Transport Level Mechanisms for Bandwidth Aggregation on Mobile Hosts

Luiz Magalhaes and Robin Kravets
Department of Computer Science
University of Illinois, Urbana-Champaign
1304 W Springfield Avenue
Urbana, IL 61801
{magalhae,rhk}@cs.uiuc.edu

Abstract

Present mobile computing does not support the simultaneous use of multiple heterogeneous network interfaces for a single transport layer connection. In this paper, we describe a solution for channel aggregation at the transport layer, which provides increased bandwidth to mobile nodes. We present R-MTP (Reliable Multiplexing Transport Protocol), a rate-based reliable transport protocol capable of multiplexing data from a single application data stream across multiple network interfaces. Due to the lossy nature of wireless links in mobile environments, R-MTP tracks packet interarrival time for discrimination between congestion-based and transmission-based losses as well as better bandwidth estimation. The challenges to such a reliable protocol lie in the coordination of packets across streams with varying channel characteristics. Our experimental results validate R-MTP's bandwidth estimation and loss characterization techniques. Successful bandwidth aggregation is demonstrated in ideal and lossy environments

1 Introduction

With the growing popularity of wireless access, coverage areas are growing and often overlapping, enabling more than one communication technology in a specific area. This fact combined with the availability of hosts with multiple interfaces enables the simultaneous use of multiple communication channels, adding an untapped source of additional bandwidth to mobile networking systems. Current channel aggregation techniques are limited to identical channels across a homogeneous link layer. The goal of our research is the aggregation of available resources from multiple heterogeneous channels to create a virtual end-to-end channel with better characteristics than each channel alone. To this end, we have designed and developed R-MTP (Reliable Multiplexing Transport Protocol), a rate-based transmission protocol for the reliable transmission of bulk data to mobile nodes. The use of R-MTP preserves end-to-end semantics, transparently providing the application with the simultaneous use of multiple channels by multiplexing data from the application across a set of available channels. Our approach exposes link-layer connectivity and per channel resource information to the transport layer, allowing R-MTP to adapt to both changes in available bandwidth on each channel and changes in availability of channels. As the available channel resources change, R-MTP adapts, changing the fraction of flow that is being sent on each channel and adding or removing channels as necessary. For each channel, R-MTP monitors delay and interarrival time for use with bandwidth estimation and congestion control. In addition, R-MTP uses information generated by its rate-based mechanism (interarrival time) for the classification of losses (i.e. congestion losses vs. transmission errors) on the wireless link.

The key contribution of our research is the aggregation of resources across multiple heterogeneous channels in a mobile environment. The simultaneous use of communication channels from multiple technologies enhances communication support for mobile devices in several ways. The first advantage is the increase in bandwidth through bandwidth aggregation across channels. In traditional wired communication, the bottleneck link to a host is often some router in the network. Additional bandwidth at the endhost will not alleviate this problem. In wireless environments, the bottleneck link is often the last hop. Combining multiple channels into a single virtual channel can increase the capacity of the last hop. The second advantage is to free the system from committing to a single choice of a link-layer channel. By providing information about currently available channels, the system can choose some subset of these channels for its communication. In addition, in the event of a blackout or congestion in any of the active link layers, the use of multiple link layers will enable the host to stay connected. Using every available link layer all the time may not always be the best solution. The addition of a resource poor channel may actually hurt performance, the complexities of which we discuss later in the paper. In

1092-1658/01 $17.00 © 2001 IEEE

general, the choice of which channels to use depends on the constraints set by the user (e.g. cost), the protocol (e.g. performance) and the operating system (e.g. power consumption).

The remainder of this paper is organized into four sections. The next section contains related research and design decisions. R-MTP is described in Section 3, where we present our channel abstraction and the mechanisms for bandwidth estimation, reliability, flow control and channel management. The experimental results are discussed in Section 4, and Section 0 contains the conclusions and future research directions.

