MMTP - Multimedia Multiplexing Transport Protocol

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

Multimedia data has special requirements that are hard to be met on mobile hosts due to potentially low bandwidth and disruptions due to host mobility. Such limited communication capabilities of mobile hosts can be offset by the simultaneous use of multiple link layer technologies. MMTP is a member of a suite of protocols that share the novel characteristic of aggregating bandwidth from multiple link-layer channels. The use of multiple channels to transport user data provides five key benefits: (1) a fatter pipe,(2) a fast feedback path, (3) the retransmission of selected lost messages, without delaying the playout of the data stream, (4) less sensitivity to minor bandwidth fluctuations on any one individual channel, and (5) smooth vertical handoffs for active data streams.

MMTP is a rate-based protocol designed for transferring multimedia data on mobile systems, and makes simultaneous use of every communication channel available to send data at the required rate. Transmission in MMTP is governed by two mechanisms. The first is a set of rate control protocols associated with each outgoing channel. The second is a scheduling algorithm that places incoming packets on the appropriate channel. MMTP is link-layer aware protocol that uses bandwidth estimation for congestion control, and relays to the application information needed for rate adaptation. In this paper, we show that the quality of data transmission can be improved through the use of MMTP through experimental comparisons with data transmitted via UDP. We also demonstrate the economy of bandwidth: MMTP only sends packets that it estimates will arrive within the packet deadline, thus decreasing the number of late packets that will be discarded at the receiver.

Keywords

Wireless communication, multimedia transport protocols, low bandwidth link.

1. INTRODUCTION

As mobile devices become more prevalent, the demand for anytime/anywhere connectivity for those devices increases, and the requirements on that connectivity become more stringent. Connectivity for mobile devices is governed by the communication technology on the device and the coverage area of that technology. To add versatility to the mobile devices, they are normally built with multiple communication technologies and the capacity for expansion. Laptops commonly come with built in infrared and with multiple PCMCIA slots, allowing for multiple communication cards. With the growing demand for mobility support, coverage areas for wireless connectivity are growing and overlapping. Due to different administrative authorities and different underlying link layer technologies, the current infrastructures only support coordination and cooperation between homogenous support stations (“horizontal handoffs”). In order to support better handoffs, communication quality and cost optimization, the mobile node’s operating system can coordinate the simultaneous use of multiple diverse communication channels, providing seamless connectivity as the mobile node migrates between coverage areas (“vertical handoffs”) [1]. If the mobile node has access to multiple communication channels, the application can be guided as to which channel is currently the most appropriate [2]. Such support is limited to the use of one communication channel per application data stream.

The goal of our research is to aggregate the bandwidth available from multiple channels to create a virtual channel with more bandwidth than each channel alone. Given that the bottleneck for most wireless communication is the last hop to the base station, the addition of bandwidth at this last hop will alleviate some of the resources constraints on the mobile node. Our approach exposes link-layer connectivity and resource information to the transport layer, allowing the transport protocol to adapt to both changes in available bandwidth on each channel and changes in availability of channels. Our solution preserves end-to-end semantics, transparently providing the application with the simultaneous use of multiple channels.

We are currently designing a framework for the aggregation of multiple communication channels, enabling transparent use for multiple and individual data streams. A main component of this framework is a protocol suite that provides diverse transport layer protocols able to operate in the context of multiple communication channels. In this paper, we present a protocol from this suite, MMTP, Multimedia Multiplexing Transport Protocol, which supports the transmission of time sensitive rate-based data streams (e.g. audio, video) that may be generated live or from stored data. MMTP is a rate-based protocol that uses bandwidth and delay estimation for both determining the available bandwidth and for congestion control. Maintenance of these estimations provides natural support for adaptive multimedia applications. Given the characteristics of the multimedia data stream in terms of frame rate and bandwidth requirements, MMTP uses any available communication channels to transmit the data. As the available channel resources change, MMTP adapts, changing the fraction of flow that is being sent on each channel and adding or removing channels as necessary. MMTP provides a best effort service. If the aggregation of available channels does not provide enough bandwidth for the application stream, MMTP will drop packets that it estimates cannot arrive on time and inform the application of the lack of necessary resources.

The use of MMTP provides five key benefits. First and foremost, there is the benefit of a fatter pipe, which enables better quality multimedia traffic. Second, MMTP provides a fast feedback path. Although the delay for data transmission will depend on the channel with the longest propagation delay, control feedback can be returned on the channel with the shortest propagation delay. Third, extra bandwidth on any of the channels may enable the retransmission of select lost messages, without delaying the playout of the data stream. Forth, due to the use of multiple channels, MMTP is less sensitive to minor bandwidth fluctuations on any one individual channel. Finally, smooth vertical handoffs for active data streams are a natural benefit of using multiple channels and of hiding link-layer connectivity.

This paper is organized into five additional sections. Section 2 presents relevant research associated with the current work in communication support for mobile computing. A real life scenario of multimedia in mobile environments is investigated in Section 3. MMTP is presented on Section 4, where we explore its startup behavior, flow and congestion control. In Section 5, the experimental results are presented, and Section 6 contains conclusions and future research directions.

