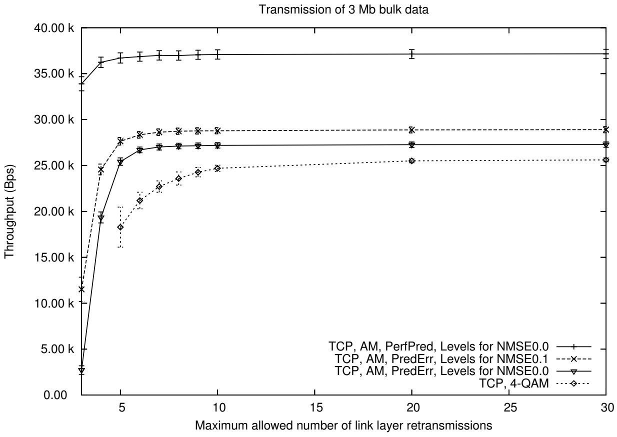

Figure 6. Adaptive modulation: TCP throughput

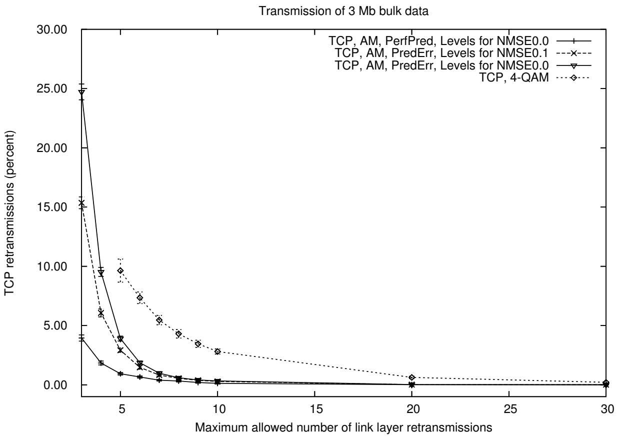

Figure 7. Adaptive modulation: TCP retransmissions

higher modulations the difference is even larger, but they also have a higher SNR requirement.

The observed asymptotic throughput for the adaptive modulation optimized for power prediction NMSE 0.1 (Figure 8) is about 29000 byte/s = 232000 bits/s. This leads to $232000 * 0.667 * 10^{-3} \approx 154$ bits/frame = $154/108 \approx 1.43$ bits/symbol. In [17], an average SNR of 16 dB corresponds to about 2.2 bits/symbol, for the one user scenario we are considering.

The differences between the measured and theoretical values can be attributed to a few causes. First, the theoretical limits are on the physical link. The measured values are on the application layer, i.e. after any transport and link layer retransmissions and overhead. The *average* SNR of the channel is 16 dB, but because of the shadow fading there are moments when the instantaneous SNR drops below what is acceptable for the given modulation. This causes link layer retransmissions even for BPSK which should operate optimally above 8 dB as mentioned above. Also, the throughput is measured as data divided by transmission time. There may be timeout events on the transport layer when no data is transmitted, which then contribute to a lower observed frame capacity compared to the theoretical limit.

The experiments with different transport protocols is shown in Figure 8, for the best modulation case (adaptive NMSE 0.1). The throughput of TCP, TCP Westwood+ and

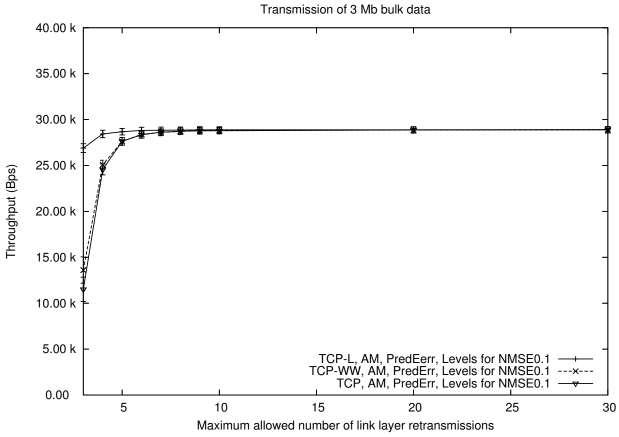

Figure 8. Adaptive modulation: TCP, TCP-Westwood+ and TCP-L

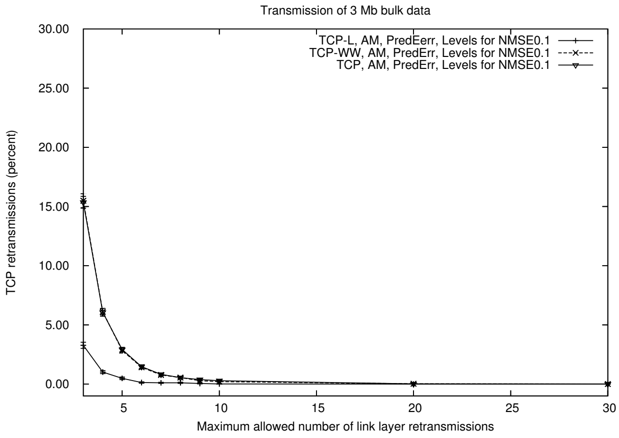

Figure 9. Adaptive modulation: TCP, TCP-Westwood+ and TCP-L

TCP-L reaches the same asymptotic limit when enough link layer retransmissions are allowed. As retransmissions are limited, TCP and TCP Westwood+ show a similar behavior in response to the increased error level. TCP-L achieves better throughput in presence of these errors, assuming that the application can handle a residual BER of approximately $2 * 10^{-4}$ in the application data. The reason for these differences in throughput can again be found in the retransmission graph, shown in Figure 9. TCP-L performs fewer retransmissions, thereby achieving a higher throughput. As mentioned in section 3, TCP Westwood+ was expected to perform better than TCP in presence of non-congestion related packet loss. As seen in the figures, the noticeable difference between the two is very small. The cause for this is subject to future research, but may depend on the particular setup, buffer sizes and loss patterns.

6. CONCLUSIONS

This paper presented the WIPEMU network emulator, the experiment environment and performed experiments. The emulator was developed to evaluate the impact of design decisions of the 4G system proposal from the Wireless IP project at Uppsala University on current Internet protocols. The experiments indicate that adaptive modulation utilizes the available channel capacity better than fixed modulation. Fixed modulation either under-utilize the capacity, or experience too many errors. Further, in the case of adaptive modulation, the results indicate that highly persistent ARQ should be allowed, considering best-effort traffic. Experiments with three different TCP variants showed little difference between TCP and TCP Westwood+ in presence of residual bit errors. TCP-L was able to recover many packets and deliver the possibly corrupted payload to the application, allowing the application to trade reliability for performance.

7. 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] Luigi 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 and S. Falahati. Maximizing throughput with adaptive M-QAM based on imperfect channel predictions. In *Proceedings of IEEE PIMRC*, Barcelona, Sep 2004.

[18] 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.

[19] 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.

[20] 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.

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

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