# A Method for Compensation of TCP Throughput Degrading During Movement of Mobile Node

Amin Salih Mohammad, and Moayad Yousif Potrus

***Abstract***—The tremendous growth of Wireless internet users in recent decade has triggered the need of efficient mechanisms and protocols which integrate the wireless with wired networks in term of data communication. In this case, the existing infrastructure of wired internet with its quality of services can be made available to wireless connected users. However, Wireless network has very different characteristics as compare to wired network which causes the TCP inefficient performance over heterogeneous environment of wired and wireless network. In this paper, a compensation method for TCP throughput degrading during moving of mobile node in wireless networks is proposed. It allows the determination of handoff event beginning by using link layer. The link layer will message transport layer about handoff events. The proposed method was compared with other techniques such as SNOOP, TCP & MCP, M-TCP and Delay & Dupacks. The method yields a better performance in regards of size of congestion windows and recovery time**.**

***Keywords—***Handoff, TCP (Transmission Control Protocol), floating window, Mobile IP

## I. Introduction

MODERN wire networks are sufficiently reliable in terms of data transfer, but wireless networks, as a rule, experience high level of data transmission errors compared to wired networks. Such level of errors has negative impacts on network application throughput. With explosive growths in wireless services and their subscribers, as well as portable and affordable computing devices; it is natural that supporting user mobility in the Internet is a hot and exciting issue that has attracted extensive efforts. As a result of these intense efforts, since early 90s, Mobile-IP (Perkins, 2002) has rapidly matured to a stage where it is being proposed as a standard by the IETF. Now that the basic mobile IP protocol is more or less standardized, researchers are beginning to focus on performance enhancing mechanisms at all layers of the networking stack in order to deliver high performance at the end-user level.

TCP (Postel, 1981) is a vital component of the Transport layer of the Internet protocol suite. It is intended to provide connection oriented reliable service over an underlying unreliable network. Therefore, TCP has received a lot of attention and fairly large number of researchers that tried to optimize and improve TCP for different environments. These environments are characterized by heterogeneous sub networks with widely different bandwidths and latencies (for instance TCP over wireless links, satellite links, slow serial links, etc.) (Bakre & Badrinath, 1995).

Figure 1 shows the tree type of errors which actually happen in wireless networks.

## II. Theoretical background and problem definition

Reliable transport protocols, as TCP, operate well in usual wire networks when the packet losses are mainly caused by congestion. Wireless networks, in turn, experience significant level of packet losses due to transmission errors and diverting of a node. TCP reacts on such error causing events by incorporating congestion avoidance algorithm that leads to degrading of the overall throughput. TCP assumes that any packet loss is merely caused by congestion where it is true in infrastructure with reliable links. However, in wireless environment, the losses are most frequently caused due to the following reasons:

1- High degree of information distortion in wireless links.
2- Temporal disconnections due to signal level decay (or other link errors) or movements of a node.
3- Diverting errors - handoffs (Own & Singh, 1997).

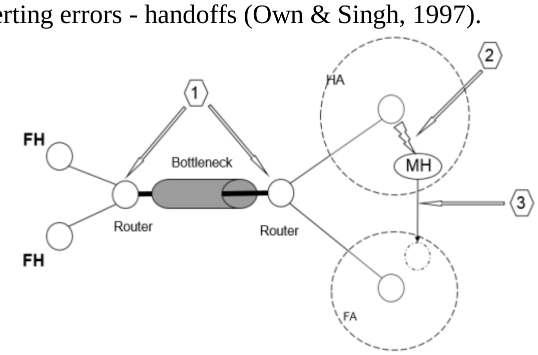



Fig 1 Type of Errors and their place of origin in wireless network, with (1)Congestion, (2) The inherently higher bit error rates of the wireless links, (3) Temporary disconnections(hand off).

For reliable data transfer, TCP uses a policy of retransmissions for lost data packets. Such approach is neither applicable nor useful for wireless and mobile networks because the main reasons that lead to errors and effectiveness

Amin Salih Mohammad, Salahaddin Uni., College of Engineering, Email ID- kakshar@gmail.com )
Moayad Yousif Potrus, Salahaddin Uni., College of Engineering, Email ID- moayad_75@yahoo.com )

degrading are bandwidth limitation congestion, link errors and mobility.