2 Communication Channel Multiplexing

The aggregation of multiple communication channels has been attempted at different layers of the protocol stack. The novel aspect our multiplexing solution is that we expose multiplexing to the transport level. The base-stations or access points do not need to be modified because they are not aware that they are carrying flows that have been aggregated into one virtual channel. The hosts at the end-points are aware of the multiplexing, thereby providing an end-to-end solution and allowing the use of heterogeneous technologies with diverse attachment points. In contrast, the current solutions work at the link layer.

At the link layer, such aggregation for serial devices has been implemented in Linux, Windows and many commercial routers. For these solutions the characteristics of all communication channels must be identical. Linux, Cisco IOS and Sun also allow for the use of multiple Ethernet adaptors in the same fashion, in what is called respectively "bonding", "etherchannel" and "trunking". This type of channel aggregation must be done between the same two endpoints because this is a link layer solution, and transparent to the upper level protocols. ATM attempts to provide a variety of rates and flexibility in allocating bandwidth for commercial services through channel aggregation. The aggregation of many lower bandwidth channels into a larger pipe is called "reverse multiplexing" [1], and is now part of the ATM specification. Since ATM depends on in-order delivery for cell reassembly, much effort has been put into maintaining the strict synchronization of different flows. Similar work has also been done in the aggregation of bandwidth [2] in wireless links using the facilities of PPP (multilink) [3]. The mechanisms described in this paper are more general, allowing the use of any interface that supports IP.

The addition of multiplexing at the transport layer introduces an increase in the occurrence of out-of-order delivery of packets due to the different delays experienced on the different channels. Large differences in delay have two effects on multiplexing transport protocols. First, a packet being transmitted over a channel with long delay may appear to be lost if the loss detection methods do not consider the channel being used and the characteristics of that channel. Gap detection can be used on a per channel basis, but has no meaning across channels. Similarly, the use of duplicate acknowledgements for quick notification of lost messages is meaningless if the sending channel is not taken into account. Second, large differences in delays across channels will affect the performance of the protocol. Application-level framing [4] may free protocols from providing resequencing and allow applications to deal with out-of-order reception, enabling the faster use of incomplete data; however, most applications currently in use require that transport protocols provide in-order delivery.

3 R-MTP

R-MTP is a protocol designed for the reliable transmission of bulk data to mobile systems that have access to multiple link-layer technologies. R-MTP is designed as a multiple channel, rate-based protocol that uses selective acknowledgements[5] for reliability, and bandwidth estimation for flow and congestion control. Rate-based protocols have a long history ([6],[7],[8]). In this section, we present R-MTP's architecture and discuss the parameters used per channel and across multiple channels. We then discuss bandwidth estimation, reliability and flow control. Due to space constraints, we omit the mechanism for adding and removing channels, and the format of the packet headers (See [9]).

3.1 Protocol Architecture and Parameters

Our approach defines the architecture of an end-to-end transport layer that supports the aggregation of resources from multiple end-to-end network layer channels. The transport layer provides a multiplexing service that maps a single virtual application layer channel across these multiple network layer channels. The application submits data units to the end-to-end transport layer for transmission; the transport layer processes them and interacts with the various network channels to determine which channel to use for transmission. The transport layer on the receiving side processes the incoming data units from all channels and presents them to the application layer in a meaningful way, where meaningful is defined by the specific application requirements.

Our goal is to design the transport layer to adapt its behavior to changes in the availability of individual link-layer channels, as well as the characteristics of the specific channels to which the transport layer has access. On startup, a collection of one-way network channels is established and each of these channels is probed via the packet pair method [10] to determine available bandwidth on the channel. This technique provides the current value of the smallest interval at which packets can be sent without experiencing queueing delays. Delay, or round trip time, is measured by recording the time a frame was sent and its acknowledgment received. These measurements define the main parameters of R-MTP, which are used by its reliability, congestion control and flow control mechanisms.

