Improving End-to-End Performance of TCP over Mobile Internetworks

Raj Yavatkar

Department of Computer Science
University of Kentucky
Lexington, KY 40506
raj@dcs.uky.edu

Namrata Bhagawat

Department of Computer Science
University of Kentucky
Lexington, KY 40506
namrata@dcs.uky.edu

Abstract

Reliable transport protocols such as TCP use end-to-end flow, congestion, and error control mechanisms to provide reliable delivery over an internetwork. However, the end-to-end performance of a TCP connection can suffer significant degradation in the presence of a wireless link. We are exploring alternatives for optimizing end-to-end performance of TCP connections across an internetwork consisting of both fixed and mobile networks. The central idea in our approach is to transparently split an end-to-end connection into two separate connections; one over the wireless link and other over the wired path. The connection over the wireless link may either use regular TCP or a specialized transport protocol optimized for better performance over a wireless link. Our approach does not require any changes to the existing protocol software on stationary hosts. *Results of a systematic performance evaluation using both our approach and regular TCP show that our approach yields significant performance improvements.*

1 Introduction

Reliable transport protocols such as TCP use end-to-end flow, congestion, and error control mechanisms to provide reliable delivery over an internetwork. However, co-existence of wireless links and mobile hosts with fixed networks poses unique problems for transport protocols. In particular, the following communication characteristics of wireless links have significant implications.

First, Maximum Transmission Unit (MTU) on a wireless link is typically much smaller than that over links in the wired network [1, 2]. Small MTU over the first link forces transmission of smaller packets over the entire end-to-end path even though wired path can accommodate much larger packets.

Second, the error rates on a wireless link are much higher than those experienced over the links in the wired network [3, 4, 5]. Higher error rates (and resulting intermittent connectivity) over a wireless link are due to a combination of factors such as multipath fading, terrain and environmental factors, and interference from other transmissions. In addition, these errors often cause a burst of packets to be lost.

Third, communication pauses during handoffs are also perceived as periods of heavy losses by transport and higher level protocols [6].

These wireless transmission characteristics together contribute to severe degradation in performance of protocols such as TCP. Use of small packets leads to under-utilization of available bandwidth in the wired network and reduces overall end-to-end throughput of a connection. Higher error rates and communication pauses during hand-off can falsely trigger congestion control mechanism of TCP [7]. For example, communication pauses and packet losses over the wireless link cause retransmission timeouts. In both cases, TCP's slow-start mechanism [8] reacts by drastically reducing the current transmission rate and TCP takes a long time to recover from such a reduction resulting in severe degradation in throughput.

We are exploring alternate approaches for optimizing end-to-end performance of TCP connections across an internetwork consisting of both fixed and mobile networks. Our approach is motivated by the following principles:

- We want to achieve performance optimization *without* modifying TCP and its existing flow and congestion control mechanisms.
- Given the widespread use of TCP/IP in fixed hosts, we would like to avoid any changes to the existing protocol software in machines on the wired Internet.
- Existing client/server applications should see no changes to the socket interface and should require no changes to execute across mobile internetworks.

The central idea in our approach is to introduce a new session layer protocol on top of TCP at both base stations (also called Mobile Support Routers or MSRs) and mobile hosts. *We require no changes to the protocol software*

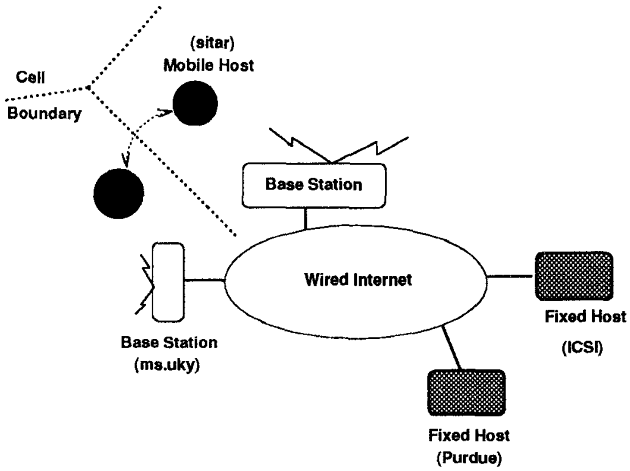

Figure 1: An example mobile internetwork.

