An Infrastructureless End-to-End High Performance Mobility Protocol

Sandeep Davu, Raid Y. Zaghal and Javed I. Khan
{sdavo, rzaghal, javed }@cs.kent.edu
Networking and Media Communications Research Laboratories
Computer Science Dept., Kent State University
233 MSB, Kent, OH 44242

ABSTRACT

Mobile IP is offers disconnection free handoff by assuming availability of infrastructure. It requires intermediate software agents in the network to be deployed ahead of time to circumvent IPs normal mode identity based routing. This infrastructure based mobility management though offers connectivity but incurs significant handoff and tunneling delays along with deployment costs. In this paper we demonstrate an alternate mobility scheme which does not require any such infrastructure and uses only end-point technique and yet provides much faster loss-free handoff. This End-to-End scheme named Interactive Protocol for Mobile Networks (IPMN) neither requires any functional changes to the network layers on the sending and receiving host machines nor an infrastructure in the network. It intelligently performs handoff based on information provided by MAC Layer. The network address change is handled by renewing the existing connections by manipulating the TCP/IP stack at the end-points. Besides, the difference in deployment scenarios, the IPMN offers blazingly fast event based handoff and much faster and simplified transport (no tunneling delay) than MIP. We provide a detail model based performance comparison between the two.

1 INTRODUCTION

Mobile IP (MIP) [1] is being used widely currently to handle mobility in TCP/IP network. In TCP/IP the identity of a node and all corresponding routing is indicated and managed by the same identifier. At application level a connection is identified using the four tuple <source address, source port, destination address, destination port>. A L3 handoff changes network address resulting in stale non-routable connections. MIP addresses this problem by adding an indirection to the routing mechanism in the form of home agent (HA) and foreign agent (FA). In MIP a handoff is detected when a Mobile Node (MN) leaves the present service area and enters a new agent's service area. The difference between the service areas is identified by sensing a form of Agent Advertisements (AA). This requires FA to periodically and continually broadcast a small signal. The next issue is to circumvent the static identity attached routing. This is done by registering the MN to a new foreign agent and updating the same at the HA. To disable the MN's identity based routing a tunnel is created from MN's HA to FA, where the actual IP packet rides inside another packet from HA to the FA. The return path from MN to Corresponding Host (CH) is directly routed through the FA without requiring any intervention of HA.

The solution logically handles the IP mobility, but also shows the following artifacts. The first is related to indirection. This creates a triangular pattern and hence this system is also referred to as triangular routing. Resulting forward path and return path also becomes asymmetric. The second is performance. The mobility detection depends on beacon. The timer based advertisements placed an upper bound on the handoff delay. These delays degrade the transport and application level performance. Too fast a timer creates excessive network overhead, on the other hand a slow beacon (which is used in most cases), delays the entire mobility management and in practice reacts adversely with the upper layer transport timers. Currently, it almost prohibits the use of connection oriented transport such as TCP [17,18]. Besides the above artifacts, MIP also requires infrastructure. It requires a significant mechanism to hide the address change from higher layers. It also dictated the strong need for an infrastructure implying greater deployment and management costs. Specifically it requires new IP layer in MN, HA & FA.

In many scenarios however, the above model of mobility handling is inconvenient. Many are sensitive to triangulated long round trip delay. For example, consider a remote surgery requiring highly reliable and time sensitive data delivery. Many advanced applications require connection oriented transport. All advanced distributed applications fundamentally require reliable connection. Even if they are force to use UDP- they have to anyway replicate the reliability and connection states at the upper level. Also in many scenarios the infrastructure might not be available at all. An example of this is a corporate business having branches spread all over the country. It cannot change the already existing infrastructure to suit the mobility management rather they would like something to work with the already existing infrastructure.

In this paper we demonstrate a mobility solution which is based on only end-point technique. Further the change at the end-points is mostly restricted at application level. It is based on a new end-point network software architecture called **interactive transparent networking** paradigm (InTRAN) [10, 11]. This new end-point network software architecture allows event based access to protocol states

by network layer processes or by L7 processes. The new scheme does not require the battery of pre-deployed FAs in the Internet (thus the need of an infrastructure!). Besides, offering advantage of being infrastructure-less, this also offer major performance advantage. It does not require HA and thus any route indirection, can avoid triangulation, is loss-free, eliminates tunneling overhead (significantly reduced jitter or delay) and above all offer much faster handoff.

Before presenting this new approach, in the following section we first present a brief review of the current mobility handling techniques in TCP/IP framework. Section 3 then explains the new interactive transparent networking paradigm. Section 4 then presents the proposed scheme. In section 5 we show the detail performance model of the scheme. For comparison we also present a model of MIP. Finally in section 6 we present a comparative performance analysis of the proposed scheme in managing selected scenarios.

2 RELATED WORK

Mobility solutions can be categorized in two broad approaches- one which handles mobility fully at the networking layers, hiding any changes in the network structure from the end systems. The others, those handle mobility at the transport layers and involves only the end systems. Below we briefly present these two approaches.