Congestion in data link occurs when the amount of transmitting data exceeds link bandwidth. In this case, TCP decreases the floating window size in order to reduce the congestion, and then calls timeout to resend lost packets.

Wireless networks, due to their structure features, increasingly subject to transmission errors or link errors, compared to wire networks. The loss of few packets from the transmitting window turns TCP to use fast retransmit/fast recovery algorithms. When the loss rate is higher, the protocol waits for a timer to expire, and then turns to slow start algorithm. In this case, we consider the third group of losses – diverting errors (in according to the above classification).

When a mobile node is moved out from the coverage area of a base station, it cannot receive packets from a source, and, therefore, all the packets addressed to the mobile node will be deleted. To solve such problem there is mobile-IP procedure that implements handoff principle. It allows to assign a new address to the mobile node and to register it in the Home Agent (HA), when the node enters coverage area of another base station (Bakre & Badrinath, 1995). At the same time, all the packets sent before such registration, are lost. To better illustrate the third case above, it should be noted that mobility is distinct from wireless connectivity. For instance, a user working in the office on a notebook wants to move (with the notebook) to a laboratory or a meeting room at the other end of a building or in the next building, where the IP addresses can be on different subnets or possibly across one or more firewalls. FTP, Telnet sessions and other connections can certainly remain alive for a few minutes which might be taken to go from one end of a building to another. The idea behind mobility is that such open connections should be retrieved seamlessly despite the move and change of the underlying IP address.

Figure 2 depicts situation before beginning of handoff event, when the packets are transmitted to mobile node MH by HA and FA agents.

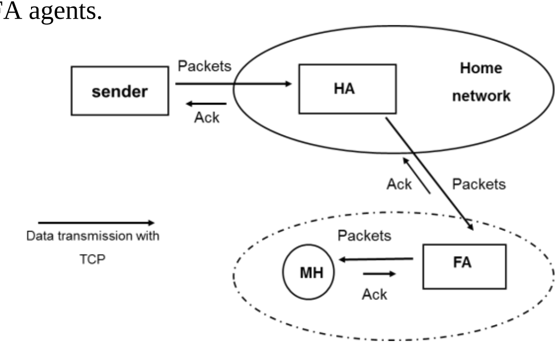



Fig 2 Process of handoff between two cells

Wireless networks are based on Mobile-IP protocol that maintains permanent address for a host to assure the stability of data transfer sessions (Perkins, 2002). When a mobile node leaves the coverage area of stationary host, its IP address becomes expired. Until the node is in the same coverage area, the data are transmitted in a standard way, but when the node moves to another coverage area, it transfers the new IP into HA using FA (re-registering process). After such procedure, all the packets arrived at HA that will be sent to FA, contain the new IP addresses.

When a node is moved between two areas, it receives a service message from FA, and requests FA to register the new IP address in HA. At the next step, registering request is transmitted from FA to HA, which executes registering process (end of handoff event). In this case, all the data packets sent to the node during its re-registering process are lost, and congestion avoidance algorithm should be started. Then after, all retransmitted packets arrive to a mobile node (MH) through HA and FA which trigger slow start algorithm (due to the losses). This process is illustrated in Figure 3. If the packets are lost during handoff, TCP starts congestion avoidance algorithm that leads to decreasing of transmitted packets number per time unit and increasing of retransmission timeout (RTO) value. Such situation negatively impacts on TCP effectiveness through the following factors:

- Long delays during transmission: If there are areas with no coverage and MH moves fast, then the time of its staying within single area is sufficiently short; therefore, until the end of a handoff, MH moves to another coverage area. It results in timeouts and RTO value is increasing.
- Slow data recovery: After expiration of RTO timer, during handoff event, size of floating window is decreased to its minimal value.

