

Supporting end-to-end applications over HSDPA by cross-layer signalling

Niels Möller*, Åke Arvidsson†, Justus Petersson†, Carlo Fischione*,
Robert Skog†, Patrik Karlsson†, and Karl H. Johansson*

*KTH, School of Electrical Engineering, SE-100 44, Sweden
Email: {niels|carlofi|kallej}@ee.kth.se

†Ericsson AB, Stockholm, Sweden

Email: {Ake.Arvidsson|Justus.Petersson|Robert.Skog|patrik.j.karlsson}@ericsson.com

Abstract—A new control structure is proposed to improve user experience of wireless Internet. Information on radio bandwidth and queue length available in the radio network, close to the base station, is used in a proxy that resides between the Internet and the cellular system. The control algorithm in the proxy sets the window size according to event-triggered information on radio bandwidth changes and time-triggered information on the queue length at the wireless link. The mechanism is compared to TCP Reno in two simulation scenarios. The first scenario models a dedicated channel with stepwise changes in the bandwidth, while the second scenario models the High-speed Downlink Shared Channel recently introduced by 3GPP. The proposed mechanism significantly reduces the amount of buffer space needed in the radio network, and it also gives modest improvements to user response time and link utilization. Reduced buffering is particularly beneficial for third-party end-to-end real-time services such as voice, video, and online gaming.

I. INTRODUCTION

Two major phenomena have dominated the telecommunications and networking industries during more than a decade: The growth of packet based services over the Internet and the growth of voice services in wireless cellular systems. Today we witness an ongoing amalgamation of these technologies that leads to packet based services in wireless cellular systems. The development is driven by users demanding ubiquitous access to Internet services and operators wanting to broaden and integrate their businesses.

The marriage between Internet traffic and wireless access is, however, not an untroubled one. The Internet protocols were developed for wireline services with links of fixed bandwidth, whereas wireless systems typically provide where both bandwidth and other properties are varying. Similarly, cellular wireless systems were developed for services like voice which are robust to losses, whereas Internet services typically are sensitive to losses.

Proposed improvements for TCP-over-wireless can be classified in three major groups: The end-to-end approach tries to change the TCP protocol to behave better over wireless

links. The link-layer approach tries to tweak the link-layer processes, such as forward error correction coding, to make the link properties more friendly to the current TCP protocol. Finally, the cross-layer uses direct signalling between the link-layer in one node and the transport layer in a distant node.

A. Main contribution

We propose a cross-layer control structure for improving the user experience of the wireless Internet. The radio network sends information about the available bandwidth and the current queue length to a web-proxy. The proxy uses this information to control its sending window, replacing the congestion control of standard TCP. Evaluation of this scheme over a dedicated channel, with stepwise changes of the bandwidth, has shown promising results [1], [2]. This paper examines the performance over the shared downlink channel in HSDPA, based on realistic simulations of the HSDPA radio layer. Compared to TCP Reno, the proposed mechanism significantly reduces the amount of buffer space needed in the radio network, and it also improves user response time and link utilization.

B. Related work

Two general overviews of the end-to-end approach are given in [3] and [4]. Among the specific protocols we mention TCP Eifel [5], and TCP Westwood+ [6]. Another line of research considers the problem of distinguishing between wireless losses and congestion losses [7].

The link-layer approach is focused on improving the wireless interface with respect to TCP performance. There are different parameters that can be tuned, namely, e.g., modulation, error correction coding, power control, and Automatic Repeat reQuest (ARQ). Because of the complex interdependence of these parameters, many contributions in literature are focused on one or a few of them. Specifically, [8] considers forward error correction coding to add redundancy to the packets. The work in [9] concerns the forward link power allocation and rate adaptation for TCP throughput maximization in a WCDMA wireless system. An optimization of the ARQ parameters for a TCP connection is proposed in [10]. The interaction

This work was partially supported by European Commission through the Network of Excellence HYCON, by the Swedish Foundation for Strategic Research through an Individual Grant for the Advancement of Research Leaders, and by the Swedish Research Council.

between power control, ARQ and TCP is studied in [11], [12]. The trade-off between end-to-end mechanisms and link-layer mechanisms are discussed in [9], [13]–[15].