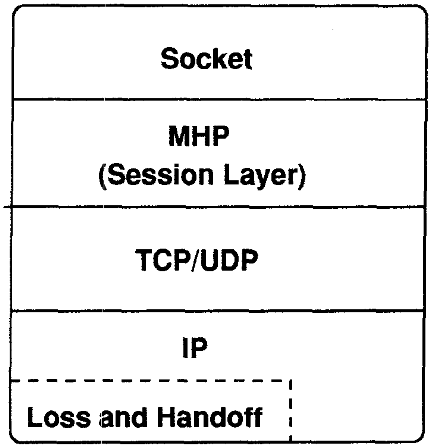

Figure 2: The protocol hierarchy assumed at base stations and mobile hosts.

on ordinary stationary hosts. The session layer protocol is designed to exploit the available knowledge about both wireless link characteristics and host migration and to compensate for highly unpredictable and unreliable link between a mobile host and its base station.

An advantage of this approach is that performance degradation in TCP is limited to a "short" connection over the wireless hop and traffic over the "long" connection over the wired network can be protected from the impact of erratic behavior over the wireless link.

We have considered two alternatives for improving performance of TCP over the wireless hop. The two alternatives can be summarized by an example using Figure 1. Let us assume that a TCP connection is desired between `sitar` and `icsi`.

Under the first alternative (called *MTCP*), the proposed session layer protocol, called **MHP** (Mobile Host Protocol), establishes two TCP connections, one from `sitar` to its base station, and another from its base station to `icsi` across the fixed internetwork. An intermediate agent at the base station acts as a relay for traffic from the first connection to another[1]. In the case of a handoff, we assume that the mobile IP protocol can pass on an indication of *handoff in progress* to higher layer protocols using an upcall through the protocol layers. When the handoff completes, MHP transfers the connection state information to the new base station and establishes a new connection between the mobile host and its new base station. No changes are, however, necessary to the connection with the remote host as mobile IP routing [9] takes care of routing packets through the new base station.

[1]If the remote host is also mobile, an additional connection must also be set up over the wireless link to its base station.

The second alternative (called *SRP*) is similar to the first alternative except that the session layer does not use TCP as its transport layer for the wireless hop. We are considering this alternative to investigate whether one can justify use of a specialized transport protocol tuned and optimized for better performance over a wireless link. Under SRP, the protocol used over the wireless hop uses its own flow and error control mechanisms designed and optimized specifically to tackle the lossy and erratic delay characteristics of the wireless link. The intermediate agent at the base station participates in the session layer protocol and forwards incoming traffic over a TCP connection to the remote host. The session layer hides the details of the first connection and provides the same application layer interface as TCP through the Unix socket library.

We have compared both alternatives against the use of normal TCP in a mobile internet testbed consisting of a simulated wireless link and the Internet. Our tests have yielded impressive results. The rest of this paper is organized as follows. Section 2 describes the protocol model in detail. Section 3 describes the experimental setup, methodology, and results of our performance evaluation. Section 4 summarizes the related work in this work and Section 5 provides concluding remarks.

2 Protocol Model

Our goal is to isolate the wired portion of the path of a connection from the impact of erratic behavior over the wireless portion and also to recover quickly from errors over a wireless link to obtain good end-to-end performance. We regard the impact of small MTU and intermittent connectivity over a wireless link as transient errors over a transport level connection and we believe that the protocol software must protect applications by transparently recovering from such transient errors. In the ISO reference model [10], the responsibility for session management including recovery and re-synchronization in data transfers lies with the session layer in the protocol stack. Transport layer protocols only provide end-to-end delivery of messages or byte streams. In keeping with the ISO reference model, we introduce a new session layer protocol called **MHP** (Mobile Host Protocol) that explicitly includes mechanisms for recovering from errors over the wireless link.

