ILC-TCP: An Interlayer Collaboration Protocol for TCP Performance Improvement in Mobile and Wireless Environments

M. Chinta and A. Helal
Department of Computer & Information Science & Engineering
University of Florida, Gainesville, FL 32611-6120, USA
helal@cise.ufl.edu

ABSTRACT

The growth of the wireless Internet has led to the optimization of network protocols to provide for a better performance. Most of the Internet traffic uses TCP, the de-facto transport layer protocol. Unfortunately, TCP performance degrades in the mobile and wireless environments. A good amount of research has been attempted to improve its performance in the unpredictable mobile and wireless environments where link disconnections, packets losses and delays are common. Most of these proposed solutions target the case where the mobile host acts as a TCP receiver. For various reasons, almost none of the solutions have reached the stage of deployment. In this paper, we propose an interlayer collaboration model for TCP performance improvement in mobile and wireless environments. We specifically target the case where a mobile acts as a TCP sender. ILC-TCP is an end-to-end approach and does not need any special support from the base station infrastructure. ILC-TCP is evaluated against the normal TCP in various scenarios. Performance results suggest that ILC-TCP performs better than the normal TCP in many scenarios involving long disconnections, frequent disconnections and in the scenarios where a mobile host moves at considerable speeds.

KEYWORDS

Mobile Networking, TCP/IP, Handoff, Network Performance, Wireless Networking.

1. INTRODUCTION

The growth of the wireless Internet gave impetus to the development of various kinds of applications and small devices that use the resources of a network. Also, this led to the development of new paradigms like Mobile Computing, Pervasive Computing, and Ubiquitous computing. Thus, the users of the wireless Internet expect the same kind of services, as do the users of the wired Internet. Unfortunately, difference in the link technologies makes it difficult to hide the difference in the level and quality of services between the wired Internet and the mobile and wireless Internet. So, a quest for optimized protocols, suitable for the wireless Internet, has begun few years ago.

Nearly 80-85% [Tho97] of the Internet traffic uses the reliable transport protocol, TCP, for their transport. Most of the dominant Internet applications like File Transfer Protocol (FTP), Hyper Text Transfer Protocol (HTTP), and Telnet use TCP for their transport.

TCP was basically developed assuming that it would run on wired networks. Wired `networks usually have less bit error rates and hence less packet loss because of such reasons. Also, since mobility was not considered, there are no packet losses and delays caused by mobility. Hence, TCP was designed to assume that any segment loss is caused by the network congestion and on seeing any segment loss, TCP invokes its congestion control measures.

Usually, the wireless links have high bit error rates. Also, temporary disconnections occur because of the factors like channel fading, and handoffs when a mobile node is in motion. The handoff period depends on both the link level handoff protocol being used as well as the IP level handoff protocol being used, if any.

The standard TCP implementations assume the cause of any packet loss to be the network congestion. Then, they reduce the congestion size to a minimum. Also, TCP invokes the slow-start mechanism. If the network links are slow, it takes long time to grow the congestion window. Thus, reduction in the size of the congestion window reduces the transmission rate and hence degrades the performance. Also, TCP undergoes a binary exponential backoff, causing long pauses of communication. Because of this, some time may be unused even after the mobile node recovers from the temporary disconnection.

Thus, TCP proved not so efficient in wired cum wireless environments and suffers from performance degradation introduced by the conditions that exist in those environments. Hence, an optimized reliable transport protocol is very important for the development of the wireless Internet.

A good amount of work has been attempted in the past to optimize TCP in the wireless environments. Also, work has been done to improve its performance in mobile conditions. Unfortunately, many of the approaches that were suggested, suffer from many drawbacks regarding their deployment, and the kind of changes they require to the existing infrastructure.

In this paper, we propose an approach to improve the performance of TCP in mobile and wireless environments in the case where a mobile host acts as a TCP sender. Many scenarios like mobile devices transferring files (both data and music) to a different place in the network, sending emails (possibly with attachments) and many such applications suggest the need for uplink TCP transactions. Also, our approach doesn't require any special support from the base stations and other network infrastructure. We propose an interlayer collaboration model and an interlayer collaboration protocol, in which a new management layer parallel to the existing network protocol stack in a mobile device is introduced. The management layer receives information from the various layers of the networking stack and attempts to assist TCP to cope with the complexities of the wireless and mobile environments.

In the following sections, work done in the past to improve its performance in those environments, the architecture and working of our protocol, ILC-TCP and the experimental evaluation of our approach.

2. RELATED WORK

A good amount of research has been done in the past to improve the performance of TCP in the conditions that prevail in mobile and wireless environments. In this section, we categorize the early solutions and briefly outline their pros and cons. Detailed survey of the earlier work done can be found in [Pen00], [Tsa02].

2.1. Split connection Approaches

The basic idea of split connection approaches is to use a separate TCP connection, one for the wireless network and the other for the wired network between a sender and a receiver. The motivation behind the use of separate connections is the fact that the characteristics of a wireless network are different from the characteristics of a wired network, hence a separate connection that is optimized to the kind of network it operates on, can be used. Usually, the connection is split at the base station. Some of the split connection approaches are I-TCP [Bak95], M-TCP [Bro97].