The end-to-end approach and the link-layer approach respects the layered architecture of the networking stack. The cross-layer explores the additional optimization possibilities offered when the constraints of the protocol stack are relaxed. A drawback is that the additional complexity of a system with cross-layer interconnections can make the system harder to understand, with complex and sometimes unexpected interactions between different layers. The pros and cons of cross-layer design are discussed in [16] and [17].

There are several papers on cross-layer approaches in wireless systems such as IEEE 802.11 (Wi-Fi) and IEEE 802.16 (WiMax). In the present context, however, we restrict ourselves to cross-layer approaches for cellular systems such as the 3GPP standard for WCDMA access. The throughput of TCP Reno over HSDPA is analyzed in [18]. A relatively early example of cross-layer approaches in such systems is found in [19] where cross-layer mechanisms are used to optimize radio resource management between base stations and to provide assured service delivery. The work in [20], [21] suggests that application layers on servers submit layer independent descriptors to link layers on radio elements.

The rest of this paper is organized as follows. Section II summarizes the problems when using TCP over a wireless network. Section III introduces the HSDPA channel, and Sec. IV proposes a cross-layer mechanism for adaptation to the bandwidth variations of the channel. In Sec. V we present the evaluation of the proposed mechanism in two simulation scenarios, and discuss the implications on buffer sizing and queueing delays, before concluding in Sec. VI.

II. THE TCP-OVER-WIRELESS PROBLEM

There are two main drivers behind the ongoing merge of the mostly-wired Internet and the wireless cellular systems. On one hand, users, together with third-party service providers, demand ubiquitous access to Internet services. And on the other hand, operators want to get the maximal value out of their infrastructure investments.

For the users, the primary quality objectives are to maximize the throughput and minimize the response time of Internet services. Secondary objectives are to minimize packet delay and jitter. From an operator perspective, there is one additional important objective: Efficient use of available resources, which can be measured in terms of radio link utilization, aggregated throughput, or the maximum number users that can be served at the same time. The attractiveness of wireless Internet access can be improved by low-cost solutions for obtaining high throughput.

From a networking perspective, a wireless link is just one of many links on the path between the communication end points. The most important properties of a link are the delay, the bandwidth and the packet loss probability. For wired links, these properties are static, and the loss probability is usually very small. Most Internet protocols, in particular TCP, were

designed with wired links in mind. The congestion control mechanisms of TCP work fairly well over wired links, but there are several problems with TCP over wireless:

- Wireless losses are misinterpreted as network congestion.
- Large delay variations triggers TCP's timeout mechanism.
- TCP adaptation to bandwidth changes is quite slow, in particular when the bandwidth increases.
- Cellular systems often exhibit fairly large delays, which slows down TCP window updates even further. The delays are reduced with recent systems such as HSDPA.

All these problems result in under-utilization of the radio resources, and decreased end-to-end throughput.

III. HSDPA TRANSMISSION

The High Speed Packet Access network (HSPA) is an evolution of the third generation cellular network specified by the 3GPP [22]. It consists of a High Speed Downlink Packet Access (HSDPA), as specified in 3GPP release 5, and an Enhanced Uplink, as specified in 3GPP release 6. It has been shown that HSPA can offer bandwidths in excess of 10 Mbit/s during favorable conditions. The peak bandwidth of HSPA is therefore on par with xDSL. The end user performance is, however, determined not only by peak bandwidth.

The HSDPA introduces three channels in order to support the high performing transmission: the transport channel High-Speed Downlink Shared Channel (HS-DSCH), which carries data of the users, and two control channels carrying signalling information: High-Speed Shared Control Channel HS-SCCH (for the downlink) and High-Speed Dedicated Physical Control Channel HS-DPCCH (for the uplink). The transport blocks are the basic units of data transmitted over HSDPA channels. A transport block is transmitted over a Transmission Time Interval (TTI) of 2 ms. Therefore, TCP segments are split into transport blocks before being transmitted over HSDPA.