2.1 Network versus Higher Layer Solutions

Mobile IP [1] provided a first effective crack of handling mobility. It is now an IETF standard and most common protocol in practice. Though this solved the problem of handling stale connections, it introduced lots of communication overhead and longer triangular routing paths. Triangulation of routes can be reduced by enhancing Mobile IP with Route Optimization [2] presented by Perkins and Johnson. The CH optimizes the route by maintaining a binding cache for each MN's care-of-address. Based on this information a direct tunnel from CH to MN is created defying any further involvement of HA. MN needs to update CH of its current care-of-address explicitly during each handoff, accruing already complex MIP. Hint based handoffs in [3] use triggers from L2 as hints to L3 about an impending handoff. This eliminated the cost and complexity of advertisement based movement detection. [20] also uses assistance form link layer to perform fast Mobile IP handoff through MAC bridging. The RAT (Reverse Address Translation) architecture [4], based on the network address translation (NAT) protocol, uses packet re-direction service between CH and MN to support IP mobility. However, both the techniques require functional enhancement of the layers involved.

Higher layer solutions mostly involving TCP handled mobility differently, using the split connection approach. Indirect-TCP [5, 9] splits the wired and wireless parts of the connection using the Base Station (BS) as a common end point. The state information of the connection is transferred between the old BS and new BS transparently during handoffs. MSOCKS [8] achieves connection redirection using split connection proxy. TCP-R[12] is based on an idea- same as ours, renewing the connection to handle the new IP address. But their scheme bases its idea on the conventional timer based approach and hence lacked the robustness and scalability. Freeze-TCP[6], though does not handle mobility directly, aids the TCP performance by freezing the connection during periods of disconnection.

2.2 Our Approach

In compared to the above approaches the solution we propose has several distinguishing characteristics. First the mobility reactions are *event based* as opposed to *periodicity dependent*. Secondly, the solution does not require any significant overhauling of the existing protocols rather works in L7. The InTraN enables networking layer protocol events to be orderly and securely subscribed and responded at L7 without requiring functional modification of the existing protocols- with much simpler protocol meta-engineering. This is an interesting diversion from previous thinking that application layer solutions may slow down the overall performance. We show by modeling and analysis that the premium is much less than suspected- almost negligible. On the other hand- since mobility tracking and corresponding reconnection responses can be generated at application layer- its deployment is much easier and also the strategies can be made much more sophisticated and powerful. Indeed the scheme we show incorporates and extends the strengths of several previous approaches. It could thus offer a dramatically improved handoff performance- far out weighing the cost.

3 INTERACTIVE TRANSPARENT NETWORKING

The framework assumes a transparent encasement of protocol functionality by making a subset of its internal protocol states and events visible to its service subscriber layer. It provides means to allow programs in upper layers to subscribe listed events in lower layers and be notified when these (state transitions) occur. Upon notification, subscriber processes can further pull up the listed service states. Typically an application program can register an event handler child process called *Transientware* to handle a specific event. The transientware is an application level programmable process and thus can perform various adaptive intelligent actions at L7. Figure-1 shows the general architecture of an example interactive version of

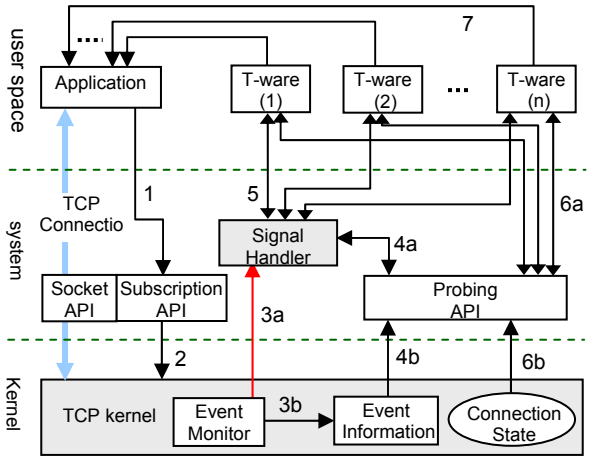


Figure 1. TCP interactive extension and API.