Though split connection approaches improve the TCP performance to certain extent in wireless networks, they may not perform well in mobile environments where frequent disconnections are common. Also, I-TCP violates TCP end-to-end semantics, as the segments are acknowledged even before they reach the mobile host. One recent problem these approaches face is the problem with the encrypted traffic as in the case where IPSec is used for encrypting the IP payload. In that case, the base station cannot even know that payload inside an IP packet is of TCP.

Link Level Solutions requiring Transport Level Support from Base Stations

Many solutions were proposed that are based at the link level but require transport support from the base station. The basic idea of these solutions is to place a smart software module in the base station, which can be used for special purposes like local recovery by performing retransmissions, suppressing duplicate acknowledgments etc.

The most prominent and famous protocol in this category is Snoop [Bal95]. It introduces an intelligent software module that performs local retransmissions. Some other solutions that use transport-aware link layer are WTCP [Rat98], Explicit Loss Notification (ELN) [Bal98]. ELN suggests a way to improve the performance of TCP in wireless environments where the mobile host acts as a TCP sender.

Certainly, the performance improves by using the above approaches, but certain problems, that were earlier described in the case of split connection approaches, like handling frequent disconnections, and encrypted traffic, still are not solved by these approaches.

2.2. Enhanced Link Layer

Since the basic factor that causes degradation in wireless environments is the nature of the transmission medium, an improvement in the link technology that operates over the physical layer can improve the overall performance. This was the idea behind enhanced link layer solutions. Mechanisms like ARQ, FEC [Cho99] were suggested to improve the performance at the lower layers which, in turn, can reduce the chances of TCP getting timed out and does not invoke its congestion control measures. Some other approaches like AIRMAIL [Aya95] come under this category.

These solutions can improve the performance to certain extent, but are not sufficient in mobile environments because they won't take into consideration the delays and losses introduced by the mobility management protocols.

2.3. End-to-End Protocols

Most of the above mentioned solutions require special support from the network infrastructure in some form or the other. Some other approaches were suggested that do not require any special support from the intermediate infrastructure and changes are confined to the end nodes. TCP Probing [Tsa00], Freeze TCP [Gof00], Fast Retransmits [Cac95] come under this category.

These solutions are not bound by the problem of encrypted traffic, and they do not assume any special support from the network infrastructure.

Though the end-to-end protocols can enhance the TCP performance to certain extent, the link layer approaches may be better than the end-to-end protocols for local recovery.

3. OUR APPROACH

3.1. Why ILC-TCP?

Though a good amount of research has been done in the past to improve the TCP performance, almost none of the above-described solutions came to the stage of actual deployment. This is because of various factors and issues like the kind of changes a solution requires to the network infrastructure, handling of encrypted IP payload, ease of implementation of the protocol, interoperability with the existing TCP. Also, most of the earlier solutions (except ELN) try to optimize the where a mobile host acts as a TCP receiver but do not consider the case of a mobile TCP sender, which can also be a common case in the future.

Relatively, very less research has been done in the past in the area of inter layer collaboration. Some approaches like Freeze TCP [Gof00] and Fast retransmits [Cac95] use cross layer signaling to certain extent.

Our argument in favor of inter layer collaboration is that the factors that cause performance degradation of a TCP based application do not stem from any single layer. Thus, the basic idea of our approach is that the information available at one layer can be used in the decision making of another, for performance enhancement.

Our solution tries to optimize the case where a mobile host acts as a TCP sender. Though our solution is not an alternative to all the work done in the past, it certainly acts as a complement to the existing solutions for achieving the goal of optimized TCP in wired, wireless and mobile environments. Also, we argue on the need for a new layer in the network protocol stack for a better performance.

3.2. ILC-TCP Interlayer Collaboration Model

Figure 1 shows our Interlayer collaboration model. We introduce a new management layer parallel to the networking stack called the State Manager. In this model, each layer shares minimal information with the state manager.

As shown in Figure 1, State manager acts as a parallel layer exchanging information with all the core layers of the network protocol stack. In our interlayer collaboration model, the LL/PHY and/or the IP layers periodically notify their status to the State Manager. TCP, in the case of a timeout, contacts the State Manager. Based on the information it gathered from the other layers, the State Manger decides on what it considers to be the best action for the TCP to do at that instance. It then notifies the TCP about the action that TCP can take.

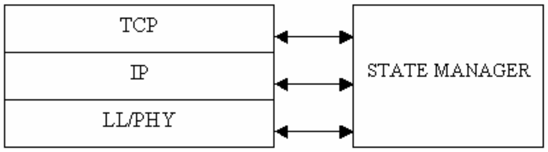


Figure 1 - ILC-TCP Interlayer Collaboration Model

The information that can be exported to the State Manager by each layer depends on the specific protocol that is being used. In our inter layer collaboration protocol, the information LL/PHY exports to the State Manager is whether the link is good state (LINK_OK) or not in a good state (LINK_NOT_OK). The information that IP exports to the State Manager depends on various factors. In our inter layer collaboration protocol, we assume the mobility protocol implemented is mobile IP. But this can be easily extended to any mobility protocol in use (e.g., cellular IP, Hierarchical Foreign Agents). In essence, the IP layer sends the information about the network level connectivity to the State Manager. The term "network level connectivity" can be explained as follows. The mobility protocol Mobile IP has a concept of agent registration. If the mobile host (MH) is registered with any foreign agent, then it has an understanding that it can receive the packets from its corresponding peers. But, if it is not registered with any of the foreign agents and if it is not in the home network, then it has an understanding that the probability of it not getting packets from the fixed host is more. Thus, the IP layer can contribute this knowledge about its current state, to the State Manager.

