Hints and Notifications

Lars-Åke Larzon Ulf Bodin Olov Schelén
Dept. of CS&EE
Luleå University of Technology
SE-971 87 Luleå, Sweden

Abstract— **With current Internet protocols, users may experience low and unpredictable forwarding quality at wireless links. This is due to varying link properties caused by changing radio conditions. Decreased forwarding quality can cause severe degradation in utilization. This is undesirable since forwarding capacity often is expensive at wireless links because of limited radio spectrum. Allowing the application and transport layers to communicate with wireless link layers can improve forwarding quality and utilization.**

We propose to enable inter-layer communication by adding hints and notifications (HAN) to the Internet architecture. Hints can be introduced and used without notifications, while notifications need hints or a similar mechanism to operate. By using IP options and ICMP messages to implement HAN, a backward-compatible partial deployment is possible. With HAN, the network layer becomes truly wireless friendly and radio spectrum can be used efficiently while supporting both real-time and traditional data applications.

I. Introduction

We propose introducing support for inter-layer communications to the Internet architecture in form of *hints and notifications (HAN)*. HAN enables spectrum-efficient data handling, which can improve service quality for individual users and utilization of wireless links. At wireless links, efficient usage of the radio spectrum is essential for reasons of cost-efficiency. HAN provides generic extensions to the network layer that improve transport of both real-time and traditional data over various wireless link technologies.

A strong trend is the increasing number of mobile devices connecting to the Internet using IP [15] over low- to medium-bandwidth wireless link technologies. Providing IP connectivity *all the way* to these devices is appealing since that purifies the infrastructure and enables simple access to the wide range of existing applications.

Constantly changing radio conditions can make the forwarding quality unpredictable at wireless links. Wireless link technologies may provide mechanisms for adaptive forward error correction (FEC) and automatic repeat request (ARQ). These mechanisms can introduce variations in delay and available bit-rate at higher layers. Moreover, the redundancy introduced by FEC and ARQ retransmissions may decrease the utilization of scarce forwarding resources at wireless links.

The radio spectrum available to any wireless link technology is a limited resource, and it is essential to utilize this scarce capacity as efficiently as possible to provide cost-efficient connectivity. Unnecessary transmissions and delays in the link layer are desirable to avoid. We identify the following four problems associated with the efficiency of radio spectrum utilization:

- *Unknown payload properties*. Properties of packet payloads are unknown at the link layer. With payload information, link utilization can be improved for bit-error tolerant applications (i.e., re-transmitting or discarding packets due to acceptable payload errors can be avoided).
- *Unknown link properties*. Link layer properties such as available bandwidth and error-rates are unknown at higher layers. Using such information, applications can adapt dynamically to suitable codecs, frame rates, etc.
- *Misinterpreted loss*. Congestion-responsive applications interpret packet losses as signs of persistent congestion and cannot react properly to losses caused by bit-errors or temporary congestion (i.e., shorter than a typical RTT).
- *Temporary disconnection*. After a temporary disconnection, congestion-responsive applications are often late in detecting re-established connectivity, which can cause the link to be under-utilized for a period.

To solve the problems listed above, information need to be exchanged between the link layer and layers above the network layer. The layered design of the Internet protocols does not however support such information exchange. History has proven that this strict layering is very useful in supporting Internetworking over the highly heterogeneous link technologies present in today's Internet. Therefore, we consider it important to preserve the layered design of IP. This can be achieved by allowing transparent operation over parts of the network where inter-layer communication is unsupported. Transparent operation also ensures full backward compatibility with the already installed IP infrastructure.

We propose adding *hints and notifications (HAN)* to the Internet architecture. By making HAN part of the network layer, it enables inter-layer communication without violating Internet layering principles. By getting *hints* from upper layers, link layers can use spectrum-conservative mechanisms to a greater extent. Hints can also improve flexibility for application designers by allowing data with different requirements to be placed in the same packet. By sending *notifications* from lower layers, applications and transport layers can react more accurately to variations in the physical medium.