2. BACKGROUND AND RELATED RESEARCH

The current Internet infrastructure was not designed with the needs of multimedia traffic in mind. The pervading best effort delivery protocol that forms the base of all Internet traffic, IP, has no built in mechanism for reservation of bandwidth or for periodic traffic. The protocols that were developed later to allow the use of the IP infrastructure for multimedia traffic do not consider mobility. Adapting the solutions used on wired hosts to mobile systems is not straightforward, because the characteristics of wireless communication channels are even less agreeable to multimedia traffic. In addition, normal reservation schemes [3] used for wired hosts will not work or may become very expensive due to changes in the location of a mobile host. However, varying the quality of the source stream to match the available bandwidth and loss rate has been successfully adapted to the mobile environment, although it falls to the application to keep track of the available bandwidth and other parameters necessary for the adaptation.

Due the periodic nature of multimedia traffic, it is commonly accepted that the best protocols for such data streams use rate-based mechanisms. Moreover, the lossy nature of wireless communication channels makes channel losses a poor congestion indicator. In response, bandwidth estimation in conjunction with congestion avoidance has been suggested for use with wireless rate-based protocols. One of the earliest examples of a reliable rate-based protocol is NETBLT [4], which was designed for the transport of bulk data and is not suitable for multimedia traffic. Recent examples of other reliable rate-based protocols are WTCP and RAP. WTCP [5] is a reliable split connection protocol that has good performance over lossy low bandwidth links that have high latency. RAP [6] is a TCP-friendly rate-based protocol for realtime streams.

The aggregation of the bandwidth from two modems has been implemented in both Linux and Windows. In both systems, the characteristics of both channels must be the same and only a simple load-balancing algorithm is used for scheduling transmission. The aggregation of many lower bandwidth channels in a larger pipe is called “reverse multiplexing” in ATM [7], and is now part of the ATM specification, as it allows a multiplicity of rates and flexibility in allocating bandwidth for commercial services. Some work has also been done in the aggregation of bandwidth [8] in wireless links by using the facilities of PPP (multi header extensions [9]. The mechanisms in MMTP are more general, working with heterogeneous interfaces. By uncoupling the transport protocol from the network protocol, transitions from one network to another are very natural in MMTP, and require no switching. The Barwan project presented the concept of “vertical handoffs”[1], transitions from one link layer to another. WTCP uses a similar model, when the mobile transitions from one area of coverage to another, there is a handoff and the older connection is relinquished. In MMTP, if an area is connectivity rich and multiple ways to access the infrastructure are available, the activation of a new interface does not cause another to be dropped. The new interface is added to the existing pool.

Adapting the bandwidth requirements of a multimedia stream to the available bandwidth of the channel has been proposed in [10]. Because this requires a close interaction between the transport protocol and the coding application, there are many proposals for integrating source coding and the transport protocol. [11] proposes a transcoding the source into a non-prioritized packet stream to ensure graceful degradation in the presence of packet loss, and describes a TCP-friendly rate based protocol and the framework for the interaction of the protocol and transcoder. [12] proposes modifications to the TPC protocol, a resilient encoder and a rate control algorithm for the same objective. While we do not delve into source coding, MMTP exposes rate changes to the application, enabling adaptation.

MMTP can be viewed as two one-way protocols, one from sender to receiver carrying data, and another from the receiver to the sender, carrying control information. RTP [13] uses different streams for data and control information, while MMTP carries control information inside the data packets, allowing for changes in the rate for presentation be communicated to the application simultaneously with the receipt of data.

Because MMTP does not back-off on lost packets, mechanisms such as RED [14] will not affect the sending rate. Although we believe that our congestion control method results in fair resource utilization, the work presented on [15] on the differentiating multimedia traffic on router to avoid congestion can be used to police MMTP traffic.

A new approach to mobility is to make mobility visible to the endpoint. While Mobile IP tries to hide host mobility by using proxies, mobility can also be achieved by letting transport protocols take care of the switch. This requires the uncoupling of the network address from the connection identifier, and a scheme for location. TCP and DNS were modified to accomplish that in [16]. The same end-to-end arguments apply to MMTP, but MMTP relies on a proxy for location. MMTP has the additional advantage in the case of multimedia traffic because it generally does not need nor accepts the added overhead of TCP reliability.

3. MULTIMEDIA IN MOBILE ENVIRONMENTS

Multimedia traffic is very sensitive to delay, jitter and bandwidth restrictions. Introducing wireless links into the path of a multimedia data stream not only increases the potential for such problems, but also brings with it problems from handoffs. In an effort to offset these negative effects, a mobile host may have access to multiple communication technologies. With the growing pervasiveness of the wireless infrastructure, in many places there will be an overlap of coverage. This presents an opportunity to tap additional resources to help the transmission of multimedia traffic. At the same time, mobility-aware protocols may account for communication artifacts created by movement, such as variations in bandwidth and changes in channel availability.