Also, in our approach, there is no restriction on which layer contributes information and which layer does not. The more information the State Manager has, the more are the chances that its decisions are accurate.

The next section describes the detailed working of the protocol.

3.3. ILC-TCP Interlayer Collaboration Protocol

The complete ILC-TCP protocol is described below in an event driven fashion, component-by-component. All the events that are relevant to our approach are described. Various events and their effects of on each component are described. We use a pseudo language to describe the protocol. The notation x $\Rightarrow$ y indicates that if an event x is received then action y is taken. Also, "//" indicates the start of a comment. The comments end again by "//". Figure 2, Figure 3 and Figure 4 show the interactions of the various layers of the network protocol stack with the State Manager.

3.3.1. System Level Interactions

System level interactions are explained in Figure 2. We need not expect each layer to contribute its information to the State Manager. We can notify the State Manager about the layers from which it will receive the information. By LINK_ONLY, we mean that the State Manager will get information only from the Link Layer and by LINK_IP, it means that both the LL and IP will contribute their information to the State Manager.

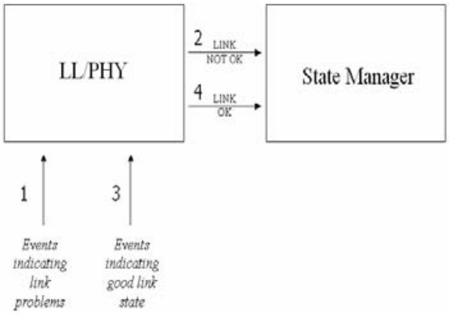
```
If only LL/PHY can export information to the State Manager
Then Send LINK_ONLY notification to the State Manager;
Else
if both LL/PHY and IP can export information to the State Manager
Then Send LINK_IP notification to the State Manager;
End If;
```

Figure 2 – ILC-TCP Protocol: At the System Level

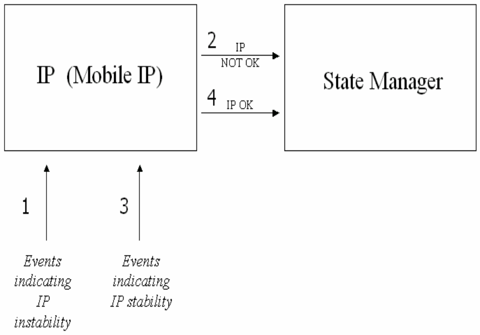
In the following subsections we describe the interactions of each of the LL/PHY, IP and TCP layers with the State Manager. All interactions are also summarized in Figure 3.

3.3.2. Interaction with the physical layer

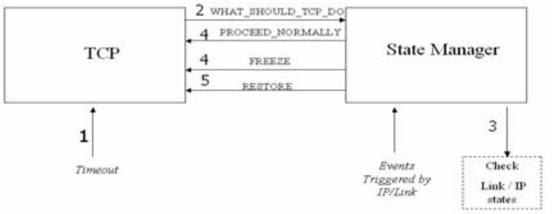
Link Layer can contribute its knowledge about the link to the State Manager. When link layer finds that the link quality is deteriorating, it can send a *LINK_NOT_OK* notification to the State Manager. This gives a hint to the State Manager that the channel is fading or a handoff is impending or of an impending disconnection. In either of those cases, packet loss can occur. If the link state was previously bad after some time the link state becomes good, link layer can send a *LINK_OK* notification to the State Manager. This helps the State Manager in restoring the state if needed. Interactions at this layer are summarized in Figure 4.



(a) LL/PHY – State Manager Interaction



(b) IP-State Manager Interaction



(c) TCP-State Manager Interaction

Figure 3 - ILC-TCP Interlayer Collaboration Model

```
Case event of
 Events indicating link problems ⇒
   Send LINK_NOT_OK notification to the State Manager;
   link_state ← NOT_OK;
 Events indicating good link state⇒
   If (link_state = = NOT_OK)
      Begin
        link_state ← OK;
        Send LINK_OK notification to the State Manager;
      End
```

Figure 4 – ILC-TCP Protocol: At the LL/PHY

```
Case event of
   Events indicating IP instability ⇒
      Send IP_NOT_OK notification to the State Manager;
   Events indicating IP stability ⇒
      Send IP_OK notification to the State Manager;
```

Figure 5 – ILC-TCP Protocol: At the IP Layer

3.3.3. At the IP Layer

In our protocol, we assume the mobility protocol to be Mobile IP. But this can be generalized to any mobility protocol. When a mobile host is not in the home network and is not registered with any of the Mobile IP foreign agent, it sends an *IP_NOT_OK* notification to the State Manager. This gives an indication to the State Manager that there is a chance of an IP level handoff and the network routes have may not yet converged. This helps the State

Manager in interpreting that the acknowledgment segments may be lost. Whenever the mobile host receives registration replies from a mobility agent, it sends an *IP_OK* notification to the State Manager indicating it about the stability at the IP level or completion of an IP level handoff. This is shown in Figure 5.