There are several proposals for inter-layer communication aiming at enhancing performance in wireless networks. Most of them are however incompatible with each other and aim at specific improvements. Deploying all of these proposals would result in an unnecessarily complex system. The generic design of HAN supports any application, transport, or (wireless) link layer. HAN can be implemented as a part of the network layer to simplify successive deployment without interfering with existing Internet nodes. Hints are independent of notifications, while notifications require usage of hints or some similar mechanism to transport notification requests with data packets.

This paper is organized as follows. In the next section, we describe related work aiming at solving parts of the outlined problems. We then specify demands from real-time and congestion-responsive applications in Section III. The HAN concept is described in Section IV, followed by illustrative examples in Section V. Finally, the paper is concluded in Section VI.

II. Related work

There is a proposal for a new UDP protocol [10], in which applications can specify part of a packet as less sensitive to errors. The purpose of new UDP is to avoid unnecessary packet drops by supporting delivery of damaged packers. Another effort within the IETF is the Datagram Control Protocol (DCP) [9], which also supports that part of a packet is insensitive to errors. However, for neither of these proposals, there is currently no solution on how to propagate information about error-insensitive payload to lower layers. Hints provide a generic mechanism to propagate such information to lower layers.

For layered data, STRIP [6] provides a mechanism where the payload of an IP packet can be layered. When congestion occurs or if a bottleneck link is reached, layers can be stripped from the packet instead of discarding the entire packet. Alas, STRIP only work for certain transport protocols since its modifications to transport protocol headers requires knowledge about their semantics. Nevertheless, STRIP is a similar concept as hints with regard to avoiding packet losses by layering data with different properties and importance.

For TCP, several extensions to improve operation over wireless links have been proposed. These proposals include I-TCP [1], MTCP [4], M-TCP [16] and Snoop [2], which all address the problem of losses caused by bit-errors or temporary congestion. The proposals introduce per-connection states in access nodes, which requires extra processing and overhead. Moreover, at a handover, states need to be either transferred to or reestablished in a new radio access node.

Other proposed extensions to TCP include Freeze-TCP [7] and TCP-Probing [17], which address the problem of wireless hosts being temporarily disconnected. These proposals do not require any per-connection states in radio access nodes (i.e., they are pure end-to-end solutions), but do not effectively handle the problem of losses caused by bit-errors or temporary congestion.

In addition to TCP extensions, there are proposals addressing the problem of losses caused by bit-errors or temporary congestion that are TCP-unaware. DDA [18] attempts to provide the same functionality as Snoop without looking at TCP segment numbers. TULIP [13] attempts to provide a specific service at the link layer for upper layers that provide reliable transport. However, these approaches also introduce states in radio access nodes.

In [8], enhancements to TCP are proposed making it capable of distinguishing between losses caused by persistent congestion and losses caused by bit-errors at wireless links. Two new ICMP message types are used to inform TCP sources of that a packet is delayed due to local retransmissions (i.e., ARQ) and of that a packet is lost due to that these retransmissions have failed. With the concept of HAN, we extend this proposal to include more message types. We also limit the amount of messages by only sending them as response to explicit requests included in hints.

Real-time and traditional data applications can use HAN over various wireless link layer technologies. HAN requires no states in network nodes and control packets (i.e., notifications) are only sent to HAN aware traffic sources. Hence, HAN provides a generic and stateless approach of extending the Internet protocols in contrast to the TCP related approaches discussed above, which each aims at a limited number of improvements for a specific transport protocol (i.e., TCP) or link layer.

III. Demands

Forwarding capacity of wireless links is expensive due to radio spectrum limitations. To achieve high utilization of the spectrum, error-rates are generally orders of magnitudes higher than in wired networks. This is because moderate increases in error-rate typically result in considerable higher bit-rates in higher layers. High bit-error rates can be hidden to higher protocol layers through local retransmissions at the link layer. This introduces however extra delay and degrades the bit-rate available in upper layers.