For reliability and sequencing, R-MTP uses a window-based reliability mechanism with selective acknowledgements. The size of the reliability window, *window size*, defines how much space is available at the receiver for active messages, where an active message is a message that may still be accepted and buffered by the receiver. The value of *window size* depends on the bandwidth and delay estimates for all channels currently in use. Since a fixed size bitmap is used to implement the selective acknowledgements, the window size is fixed during the lifetime of the connection to maintain a constant header size. Processing is simplified if the header is word aligned, therefore, *window size* is defined as the minimum multiple of 32 that is greater than twice the sum of the bandwidth delay product of each channel. The bandwidth delay product is the number of packets in flight at each time. The reliability window must accommodate enough packets so protocol processing does not stop if a packet is lost. This requires the window to be large enough to accommodate all the packets transmitted from the time a packet was lost until a replacement arrives.

R-MTP tracks the available buffer space at the receiver with the variable *free buffer space*. Similarly to TCP, the receiver maintains a receive buffer, equal in size to the reliability window, for buffering messages that have already been processed by R-MTP, but not yet passed up to the application. The amount of free space in this buffer dictates how far the sender can slide its send window upon receipt of acknowledgements from the receiver. If there are no more free buffers, R-MTP can continue processing messages in its reliability window, but will not be able to advance its reliability window. R-MTP uses a field in its header to report the value of *free buffer space* to the other side of the protocol and maintains the parameter *maximum receiver rate* to track the maximum rate at which a receiver can process frames. If transmission is limited by the processing power of a receiver, R-MTP caps its transmission rate at *maximum receiver rate*.

R-MTP tracks the maximum bandwidth per channel both to avoid trying to send too much data across a channel and to make sure a channel is used to its capacity. Since R-MTP is a rate-based protocol, R-MTP translates this information into a *maximum rate* that is sustainable on the channel. To use all available bandwidth, R-MTP continuously probes active communication channels. Probing may cause packet loss if it is done when a channel is operating at its limit, since probing beyond the maximum available bandwidth will momentarily exceed that maximum. If the maximum channel bandwidth is known, and the *maximum rate* is set to that value, R-MTP will stop probing when the maximum bandwidth of a channel is achieved.

The ideal rate is an elusive target. It should be as close as the maximum allowed by the channel before causing congestion, but since channels are shared, the ideal rate will change over time. The interarrival time measured at startup is used to calculate the *initial rate*. The ideal rate is the inverse of the minimum period that the channel can support without causing congestion, which is given by the interarrival time measured by the packet pair method. Since this measurement may contain errors, the *initial rate* is defined as half the calculated rate. The rate will track the available bandwidth, and eventually converge to the inverse of the interarrival time or the *maximum rate*.

3.2 Bandwidth estimation

Bandwidth estimation is at the heart of the protocol. It sets the rate at which each channel sends frames, defines how information is multiplexed on the available channels, and is the major input to infer if losses are congestion related or caused by the medium ([11],[12],[13]). The protocol keeps track of the *interarrival times* and *jitter* for each frame received, and a cumulative *long run jitter*. The *interarrival time* of a frame is the difference in the arrival time of the previous frame and the current frame in this channel. If a frame is lost then it has no meaning, and it is not calculated. The *jitter* is the difference between the measured *interarrival time* and the *rate* of the channel, which is calculated at the receiver and sent in the frame header. The *long run jitter* should hover around zero, because the value of the jitter may be positive or negative, when delays in the previous frame may case the current frame to appear to arrive early (Figure 1). If the *long run jitter* starts to grow, the system is experiencing congestion, since each individual jitter is getting larger without a negative jitter to cancel out the increase.

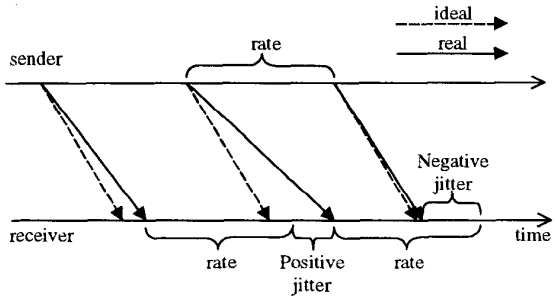

Figure 1: Negative and positive jitter values caused by the variance in propagation delays

By tracking the values of *jitter* and *long run jitter*, R-MTP infers if losses are due to congestion or due to the medium. If *long run jitter* is positive and the last *jitter* value was large when a loss was detected, then the loss is deemed a congestion loss and the protocol slows down its sending rate by changing its rate on the receiver and sending the new rate to the sender. Otherwise, the loss is considered a medium loss, and the rate is unchanged.