3.3.4.At the TCP layer

In case of a timeout, TCP contacts the State Manager regarding the action it needs to take at that instance. The State Manager, in such conditions, can suggest one of the two actions. One is PROCEED_NORMALLY, i.e., the State Manager is not in a position to suggest anything from the information it has. In that case, the TCP behaves as normal TCP. Another action that can be suggested by the State Manager is FREEZE. After sending the FREEZE action, the State Manager registers TCP. By registering, we mean that when conditions become normal, it the responsibility of the State Manager to restore TCP to its normal state. When it receives FREEZE, the TCP does not Slow Start and it does not reduce its congestion window and it does not backs off its retransmission timer. It waits for itself to be restored by the State Manager. Interactions at TCP level are shown in Figure 6.

3.3.5.At the State Manager Layer

State Manager is the main entity in our approach. It is responsible for the inter layer collaboration. It receives events from different layers about their knowledge about the state of the system and the network. It keeps track of the different events like LINK_ONLY, LINK_IP, LINK_NOT_OK, LINK_OK, IP_OK, and IP_NOT_OK. When it receives a WHAT_SHOULD_TCP_DO query, then the State Manager can one of the two replies to TCP. First, the State Manager checks if it is configured to get the information from both the LL/PHY and IP or only LL/PHY. If it is configured to get information from only LL/PHY, then it checks the latest notification it received from the LL/PHY. If it is LINK_NOT_OK, it means that the Link Layer has notified about the possibility of the link being down. Then the State Manager comes to a conclusion that there is a chance of the acknowledgments of TCP segments getting lost and it may not be because of the network congestion. So it sends the FREEZE signal to the TCP. Also, it registers the TCP so that it can restore sometime later when the link becomes up again. If it did not receive a *LINK_NOT_OK* from the link layer, then it comes to a conclusion that the timeout is caused by something else and suggests the TCP to proceed in its normal way by sending a *PROCEED_NORMALLY*. Similarly, if it is configured to get information from both the IP and link layer, then it checks whether both the link and IP are ok or not. If not, then it sends the FREEZE notification or else it sends the *PROCEED_NORMALLY* notification to the TCP.

Also, when it gets LINK_OK and IP_OK notifications, it understands that the link is in good state and that the IP is back to normal. It then performs the checks to determine whether or not it can restore TCP back to its normal state. The checks involve seeing whether the link state is ok or not and whether the IP is ok or not. If the State Manager finds both the link and IP to be up and if it has TCP registered with it, then it sends RESTORE notification to TCP.

```
Case event of
  Timeout ⇒ Send WHAT_SHOULD_TCP_DO query to the
            State Manager and determine the Action;
            If  (Action = = PROCEED_NORMALLY) then
                Default back to Normal TCP behavior;
            Else if (Action = = FREEZE) then
                Freeze the TCP state;
                // Do not do any TCP control measures; //
                // Do not back off the TCP retransmission
                                                  timer //
            EndIf
  Restore ⇒  // Assertion: TCP state is the same as
             it was before it went into Freeze Mode //
             Retransmit the first unacknowledged segment;
             Henceafter, follow the normal TCP behavior
EndCase;
```

Figure 6 – ILC-TCP Protocol: At the TCP Layer

4. Experimental Evaluation of ILC-TCP

We have implemented the ILC-TCP in the Network Simulator-ns2. The source code was modified to incorporate the State Manager component. Also, configuration options were provided to specify the collaboration model. That is, whether the State Manager receives information from both IP and LL/PHY or it receives information from only LL/PHY. Thus, we have a way to compare the performances of both the approaches. The link state was assessed by measuring the power of frames. We used the heuristic that low powered frames are an indication of impending link disconnection.

We conducted a series of experiments to determine quantitatively how much performance gain do we achieve by employing our approach when compared to the normal TCP. The experiments comprise of various scenarios like the ones with frequent disconnections, long disconnections, networks having slow links, different file sizes, etc. Also, we compare the performance of ILC-TCP with the normal TCP in mobile environments where the mobile host moves with varying speeds. The following sections briefly describe each experiment and the results obtained. Also, ILC-TCP1 refers to our approach in which only link layer contributes its information to the state manager and ILC-TCP2 or ILC-TCP refer to our approach in which both the link layer and the IP layer contribute their knowledge to the state manager. In all experiments, we have used both the standard Reno and Tahoe TCP implementations. In all experiments, only one mobile host is considered. In most of the experiments, the topology looks similar to the Figure 8.

```
Case event of
        LINK_ONLY ⇒
                Make of note of this event;
        LINK_IP ⇒
                Make of note of this event;
        LINK_NOT_OK ⇒
                Make a note of this event;
        LINK_OK ⇒
                Make a note of this event;
                Invoke restore_if_needed function;
        IP_NOT_OK ⇒
                Make a note of this event;
        IP_OK ⇒
                Make a note of this event;
                Invoke restore_if_needed function;
        WHAT_SHOULD_TCP_DO ⇒
          If (LINK_ONLY) Then
          Begin
            If (LINK_DOWN) Then
              Send FREEZE command to TCP;
            Else
              Send PROCEED_NORMALLY command to TCP
            EndIf;
          End
          Else If (LINK_IP) Then
          Begin
            If (LINK_UP and IP_UP) Then
              Send PROCEED_NORMALLY command to TCP;
            Else
              Send FREEZE command to TCP;
              TCP_REGISTERED = true;
            End If;
          End;
        End If;
EndCase;

    Function restore_if_needed
    Begin
      If (LINK_ONLY) Then
      Begin
        If (LINK_UP and TCP_REGISTERED) Then
              Send RESTORE notification to TCP;
              TCP_REGISTERED = false;
        EndIf
      End
      Else If (LINK_IP) Then
      Begin
        If (LINK_UP and IP_UP and TCP_REGISTERED) Then
              Send RESTORE notification to TCP;
              TCP_REGISTERED = false;
        EndIf
      End
    End;
```