Radio conditions vary at different time-scales, ranging from milliseconds up to minutes. Upper layers will be exposed to these errors and must compensate for them. In this section, we discuss demands from protocols at higher layers on mechanisms that increase the utilization of the expensive forwarding capacity over wireless links.

A. Real-time applications

Today, there are applications that suffer more than others from packet losses in the Internet. Audio and video codecs can often cope with some degree of errors in the received data and still find it useful. For applications using such codecs, receiving packets with errors in the payload may be preferred over loosing packets or having them delayed. The packet will however be discarded if there are errors in the packet header. The error tolerance may differ among blocks of data in the payload. Consequently, information about data properties (e.g., error tolerance and delay requirements) is needed to support cost-effective usage of the radio spectrum allocated for wireless links.

Cost-effective usage of radio spectrum can be achieved through less protective link layer FEC and ARQ strategies for error tolerant data blocks. Moreover, unequal error protection (UEP) [11] can be used when data blocks have different error requirements [5][19].

In the current Internet architecture, there is no straightforward way for applications using bit-error tolerant codecs to pass information about data properties to the link layer. Mechanisms for service differentiation allow for multiple levels of priority between packets (e.g., mechanisms specified in the Differentiated Services architecture [3]). These mechanisms do not however support multiple priority levels for data blocks within individual packets. We refer to this as the *unknown payload properties* problem.

Real-time applications could also benefit from getting information about current link layer conditions (e.g., available bandwidth, error rates, etc.). This could help a real-time transmission of video or audio to adapt dynamically to suitable codecs, frame rates, etc. We refer to this as the *unknown link properties* problem.

B. Congestion-responsive applications

An application is said to be congestion-responsive if it responds to packet loss by reducing its sending rate. Since congestion-responsive applications rely on feedback from receivers, they adjust sending rates at time-scales longer than their RTTs.

Common for congestion-responsive applications is that they interpret packet losses as signs of persistent congestion. However, at wireless links, a considerable number of packets are lost due to changing radio conditions. Moreover, packets may be interpreted as lost if they are delayed due to link layer retransmissions.[1] Consequently, congestion-responsive applications may underutilize wireless links with high bit-error rates, available bit-rates that vary at short time-scales (i.e., time-scales shorter than a typical RTT), or both. We refer to this as the *misinterpreted loss* problem.

Another problem associated with wireless links is that hosts may be temporarily disconnected. Congestion-responsive applications that rely on feedback from receivers are late in detecting disconnections. Moreover, they might detect re-established connections late and need additional time to find the appropriate sending rate, causing the link to be under-utilized for a period. This becomes particularly problematic if the sending rate is reset after a re-establishment. We refer to this as the *temporary disconnection* problem.

To avoid the problems described above, congestion-responsive applications must be able to distinguish between losses related to persistent congestion, temporarily congestion, bit-errors, and disconnections. Applications should also be able to distinguish between packets being lost and packets being delayed. Moreover, as a wireless connection is re-established congestion-responsive applications should be able to immediately find the appropriate sending rate. The performance of these applications can be improved with a mechanism that notifies sources explicitly on the appropriate sending rate, why packets are dropped and whether they have been delayed due to local retransmissions.

IV. HINTS AND NOTIFICATIONS (HAN)

The Internet protocols were originally designed with the implicit assumption that bandwidth, delay, error-rates and jitter do not vary much at the link layer. When exposed to networks where these metrics vary, these protocols experience problems and perform poorly. HAN addresses these problems and aim at improving the performance of the Internet protocols when used under these conditions.

The need for information exchange between the link and transport layers outlined in section III can be satisfied locally by violating layering principles. We believe however

[1] For TCP, this might cause a timeout, which forces TCP into slow-start.

that the layering of the successful Internet architecture should be preserved to not increase the risk of malfunction due to many complex additions, where each addition only supports a few combinations of link layers and applications or transport layers. To preserve the layering, we propose to place information exchange mechanisms in the network layer itself. Thereby, link layers and transport layers or applications need not know each other's semantics in order to exchange information.