The HS-DSCH is a time shared channel among users. It is designed to handle fast link variations at the base station. The HS-DSCH channel is terminated at the base station, and it is mapped onto a number of physical channels, called High-Speed Physical Downlink Shared Channels (HS-PDSCHs). Each HS-PDSCH channel is characterized by a channelization code having spreading factor 16. Since it is possible to select among 15 codes, there are 15 HS-PDSCH channels available. Data of users are then actually transmitted over the HS-PDSCH channels by associating a channelization code to a user. This means that more than one user may receive data during a TTI. Furthermore, HSDPA scheduler can assign more than one HS-PDSCH channel to the same user. For each TTI, the scheduler determines which user should use the channel and at which rates, by assigning the HS-PDSCH channelization codes. If not all power or all multiplexing codes are consumed by a single user, the remaining resources can be allocated to other users in the cell. When user data are transmitted over the HS-DSCH, the data rate during each TTI depends on a combination of factors:

- Cell load. Higher load results in lower rate per user.

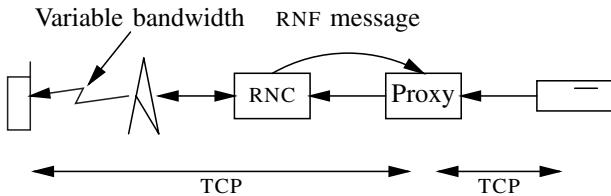


Fig. 1. Radio network feedback architecture. The mobile terminal on the left downloads a file from the server on the right, via the proxy. During the transfer, the RNC generates cross-layer RNF messages including information about the current bandwidth over the radio link, and the current RNC queue length. The proxy uses this information to adjust its window size.

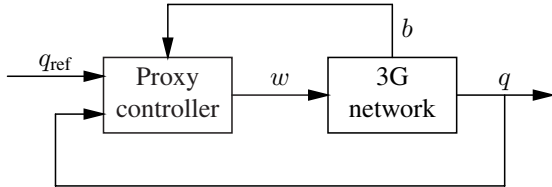


Fig. 2. Control structure. The controller uses cross-layer signalling: Event-triggered feedforward of the available radio link bandwidth b , and time-triggered feedback of the RNC queue length q . Note that both the RNC and the proxy belong to the operator's administrative domain.

- Downlink power. Higher power increases the rate.
- Downlink radio quality.

Users connected to HSDPA are constantly measuring the downlink radio quality for HSDPA transmission. The quality varies depending on location (which determines the fading conditions) and surrounding interference. The results of the measurements are fed back over the uplink control channel, the HS-DPCCH, to the system on a regular basis ranging from every TTI to every 500 TTI. This information are used for power control for selection of modulation and channel coding scheme. Users with poor radio conditions will receive fewer data bits during a TTI than users with good radio conditions.

IV. PROPOSED RNF MECHANISM

This section, together with the simulation results in Sec. V, give the main contribution of this paper. We consider web-browsing, or the download of files from a web-server on the Internet, to a mobile terminal. This system is illustrated in Fig. 1. We split the connection into two TCP connections, one from the web server to a web-proxy, and another from the proxy to the terminal. On the path from the proxy to the terminal, the most important node is the Radio Network Controller (RNC), which controls the link layer for the radio channel from the base station to the terminal.

To improve the TCP sender's adaptation to varying radio conditions, we propose explicit cross-layer signalling (Radio Network Feedback, RNF) from the RNC, which knows the radio channel and link state, to the TCP sender. A feedback control system view is given in Fig. 2. The control signal is the window size w , and the output is the RNC queue size q , which we want to keep close to the reference value q_{ref} .

The control is based on the information in the RNF messages from the RNC, and divided in two parts. We use feedforward from the link bandwidth b , and feedback from the queue size q .

A. Feedforward of bandwidth changes

When the bandwidth of the radio link changes, the RNC generates an RNF message with the new bandwidth to the proxy. When the message is received, the proxy updates its window size according to the control law

$$w = b\hat{\tau} + q_{\text{ref}}$$

where $\hat{\tau}$ is the proxy's estimate of the roundtrip time, τ . With this window size, the queue size will converge to

$$q^* = q_{\text{ref}} + b(\hat{\tau} - \tau)$$

Clearly, any estimation error in $\hat{\tau}$ implies a bias in the equilibrium queue size. Any measurement errors in the RNCs bandwidth also results in a bias in the queue size.