Figure 2 shows the proposed protocol hierarchy for networking software on mobile hosts and their base stations (also called Mobile Support Routers or MSRs). We assume that no changes are necessary to the existing network protocol software on fixed hosts. The protocol software on mobile hosts and their base stations is now augmented with a new session layer protocol called MHP (Mobile Host Protocol) that retains the same API (such as BSD socket interface [11]) as that offered by TCP.

2.1 Connection Establishment

Figure 3 shows an example interaction involving a mobile host and a remote, stationary host. We assume that protocol software on base stations and mobile hosts consists of an MHP layer that manages the transport level connections. In the following, we describe how MHP layers at base stations and mobile hosts cooperate to support an end-to-end connection.

- When a TCP application on the mobile host X issues a `connect` call to request a connection to a remote destination Y at `<destIPaddr, destPort>`, the MHP layer at X (MHP_X) intercepts the call and instead requests a transport level connection ($Connection_1$ in Figure 3) with its peer at its current base station. MHP peer on the base station sets up a surrogate or MHP agent (MHP_BS1) on behalf of the requested connection. The surrogate, MHP_BS1, in turn, establishes a TCP connection ($Connection_2$ in Figure 3) with Y at the address `<destIPaddr, destPort>` on behalf of the endpoint on X. One endpoint of $Connection_2$ is still marked as `<X_IPaddr, X_srcPort>` and all the TCP traffic from Y to X is intercepted and forwarded to the surrogate MHP_BS1 at the base station. As described in Section 2.2, MHP_BS1 simply acts as a relay for the traffic between X and Y in both directions.

- When a TCP application on a stationary host (Y) requests a TCP connection to a mobile host (X) at address `<MHaddr, MHPort>`, the connection request is intercepted and forwarded to the surrogate MHP_BS1 at the base station. The surrogate then completes the TCP connection establishment with Y and establishes a new connection with its peer (MHP_X) at the given address.

- A TCP connection between endpoints on two mobile hosts is handled similarly except, in this case, three separate connections are established.

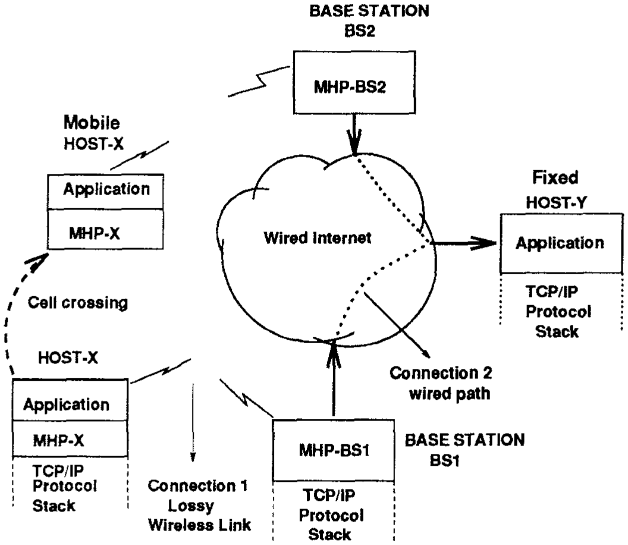

Figure 3: An example of connection establishment and handoff involving a mobile host (X), stationary destination (Y), and two base stations. Initially, X establishes a connection to Y through the MHP agent at base station BS1. After a cell handoff, a new connection is established between X and the new base station BS2 and the endpoint of connection 2 is transferred to MHP agent on BS2.

2.2 Data Transfer

Data transfer to and from the mobile host X and remote destination host Y proceeds as follows. When a TCP application on X sends data, MHP_X uses the first connection to send that data to MHP_BS1. In particular, MHP_X sends data in small segments to match the smaller MTU over the wireless link. MHP_BS1 receives the data and buffers it to assemble these smaller packets into larger TCP segments before forwarding them over the connection to Y. Similarly, when MHP_BS1 receives TCP segments from Y, it first breaks them into smaller fragments to match the MTU over the wireless link, forwards smaller TCP segments to X.