Hints and notifications are placed in the network layer headers where they are parsed by surrounding layers[2]. We propose to use an IP option [15][3] for hints and notification requests, and ICMP messages [14] for two different types of notifications. This simplifies a partial deployment of HAN in the existing Internet. Nodes capable of parsing hints do so, and provide their link layers with the information. For nodes unaware of the concept with hints, packets will be forwarded as usual with the IP option carrying the hints intact. Using ICMP enables deployment without inventing a new protocol to provide notifications.

A disadvantage of using the IP options/ICMP solution is that packets might be processed in the slow path of intermediate routers. An alternative could be to use a bit in the IP header to indicate the presence of HAN information. Core routers where HAN might not be used as frequently can then process IP packets in the fast path. Though this solution would be better from a performance perspective, it would be harder to deploy - mainly because of political issues.

A. *Hints*

Today, varying physical conditions on a link can prevent queued packets from being transmitted immediately. By providing hints from upper layers on how to treat packets at periods of bad radio conditions, unwanted and unnecessary packet losses and delays can be avoided. In its simplest form, a hint includes information about the boundaries between different parts of a packet and the acceptable error-rate and delay requirements for each part. The link layer can use this information to choose suitable modulation, error correction and verification mechanisms on a per-packet basis. This makes it possible to quickly react on short-time variations.

There are no strict rules of how the hints are used by the link layer, i.e., upper layers can not exactly instruct the link layer of how to treat frames. Neither is there any requirement of what mechanisms for error correction or recovery that the link layer must provide in order to use the hints. Instead, the hints are, as the name indicates, merely additional information of packet internals that the link layer may use as it finds suitable.

[2]In a practical implementation, the information processing could be done in the network layer and use out-of-band signaling between the network layer and other layers

[3]For IP version 6, this IP option is a per-link option.

B. *Notifications*

As mentioned in Section III-B, congestion-responsive applications might suffer from the misinterpreted loss and the temporary disconnection problems. To handle these problems, *link action notifications* can be used to inform sources of why packets are dropped and to inform sources of long packets delays caused by local retransmissions.

Furthermore, as mentioned in Section III-A, real-time applications might suffer from the unknown link-properties problem. To handle this, *link status notifications* can be used to inform sources of current and maximum available bit-rate at a wireless link. These notifications can also be used to detect re-established connectivity after a temporary disconnection.

In the following two sections, we present the two different types of notifications — link action notifications and link status notifications.

B.1 Link action notifications

Link action notifications are messages that origin from actions associated with individual packets, e.g., local retransmissions of link layer frames causing a packet to be delayed.

We have identified the following link actions that can be used to generate notification messages.

- *Packet delayed due to local retransmissions*. This information can, e.g., be used by a congestion-responsive source to avoid an unnecessary time-out, redundant retransmission at a higher layer, or both.
- *Packet dropped due to bit-errors* and *Packet dropped due to temporary congestion*. These can, e.g., be used by a congestion responsive source to avoid an unnecessary reduction of sending rate.
- *Packet dropped due to persistent congestion*. This can be used as a fast congestion notification (FCN).
- *Packet dropped due to that the wireless host is disconnected*. Upon reception of this information, an application can pause the data transmission until the the connection is re-established.

Link action notifications are put in separate packets to avoid having link layers monitoring transport and application layer headers, which would require per-flow states. To reduce the amount of notification traffic, notifications are sent only to notification-aware sources. Notification awareness is indicated in outgoing IP packets (e.g., as part of hints).

B.2 Link status notifications

Link status notifications are messages that contain information of static and dynamic link properties. For example, the maximum bit-rate of a wireless link is often a static parameter while the bit-error probability may vary.

We have identified the following link status notification messages.

- *Currently available bit-rate*. This information can for example be used by a TCP source to select an initial sending window after re-establishing a temporarily disconnected connection.
- *Maximum available bit-rate*. This information can be used as an alternative to probing to estimate link capacity. TCP, which normally reduce the sending speed significantly after exceeding the link capacity, could benefit from this.