Consider the following scenario: a user is waiting for a train and wants to watch the news. The train station terminal has both short-range radio and infrared connectivity. The user has a cellular modem in addition to the infrared and short-range ratio interfaces on the palmtop. Currently, the user would have to choose which interface to use to access his/her favorite news source. Even if the user chose the interface with the best qualities, any fluctuations in service characteristics would directly impact reception. When the user boards the train, the local connectivity (short-range radio and infrared) becomes unavailable. If the user had chosen any of the local connections, there would be a gap in the transmission as the connection was being handed off to the cellular modem. The solutions presented in this paper improve this scenario in two ways: better channel quality and seamless handoffs.

The improvement of quality comes with the use of all available channels. This not only aggregates bandwidth, permitting a better quality in the multimedia data stream, but also smoothes out variations in a single channel that would impact data presentation by spreading consecutive packets in different media. We enable the simultaneous use of multiple channels by exposing the underlying link layer to the transport protocol. By creating a transport protocol that is mobility-aware, and knows the existence of multiple interfaces, we can offer the illusion of a single “fatter” pipe with better transmission qualities to the multimedia application. This aggregated channel has interesting qualities. Although the individual channel with longest propagation delay dominates the propagation delay for data, as will be shown in the next section, the control data may take advantage of the channel with lowest propagation delay, helping adaptation.

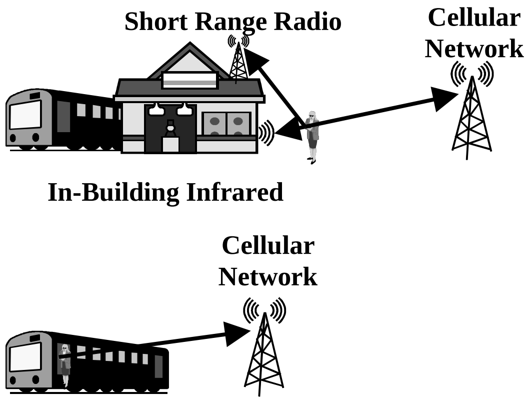

Figure 1. Wireless Mobility

The use of overlapping communication channels allows for smooth handoffs as the mobile node migrates between coverage areas. If the user moves slowly out from an area covered by a certain communication means, the degradation in quality is perceived by the protocol, and the channel is slowly phased out, with no interruptions. When entering an area covered by a new link layer, the protocol adds the new channel to its processing, with the entailing gains in quality.

4. MMTP

MMTP supports the transmission of time sensitive rate-based data streams that may be generated live or from stored data. Given the characteristics of the data streams in terms of frame rate and bandwidth requirements, MMTP uses any available communication channels to transmit the data. As the available channel resources change, MMTP adapts, adding or removing channels as necessary. MMTP provides a best effort service. If the aggregation of available channels does not provide enough bandwidth for the application stream, MMTP will drop packets that it estimates cannot arrive on time and inform the application of the lack of necessary resources.

MMTP is a rate-based protocol that multiplexes data packets across multiple communication channels. The main task of MMTP is the decision as to which channel to use for transmitting the current packet. This decision is based on estimations of the bandwidth and delay characteristics of each channel. After startup, two control mechanisms are used to adapt the sending rate to the channel bandwidth: rate decrease messages and channel probe. In this section, we present the protocol parameters and describe the operation of MMTP.

4.1 Communication Framework

MMTP was designed in the context of a communication framework that provides support for the simultaneous use of multiple link-layer communication technologies. Relevant to this protocol, the framework provides the following functionality. First, the framework monitors for the availability of communication channels for each communication interface on the mobile computer. Dynamic probing techniques are used to find new channels and maintain information about existing channels. Second, the framework probes idle available channels for information about channel parameters such as available bandwidth and propagation delay. Information about active channels is collected from the protocols using them. The framework continues monitoring dynamically, providing information about available and active interfaces to interested protocol such as MMTP. This communication framework is currently being designed in conjunction with the protocols that will be using it. The details of the framework are beyond the scope of this paper.

4.2 Data Characteristics

Multimedia data streams can be generated on the fly at the rate at which they need to be played out or retrieved for storage. Applications using on-the-fly streams often have very little tolerance for delay, while applications using stored media may be more tolerant. In addition, for the former, frames are only accessible for transmission as they are generated, while in the later, all frame are accessible at the same time. Frames from multimedia data streams often exceed the maximum transmission size and must be fragmented into multiple packets. Intelligent fragmentation can be done that enables the reception and processing of pieces of the frame, even if the entire frame does not arrive, supporting the concept of application level framing [17]. In the rest of this paper, we discuss MMTP in the context of on-the-fly data with one packet per frame.

4.3 Startup Behavior

On startup, the application defines the requested *frame rate*. The protocol queries the communication framework to learn the number of available channels, and an estimate of the *propagation delay* and *packet rate* for each. To see if the required *frame rate* can be met, the protocol calculates the available aggregated channel rate. There are two cases:

1. ***Frame rate* > ∑ packet rate(i)**
   - the application is notified that packets will be dropped. The application may decide to abort the transmission, to change the *frame rate* or just continue.
2. ***Frame rate* < ∑ packet rate(i)**
   - the estimation indicate that there is enough bandwidth for the transmission.