B. Feedback of the RNC queue size

To reduce the the bias in the queue size, it is natural to introduce feedback of the actual queue size. Unlike the feedforward, which is event-triggered, we use a queue feedback with a fixed fairly large sampling time, around 1 s. We use the simple feedback law

$$w_{k+1} = w_k + q_{\text{ref}} - q_{k+1}$$

where q_k are the queue size measurements, provided by the RNF messages. The stability and convergence speed of these control laws was established in [2].

C. Generation of RNF messages.

The RNC node will send RNF messages when it detects a change on the link layer. Example of such a change is the change of the available bit-rate over the air. The question is when to send the message.

For a dedicated channel, switching between a small number of states corresponding to different bandwidths, it is natural to send an RNF messages whenever the channel state changes.

For a shared channel such as the downlink in HSDPA, where are no distinct states, instead, the bandwidth available to a user varies continuously depending on radio conditions and cell load. One simple approach is to select a sampling time, e.g., 0.5 s. For each sample interval, the RNC computes the average bandwidth for the user. At the end of the interval, it sends an RNF message with this average, where it is used by the proxy to predict the bandwidth during the next interval.

The RNC sends the RNF messages as ordinary UDP packets to a specific IP address and UDP port. The RNF message typically contains the following information: Time stamp, current bit rate on radio channel, and the current RNC queue size. In order for the proxy to distinguish between several data flows to several terminals, the RNF message must also identify the terminal, e.g., by its IP address. The RNC also needs to find out where to send the RNF-message. This can be done in several ways. The terminal can tell the RNC which IP address and port should be used, or we can let the proxy insert this information in IP header options. It is also possible to use explicit signalling, where the RNC extracts the needed information from packet inspection.

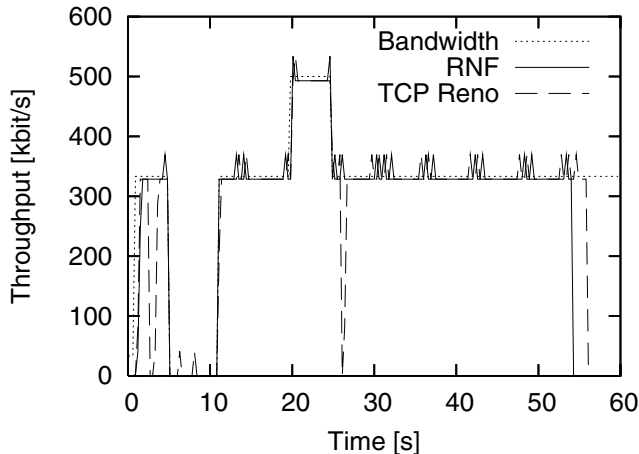


Fig. 3. Resulting throughput for the dedicated channel scenario. The dotted line is the time-varying bandwidth to the terminal, the dashed line is the throughput for TCP Reno, and the solid line is the throughput using the proposed RNF controller.

D. New window update mechanism

Some buffering before the radio link is essential for high link utilization, to even out short-term variations in the available bandwidth and propagation delay. When the proxy uses the RNF information to adjust its sending window, the need for buffering at the RNC is reduced.

Adding the proxy also introduces a new buffer into the system. Data that are received from the web-server adds to buffer, and data in the buffer are transmitted to the terminal under the control of received ACKs from the terminal and the RNF messages received from the RNC. The proxy allocates a fixed size buffer for each flow. It uses the standard TCP flow control, the advertised window, to control the data flow from the web-server and prevent buffer overflow.

V. SIMULATION STUDY

We evaluate the proposed model in two scenarios, where the first scenario models a dedicated channel, with stepwise changes of the bandwidth, while the second scenario models the shared downlink channel of HSDPA. In each scenario, we evaluate the performance for download via a web-proxy, where the proxy uses either the proposed RNF controller or standard TCP Reno.