B.3 Implementation

By defining notifications to be initiated upon request from sources aware of notifications, no states are needed in network nodes. Information about notification awareness and what types of notifications the source is interested in can be included in hints, or sent in separate packets. The stateless approach is attractive when users are highly mobile, since access points do not have to manage neither per-host or perflow states, nor state migrations. If notifications are sent on request only, sources that are interested in notifications without any traffic to send must periodically probe the network, e.g., with ICMP packets.

V. Example scenarios

In this section, we present two example scenarios that illustrate the use of hints and notifications. First, a scenario for real-time applications is provided in Section V-A. Next, a scenario for congestion-responsive applications is provided in Section V-B.

A. Real-time applications

Figure 1 illustrates a real-time application which use hints to avoid unnecessary packet drops and delays in the link layer. The application sends data which is logically divided into two parts, *D1* and *D2*, with different error tolerance.

In the link layer, one of four actions can occur. First, the packet could be sent unmodified. Second, the packet could be discarded - possibly after a delay. If requested, a link action notification of the packet loss is sent back to the network layer. These two actions can occur even if hints are not used, with the exception that the hints can specify the maximum

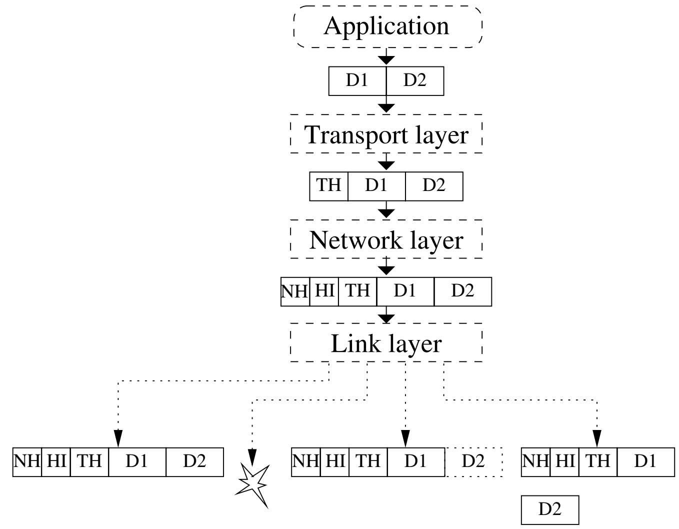

Fig. 1. Example usage of hints.

acceptable delay. The last two types of actions are enabled by usage of the hints.

In the third action, the packet is transmitted, but with a higher error probability for *D2*[4]. A similar action is the fourth one, in which *D2* is separated from the rest of the packet and sent on its own with possibly other coding and/or error probability than the first part of the packet.

What without hints is restricted to the two leftmost actions has by the introduction of hints been extended to four possible actions. Also, delayed transmission of packets that will be unusable at the receiver can be avoided to a greater extent.

B. Congestion-responsive applications

This example scenario illustrates how a TCP implementation can improve its operation over wireless links using HAN. Assume that an application at a mobile host is downloading data from a host connected to the Internet via a wired link (Figure 2). To optimize the TCP transmission efficiency, the mobile host informs the data source of that HAN should be used. If the data source is aware of HAN, it applies a hint to its first packet and sets a link status query flag in this hint (1). The base station initiates a link status message that contains information of the maximum available bit-rate (MABR) at the link (2). This information is used to limit the sending rate by dividing MABR with the estimated RTT. The data source also adds hints that requests link action notifications to all

[4] Some link layers that can divide frames in multiple segments with different error protection and correction properties.

packets (3).