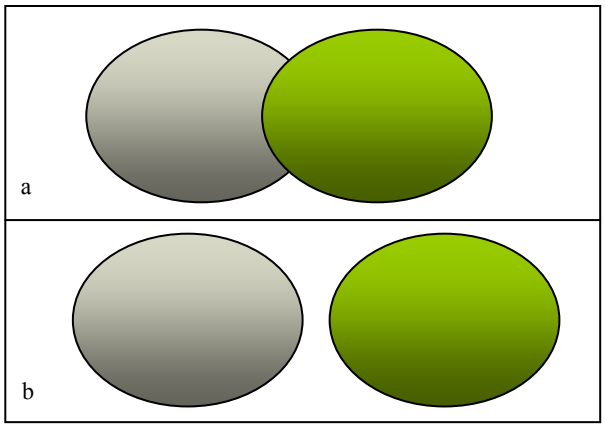


Fig 2 Cell Boundaries a) Overlapping b) Non-Overlapping

TCP. Upon opening the socket, an adaptive application may bind a Transientware module to a designated TCP event by subscribing with the kernel. This is represented by arrows 1 and 2 in figure-1. The binding is optional; if the application chooses not to subscribe, the system defaults to the silent mode identical to TCP classic. When the event occurs in TCP, the kernel sends a signal to the upper layers (3a) and at the same time it saves the event information (3b). A special handler catches the signal and probes the kernel for the event type (4a, 4b), The handler then invokes the appropriate Transientware module to serve the event (5). A Transientware module can also use the probing API to access the kernel state (6a, 6b) or to pass some information to the subscriber application itself (7). A freeBSD version InTraN kernel and corresponding interactive protocols including iTCP has been recently implemented [10, 11].

4 INTERACTIVE PROTOCOL FOR MOBILE NETWORKS (IPMN)

4.1 Overview of Wireless Environment

In a wireless scenario L2 handoff is initiated by the MN when the Signal/Noise ratio and Signal strength of the current Access Point (AP) falls behind a certain threshold. To have a better understanding of the architecture let us first explain briefly the two wireless scenarios existing and how our architecture robustly handles both while performing a L2 handoff.

When an MN receives service from more than one access point at any point of time then their areas of coverage are overlapping, which generally happens at the boundaries of the cells and this is called *Overlapping cell boundaries.* Overlapping cell boundaries as shown in figure 2(a) have better performance in our approach as applications probing link layer yields information about

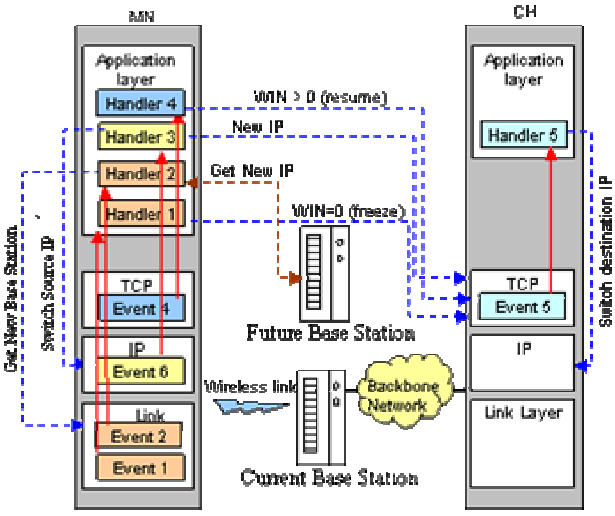


Figure 3. IPMN–Architecture and event sequencing.

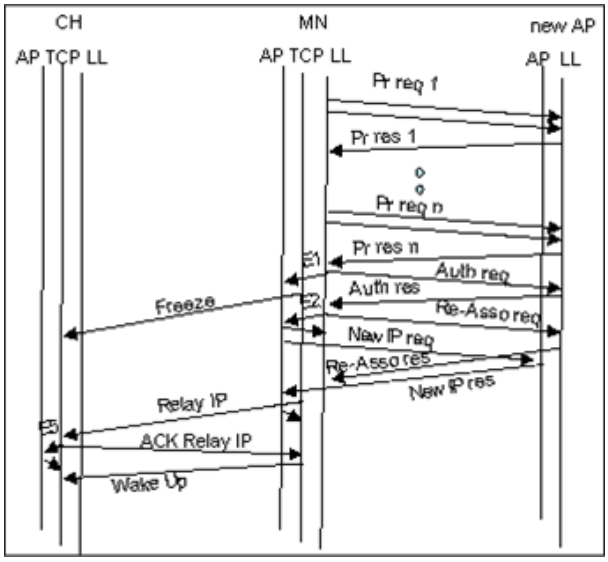
the next possible AP that would serve MN after the handoff. This information aids faster L3 handoffs which would be discussed in detail later in the section.

At times MN may encounter temporary layer2 disconnections, encountered due to disjoint coverage area as shown in figure 2(b). This type of coverage is called *Non-Overlapping cell boundaries*. Probing the link layer will not yield any useful information.