A. Dedicated channel scenario

The dedicated channel is characterized by a bandwidth that varies between discrete values, depending on the current radio conditions. In this simulation, we use a link with a nominal bandwidth of 1 Mbit/s, which is subject to an outage of 6 s, starting at time 5 s, and an excellent radio environment where the bandwidth is increased to 1.5 Mbit/s, for 5 s starting at time 20 s. The roundtrip delay between proxy and terminal is 223 ms. The packet size is 1500 bytes, and the RNC buffer size is 30 packets, corresponding to the bandwidth-delay product during the time interval with the largest bandwidth.

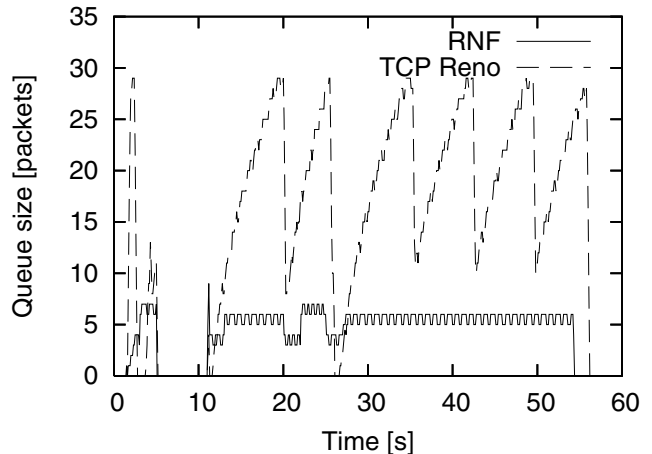


Fig. 4. Resulting RNC queue size for the dedicated channel scenario. The dashed line is the queue size with TCP Reno, and the solid line is the queue size with the RNF controller. With RNF, the queue stays small, with small fluctuations.

Figure 3 shows the resulting throughput as a function of time, and Fig. 4 shows the evolution of the RNC queue. In Fig. 3, note that the download finishes 2 s faster, which is a modest improvement of user response time. In Fig. 4, we see that the queue stays small, with small fluctuations. Due to errors in the roundtrip time estimate, the feedforward from the bandwidth changes results in a queue size slightly different from the target value $q_{ref} = 6$, but this error is corrected by the feedback from the actual queue size.

B. Shared channel scenario

To simulate a realistic scenario of how the rate fluctuates for an end-user it is important that all the aspects which are impacting the end-user rate are considered. As described in Sec. III, the rate fluctuates due to variations in radio quality and cell load. The load variation in number of users depends on statistical variations in call originsations but also due to mobility when users move between cells.

We use a two stage simulation: First, the radio system is simulated in a fairly loaded scenario, where users move around and download files using TCP/IP. From this radio level simulation, we extract the available bandwidth per user in one of the cells, which is a time varying signal. The second stage of the simulation implements a link with this particular bandwidth variation in the ns-2 network simulator [23], for evaluation of the RNF mechanism.

The setup for the radio layer simulation was as follows: 7 sites and 21 cells, 1500 m site-to-site distance, realistic radio and fading models, and end-users moving at 1 m/s while downloading files of 50 kbyte via TCP/IP. The output result from the radio layer simulations, the end-user rate level and fluctuation, is of course different depending on how high system loaded that is used in the simulation. The result is also depend on which services that are used. However, we believe that on the time scales that matter for TCP-traffic, the character

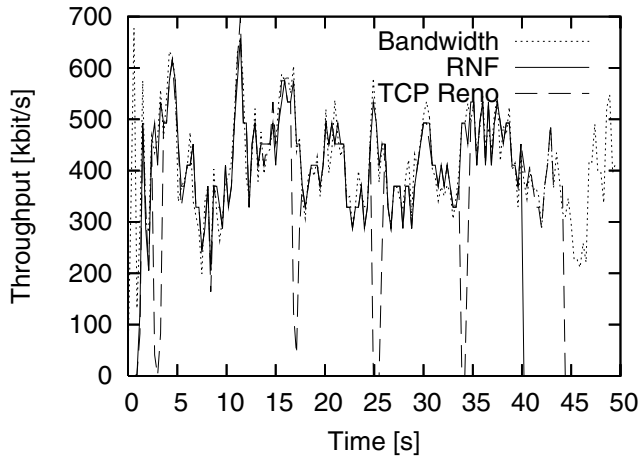


Fig. 5. Resulting throughput for the HSDPA scenario. The dotted line is the time-varying bandwidth to the terminal, the dashed line is the throughput for TCP Reno, and the solid line is the throughput using the proposed RNF controller.