Figure 7 – ILC-TCP Protocol: At the

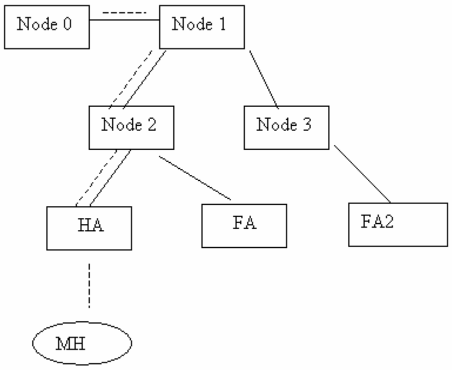


Figure 8 - Most common Topology used in the experiments

4.1. Experiment 1: Link and IP level Handoffs

In this experiment, we ran a number of simulation scripts in ns-2 to study the effect of handoffs as well as link disconnections on the TCP performance of the normal TCP and the TCP with inter layer collaboration. The disconnection is caused at the link layer. But, even if the mobile host is connected to a new node, it may not able to receive the acknowledgment packets till some time because of the mobile IP registration process. The possible loss of the acknowledgments will cause the TCP to interpret it as a sign of congestion in the normal TCP.

In each simulation, a 10MB stream of data was sent from the mobile host to the correspondent host 3 hops away. In each of the simulations in this experiment, we had two disconnections, the length of disconnection being a parameter to the simulation.

The file transfer starts after 1sec the simulation starts. The mobile host starts moving towards the foreign agent, FA. After few seconds, the mobile host registers with the foreign agent at the foreign agent FA. The mobile host moves at 1m/s. All the links have a bandwidth of 1Mb/s. The performance results are shown in Table 1.

In the above experiment ILC-TCP1 represents the case when the state manager gets information from only the link layer. But, in ILC-TCP2, the state manager gets the information from both the IP as well as the link layer.

The results show that we can achieve almost 12-18% performance improvement even if only one disconnection is considered. If the number of disconnections is more, the performance gain increases. Also, in all the cases, we find that the ILC-TCP2 performs better than ILC-TCP1. This is expected because the State Manager can have a better estimate of the state of the mobile host if it has more information. Even in ILC-TCP1, TCP performance can degrade because the delay and loss introduced by the mobile IP handoff is considered to be as a sign of network congestion.

Disconnect Time (Secs)	Transfer Time (Secs)		
	Normal TCP	ILC-TCP1	ILC-TCP2
0	69.53	69.53 (0%)	69.53 (0%)
1	89.57	81.35 (+9.2%)	78.45 (+12.4%)
2	89.67	82.36 (+8.2%)	78.4 (+12.6%)
3	89.64	79.96 (+10.8%)	78.41 (+12.6%)
4	95.72	81.02 (+15.4%)	78.47 (+18%)
5	95.65	81.98 (+14.3%)	78.4 (+18%)
6	89.69	81.02 (+9.7%)	78.42 (+12.6%)

Table 1 - Comparison of ILC-TCP1, ILC-TCP2 and Normal TCP; Transfer Size = 10MB Stream; Correspondent Host 3 hops away from the Mobile Host; Mobile Host moving at 1m/s; Intermediate Link Bandwidth = 1 Mb/s

4.2. Experiment 2: Link Disconnection

In this experiment, we do not consider any IP level handoff, but do consider link level disconnections. We simulated the link level disconnection simulated by keeping the mobile host out of the range of the base station for a period equal to the disconnection time and bringing it again in the range of it. We considered a 500KB file transfer for all the simulations in this experiment. As in the previous example, the TCP receiver on the fixed host was located 3 hops away from the mobile sender. But, this time the link bandwidth between the Node 2 and the HA was changed to 2 Mb/s. All the other links have a bandwidth of 5Mb/s and all the links have a delay of 20ms. In this experiment, only one link disconnection was considered. The results are shown in Table 2.

The transfer times of these simulations are less, so we could not find much difference between the normal TCP and the ILC-TCP for disconnection times of less duration. But for longer duration disconnections, the performance improvement is good. Also, better results can be expected if multiple disconnections are considered.