Initial delay is a playout parameter that tells the receiving application how long to wait before playing the first frame. *Initial delay* can be chosen between the *maximum initial delay* given by the application and the *minimum initial delay* calculated by the protocol given the *propagation delays* of active channels. The larger the *initial delay*, the more slack the protocol has to compensate for jitter, because it adds time to the deadline of each frame. The receiving application will buffer a number of frames proportional to the ratio between *initial delay* and *frame period* (the inverse of *frame delay*) before initiating playout.

Initial delay is always greater or equal the longest propagation delay for the channels being used and is not dependent on which channel is used to send the first packet. The examples in Figure 1 depict the startup behavior assuming two channels and one packet per frame. Figure 1a shows *initial delay* when the first packet is sent on the channel with longest propagation delay, and Figure 1b shows *initial delay* when the first packet is sent on the other channel. In the both examples, the sender receives frame 0 at time *t* and frame 1 at time *t + 1/frameRate*. In Figure 1a the first frame was sent on the channel with longest propagation delay, so playout can start as soon as frame 0 is received, since we know that frame 1 will have arrived at the end of frame 0 play period. If propagation delays of the two channels are very different, it is possible that frame 1 will arrive before frame 0. In this situation, we buffer frame 1 and still begin playout upon receipt of frame 0. In Figure 1b, frame 0 arrives first, but playout must be delayed until we are within one play period of the receipt of frame 1, or else there may be a gap at the end of the playout of frame 0. Because MMTP uses estimation for both *available bandwidth* and *propagation delay*, it is difficult to implement option 2, since *initial delay* depends directly on the estimated value for *propagation delay* in both channels. Sending the first packet on the channel with longest propagation delay allows the playout to be self-clocked

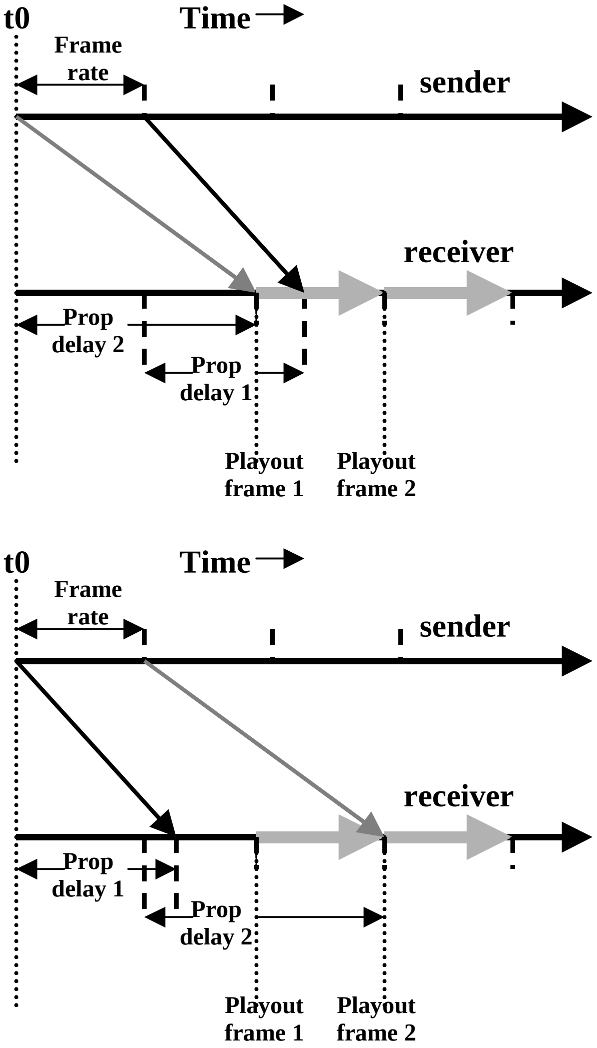

Figure 2. Startup Behavior for MMTP with Two Channels

4.4 Flow Control

Flow control for MMTP can be modeled as a set of rate control protocols, one for each channel, all servicing a single queue.

Tokens are generated for each channel based on estimates of *available bandwidth* and tagged with estimates of *propagation delay*. Packets are inserted into the queue at the frame rate of the source and need to be transmitted on one of the available channels. This type of resource management problem closely models the scheduling of processes in real-time multiprocessor operating systems [18], where processing time is mapped to transmission time for each channel. Packets are transmitted on a channel if there is a token and if the *propagation delay* ensures that the frame will arrive on time. Tokens are used for each transmission on a channel and new tokens are not generated until old tokens are used.

When a frame arrives, there are three possibilities:

1. No tokens are available: the frame has to wait until a token is generated on a channel that will deliver it on time. If no token is generated before the frame's deadline expires, the frame is discarded.
2. Exactly one token is available: if the corresponding channel can deliver the frame on time, the frame is sent. Otherwise, this case reduces to case 1.
3. Multiple tokens are available: if more than one channel can deliver the frame, the channel with longest propagation delay is chosen. Maintaining the channel with longest propagation delay filled helps creating a fast response path. If no channels can satisfy the frame's deadline, the frame waits as in case 1.

Even if none of the available channels can currently send a frame, queued frames may still be transmitted if a new channel with smaller propagation delay is added or a large decrease in the expected propagation delay of one of the current channels enables successful packet transmission.