Using a congestion avoidance mechanism similar to WTCP's [7], R-MTP does not depend solely on packet loss to sense congestion. If two consecutive values of the *jitter* show a growing trend and *long run jitter* is positive, the protocol reacts to the incipient congestion. The growing jitters combined with a net increase in the *long run jitter* may mean that somewhere along the path a router queue is growing, increasing packet latency. In this case, the protocol slows down to avoid congestion.

Measuring *interarrival times* and *jitter* allows the protocol to sense bandwidth scarcity. It does not allow the protocol to detect bandwidth abundance. If more than enough bandwidth is available to support a certain *sending rate*, the *interarrival times* will converge to the *sending rate*, but will never surpass it. To use all available bandwidth the protocol must probe the channels periodically and adjust the rate accordingly. The method used to probe available bandwidth is the packet pair method. Two packets are sent back-to-back to probe the channel and measure the current *minimum interarrival time*. If the *minimum interarrival time* is smaller than the current period (inverse of the current *rate*), the current *rate* is increased by a fraction of the difference. If the available bandwidth is a constant, then the *rate* will converge to the optimal rate.

In general, the available bandwidth is a dynamic value, changing according to the channel usage. Measurements are made at the receiver, and fed back to the sender. Thus, the protocol will always trail changes. To minimize time under non-optimal channel usage, it would be good to probe often. Since probing disrupts channel regularity, over which congestion estimation is made, the maximum *probing rate* is once every four packets. The *probing rate* is adjusted to match channel conditions. When the protocol drops its rate because of congestion, the *probing rate* also drops. If the channel rate reaches the *maximum rate* defined for the channel, probing is disabled.

3.3 Reliability

R-MTP uses retransmission-based reliability and gap-detection for identifying losses. The sender is notified that frames have arrived at the receiver by receiving acknowledgements. There are two types of acknowledgments: cumulative and selective. The cumulative acknowledgment carries the number of the next expected frame (last received + 1). The selective acknowledgment is a bit map of the state of the received queue, with bit 0 being the position of the next expected frame. The sender keeps an individual bit map for each channel to record which frames were sent on the channel.

Gap detection works by checking the selective acknowledgement for gaps in the sequence of each channel. Every time a frame is sent through a channel, it is marked in that channel's bitmap. When the selective acknowledgement is received, it is compared with each channel bitmap. If gaps are found, the missing frames are retransmitted. Once a frame is retransmitted, the protocol can no longer rely on sequencing to detect gaps, since the retransmitted frame is out of order. Two additional global bit maps are kept to track information about retransmissions. One bitmap, *retransmitted frames*, tracks all retransmitted frames to prevent multiple retransmissions of the same packet before it has had a chance to be successfully received and acknowledged. In order to determine when a retransmission has been lost, R-MTP maintains a second bit map for *marker frames* that is used to indicate the frame that was sent immediately after a retransmission. If an acknowledgement for the *marker frame* arrives before the acknowledgement for the retransmitted frame, the retransmitted frame is deemed lost and sent again.

Marker frames are needed because even after a gap is found and a lost frame retransmitted, all acknowledgements generated before the lost frame arrives at the receiver will contain the same gap. If a frame were retransmitted every time a gap was detected, there would be many duplicated retransmissions. To prevent unnecessary retransmissions, a frame in the *retransmitted frames* bitmap can only be retransmitted again after its

marker frame in the *marker frames* bitmap has been cleared. The position in the *marker frames* bitmap is cleared when an acknowledgement or a gap for that position is found. This allows the protocol to do all its retransmissions without using explicit timers.