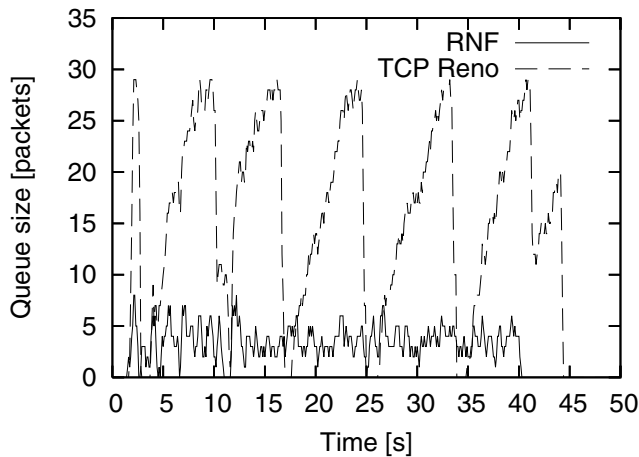


Fig. 6. Resulting RNC queue size for the HSDPA scenario. The dashed line is the queue size with TCP Reno, and the solid line is the queue size with the RNF controller. Like in the dedicated channel scenario, the RNF controller results in a small queue.

of the bandwidth variations are similar for a wide range of load levels and services.

As discussed in Sec. IV-C, the bandwidth varies continuously, and there are no distinct stepwise changes of the bandwidth. In this scenario, we base the bandwidth feedforward on regular sampling instead of the event-based signalling in the previous scenario. We use a sample time of 0.5 s, and for each sample, the RNC calculates the averaged bandwidth available to the user during the previous 0.5 s interval. The queue feedback is omitted—it was intended to operate on a time scale faster than the feedforward control and slower than the roundtrip time, and is not applicable here.

Figure 5 shows the resulting throughput as a function of time, and Fig. 6 shows the evolution of the RNC queue. The results are similar to the case of a dedicated channel: The RNC queue stays small, and the total download time is a few

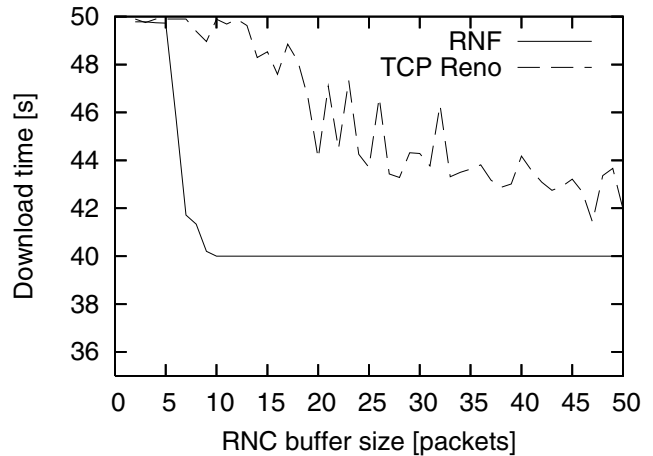


Fig. 7. Total download time, as a function of RNC buffer size. A download time of 40 s corresponds to almost full utilization of the radio link.

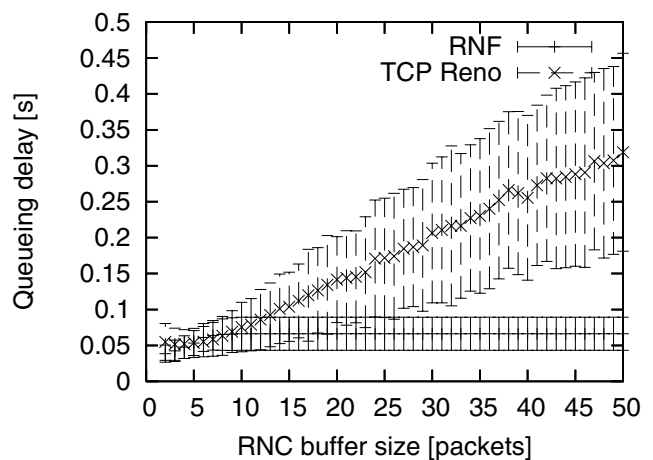


Fig. 8. Average and standard deviation of the one-way packet delay during the download, as a function of RNC buffer size. With TCP Reno, both average delay and the delay jitter increases with the buffer size.

seconds shorter than with TCP Reno.