4.5 Congestion Control

In MMTP, congestion control is implemented as a reactive technique based on bandwidth estimation. Both the *propagation delay* and *available bandwidth* in a channel will vary due to routing changes and due to interference caused by other traffic. The *available bandwidth* is equal to the difference between the maximum bandwidth of each link and the usage of each link. *Propagation delay* is composed of two parts: the actual propagation delay of the bits in the transmission lines and plus the time spent during processing in each router on the path. The first part is fixed in the absence of route changes, but the second will grow according to the size of the queues in the intervening routers.

To avoid congestion, the protocol tries to keep the requested bandwidth below *available bandwidth* in a channel. This is done by measuring *available bandwidth* and changing the rate packets are sent to a compatible value. *Available bandwidth* is inferred by measuring the inter-arrival times of packets at the receiver, and feeding the measurements back to the sender. Because packets are being sent at regular intervals in each channel, the inter-arrival time should converge to the period frames are being sent on that channel if sufficient bandwidth is available. If the inter-arrival times start to grow, somewhere in the path a router is running above capacity, and queuing packets.

4.5.1 Parameter Estimation

To estimate the basic parameters, *available bandwidth* and *propagation delay*, the inter-arrival time of packets is tracked at the receiver for each channel. Inter-arrival time is compared to the period frames are being sent on that channel (present on each packet header). With stable queueing delays, inter-arrival time should converge to this period. As queueing delays change, jitter is introduced and inter-arrival time will vary. A running total for jitter should be zero if there are no changes in channel characteristics, stabilizing the average inter-arrival time. A packet that arrives late causes the next packet to appear to arrive earlier, so the sum of the jitters should cancel. The accumulated jitter is a moving average of the last n packets, where n is an implementation parameter. The size of n should be such that it has the same temporal granularity of the keep-alive messages, so those can carry meaningful data back to the sender.

There are two special cases when running total for jitter will not be zero:

- When there is incipient congestion, the queue sizes on the routers grow, and this is seen at the receiver as a positive increase in the accumulated jitter. In this case, the receiver sends back a message asking for a decrease in the sending rate, so the transmission will not cause congestion. This is actually a better mechanism than decreasing bandwidth usage upon dropped packets. Using dropped packets as a measure of congestion is a reactive technique, used when congestion is already present. By using the accumulated jitter to signal approaching congestion, the protocol can try to prevent packets from being dropped. Moreover, dropped packets are not a good measure of congestion for mobile hosts, where wireless links have much higher loss rates than normal wired lines.
- When propagation delay drops, there will be a drop in the inter-arrival time in one time-slot. After that, the perceived rate will again converge to the sending rate. To detect that this drop is not an artifact cause by a previous packet that was very late, the jitter history is analyzed. If the pattern shows a large positive value followed by a large negative value (which would cancel out), then this event is ignored. If the trend is stable, and the early packet is not an artifact, then the next keep alive message will carry an indication that extra bandwidth may be available on that channel.

The measurement of inter-arrival times works well to prevent congestion, but it does not work to measure excess or idle bandwidth in a channel. The problem is that at every sending rate, the maximum bandwidth measured by the application is equal to the bandwidth being pumped at the sender. Therefore, we can decrease the sending rate, but not increase it with this method. Another mechanism is needed to measure available bandwidth above what is being used.

4.5.2 Probing

The mechanism MMTP uses to measure an increase in available bandwidth is probing. There are two easy ways to implement probing. The first one is additive increase: to continually increase the send rate by small amounts in the absence of rate decrease requests, and keep normal inter-arrival time measurements. If the inter-arrival times start to grow, that means we overshot or we are experiencing congestion, and the protocol has to throttle back the rate. The problem with this approach is that the virtual bandwidth gains are not tested until needed, as the packet rate is bound by the frame rate of the multimedia stream. In addition, when needed the virtual bandwidth measured by the rate increases may not be there. The second is to use probe-packets. A probe packet is sent back to back with a normal packet. The inter-arrival time of the packet and the probe should be zero, as they were sent back-to-back. However, normally this value will not be zero, but instead measure the queuing delay inserted by the routers.

Probe packets should carry useful payload, as they will take place of a normal packet. The problem is that there may not be data available to send two packets back-to-back. In this case, an empty packet can be sent. This type of probing can be harmful if the system is working at the limit: the bandwidth lost to the probe packet may cause a useful packet to be dropped.

4.6 Adding and Removing Channels

The initial list of available channels is received from the communication framework when the protocol starts. As time goes by, new channels may become available, and old channels may be lost. The communication framework notifies MMTP of those events, and gives ancillary information as estimates of *available bandwidth*.

When a new channel is added to protocol processing, the *available bandwidth* has to be measured if this data is not present. To do the initial probing, the protocol used the packet pair method [19], the same mechanism as the normal protocol probing: two messages are sent back to back and their inter-arrival time is measured at the receiver. This is used as the estimation of maximum bandwidth on that link, and fed back to the sender. When a channel is lost, a de-registration message is sent to the peer using the channel with lowest delay. This takes out that interface from the protocol processing. The message also contains the last packet received on that channel. Outstanding packets are put on the queue for retransmission, with the same constraints of the other packets (they will not be sent if they cannot meet the deadline.)