The reliability window must buffer enough frames to allow the acknowledgement for a frame to be received in time to prevent its retransmission. If more than one channel is being used, then the Minimum Window Size (MWS) is given by a ratio of the rates and the longest propagation delay. The MWS is defined as the number of buffers needed to accommodate all packets that are transmitted between the time a packet is sent on the channel with longest propagation delay and the time that its acknowledgement is received. The total propagation time is the one-way propagation delay of the slowest channel (packet) plus the one-way propagation delay of the fastest channel (ack). The number of packets transmitted in this interval in a channel is the total propagation time times the rate of the channel. The total of packets is the sum of the number of packets transmitted in each channel. Therefore, MWS is given by the expression below:

$$MWS = \sum_{i} \begin{array}{l} (delay\_slowest\_channel + \\ delay\_fastest\_channel) * \\ rate\_channel\ (i) \end{array}$$

3.4 Flow Control

There are two mechanisms used for flow control. One is the *maximum rate* negotiated on startup. The *maximum rate* defines a hard limit on the number of packets sent per second – even if the protocol measures that the available bandwidth is greater than it is currently using, it will not surpass the *maximum rate*. When the protocol reaches the *maximum rate* it will stop probing until the rate drops. The second mechanism is the receiver queue. This is an ancillary queue that smoothes the bunching effect that the mismatch of rates and delays generates, and is implemented as a part of the receiver window. The flow control works in the same way as TCP, but with a fixed size queue: the packet header carries how many slots are left on the receiver queue. The sender will stop sending data frames if the queue gets full, halve the rate and send only acknowledgements until it gets acks from the receiver that shows there is space again in the receiver queue, at which time normal processing resumes. The flow control and reliability queues are two separate queues, the data in the flow control queue having already been acknowledged.

4 Experimental Results

R-MTP provides a reliable transport service using bandwidth aggregation and is optimized for wireless environments. The experiments in this section are designed to demonstrate the effectiveness of R-MTP as a reliable transport protocol, and R-MTP's bandwidth aggregation. We describe our test setup, and the experiments conducted. Due to space constraints, we selected two sets of experiments. In the first set, we address issues related to the performance of R-MTP in lossless environments. The experiments compare the performance of R-MTP to the performance of TCP over various link layer technologies. The second set demonstrates the effectiveness of R-MTP's bandwidth aggregation techniques, both in lossless and lossy environments. Other experiments (found in [9]) attest that R-MTP is TCP-friendly, and analyze in greater depth its reaction to loss and congestion.

The experiments use a user-space implementation of R-MTP that encapsulates its data in UDP packets. UDP already offers a *port number* and *checksum* so these variables were omitted from R-MTP's header. For each run, the test program sends one thousand packets of 1400 bytes, using both R-MTP and TCP. The receiving program records the arrival time of each packet, and the graphs plot the cumulative time of arrival of each packet and their interarrival time. The tests cover a sample of wireless link layer technologies: three types of infrared devices and two types of wireless Ethernet. Infrared has two different standards for point-to-point communication: IRLan, a LAN emulation over infrared which is an IRDA standard, and IRNet.

4.1 Performance of a single link layer under lossless conditions

Table 1 presents a comparison of R-MTP and TCP over wireless channels with no external losses. In general, we expected TCP to perform better than R-MTP since R-MTP is implemented in user space. Surprisingly, R-MTP performs better than TCP even when no external losses are introduced. On slower network interfaces, a node's CPU can feed data to the network interface faster than the data can be transmitted on the link. This allows a single transport layer connection to use all available bandwidth of this link. Under this condition, TCP's bandwidth probing can cause losses by congesting the link. This results in a net performance below R-MTP's, whose

bandwidth probing rarely causes losses in these experiments.

Link Layer Technology	Total Run Time R-MTP (sec)	Total Run Time TCP (sec)
IRLan	159.5 ± 3	160.9 ±2
IRNet	156.0 ± 4	160.0 ± 3
Wavelan	10.69 ± 0.7	9.89 ± 0.4
Aviator	8.64 ± 0.5	9.03 ± 0.6