4.3. Experiment 3: Link Disconnection, Low Bandwidth Link

In this experiment, a number of simulations were run involving the transfer of a 500KB file from the mobile host to the corresponding 3 hops away. But, this time the link bandwidth between the Node 2 and the HA was changed to 0.5 Mb/s. All the other links have a bandwidth of 5Mb/s and all the links have a delay of 20ms. In this experiment, only one link disconnection was considered.

Table 3 shows the results. The results are similar to what we have seen in the previous experiment. But the normal TCP seems to perform a little better then the ILC-TCP in the case when disconnection time is 1 second. After analyzing the trace files of that simulation, the reason for that was found to be the time taken for the state manager to get the information about the link re-connection.

Disconnect Time (Secs)	Transfer Time (Secs)	
	Normal TCP	ILC-TCP
300ms	4.92	4.92(0%)
1s	6.75	6.70 (+0.7%)
2s	7.53	7.33 (+2.6%)
3s	9.53	8.3 (+12.9%)
5s	13.53	10.3 (+23.8%)
6s	13.53	11.33 (+16.3%)
9s	21.53	14.33 (+33.4%)

Table 2 - Comparison of ILC-TCP and Normal TCP; Transfer Size = 0.5 MB Stream; Correspondent Host 3 hops away from the Mobile Host; Intermediate Link Bandwidth = 5 Mb/s.

This problem is specific to ns-2 implementations. But in real implementations, the state manager can get the information more accurately and in time by measuring the signal strengths in the wireless antennas.

4.4. Experiment 4 – Multiple Disconnections

The goal of this experiment to see how much improvement can we get from out TCP in the scenario when multiple disconnections occur. The simulation setup is the same the one used in experiment 3, but this time the disconnection time was fixed to 3 seconds and a 2 MB stream of data was sent from the mobile host to the correspondent host in each of the simulations. Also, results were obtained for different number of disconnections. Results are shown in Table 4 and Figure 9.

We find almost 20-27% improvement in the performance by using ILC-TCP. Figure 9 show that the improvement in performance we obtain by using our approach increases as the number of disconnections increase.

Discon Time (Secs)	Transfer Time (Secs)	
	Normal TC	ILC-TCP
300ms	9.67	9.67 (0%)
1s	10.53	10.8 (-2.56%)
2s	12.53	11.9 (+5.0%)
3s	15.73	12.9 (+17.9%)
5s	15.74	14.9 (+5.3%)
6s	22.13	15.9 (+28.15%)

Table 3 - Comparison of ILC-TCP and Normal TCP; Transfer Size = 0.5 MB Stream; Correspondent Host 3 hops away from the Mobile Host; Intermediate Link Bandwidth = 5 Mb/s; One Link is 0.5 Mb/s; Only one link disconnection considered

Num of Discon	Transfer Time (Secs)	
	Normal TCP	ILC-TCP
1	40.19	32.03 (+20.3%)
2	45.53	35.19 (+22.71%)
3	50.79	38.5 (+24.5%)
4	56	41.5 (+25.89%)
5	61.21	44.6 (+27.13%)

Table 4 - Comparison of ILC-TCP and Normal TCP; Transfer Size = 2 MB Stream; Correspondent Host 3 hops away from the Mobile Host; Intermediate Link Bandwidth = 5 Mb/s.

Num of Discon	Transfer Time (Secs)	
	Normal TCP	ILC TCP
1	92.84	88.26 (+4.93%)
2	98.63	94.54 (+4.15%)
3	110.89	96.65 (+12.84%)
4	117.09	98.38 (+15.98%)
5	127.51	97.53 (+23.5%)

Table 5 - Comparison of ILC-TCP and Normal TCP; Transfer Size = 2 MB Stream; Correspondent Host 3 hops away from the Mobile Host; Disconnection Duration = 1 sec; Total Link Delay between Correspondent Host and Mobile Host = 100ms.

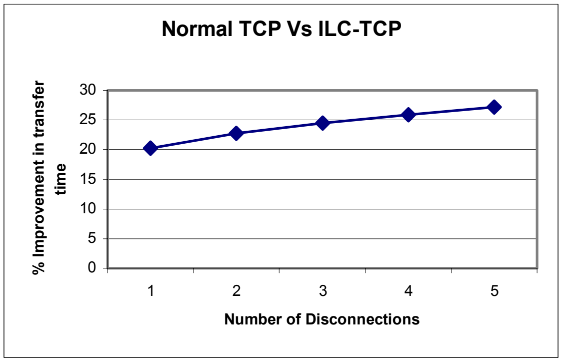


Figure 9 - Percentage Improvement in Transfer (Time Experiment 4)

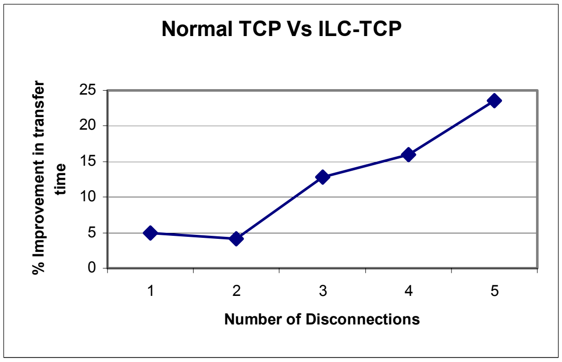


Figure 10 - Percentage Improvement in Transfer Time – Experiment 5

4.5. Experiment 5 – Multiple Disconnections, Increased Link Delays

The goal of this experiment is to see the behavior of our protocol in networks with slow links. The delays of the various links between the mobile host and the correspondent host were increased to 100ms. Also, this time the disconnection duration was only 1 second. The results are shown in Table 5.