C. Discussion

In these simulations, we see that the RNF-mechanism reduces the user response time, which also corresponds to higher link utilization and fewer queue underflows. An alternative way to increase link utilization is to simply increase the RNC buffer size. This is illustrated in Fig. 7, which shows the response time as a function of the queue size.

However, when the RNC buffer size is increased, queue delay and jitter also grows, as shown in Fig. 8. We see that with TCP Reno, we have a difficult trade-off between high utilization and short packet delay, while the RNF solution achieves high utilization with a fairly small buffer.

A large RNC queue is problematic for several reasons. At hand-over to a different RNC or to a different system, e.g., a local WLAN, a large RNC queue means that there is more data that has to be transferred or discarded, making a smooth

hand-over more difficult. When a user cancels a download, e.g., by clicking the browser stop button and then clicking another link, all data in the RNC queue will still be transmitted over the radio link, and discarded by the user terminal. In this way, a large RNC queue increases the response time to the user's browsing, and wastes radio resources. Furthermore, higher queueing delay and more jitter implies degraded quality for any real-time traffic sharing the link.

Real-time services provided by the operator, such as voice, need not suffer from a large RNC queueing delay, since the operator is in control of the link layer, and can ensure that this real-time traffic is given priority.

However, no other players have this control. Any third-party real-time service, such as voice, video, or online games, is restricted to end-to-end solutions: Software is installed at the communication end-points, but modifications or tweaks to the network infrastructure are impossible. Therefore, this traffic will most likely share the same RNC queue with web-browsing and downloads.

It is remarkable that by violating the end-to-end principle for one service, web-browsing, we improve the quality for other services which adhere to the end-to-end principle.

VI. CONCLUSIONS

We have proposed a cross-layer design to improve user response time, and link utilization, for web-browsing and downloads to a mobile terminal. The Radio Network Controller (RNC) sends information about the available bandwidth and the current queue length to a web-proxy. The proxy uses this information to control its sending window, replacing the congestion control of standard TCP. For ease of deployment, the modifications are isolated to the RNC and the web-proxy, both belonging to the operator's administrative domain.

The mechanism is evaluated by comparing the resulting throughput to TCP Reno in two scenarios. The first scenario models a dedicated channel which switches between different discrete values of the bandwidth. The second scenario is based on simulations of the High-speed Downlink Shared Channel recently introduced by 3GPP, where a user's share of the available bandwidth is varying continuously.

In both scenarios, we find modest gains in user response time and link utilization. Furthermore, the proposed mechanism achieves the high link utilization with a much smaller RNC queue than is needed with TCP Reno. This reduced queueing is particularly beneficial for third-part real-time services.

A. Ongoing work

To further evaluate the mechanism, we are comparing it to other proposed mechanisms in the literature.

Using the average bandwidth over the previous interval to predict the bandwidth in the near future is a very simple mechanism. When introducing more sophisticated bandwidth estimators, care must be taken so that the time delay for estimator convergence doesn't make the information too out-of-date. Another possible approach is to use the queue size as a trigger, and generate an RNF message only when the queue size crosses an upper or lower threshold value.