Table 1: Comparison of R-MTP and TCP over various wireless technologies

4.2 Bandwidth Aggregation

Dealing with losses provides a good argument for using multiple link layers simultaneously, even when bandwidth aggregation does not seem compelling. Bandwidth aggregation may significantly increase the total bandwidth available to a mobile host if the link layers have similar capacities; (i.e. aggregating two 115Kbps channels is a good choice). On the other hand, it is not intuitive that aggregating a 1.2 Mbps and an 115Kbps channel will add enough bandwidth to offset the cost of multiplexing. The result of both bandwidth aggregations can be seen in Figure 2 and Figure 3 below. Figure 2 shows the aggregation of two infrared channels at 115Kbps, an IRLAN channel and an IRNET channel. The aggregate channel is considerably faster than each individual channel. This is not the case for the example shown in Figure 3, which shows that the channel resulting from the aggregation of an Aviator wireless Ethernet and an infrared IRLAN connection is only slightly faster than the individual Ethernet channel. In fact, the multiplexing overhead makes the composite channel slower at some points.

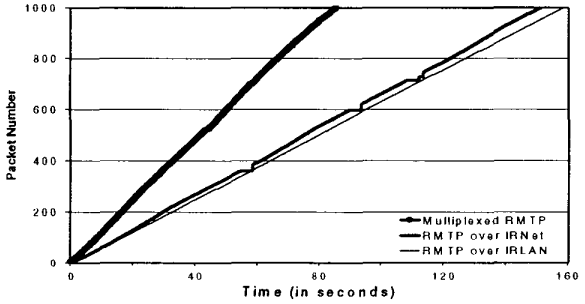

Figure 2: Bandwidth aggregation on IRNet and IRLAN

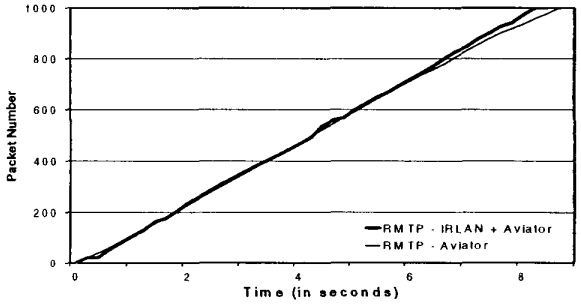

Figure 3: Bandwidth aggregation on Aviator and IRLAN

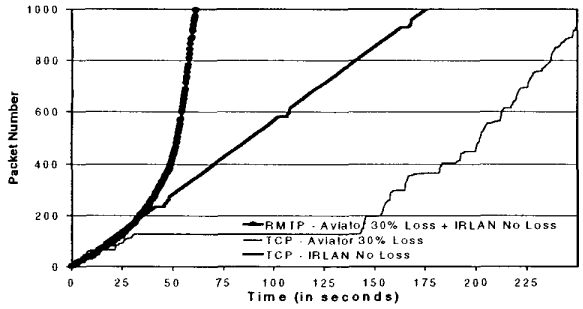

Figure 4: Comparison of TCP over Aviator with 30% loss and over IRLAN with no loss to R-MTP multiplexed over the same two channels

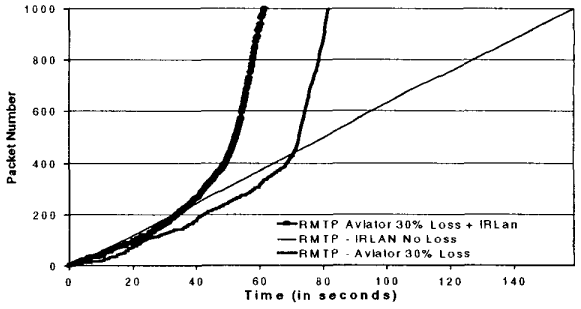

Figure 5: Comparison of R-MTP over Aviator with 30% loss and over IRLAN with no loss to R-MTP multiplexed over the same two channels

Bandwidth aggregation with such mismatching channel characteristics can be justified in certain scenarios. Figure 4 shows an experiment with the same infrared interface and wireless Ethernet of Figure 3, but with a 30% loss rate on the Ethernet. A TCP run, which normally takes 10 seconds to complete on the wireless Ethernet link, now takes 25 times longer. Now the IRLAN link, lossless in this example, is performing better than the Ethernet link, and completes the run in 160 seconds. Figure 5 shows R-MTP using Ethernet, IRLAN and both multiplexed. While R-MTP has better behavior on the lossy link, finishing the run on the wireless Ethernet alone in 81.5 seconds, adding the much lower bandwidth link actually improves the performance by 25% in this case (61.2 seconds).