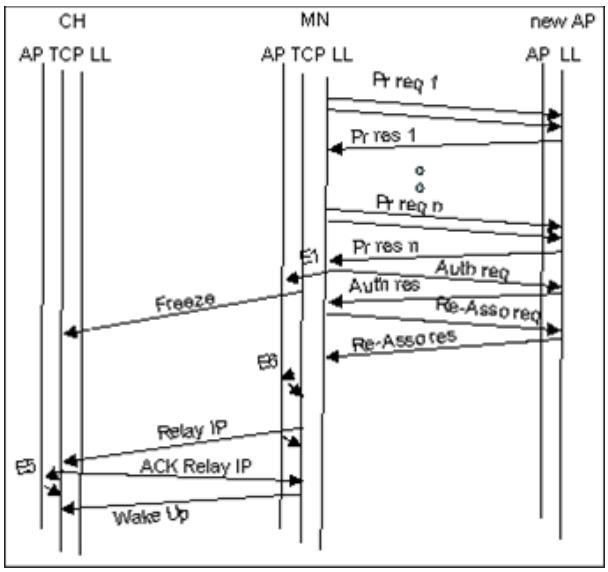
4.2 Architecture

In this section we will discuss in detail the architecture of our approach which we have named Interactive Protocol for Mobile Networks (IPMN). This event-driven model has network layer events (like congestion, retransmission, handoff, etc.,) that the application can subscribe so that the application can be notified upon occurrence of that event in the network layer. L2 handoff consists of three steps a) probing, b) authentication and c) re-association explained in detail in the next section. The handoff procedure is

started when the application receives a L2 trigger notifying an impending handoff. Application would then let the TCP layer advertise a zero window temporarily interrupting the data flow. Another signal triggers the application to notify the completion of the authentication process in L2 layer. Here the overlapping scenario and non-overlapping scenario follow different approaches. In overlapping case probing the link layer



(a) Overlapping Cell Boundary



(b) Non-Overlapping Cell Boundary

Figure 4 : Timing Diagram for IPMN Architecture

now would give information about the Future Access Point (FAP). This information is used to registers an IP address for MN even before the completion of L2 handoff. After getting the address the same connection is renewed by manipulating the TCP/IP stack at both the end points. The application unfreezes the connection after manipulating the TCP stack. Since IP address is attained almost in parallel with L2 handoff we reduce the handoff latency by a great extent. In non-overlapping case the system waits until it gets an IP address normally through DHCP. After obtaining the new IP the rest of the process follows as explained earlier. Freezing the connection during handoff avoids congestion control algorithms and increasing the performance of the transport layer.

We are trapping 4 events, two state variable update from three layers and corresponding 5 transientware processes in order to provide mobility support. Figure-3 gives the architecture and event sequencing of IPMN. Probing in L2 is initiated when the Signal-to-Noise Ratio (SNR) of the current AP falls below a certain threshold. At the MN, the subscribing application (L7) is notified of an impending handoff after probing (event 1) is completed. When the event is received at the L7, a Transientware module (Handler 1) is activated immediately; this module makes a system call *Freeze* which lets TCP advertise a zero window temporarily interrupting the data transmission. Freezing the connection during handoff avoids TCP congestion control algorithms by invoking the persist timer. In overlapping case successful L2 authentication (event 2) triggers L7 about the same. Handler2 then probes L2 to identify the cell boundary condition (if the probe yields at least one AP then the cell boundaries are overlapping. Here we assume that the current connection is still valid) and extract information about the Future Access Point (FAP). MN uses this information to register with Future AP and attain an IP address (L3 handoff) in parallel with L2 handoff. Handler3 upon initiation transmits the IP address to the CH through a system call-*relayIP* and also manipulates the '*SourceIP*' field in the TCP substrate. A special TCP segment with option = SWITCH_IP' is sent to the CH. On reception of this special segment at CH (event 5), Handler5 is activated, which manipulates the '*DestinationIP*' field in the TCP substrate with the newly received IP address. MN on reception of an 'ACK' for 'SWITCH_IP' segment (event 4), invokes Wakeup Handler allowing MN to resume the communication by advertising a non-zero window to the CH. In non-overlapping case MN waits until L2 handoff is completed and a DHCP IP address (event 6) is assigned. Handler3 is then invoked which probes IP layer to get the IP address. The same process as explained in the previous case follows when RelayIP Handler is initiated. The timing diagrams of both overlapping and non-overlapping cases are depicted in the timing figures 4(a) and 4(b) respectively. E1, E2 and E5 are events that trigger the handlers in overlapping case while E1, E5 and E6 trigger the respective handlers in non-overlapping scenario.

5 FORMULATION OF HANDOFF LATENCY

We divided the handoff process into two subsets and formulated the latency for each of the subset – Layer2 handoff and Layer3 handoff. Depending on the cell

boundary condition the latency would change yielding different results.

5.1 Link Layer Handoff Latency

Link layer (802.11) handoff can be classified into 3 categories according to [13, 14, 18] a) Probing, b) Authentication and c) Association/Re-Association. Each of these categories would constitute a delay parameter to L2 handoff latency.