To recover from handoffs, the MHP layer must maintain some state information on the segments in transit to and from the wireless link. Therefore, the MHP layer maintains state information on the segments in its buffers and also accesses the connection state information maintained by its underlying transport protocol. The state information accessed includes connection parameters such as current window sizes and sequence numbers for window edges.

2.3 Error Recovery

To recover from errors due to high bit error rates of the wireless link and handoff, we have investigated two alternatives.

Under the first alternative called **MTCP** (Multiple TCP), MHP uses regular TCP as the transport protocol for the connection over the wireless link.

Under the second alternative called **SRP** (Selective Repeat Protocol), MHP uses a specialized transport protocol designed to recover quickly from higher and sometimes bursty packet losses experienced over the wireless link. SRP uses a selective repeat algorithm in which a receiver returns a selective ACK (SACK) when an out of sequence segment is received. The SACK specifies the missing segments using a bitmap, the sequence number of the latest segment received, and the sequence number of the last segment received in sequence. On receiving a SACK, the sender retransmits all the missing segments specified in the SACK. Using this alternative, unlike TCP, SRP can recover more than one segment in one round trip time and can yield better throughput over the wireless link.

Section 3 compares the performance of two alternatives when used over a mobile internet.

2.4 Handoff Management

When the MH moves and crosses the current cell boundary, it gets attached to another base station (BS2) and the IP datagrams for TCP segments over an existing connection start getting forwarded to the new base station. During the cell handoff, we must also make sure that the existing transport connections get transferred to a new MHP agent.

We assume that the MHP layer at the mobile host registers an upcall function with its IP layer. When a handoff is completed, IP layer on MH informs the MHP layer of handoff using the upcall function and passes the address of the new base station (BS2) to it. The MHP layer (MHP_X) then contacts its peer at the new base station to initiate a *handoff management* procedure that consists of the following steps:

- On receiving the upcall, MHP_X first suspends the ongoing data transfer across its transport connections, contacts its peer at BS2 giving it the address of the previous surrogate MHP_BS1, and then waits for a *connection resume* message from BS2.

- The MHP peer at BS2 establishes a new MHP agent or surrogate (MHP_BS2) for the connection. The new surrogate then sends a *handover* message to the old surrogate (MHP_BS1) requesting the state information for two connections.

- MHP_BS1 responds with the connection state information and, in addition, also forwards the TCP segments in transit that it has buffered for traffic in each direction. When MHP_BS2 receives the state information, it re-creates the state information for connections with MHP_X and Y and then sends a *connection resume* message to MHP_X.

- Data transfer to and from MHP_X then resumes and the remote stationary host observes no changes in the state of its connection except possibly for some transport layer retransmission of data lost during handoff.

3 Performance Evaluation

We have conducted a systematic performance evaluation of our approach using a wireless internet testbed. In the following, we describe the testbed, experiments performed, and results obtained.

3.1 Experimental Testbed

The testbed consists of two parts. The first part consists of a wireless network simulated over an ethernet segment and Sun sparcstations running a modified SunOS kernel acting as *mobile sparcstations*. The second part consists of our campus network attached to the rest of the Internet over a T1 link. Some sparcstations on the campus network act as base stations.

We have implemented MHP in two different versions. To test our ideas, we first implemented MHP as a user level library and rest of this paper reports results obtained using the user level MHP implementation. We also have a kernel implementation of MHP on mobile hosts and base stations that resides below the socket layer (above TCP) and provides the same interface as TCP through the socket interface.

The IP software in the SunOS kernel of the mobile sparcstations has been modified to simulate a wireless link as follows:

1. In *mobile* sparcstations, we have modified the IP layer to use a smaller MTU (128 or 256 octets). In addition, the IP software simulates packet losses and handoffs. Delay and loss characteristics simulated are taken from the experiences reported in the published literature [1, 4, 3, 12].

2. IP software simulates bursty losses over the wireless link. The bursty loss simulation models the interburst gap using an exponential distribution around a mean inter-burst interval (IBG) and the size of each burst is modeled using a geometric distribution around a mean burst size (BS) value. The values of IBG and BS were chosen for each experiments based on the the average packet loss desired for the experiment. We

have simulated packet losses of 0, 5, and 10% for different testcases.