In this experiment, we can observe the effect of disconnection on the congestion window. Since the links are slow, it takes more time to grow the congestion window. So multiple disconnections can slow the rate of data transfer considerably. Figure 7 compares the percentage improvement time for different number of disconnections. We can observe that, usually, the performance improvement is more when the number of disconnections is more.

4.6. Experiment 6 – Mobile Host Moving with Different Speeds

The goal of this experiment is to compare the performance of our approach and the normal TCP under mobile environments. We ran different simulations in which the mobile host moves with different speeds. We compare the performances of both in each of those cases. The topology is shown in Figure 7.

A trust delegation can cause a malicious node to be installed, and the compromise may go for a significant period of time. Besides, applying a trust delegation to a reactive routing uses a flooding method so that it is not a scalable approach if the entire network has many nodes.

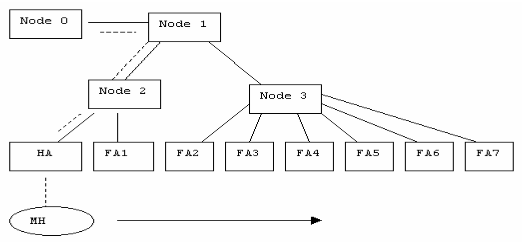


Figure 11 - Topology used in the Experiments

Speed (m/s)	Transfer Time (Secs)	
	Normal TCP	ILC TCP
5	394	301 (+23.6%)
10	474	354 (+25.3%)
20	480	389 (+18.9%)

Table 6 - Overlapping Base Station Ranges

4.7. Case 1:. Non Overlapping Base Station Ranges

In this case, we are considering that the area is fully covered by the base stations. That is, the link level disconnection time is zero. But delay occurs in communication because of the mobile IP registration procedure. The performance comparison results are shown in Table 6.

The results show that the performance is much higher in environments where the mobile host moves with considerable speed. This is because of the frequency of the handoffs and the delay and packet loss caused by them.

4.8. Case 2: Non Overlapping Base Station Ranges

In this case, the base station ranges are not overlapping. We have considered different values for separation of ranges and performance evaluation is done. The results are shown in Table 7.

The results shown in Table 7 show that the normal TCP performs better in the case when the base stations are not overlapping when compared to the case in which the base stations are overlapping. This is explained by the fact that in the latter case, the mobile host undergoes one IP level handoff more than it undergoes in the former case and an IP level handoff introduces more delay than the delay introduced by the link level disconnection caused because of the separation between the base station ranges.

From the Table 9, we find that the mobile host moving at 10m/s performs worse than the mobile host moving at the same speed in the overlapping base station ranges case. In this case, the mobile undergoes the same number of IP level handoffs as in the overlapping base station ranges case. Apart from that, it undergoes several link level disconnections

Speed (m/s)	Transfer Time (Secs)	
	Normal TCP	ILC TCP
5	394	301 (+23.6%)
10	474	354 (+25.3%)
20	480	389 (+18.9%)

Table 6 - Overlapping Base Station Ranges

Speed (m/s)	Transfer Time (Secs)	
	Normal TCP	ILC TCP
5	354.26	280.66 (20.9%)
10	469.19	366.51 (21.9%)
20	480.88	370.93 (22.9%)

Table 7 - Distance of Separation of Ranges = 4m

Speed (m/s)	Transfer Time (Secs)	
	Normal TCP	ILC TCP
5	353.86	294.46 (+16.8%)
10	476.87	371.43 (+22.1%)
20	461.26	362.65 (+21.3%)

Table 8 - Distance of Separation of Ranges = 10m

Speed (m/s)	Transfer Time (Secs)	
	Normal TCP	ILC TCP
5	353.76	290.29 (+17.9%)
10	500.89	377.21 (+24.7%)
20	463.5	373.94 (+19.3%)

Table 9 - Distance of Separation of Ranges = 25m

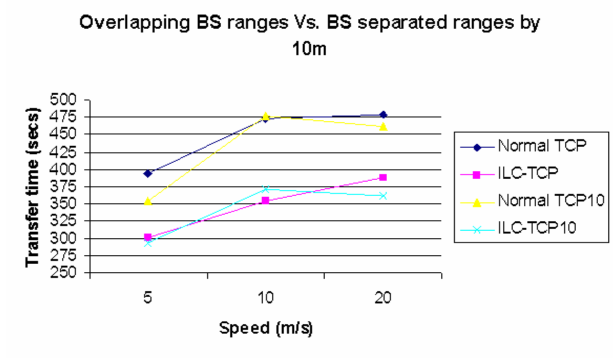


Figure 8 - Most common Topology used in the Experiments

Figure 8 shows the comparison between the overlapping base station ranges case and the case where the base station ranges are separated by a distance of 10m.