Every active channel sends keep-alive messages periodically. Besides notifying the peer that the channel is still open, these messages carry updates of the measures performed at the receiver, the inter-arrival times of the messages, the estimated *available bandwidth*, the number of packets late and lost. If a keep-alive is not received, the sender sends a query message to assure that the channel is still open or if a control message was lost.

A channel may be present in processing but not carry any data if the channel delay is greater than the slack on the channel. This may happen if transmission started before the channel was added to the processing. If the delay on the new line is greater than D0 (the initial delay), the channel can never be used to carry data for that transmission. It can still carry control messages, but it will only be used if no other channels are available. Normal keep-alive messages and probing are done in the channel, in case the delay drops to a usable level. A channel may also be phased out if the delay or bandwidth drops to very low levels. This may happen as the user moves out of the coverage area and the link layer conditions deteriorate, or by congestion. While no de-registration message arrives, keep-alive and probing will continue. If only one channel is available and keep-alive messages are not being received, the protocol will signal that the link may be broken.

5. EXPERIMENTAL RESULTS

We built our experiments to show two important aspects of MMTP. The first is bandwidth aggregation: we show that we can send better quality streams if we use more than one channel. This was accomplished by measuring the number of packets that arrived on time. The second aspect of MMTP is the economy of bandwidth. MMTP will only send packets that it estimates will arrive within the packet deadline. This way no resource is wasted. This is shown by the goodput, or ratio of packets that arrived on time by the total number of packets sent. This was shown indirectly by measuring the number of packets that arrived late.

5.1 Experimental Testbed

Our test setup consists of two Sony laptops, a PCG-505TR and a PCG-F450, both running RedHat 6.2 linux with 2.2.15 kernels, patched for infrared, and with PCMCIA package version 3.1.20. The first is connected to the network using a 3Com 574 PCMCIA adapter at 10Mbps. The second has two connections, an IRLan connection to an HP NetBeamIR using an Actisys 2000+ dongle on the serial port, and a Rangelan2 PCMCIA adapter, as shown on Figure 3. The first laptop is running the proxy software and the mobile the client side.

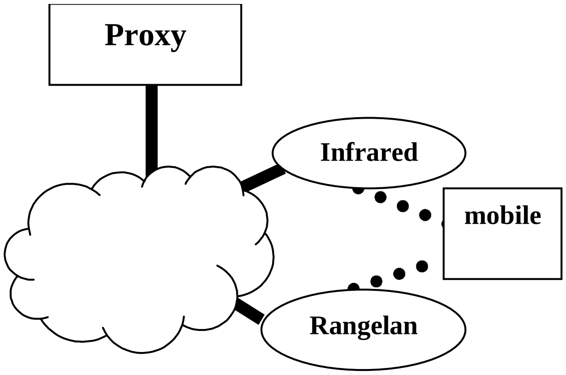

Figure 3. Test Setup

Infrared and Rangelan have very different characteristics. Infrared offers a point-to-point reliable link layer, with link speeds up to 4Mb/s. The use of a serial dongle limited the speed to 115Kbps, and the protocol overhead further limited throughput to 9KBps. Rangelan is an old radio technology, the radio link is subject to burst errors, and throughput varies with medium usage. The best throughput measured was on the order of 36KBps. The major characteristics are given in Table 1.

Table 1. Characteristics of the Wireless Networks

Delay for packets of 1400 bytes	Minimum (msec)	Average (msec)	Throughput (KB/sec)
Infrared	315	465	9.2
Rangelan	176	291	36.4

We assume that there is a source generating frames of 1400 bytes with a certain periodicity. We tested our protocol against a naïve program that sends frames using UDP as they are being generated. The results for the test program follow.

5.2 Rangelan

These are typical results since there are small fluctuations due to traffic and errors on different runs. For each run, the source program generated 1000 frames of 1400 bytes each. A frame was generated every x microseconds, as shown on the table. Frames were deemed late if they arrived more than one second after they were generated. As an example, if our cutoff value was 6 seconds, all frames on the run with period equals 35000 microseconds would have arrived on time, as this is the largest value for the delay, and no frames were lost. Of course, this is a limited run, and for unbounded media if there is not enough bandwidth the delay would keep on growing, making frames late. Frames are normally lost in bursts – the program recorded both the numbers of bursts (blocks of lost frames) and the total number of frames that were lost.

Table 2. UDP Flooding on Rangelan

Periodicity (in msec)	Late frames (delayed more than one second)	Lost frames/ blocks	Frames that arrived within one second	Total frames that arrived
10000	10	684/35	306	316
15000	582	352/37	66	648
20000	6	350/38	644	650
25000	753	68/10	179	932
30000	0	101/15	899	899
35000	818	0/0	184	1000
40000	18	20/3	962	980
50000	1	21/3	978	979
100000	0	17/1	983	983

It is interesting to note that the number of “good” frames seesaws as the periodicity varies (as seen on chart 1). For a multimedia flow, late frames are useless, so the important number is given on the fourth column – even though more frames may have arrived, if they are late they are wasted. A frame that arrived late also used bandwidth and time – making other frames that follow it late. Ideally, dropping frames should happen at the source. Another way of using more frames is changing the allowed delay, by buffering frames so the deadline of each frame is postponed. In some runs, the UDP frames flooded the wireless link, and these frames were dropped, allowing more frames to arrive in time.