Probing: Probing is done by a MN to determine the availability of a wireless network and perform a L2 handoff. A MN sends a probe request and waits for a reply governed by two timers. *MinChanneltime*(T_e) – minimum time that a MN needs to wait after sending a probe request. If there is no activity in the channel for *MinChanneltime* then the channel is considered empty. *MaxChanneltime* (T_u) – time for MN to get a response from a used channel. In [14] Velayos and Karlsson et al., tried to estimate the various 802.11 handoff delays and tried to optimize the handoff latency. If *u* and *e* are the number of used and empty channels respectively, then the total time to probe the network called the Scan Delay S_d can be computed as in Eq 1(a).

$$S_d = u * T_u + e * T_e \quad \text{.....1(a)}$$

Furthermore both T_u *and* T_e send probe requests twice so as to minimize the possibility of these probe packets being lost. Values of T_u *and* T_e are directly dependent of the transmission delay T_d governed by the load of the channel.

$$T_u = 2 * T_d + MaxChannelTime$$
$$T_e = 2 * T_d + MinChannelTime \quad \text{.....1(b)}$$

The transmission delay T_d can be modeled as a function (Eq 1(b)) as the contention for the channel and retransmission (both being the critical parts of T_d) always follow the randomized binary exponential backoff. Equations 1(a), 1(b) and 1(c) will give the total probing delay.

Authentication: This is the process that validates whether the MN can use the services of an AP.

$$T_d = \int_0^{\infty} f_t(t)dt \quad \text{.....1(c)}$$

If a mutually acceptable level of authentication has not been established between an AP and MN then, association/re-association would not be established. The MN after probing the channels and obtaining the list of Access Points in range tries to prioritize them based on numerous criteria. The MN will send an authentication request to the first entry in the list of prioritized APs and waits for an authentication reply.

$$A_d = \sum_{i=1}^{n} (2 * T_d + P_{Ta}) Where \ \ 1 \le n \le u \quad \text{1(d)}$$

If the reply indicates a successful authentication then re-association procedure is started. If there is an unsuccessful authentication then the same procedure is repeated with the next entry until successful authentication.

Re-Association: After successful authentication the MN's state information is transferred from the old AP to new AP. Re-association is helpful in knowing the current attachment point of the MN as it is moving from AP to AP. Re-Association delay (RA_d) is the request/response sequence *RTT* plus the time to exchange state information between the old and the new AP(P_{Tr}).

$$RA_d = 2 * T_d + P_{Tr} \quad \text{.....1(e)}$$

5.2 Handoff Latency in Higher Layers.

There are 5 system calls that are used to invoke handlers. The latencies of these system calls range from 10µs to 100µs. Each system call will have latency Syc_d given by a random variable X between the aforementioned intervals.

$$Syc_d = X\mu s \quad where \ \ 10 \le X \le 100 \quad \text{2(a)}$$

In the same way the triggering latencies are between 5 and 20µs which is represented as Tg_d and governed by a random variable Y ranging form 5 to 20.

$$Tg_d = Y\mu s \quad where \ \ 5 \le X \le 20 \quad \text{.....2(b)}$$

MN would directly contact the Future AP to pro-actively register itself and get an IP address. This delay termed as proactive registration delay(PR_d) constitutes of the RTT between the MN and the PAP through the present AP $T1_d$.

$$PR_d = 2 * T1_d \quad \text{.....2(c)}$$

Where $T1_d$ is

$$T1_d = \int_0^{\infty} f1_t(t)dt \quad \text{.....2(d)}$$

We have observed roundtrip times for various demographic distances up to 400 miles and observed the fact that it is practically impossible to have access points whose demographic distance is more than 400 miles. This implies that RTT latency would fall between 100µs to 10ms depending on the distance. Table 1 gives the handoff delay in both overlapping and non-overlapping cases.

From Eqs 1(d), 1(e), 2(a) and 2(b) we have the delay of Overlapping cell boundary given by Equation (3).

$$T_0 = S_d + Tg_d + \max(A_d, Syc_d) + 2 * Tg_d + \max(RA_d, 3 * Syc_d + PR_d + Ep_d) + Syc_d \quad 3$$

Where *max(a1, a2)* gives the maximum value of *a1* and *a2*, and represents the overlapping latencies. EP_d is the

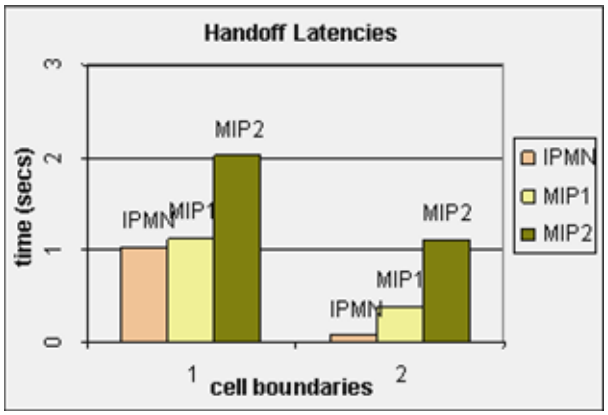


Figure 5: Handoff latencies of Mobile IP and IPMN. MIP1 has an AA lifetime of 100ms MIP2 1s.

Round Trip Time between end points of the connection governed by Relay Handler.