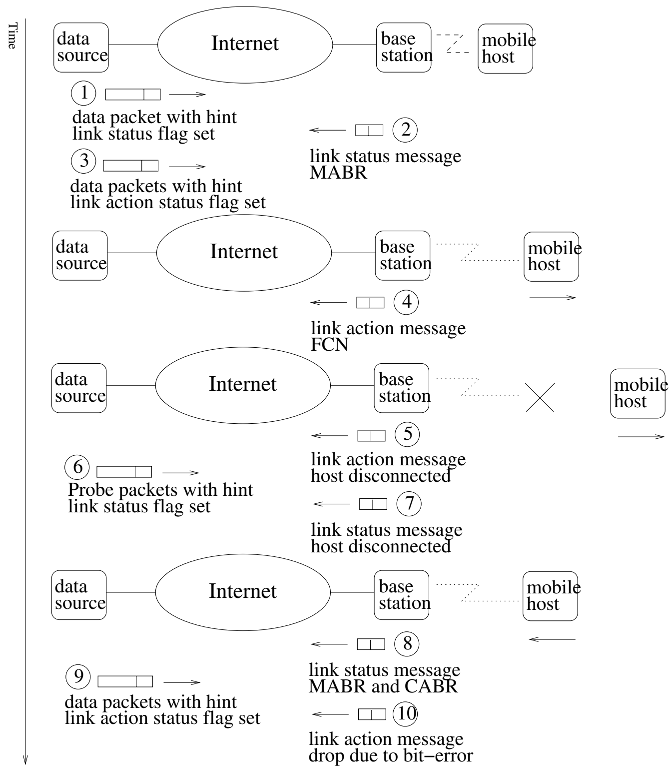

Fig. 2. Example usage of notifications.

After some time, the link quality decreases as the mobile host moves. The available bit-rate decreases and a packet is dropped to signal congestion. When this happens, the base station initiates a link action message indicating that a packet has been dropped due to persistent congestion (4). This can be interpreted as a fast congestion notification (FCN) by the data source, which enables the TCP source to invoke fast recovery before receiving duplicate ACKs and thereby reduce the risk for additional packet drops.

As the mobile host continues to move, it eventually loses contact with the base station, which causes packets to be dropped. When the base station detects the lost connection, it initiates a link action message for each dropped packet containing information of that the host has been disconnected (5). Upon receiving these messages, the source halts the data transmission and starts to send probe packets periodically (e.g., once every second). Hints are applied with a link status query flag set to these probe packets (6). The base station initiate link status messages containing information of that the host is still disconnected as responses to these probe packets (7) until the mobile host gets connected again (possibly via a new base station[5]). Then, a link status message is initiated to inform the source of that the connection is re-established (8). This message also contains information of the current available bit-rate (CABR) and the MABR at the re-established connection. Upon reception of the link status message the source resumes data transmission. The sending window is set to the smallest of the size before the disconnection and the CABR divided with the RTT of the link status message. The sending window limit is updated to fit the new MABR. As before, the data source also applies hints with the link action query flag set to the packets sent (9).

Unfortunately, the new connection suffers from high bit-error rates and a packet eventually needs to be dropped due to failed local retransmissions over the wireless link. This triggers a link action message to be initiated that containing information of that the drop is due to bit-errors and not congestion (10). The message enables the data source to avoid reducing its sending window. Instead, it can just retransmit the lost packet and continue in the congestion avoidance phase.

The messages containing information about disconnection and re-establishment can be used by TCP-probing [17] to improve the detection time for these events. The information that a packet has been dropped due to bit-errors addresses the misinterpreted loss problem, which is also addressed by e.g., I-TCP [1], MTCP [16], M-TCP, Snoop [2], DDA [18], TULIP [13], and more. The message provided by HAN is possibly less effective, but could with HAN deployed simplify development of mechanisms as specific as the ones mentioned.

VI. Conclusions

In this paper, we define the concept of hints and notifications (HAN). HAN is justified by needs for spectrum-efficient data handling, improved service quality for individual users and high utilization of wireless links. Limited radio spectrum makes forwarding capacity expensive at wireless links, which emphasizes these needs.

HAN enables application and transport layers to communicate with various link layer technologies without violating layering principles. Thereby, HAN fits naturally into the layered Internet architecture. Inter-layer communication can improve forwarding quality and spectrum utilization. With hints, real-time applications can inform the link layer of delay requirements and error tolerance. In addition, notifications enables feedback of link layer actions and current sta-

[5]Probe packets are assumed to be routed via the new base station as soon as the connection is established

tus to application and transport layers. This information can help congestion-responsive applications increase link utilization by reacting properly to different types of packet loss.