5.3 Infrared

Because infrared offers a reliable link, all dropped frames are the result of UDP flooding. The maximum delay on infrared is smaller than on Rangelan, due probably to a smaller buffer on the base-station. If the cutoff were 3.5 seconds, all frames that arrived would be on time. The arrival of the frames was a monotonous increasing function, the first frames arriving on time until the 1-second cutoff was exceeded, and all subsequent frames were late.

Table 3. UDP Flooding on Infrared

Periodicity in (in msec)	Late frames (delayed more than one second)	Lost frames/ blocks	Frames that arrived within one second	Total frames that arrived
10000	84	909/64	7	91
50000	346	646/322	8	354
100000	671	312/309	17	688
150000	309	0/0	691	1000
160000	0	0/0	1000	1000

As soon as there is enough bandwidth to carry the traffic, both losses and late frames drop to zero. The intervals are shown are not the same as Rangelan due to the differences in bandwidth and delay.

5.4 MMTP

The test setup for MMTP has a source process that creates frames at the given periodicity and sends them to the proxy. The proxy sends these frames to the receiver on the mobile. The client process does not record the same information on lost frames, so the third columns of the tables are not comparable. Most of the frames shown on the third column were discarded at the source and not lost in the network. Changing the cutoff time would change the number of sent frames, as the protocol would assume that more frames could meet their deadline.

Table 2. MMTP on Infrared and Rangelan

Periodicity in (in msec)	Late frames (delayed more than one second)	Lost/ discarded frames/ blocks*	Frames that arrived within one second	Total frames that arrived
10000	0	761/21	178	178
15000	25	562/17	413	438
20000	0	548/27	452	452
25000	73	269/3	648	731
30000	10	86/10	904	914
35000	0	14/5	986	986

The current version of MMTP is still in the development stage, so the performance is below the theoretical maximum. It is too conservative on the low range, dropping too many packets, and seems to perform worse than the naïve protocol at certain periodicities. However, when comparing the performance of MMTP and simple UDP flooding, it must be pointed out that flooding is actually wasting resources both on the landline and on the wireless link. The high-speed links of the University infrastructure mask this effect, and allow flooding to course through the network until packets are dropped on the wireless base-stations. This would not happen on the Internet at large. MMTP drops the packets at the source, so only packets that are expected to arrive on time are actually sent. Chart 2 below shows the number of wasted frames. Those frames were received after their deadline had expired.

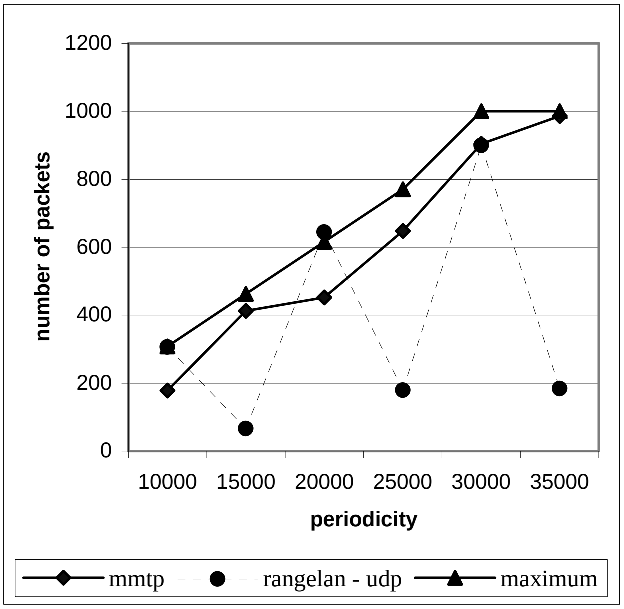

Figure 4. Comparison of UDP, MMTP and Theoretical Maximum for Packets versus Periodicity

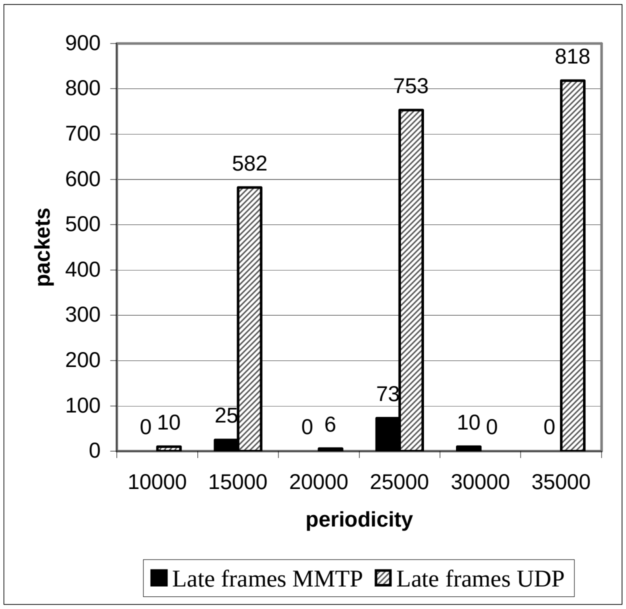

Figure 5. Packets Received after their Deadline

Other than losses caused by errors on media, when the periodicity hits 35000 microseconds the joint channel created by MMTP has enough bandwidth to carry all traffic. This is not true to any of the channels taken singly. That means that even in the current form MMTP allows a better quality stream with more reliability than any one of the channels used alone