In Non-Overlapping cell boundaries the probe delay S_{dno} prolongs until it hears from at least one AP, also taking into account the time to traverse between the coverage areas. From Eqs 1(d), 1(e), 1(f), 2(a) and 2(b) we have the total handoff delay for non-overlapping cells T_{no} is as given in Equation (4).

$$T_{n0} = S_d + Tg_d + \max(A_d, Syc_d) + 2 * Tg_d + RA_d, 4 * Syc_d + PR_d + Ep_d + Syc_d \quad .....4$$

Performance evaluation is then done in by comparing our IPMN handoff model with the Mobile IP handoff model. In order to do this we also modeled Mobile IP handoff latencies.

Similar efforts have been realized in modeling Mobile IP's handoff latency. MIP handoff delay H_d is given by Equation (5)

$$H_d = L_d + M_d + R_d + IT_d \quad .....5$$

Where L_d is the total Link Layer delay modeled earlier. M_d is the Movement Detection delay caused by MIP consisting of the outgoing and incoming agents beacon timers. R_d *and* IT_d are the registration and tunneling delays respectively. Detailed description on modeling of Mobile IP can be found in the technical report[16].

6 PERFORMANCE EVALUATION

We have simulated the models for IPMN and MIP. Since most of the delay for IPMN is the signaling costs and system calls, it incurred only a very little handoff delay. Figure 5 shows the latencies of non-overlapping (right side) and overlapping (left side) cell boundaries. MIP1 and MIP2 are versions of MIP that use different agent advertisement life times. MIP1 uses a life time of 100ms and MIP2 uses 1 sec. Handoff delay for MIP1 and IPMN is in the order of 100ms. Though MIP1 promises better performance, it actually degrades the performance due to bandwidth monopolization. As evident in figure 2, MIP2 fails to deliver real-time voice and video traffic. MIP2 and IPMN have the handoff delay difference up to 2 seconds.

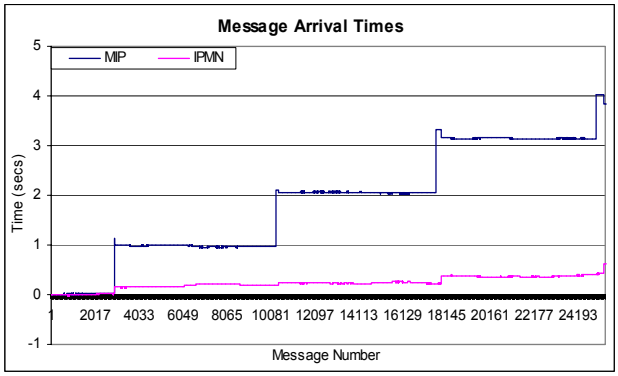


Figure 6: Message arrival times in an ideal channel without BER and congestion.

In developing the simulation we assumed that each message is of size 1KB and the total transfer size is 30MB. We also assumed the expected packet inter arrival time of 8 ms as bandwidth consideration. We also scheduled a handoff every one minute and observed the packet arrival times. Figure 6 gives the difference between IPMN and MIP message arrival times. MIP experiences a delay of around 1 second at each handoff while IPMN only has a delay of 70 milliseconds. A wireless environment is prone to higher Bit Error Rates (BER) which can be even upto10% of the transmission rates. BER can be modeled as a Log-Normal distribution [21].

We have also considered delay due to congestion in our model. This was simulated differently for different congestion constraints. Many factors such as BER in the physical layer, or cross traffic in the network layer, causes congestion. Congestion incurred due to BER is induced as an exponential distribution. Congestion caused due to cross traffic is explained in detail in later sections separately for various traffic patterns. Various traffic patterns was generated using the models given by NetSpec [19]. We subtracted each packet's arrival time from its expected arrival time. This would be the delay due to either BER, congestion, handoff or a combination of these factors. To give a better understanding we have also plotted the Normal case (where there is no handoff) along with the Mobile IP and IPMN cases. The delay in this normal case is only due to BER and/or congestion.

6.1 Ftp Cross Traffic

Figure 7 shows the performance analysis of MIP IPMN to that of Normal case. Here the congestion is generated due to ftp transfers suddenly initiated while the data is being transferred to the MN. Ftp traffic and many other traffic patterns follow a two step model. The first is the Packet inter arrival times or packet size called Session Level. The other is the Session duration or Session inter-arrival time called Call Level. We have used the traffic modeling of NetSpec [19] a traffic generator developed by

University of Kansas. In Netspec Ftp follows an exponential distribution for Session inter-arrival times and a log-normal distribution for the item sizes in Session Level. We used these density functions to generate cross traffic by incorporating these delays in RTT measurements. We used $\lambda = 0.05$ for exponential distribution *mean* = *6* and *standard deviation* = *2* for Log-Normal distribution. The burst indicated in figure 7 is the delay caused in packet arrival times due to ftp cross traffic. This has been induced as congestion into the packet arrival times in all the three cases. As evident, in MIP this delay compounds whenever there is a handoff. The handoff in MIP causes congestion in higher layers and cumulatively worsens the packet arrival time. This is evident from the figure 7. After every handoff there is an exponential increase of packet arrival times. This tends to be linear as the transmission progresses. Our metric was packet arrival times with respect to expected packet arrival. MIP triggers TCP