Beyond the significant improvement in performance by using two seemingly mismatched interfaces, these results make a case for using all available interfaces all the time. While this must be offset by power and cost considerations, because it is impossible to measure a priory if an interface is lossy, or to predict noise, using all communication resources that are available creates a virtual channel that is more resilient than any channel alone.

5 Conclusion and Future Work

Mobile nodes using wireless communication operate in less than ideal environments. Low bandwidth links limit application throughput, lossy channels challenge transport protocols optimized for congestion-based losses, and constantly changing channel availability causes breaks in transmission. The solutions presented in this paper address these problems through the use of end-to-end channel aggregation. With such techniques, additional bandwidth can be added to the mobile host's bottleneck link. The use of multiple channels can also alleviate the effect of losses or complete outages on one particular channel.

Our experiments demonstrate the performance of R-MTP in various wireless environments. The success of bandwidth aggregation across similar links supports our intuition that adding bandwidth to the bottleneck link increases throughput for the application. Most importantly, we show that aggregation across two channels with an order of magnitude difference in capacity is beneficial if the high bandwidth channel is experiencing significant loss.

Two very important issues are part of our future work. We understand the use of multiple interfaces will have a large impact on the energy consumption of a mobile node, and we are currently investigating the effect of using R-MTP on energy consumption. Our intuition is that the use of multiple interfaces can minimize transfer time and so conserve the total energy used by a mobile (system+network) in some scenarios. The second is related to mobility. Additional testing must be done to gauge R-MTP's success in mobile environments. Comparison with a solution using Mobile IP and TCP, and the changes proposed in TCP to make it mobility-aware [14], can demonstrate the effect of R-MTP's transport layer mobility solution.

6 REFERENCES

1. Chiussi, F.M., D.A. Khotimsky, and S. Krishnan. *Generalized inverse multiplexing of switched ATM connections.* in *Proceedings of the IEEE Conference on Global Communications (GlobeCom '98).* 1998.
2. Snoeren, A.C. *Inverse Multiplexing for Wide-Area Wireless Networks.* in *Proceedings of the IEEE Conference on Global Communications (GlobeCom '99), Global Internet Symposium.* 1999.
3. Sklower, K., et al., *The PPP Multilink Protocol, RFC1990.* 1996.
4. Clark, D.D. and D.L. Tennenhouse, *Architectural Considerations for a New Generation of Protocols,* in *Proceedings of the SIGCOMM '90 Symposium.* 1990. p. 200-208.
5. Mathis, M., et al., *TCP Selective Acknowledgement Options.* 1996, Internet Engineering Task Force.
6. Clark, D.D., M. Lambert, and L. Zhang, *NETBLT: A High Throughput Transport Protocol.* 1988.
7. Sinha, P., et al. *WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks.* in *ACM Mobicom '99.* 1999.
8. Rejaie, R., M. Handley, and D. Estrin. *RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet.* in *IEEE Infocom 99.* 1999.
9. Magalhaes, L., *R-MTP: Reliable Multiplexing Transport Protocol.* 2001, University of Illinois at Urbana-Champaign.
10. Keshav, S. *A Control-Theoretic Approach to Flow Control.* in *Proceedings of the SIGCOMM '92 Symposium.* 1992.
11. Balakrishnan, H., et al., *Improving TCP/IP Performance over WIreless Networks,* in *First ACM International Conference on Mobile Computing and Networking (MOBICOM).* 1995.
12. Biaz, S. and N. Vaidya, *Discriminating congestion losses from wireless losses using inter-arrival times at the receiver.* 1998, Technical Report 98-014, CS Dept., Texas A&M University.
13. Caceres, R. and L. Iftode, *Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments.* IEEE Journal on Selected Areas in Communications, 1995. **13**(5).
14. Snoeren, A. and H. Balakrishnan. *An End-to-End Approach to Host Mobility.* in *ACM Mobicom '99.* 2000.