MobiCom Poster: A Transport Layer Approach to Host Mobility

Luiz Magalhaes
magalhae@cs.uiuc.edu
Robin Kravets
rhk@cs.uiuc.edu
Department of Computer Science, University of Illinois at Urbana-Champaign

This poster describes a new approach to host mobility, where the responsibility of maintaining an end-to-end pipe that survives changes to the host network attachment point is moved to the transport layer.

Mobility, from the perspective of a network, is the ability to change network points of attachment. Support for mobility can be provided at the link layer (e.g. cell phones, 802.11) or at the network layer (e.g. Mobile IP). Link layer solutions are limited to point-to-point connectivity choices between the mobile and a base station. Network layer solutions are limited to information about the points of attachment of the host to the network. In this research, we present a transport layer approach to host mobility. Such a solution allows mobility decision to be based on end-to-end channel characteristics, as well as local connectivity information. Transport layer mobility support is link layer independent in that it can support heterogeneous technologies, but link layer-aware in that it tracks the presence of available link layer connections and determines which, if any, should be used to support the communications of the mobile host.

The key benefits of our solution are reduced handoff delay, increased bandwidth and reduced effect of losses. Our approach provides seamless mobility support through the simultaneous use of multiple overlapping technologies. As an added benefit, out solution can take advantage of the multiple local base stations of different technologies, enabling bandwidth aggregation. Using inverse multiplexing, the transport layer enables a single application-level flow to be transmitted through multiple link layers, with gain in bandwidth, robustness and handoff delay. The complete mobility solution is composed of three parts. The first is the Link Layer Manager (LLM), which is an entity responsible for link layer discovery and IP layer configuration. The second is a suite of transport protocols designed with the above requirements and adapted for the traffic they will be carrying. The third is a location service to allow mobile hosts to be contacted when away from their home network.

To enable the simultaneous use of multiple link layers for a single data flow, we define a *channel*, which is an end-to-end, transport layer connection that encompasses all available link layers and multiplexes the data of a single flow into these links. The sending application sees a single *channel*, one transport layer interface that remains stable. The transport layer protocol receives the data from the application and sends it through *sub-channels,* network layer sockets mapped to different link layers. At the other end, the transport layer gathers the data from the *sub-channels* and delivers it to the peer application. To create a *channel*, we need information about what link layers are available. The transport layer has to be *link layer aware*, although it will not communicate directly to the link layer, relying instead on the abstractions offered by the network layer. The LLM intermediates the communication between layers, working as a database, event channel and interface for inter-layer communication.

Two protocols were implemented that instantiate the *channel* abstraction. The first is a protocol for multimedia traffic, and the second is a reliable protocol. Both share the underlying characteristics, although they are designed for very different tasks. The first protocol in the suite supports multimedia traffic. The Multimedia Multiplexing Transport Protocol (MMTP) is a rate based multiplexing protocol designed to carry packets with hard deadlines. 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 creates a *channel* that multiplexes the data into any available communication *sub-channel*. As the available *sub-channel*s change, MMTP adapts, adding or removing *sub-channels* as necessary. MMTP provides a best effort service. If the aggregation of available *sub-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 main task of MMTP is the decision as to which sub-channel to use for transmitting the current packet. This decision is based on estimations of the bandwidth and delay characteristics of each sub-channel. After startup, two control mechanisms are used to adapt the sending rate to the sub-channel bandwidth: rate decrease messages and channel probe. Rate decrease messages are sent to prevent congestion when the receiver notices that the channel bandwidth is below the sender's rate. Probing is used to track increases in bandwidth.

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, fixed window size protocol that uses selective acknowledgements for reliability, and bandwidth estimation for flow and congestion control.

Multiple communication *sub-channels* will coexist if the mobile has multiple network interfaces, these interfaces are active simultaneously, and have acquired one exclusive IP address. To multiplex data into those multiple *sub-channels* it is necessary to know the *sub-channel* characteristics, particularly the available bandwidth, for load balancing. We use the packet pair method for measuring bandwidth, and the rate-based transmission mechanism used in R-MTP keeps the regularity of the traffic generated by the protocol. The regularity in which packets are transmitted in R-MTP also allows the interarrival time to be used as an aid to differentiate congestion losses from medium losses. Because R-MTP is a protocol for mobile systems, it will be used mostly in wireless environments, where losses caused by transmission errors are orders of magnitude greater than in wired environments. One of the problems of using TCP in mobile systems is the well know mechanism of slowing down transmission in the presence of losses, which are used an indicator that the protocol exceeded the available bandwidth (i.e., it is creating congestion on a link). TCP has no way of discerning the cause of the loss because it sends packets in bursts. R-MTP analyses the interarrival times to discern if a loss was most likely caused by the wireless medium or if the link is congested. Because packets are spaced regularly, channel jitter is canceled out, and an increase in the interarrival time signals channel congestion.

R-MTP measures the minimum interarrival time of each channel and multiplexes packets on the channels according to the resulting periods. Multiplexing gives a very good abstraction for dealing with mobility: if we assume that all channels are available all the time, but their period is infinite, adding a channel is just changing the period from infinite to a finite value, and deleting a channel is just setting the period to infinity. On the other hand multiplexing requires special attention on the reliability algorithms. Out of order delivery is a very common occurrence due to the different transmission delay on each channel, so we decided to use selective acknowledgements to allow each channel to do its own gap detection, although each packet that is lost is put on the common queue to be retransmitted by the first available interface.

Because traffic conditions on communication channels are in general not static, the first bandwidth measurement done at startup will not be valid for long. R-MTP changes the sending periods on individual *sub-channels* according to the available bandwidth of that *sub-channel*, increasing the period in case of congestion, and decreasing the period if more bandwidth becomes available. Congestion is signaled by increases in the interarrival time and by losses. R-MTP reacts mildly to increasing interarrival times, by increasing the period on the channel with increasing interarrival times an amount corresponding to this increase. If a loss occurs in this condition, the sending period of the channel is doubled. This technique avoids congestion, because no losses are necessary to indicate channel congestion. To track increasing bandwidth, R-MTP probes the channels regularly.

Recently we have been studying two important aspects of the protocols through simulation: the congestion control algorithm and the loss discrimination heuristic. Preliminary results that show support the feasibility of using the homeostatic principle for bandwidth estimation as a means of congestion control are shown on the poster.

More information can be found at our web site: mobius.cs.uiuc.edu