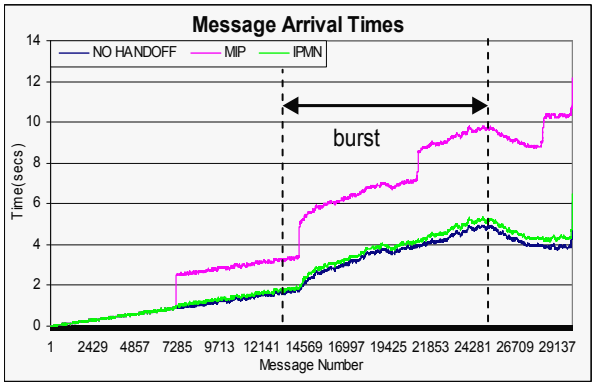


Figure 7: Message arrival times 10% BER and congestion due to ftp cross traffic.

congestion control every time a handoff is initiated. In contrast, IPMN does not have this delay because of the explicit event notification and the persist timer. Overall MIP takes around 5 seconds more than that of IPMN to complete the data transfer.

6.2 Voice Cross Traffic

Voice has a Constant Bit Rate (CBR) traffic characteristic and its typical sampling rate is 8 kHz and each sample consisting of is 8 bits. This gives the standard bit rate of 64 Kb/sec for acceptable voice quality. Call inter-arrival times are modeled to be exponentially distributed. Figure 8 shows the performance when the congestion is due to background voice call. We have used $\lambda = 0.00333$ and incorporated the congestion factor in packet arrival times. As observed in figure 8 MIP has lot of performance degradation due to cross traffic. It also suffers a ripple effect and the packet arrival times tend to increase after every handoff. Handoff also has delay induced as

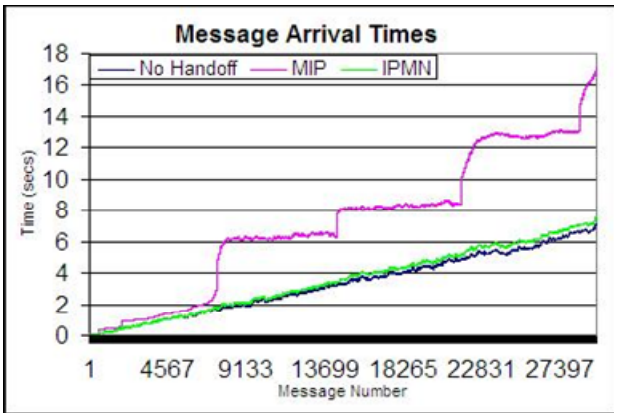


Figure 8: Message arrival times at MN with 10% BER and congestion due to voice cross traffic.

congestion into packet arrival times. The difference in the packet arrival times of MIP and IPMN is almost 10 seconds in this case which gives a clear indication of how MIP would fail delivering voice traffic. IPMN on the other hand has no congestion buildup due to handoff. Compared to IPMN is again almost linear and smooth. MIP cannot handle real time traffic like voice and video while IPMN delivers the same efficiently with very little or no quality degradation.

6.3 WWW Cross Traffic

Interactive traffic has the document size as a Power Law or Pareto distribution which is heavy tailed. The probability density function and cumulative distribution function of a Pareto distribution are given by eq 18. Since this is a heavy tailed distribution WWW possesses self similarity in network traffic. Call Level of the WWW traffic has a mean request time of 5.75 and can be modeled as a homogeneous Poisson distribution over one hour period implying that the inter-request time is an exponential model.

We have generated the distribution using the probability density function and the value of $\alpha=0.45$. This delay was also introduced as congestion into the packet arrival times in between the data transfer shown in figure 9 as burst. MIP has longer handoff jumps and as a result

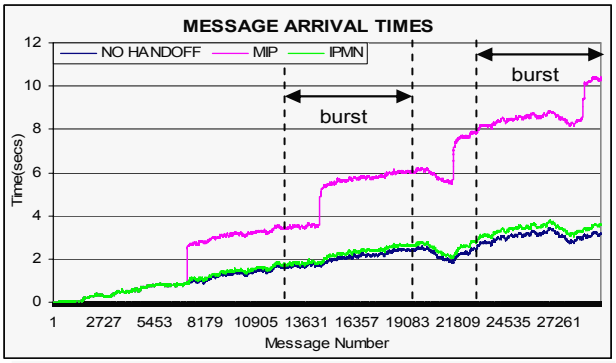


Figure 9: Message arrival times at MN with 10% BER and congestion due to WWW cross traffic.

introduces its own congestion along with the heavy tailed nature of cross traffic. This leads to longer arrival times of packets and the delay creeps up as every handoff will add more and more latency in packet arrival times into WWW interactive traffic. MIP takes 7 seconds more than IPMN for the same data transfer.