REFERENCES

- [1] N. Möller, I. C. Molero, K. H. Johansson, J. Petersson, R. Skog, and Å. Arvidsson, "Using radio network feedback to improve TCP performance over cellular networks," in *IEEE Conference on Decision and Control and European Control Conference*, Seville, 2005.
- [2] N. Möller, K. H. Johansson, and K. Jacobsson, "Stability of window-based queue control with application to mobile terminal download," in *Mathematical Theory of Networks and Systems*, Kyoto, July 2006.
- [3] A. DeSimone, M. Chuah, and O. Yue, "Throughput performance of transport-layer protocols over wireless LANs," in *IEEE Globecom'93*, 1993, pp. 542–549.
- [4] G. Xylomenos and G. C. Polyzos, "Internet protocol performance over networks with wireless links," *IEEE Network*, vol. 13, no. 4, pp. 55–63, 1999.
- [5] R. Ludwig and R. H. Katz, "The Eifel algorithm: Making TCP robust against spurious retransmissions," *ACM Computer Communication Review*, vol. 30, no. 1, January 2000.
- [6] S. Mascolo, L. A. Grieco, R. ferorelli, P. Camarda, and G. Piscitelli, "Performance evaluation of Westwood+ TCP congestion control," *Performance Evaluation*, no. 55, pp. 93–111, January 2004, special Issue with Selected papers from Globecom 02.
- [7] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Trans. on Networking*, vol. 11, no. 5, pp. 703–717, 2003.
- [8] S. Liu, T. Basar, and R. Srikant, "Controlling the Internet: A survey and some new results," in *IEEE Conference on Decision and Control*, Maui, Hawaii USA, 2003, pp. 3048–3057.
- [9] E. Hossain, D. I. Kim, and V. K. Bhargava, "Analysis of TCP performance under joint rate and power adaptation in cellular WCDMA networks," *IEEE Trans. on Wireless Comm.*, vol. 3, no. 3, May 2004.
- [10] F. Lefevre and G. Vivier, "Optimizing link layer parameters for a TCP connection," in *Proceedings of VTC 2001 Spring*. IEEE, 2001.
- [11] N. Möller, K. H. Johansson, and H. Hjalmarsson, "Making retransmission delays in wireless links friendlier to TCP," in *IEEE Conference on Decision and Control*. IEEE, 2004.
- [12] C. Fischione, F. Graziosi, F. Santucci, N. Möller, K. H. Johansson, and H. Hjalmarsson, "Analysis of TCP over WCDMA wireless systems under power control, MAI and link level error recovery," in *IWCT*, June 2005.
- [13] C. Barakat and E. Altman, "Bandwidth tradeoff between TCP and link-level FEC," *Comput. Networks*, vol. 39, no. 5, pp. 133–150, 2002.
- [14] T. Chahed, A.-F. Canton, and S.-E. Elayoubi, "End-to-end TCP performance in W-CDMA/UMTS," in *ICC*, Anchorage, May 2003.
- [15] D. Barman, I. Matta, E. Altman, and R. E. Azouzi, "TCP optimization through FEQ, ARQ and transmission power tradeoff," in *International Conference on Wired/Wireless Internet Communications WWIC*, february 2004.
- [16] G. Carneiro, J. Ruela, and M. Ricardo, "Cross-layer design in 4G wireless terminals," *IEEE Wireless Communications*, vol. 11, no. 2, pp. 7–13, April 2004.
- [17] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 3–11, Februari 2005.
- [18] Mohamad Assaad and Badiou Jouaber and Djamel Zeghlache, "TCP performance over UMTS-HSDPA system," *Telecommunication Systems*, vol. 27:2-4, pp. 371–391, 2004.
- [19] S. Kavirat, S. Ponnekanti, and C. Wijting, "Effective cross layer radio access design assisted by the location manager for systems beyond 3G," in *London Communications Symposium 2003*, London, U.K., 2003.
- [20] F. Kalleitner and M. Konegger, "A cross-layer framework for content based fine-grained scheduling of audiovisual streams over wireless network," in *IASTED CIIT 2005*, Cambridge, MA., U.S.A., October 2005.
- [21] L.-C. Wang, M.-C. Chen, and Y.-C. Chen, "A cross-layer cost-function based rate adaptation mechanism for the wcdma system with multi-class services by transport format selection," *Wireless Communications and Mobile Computing*, vol. 5, no. 4, 2005.
- [22] "3rd Generation Partnership Project; technical specification group radio access network; High Speed Downlink Packet Access (HSDPA), overall description," 3GPP, TS 25.308 V7.0.0 (2006-03).
- [23] *The Network Simulator ns-2*, Information Sciences Institute, University of Southern California, <http://www.isi.edu/nsnam/ns>.