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have shown that by exposing information about link-layer connectivity to the transport layer it is possible to use multiple channels concurrently, and adapt both intra-channel by varying the rate in which data is transmitted and extra-channel by adding and deleting channels from the pool of available channels. The resulting virtual channel has many characteristics that are more amenable to multimedia traffic than each of the channels taken alone. The virtual channel has more bandwidth, less overall variability and more resilience. Added bonus is the built-in mechanism for vertical handoffs.

The simultaneous use of multiple interfaces brings many opportunities and challenges. MMTP was designed for multimedia traffic, and reflects the constraints of this type of data stream. We are currently exploring the design of other transport protocols that can use multiple link-layer channels suitable to various types of data. It is clear that a complementary communication framework is necessary to take advantage of the benefits of such a protocol. The communication framework can help the mobile locate the available communication services and offer a uniform API for MMTP and other protocols. The path for future work lies in the integration of such a communication framework with MMTP and other protocols able to take advantage of knowledge of multiple link-layer channels. In our current research, we are exploring the possibilities opened by the excess of bandwidth on multiple channels, enabling retransmission of select packets without adversely affect the playout of the data stream.

7. ACKNOWLEDGMENTS

This work was sponsored in part by NSF ITR grant ANI-0081308 and by Conselho Nacional de Desenvolvimento Cientifico e Tecnologico, CNPq, Brazil, grant 200558/94-3.

8. REFERENCES

[1] M. Stemm and R. H. Katz, Vertical Handoffs in Wireless Overlay Networks. *ACM Mobile Networking (MONET), Special Issue on Mobile Networking in the Internet*, 1998.

[2] X. Zhao, C. Castelluccia, and M. Baker, Flexible Network Support for Mobility. presented at Fourth ACM International Conference on Mobile Computing and Networking (MOBICOM'98), 1998.

[3] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, RSVP: A New Resource Reservation Protocol. *IEEE Network Magazine*, pp. 8-18, 1993.

[4] D. D. Clark, M. Lambert, and L. Zhang, NETBLT: A High Throughput Transport Protocol.,, 1988.

[5] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, WTCP: A Reliable Transport

Protocol for Wireless Wide-Area Networks. presented at ACM Mobicom '99, 1999.

[6] R. Rejaie, M. Handley, and D. Estrin, RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. presented at IEEE Infocom 99, 1999.

[7] F. M. Chiussi, D. A. Khotimsky, and S. Krishnan, Generalized inverse multiplexing of switched ATM connections. presented at Proceedings of the IEEE Conference on Global Communications (GlobeCom '98), 1998.

[8] A. C. Snoeren, Inverse Multiplexing for Wide-Area Wireless Networks. presented at Proceedings of the IEEE Conference on Global Communications (GlobeCom '99), Global Internet Symposium, 1999.

[9] K. Sklower, B. Lloyd, G. McGregor, D. Carr, and T. Coradetti, The PPP Multilink Protocol, RFC1990., 1996.

[10] J. Bolot and T. Turletti, Experience with Rate Control Mechanisms for Packet Video in the Internet. *ACM Computer Communications Review*, vol. 28, pp. 4-15, 1998.

[11] K.-W. Lee, R. Puri, T. Kim, K. Ramchandran, and V. Bharghavan, An Integrated Source Coding and Congestion Control Framework for Video Streaming in the Internet. presented at IEEE Infocom 2000, 2000.

[12] S. Servetto and K. Nahrstedt, Broadcast Quality Video over IP. *IEEE Journal on Selected Areas in Communications, Special issue on QoS in the Internet*, 1999.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, RTP: A Transport Protocol for Real-Time Applications. Internet Engineering Task Force Internet Draft, July 1994 1994.

[14] S. Floyd and V. Jacobson, Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1993.

[15] K. Jeffay, Towards a Better-Than-Best-Effort Forwarding Service for Multimedia Flows. *IEEE Multimedia*, vol. 1999, 1999.

[16] A. Snoren and H. Balakrishnan, An End-to-End Approach to Host Mobility. presented at ACM Mobicom '99, 2000.

[17] D. D. Clark and D. L. Tennenhouse, Architectural Considerations for a New Generation of Protocols. in *Proceedings of the SIGCOMM '90 Symposium*, 1990, pp. 200-208.

[18] E. D. Jensen, C. D. Locke, and H. Tokuda, A Time-Driven Scheduling Model for Real-Time Operating Systems. in *IEEE Real-Time Systems Symposium*, 1985.

[19] S. Keshav, A Control-Theoretic Approach to Flow Control. presented at Proceedings of the SIGCOMM '92 Symposium, 1992.