7 CONCLUSION

In this paper we have presented an infrastructure-less high performance mobility protocol which uses explicit End-to-End notification mechanism to handle mobility. This eliminated the distributed nature of MIP's movement detection and hence the need for a pre-deployed infrastructure. Thus MIP requires new software at one end-point (MN), one new entity at MN's traditional base station from which it gets identity (HA) and also a battery of network deployed entities (FA). Copies of FA have to be available ahead of time in all parts of the Internet where the mobile node might move. In comparison, the IPMN does not require HA, does not require the battery of pre-deployed FAs. Instead it requires the sending end-point to be mobility aware. MIP does not need this. Naturally, there will be mobility scenarios where one will be more appropriate than the other.

When both options are feasible it is worthwhile to consider the tradeoff. Tradeoff for IPMN is that it offers much greater performance on several counts. Besides, the difference in deployment scenarios, the IPMN offers blazingly fast handoff (based on local event) compared to MIP (based remote timer/beacon). It also uses simplified routing. A packet is directly routed in both directions (does not have to go through HA). We have found that just the hop count is reduced approximately by half. There is also no tunneling (packing/unpacking) delay/jitter added to packets. With the elimination of infrastructure it also drastically reduces the deployment and maintenance costs. The performance advantage is likely to make IPMN a candidate technology for connection oriented mobility. This is currently very difficult on MIP.

REFERENCES

[1] Perkins C., "IP Mobility Support," RFC2002, IETF, Oct 1996.

[2] Charles Perkins, David B. Johns "Route Optimization in Mobile IP", IETF, February 1999

[3] Fikouras N. and Görg C., "Performance Comparison of Hinted and Advertisement Based Movement Detection Methods for Mobile IP Hand-offs," In Proc. of the European Wireless 2000, Dresden, Germany, September 2000.

[4] Singh R., Tay Y., Teo W., and Yeow S., "RAT: A Quick (And Dirty?) Push for Mobility Support," 2nd IEEE Workshop on Mobile Comp. Systems and Applications, pp. 32, Feb. 1999.

[5] A. Bakre., and B.R. Badrinath., " Handoff and system support for Indirect TCP/IP, Proc. Second Usenix Symp. On Mobile and Location –Independent Computing 1995.

[6] Goff T., Moronski J., Phatak D., Gupta V., "Freeze-TCP: A True End-to-End TCP Enhancement Mechanism for Mobile Environments," INFOCOM'00, Tel-Aviv, Israel, pp. 1537-1545, 2000.

[7] Gustafsson E., et al. "Mobile IPv4 Regional Registration" draft-ietf-mobileip-reg-tunnel-05, IETF, September 2001.

[8] MALTZ D., AND BHAGWAT P., "MSOCKS: An architecture for transport layer mobility". In *Proc. IEEE Infocom '98,* March 1998".

[9] Bakre A., and Badrinath B.R., "I-TCP; Indirect TCP for Mobile Hosts", In Proc. Of 15th International conference on Distributed Computing Systems, May 1995.

[10] Khan J., Zaghal R., and Gu Q., "Symbiotic Streaming of Elastic Traffic on Interactive Transport," IEEE ISCC'03, Antalya, Turkey, July 2003.

[11] Khan J. and Zaghal R., "Protocol Modeling with Transparent Networking", CCCT'04, Austin, TX, August 2004.

[12] Funato D., Yasuda K., and Tokuda H., "TCP-R: TCP Mobility Support for Continuous Operation", Proc. International Conference on Network Protocols, 1997.

[13] Mishra A., Shin M., Arbaugh W., "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process," Dept. of Computer Science, University of Maryland, technical report number CS-TR-4395.

[14] Velayos H., Carlson G., "Techniques to Reduce 802.11b MAC Layer Handover", Technical Report, April 2003

[15] Perkins C., "Mobile IP, Design Principles and Practices," Addison Wesley, 1998.

[16] Davu S., "Handoff Model of Mobile IP", Technical Report, Kent State University, 2005.

[17] Postel J.B., "Transmission Control Protocol", RFC 1793, September 1981.

[18] Carvalho M., Aceves J., "Delay Analysis of IEEE 802.11 in Single-Hop Networks", ICNP, November 2003.

[19] Lee B., Frost V., "Wide Area ATM Network Experiments Using Emulated Traffic Sources",Technical Report, Jan. 1998.

[20] Yokota H, et al., "Link Layer Assisted Handoff Method over Wireless LAN Networks," Proc. of MOBICOM '02, Sept. 2002.

[21] Carvalho M., Aceves J., "Delay Analysis of IEEE 802.11 in Single-Hop Networks", ICNP, November 2003.