By using IP options and ICMP messages to implement HAN, a backward-compatible and partial deployment in the current Internet is possible. Hints can be introduced and used without notifications, while notifications need hints or a similar mechanism to operate.

We plan to implement selected parts of HAN in the network simulator (NS) [12] to evaluate it further for different scenarios. Moreover, we intend to produce an open implementation of HAN to perform application tests (i.e., to examine the interaction between HAN and its target applications).

REFERENCES

[1] Ajay Bakre and B. R. Badrinath. I-TCP: Indirect TCP for mobile hosts. Technical report, 1995.

[2] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy H. Katz. Improving TCP/IP performance over wireless networks. *In proc. 1st ACM Int'l Conf. on Mobile Computing and Networking (Mobicom)*, page 10, 1995.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. Request For Comments RFC 2475, DARPA, December 1998.

[4] Kevin Brown and Suresh Singh. M-TCP: TCP for mobile cellular networks. *ACM Computer Communication Review*, 27(5), 1997.

[5] A. R. Calderbank and N. Seshadri. Multilevel codes for unequal error protection. *IEEE Trans. Inform. Theory*, 39:1234–1248, July 1993.

[6] Mattias Engan, Pierre Fransson, and Olov Schelen. Selective truncating internetwork protocol: experiments with explicit framing. *Proc. of the International Conference for Computer Communication and Networks (ICCCN 2000)*, 2000.

[7] T. Go, J. Moronski, D. Phatak, and V. Gupta. Freeze-tcp: A true end-to-end enhancement mechanism for mobile environments, 2000.

[8] Samir Goel and Dheeraj Sanghi. Improving performance of tcp over wireless links. *In Proc. TENCON 98 - 1998 IEEE Region Ten Conference on Global Connectivity in Energy, Computer Communication and Control, December 17-19*, page 4, 1998.

[9] Eddie Kohler, Mark Handley, Sally Floyd, and Jitendra Padhya. Datagram Control Protocol (DCP). Internet-Draft (work in progress) , draft-kohler-dcp-01.txt, ACIRI, December 2001.

[10] Lars-Åke Larzon, Mikael Degermark, and Stephen Pink. Efficient use of wireless bandwidth for multimedia applications. In *Proceedings of the IEEE Mobile Multimedia Communications (MOMUC) Workshop*. IEEE, November 1999.

[11] B. Masnick and J. Wolf. On linear unequal error protection codes. *IEEE Trans. Inform. Theory*, IT-13:600–607, October 1967.

[12] Ns homepage. URL: http://www.isi.edu/nsnam/ns/.

[13] C. Parsa and J. J. Garcia-Luna-Aceves. Improving TCP congestion control over internets with heterogeneous transmission media. *Proc. IEEE ICNP 99: 7th International Conference on Network Protocols*, 1999.

[14] Jon Postel. Internet Control Message Protocol. Request For Comments RFC 792, DARPA, September 1981.

[15] Jon Postel. Internet Protocol. Request For Comments RFC 791, DARPA, September 1981.

[16] Injong Rhee, Nallathambi Ballaguru, and George N. Rouskas. MTCP: Scalable TCP-like congestion control for reliable multicast. Technical Report TR-98-01, 2, 1998.

[17] Vassilios Tsaoussidis and Hussein Badr. TCP-probing: Towards an error control schema with energy and throughput performance gains. In *Proceedings of the 8th IEEE Conference on Network protocols*. IEEE, November 2000.

[18] N. Vaidya, M. Mehta, C. Perkins, and G. Montenegro. Delayed duplicate acknowledgements: A TCP-unaware approach to improve performance of TCP over wireless. *Technical Report, Computer Science Dept., Texas A&M University*, 1999.

[19] L.-F. Wei. Multilevel codes for unequal error protection. *IEEE Trans. Commun.*, 41:1439–1449, October 1993.