4.9. Summary of Experimental Results

After considering a number of scenarios, we conclude that the performance improvement achieved by using ILC-TCP varies from scenario to scenario. The performance improvement increases with an increase in the number of disconnections. Also, if the links are slow, then ILC-TCP can improve the performance of TCP by preventing the congestion window from being reduced in event of disconnection. Also, it has been observed that in the case of mobile host moving at considerable speeds, the performance improvement is more. Also, the results depend on how fast the State Manger gets the information about the changes in the link and in the network

5. CONCLUSIONS AND FUTURE WORK

The proposed ILC-TCP approach improves the performance of TCP in mobile and wireless environments involving frequent, long disconnections and in the environments where a mobile host moves at considerable speeds. It is a true end-to-end approach and does not require any cooperation from the intermediaries like base stations. No change is required at the TCP receiver in the fixed host. Also, the overhead involved is minimal.

However, there are certain open issues and requirements of this approach. Whether or not a mobile host should transfer at the same rate, as before, even after the network level handoff, is an open issue. This is because TCP does not know the properties of the new network and hence transferring at the same rate may cause network congestion. Also, our results are based on simulations done using the Network Simulator – ns2. The protocol needs to be implemented in the network stack of a real operating system like Linux. Also, more number of experiments needs to be conducted to evaluate the protocol performance in many other scenarios. Other interlayer collaboration protocols can be developed to improve the performance of other network protocols, for e.g., the model can be used to develop a protocol to improve the performance of Mobile IP, which involves the collaboration between layer 2 and layer 3 of the network protocol stack.

6. REFERENCES

[Aya95] Ayanoglu, E., Paul, S., LaPorta, T.F., Sabnami, K.K., Gitlin, R.D., "AIRMAIL: A Link-Layer Protocol for Wireless Networks," ACM Wireless Networks, Vol. 1, No.1, 1995, p. 47-60.

[Bak95] Bakre, A., Badrinath B.R., "I-TCP: Indirect TCP for Mobile Hosts," Proceedings of the 15th International Conference on Distributed Computing Systems (ICDCS), May 1995, p. 136-143.

[Bal98] Balakrishnan, H., Katz, R., "Explicit Loss Notification and Wireless Web Performance," Proceedings of IEEE Globecom Internet Mini-Conference, Sydney, Australia, November 1998.

[Bal95] Balakrishnan, H., Seshan, S., Amir, E., Katz, R., "Improving TCP/IP Performance over Wireless Networks," Proceedings of the 1st ACM Conference on Mobile Computing and Networking (Mobicom), Berkeley, CA, November 1995, p. 2-11.

[Bro97] Brown, K., Singh, S., "M-TCP: TCP for Mobile Cellular Networks," Computer Communication Review, Vol. 27, No. 5, October 1997.

[Cac95] Caceres, R., Iftode, L., "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments," IEEE Journal on Selected Areas in Communications, Vol. 13, No. 5, June 1995

[Cho99] Chockalingam, A., Zorzi, A., Tralli, V., "Wireless TCP Performance with Link Layer FEC/ARQ," Proceedings of IEEE ICC'99, June 1999, p.1212-1216.

[Gof00] off, T., Moronski, J., Phatak, D.S., Gupta, V., "Freeze TCP: A True end-to-end enhancement mechanism for mobile environments," Proceedings of IEEE 19th Infocom 2000, p.1537-1545.

[Mon98] ontenegro, G., Dawkins, S., "Wireless Networking for the MNCRS," Internet Draft, work in progress, August 1998. http://www.globecom.net/ietf/draft/draft-montenegro-mncrs-00.html, accessed in April 2002.

[Pen00] Pentikousis, K., "TCP in Wired-Cum-Wireless Environments," IEEE Communications Surveys and Tutorials, Vol. 3, No. 4, Fourth Quarter 2000.

[Per96] Perkins, C.E., Editor, "IP Mobility Support," RFC 2002, October 1996. ftp://ftp.isi.edu/in-notes/rfc2988.txt, accessed in April 2002.

[Pos81] Postel, J.B., “Transmission Control Protocol,” RFC 793, September 1981. ftp://ftp.isi.edu/in-notes/rfc793.txt, accessed in April 2002.

[Rat98] Ratnam, K., Matta, I., “WTCP: An efficient Mechanism for Improving TCP Performance over Wireless Links,” Proceedings of the Third IEEE Symposium on Computers and Communications, p. Athens, Greece, June 1998, p. 74-78.

[Tho97] Thomson, K., Miller, G.J., Wilder R., “Wide-area Internet Traffic Patterns and Characteristics,” IEEE Network, Vol. 11, No. 6, November 1997

[Tsa00] Tsaoussidis, V., Badr, H., “TCP-Probing: Toward an Error Control Schema with Energy and Throughput Performance Gains,” Proceedings of 8th IEEE Conference on Network Protocols, Japan, November 2000.

[Tsa02] Tsaoussidis, V., Matta, I., “Open Issues on TCP for Mobile Computing.” http://citeseer.nj.nec.com/tsaoussidis02open.html.

[Vai99] Vaidya, N., Miten, M., Perkins, C., Montenegro, G., “Delayed Duplicate Acknowledgements: A TCP-Unaware Approach to Improve Performance of TCP over Wireless,” Texas A&M University, Technical Report 99-003, February 1999.