3. Sparcstations located on campus subnets act as base stations where each subnet is considered a different cell and subnets are separated by a campus router.

4. The IP layer simulates a handoff at a mobile host by simply pausing for the handoff duration and dropping all outgoing and incoming packets during the pause. We have simulated handoff pauses of 1, 2.8, and 5 second durations based on figures taken from [6].

   We have also carried out tests using multiple handoffs in which successive cell handoffs are simulated spaced at different intervals ranging from 5 to 10 seconds.

3.2 Methodology

In our experiments, we use a user-level test program that establishes a connection between a mobile host and a remote stationary host and transfers data in a file of fixed size to its peer at the destination. Tests were carried using stationary hosts either located in the local area (on a campus subnet) or located across the Internet at ICSI and UC Berkeley, Purdue University, and Washington University in St. Louis.

Once the connection is in progress, mobile IP software simulates a handoff pause duration starting after a fixed, predetermined interval (typically 4 to 8 seconds, an experimental parameter) after the connection starts and contacts the mobile IP software at a new base station at the end of the handoff pause to complete the handoff.

For each test, we repeated the experiments over a two week period on weekdays between 1 and 3 pm EST to obtain results under similar Internet traffic conditions[2]. Using samples from 40 independent runs, we carried out a confidence interval analysis with 95% probability and have tabulated the confidence intervals along with average values.

3.3 Representative Results

Tables 1 through 3 show a representative sample of results. We have also conducted tests involving remote hosts located at Purdue University and Washington University and have obtained similar results.

Table 1 shows the results for the base case used for comparison with our approaches. The entry in upper left hand corner (no pause, no losses) shows the results in the absence of mobility (no handoff pause, no losses due to mobile link) and, as can be seen clearly, performance degrades as a single handoff pause and packet losses due to wireless link are introduced.

[2]We have also conducted tests late night to evaluate performance under different Internet traffic conditions.

Results Using Regular TCP

Handoff	Packet loss in Percent		
Pause	0 %	5 %	10 %
0 sec	21.7 [19.3, 24.1]	34.4 [30.6, 38.2]	63.3 [53.0, 73.6]
1 sec	31.7 [27.6, 35.9]	44.6 [40.9, 48.3]	56.6 [50.5, 62.7]
2.8 sec	32.6 [29.2, 36.0]	52.1 [45.6, 58.6]	88.7 [77.6, 99.7]
5 sec	36.7 [34.0, 39.3]	69.8 [60.1, 79.6]	99.9 [86.6, 113.1]

Table 1: Mean time to transfer a file of size 100k bytes with a single, normal TCP connection between the mobile host sitar.dcs.uky.edu and the remote destination icsib16.icsi.berkeley.edu. The confidence Interval of 95% is shown in square brackets.

Results using MTCP

Handoff	Packet loss in Percent		
Pause	0 %	5 %	10 %
0 sec	16.3 [16.0, 16.6]	20.1 [19.0, 21.2]	27.6 [24.8, 30.4]
1 sec	20.9 [20.2, 21.6]	26.6 [24.4, 28.8]	28.4 [27.0, 29.9]
2.8 sec	23.0 [21.0, 25.0]	26.0 [23.3, 28.6]	31.9 [29.6, 34.1]
5 sec	26.9 [26.0, 27.7]	30.6 [29.3, 31.9]	32.6 [30.9, 34.3]

Table 2: Results for tests carried out for the same case as Table 1, but using MTCP.

Results using SRP

Handoff	Packet loss in Percent		
Pause	0 %	5 %	10 %
0 sec	12.7 [11.7, 13.7]	19.6 [18.7, 20.5]	22.4 [21.0, 23.9]
1 sec	13.9 [12.5, 15.3]	20.1 [18.4, 21.7]	26.6 [24.7, 28.6]
2.8 sec	21.1 [19.7, 22.5]	27.3 [25.5, 29.1]	29.2 [26.9, 31.4]

Table 3: Results for tests carried out for the case same as for Table 1, but using SRP.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.