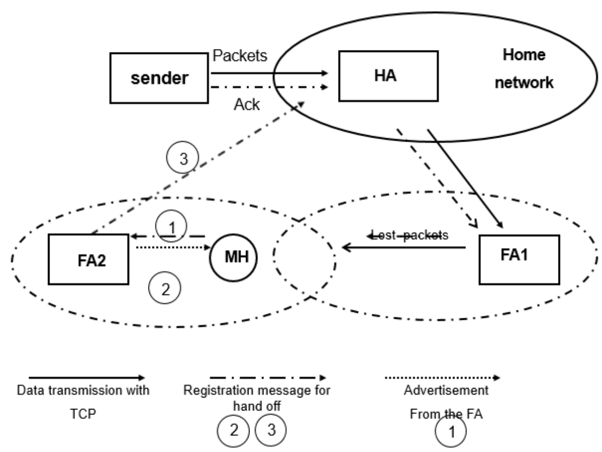



Fig 3 Retransmission of data to MH through FA2

In all cases, the receiver side controls the data flow which depends on the basis of desired amount of data from sender-TCP floating window size. However, if acknowledgement packet is lost, a sender and the receiver wait for each other. The receiver waits for data packets (since it sent nonzero window to sender), and sender waits for updating of the window that allows to continue transmission. To avoid such situation, sender uses special timer. In according to the timer, the sender periodically requests the receiver window size increment. Segments used in such requesting are called window probes (Balakrishnan, Seshan, & Katz, 1995).

The goal of Congestion Coherence (CC) method (Vaidya, Mehta, Perkins, & Montenegro, 2002) is in defining the cause of loss (congestion, transmission error, link error, or diverting) with further applying an appropriate recovery policy. This method, in contrast to other TCP methods, which recognize congestion by packet losses, uses explicit congestion notification (ECN) policy to decrease losses number. The disadvantage of CC method is that, it does not considering neither the movement of a node between coverage areas nor the losses due to handoff.

Currently there are various methods under development that can reduce negative impact of non-congestion losses on throughput of TCP in wireless links. Such research can be divided into the following groups concerning:

1. Split connection;
2. Link layer (Postel, 1981);
3. TCP modifications;
4. Interlayer solutions.

## III. Strengths and Drawbacks of Existing Solutions

There are many major factors (not necessarily in the order of importance) that should be considered in assessing any TCP enhancement scheme (Sun, Wen, & Guo, 2008). One of the main considerations is inter-operation with the existing infrastructure. To realize this goal, ideally, there should not be any change required at intermediate routers or the sender because these are likely to belong to other organizations, making them unavailable for modifications. All approaches that split the connection into two parts (this includes all the schemes mentioned in the subsection above, except (Balakrishnan, Seshan, & Katz, 1995) and (Gerla, Ng, Sanadidi, Valla, & Wang, 2004)) requires substantial modification and processing at an intermediate node (BS). Some schemes, such as EBSN (Bakre & Badrinath, 1995), also require modifications at the sender side. This makes it difficult for these schemes to inter-operate with the existing infrastructure.

The second important factor is encrypted traffic. As network security is taken more and more seriously, encryption is likely to be adopted very widely. For instance, IPSEC is becoming an integral part of IPv6, the next generation IP protocol. In such cases the whole IP payload is encrypted, so that the intermediate nodes (be it the base station or another router) may not even know that the traffic being carried in the payload is TCP. Any approach (such as SNOOP (Balakrishnan, Seshan, & Katz, 1995), I-TCP & M-TCP (Goff, Phatak, & Gupta, 2000), M-TCP (Own & Singh, 1997) ...) which depends on the base station doing a lot of mediation will fail when the traffic is encrypted. Even more serious factor, sometimes data and ACKs can take different paths (for instance, in satellite networks). In such case, schemes based on "intermediary" involvement will have serious problems.

Yet another considered factor is maintaining true end-to-end semantics. I-TCP and MTCP do not maintain true end-to-end semantics. M-TCP in (Own & Singh, 1997) does maintain end-to-end semantics, but requires a substantial base-station involvement nonetheless. Thus there is a need for true end-to-end signaling without involving any intermediary.

As a matter of fact, even if one assumes that the above mentioned issues are not relevant, and that an intermediary (such as a base station) can be brought in for performance enhancements; there is still a need to consider whether the intermediary will become the bottleneck.

It is clear that the base stations (BS) in SNOOP, I-TCP & M-TCP, M-TCP will all have to buffer at least some amount of data (to perform local retransmission, etc.) and do some extra processing for each connection going through them (Bonde, Jharia, & Shrivastav, 2010). If hundreds or thousands of nodes are mobile in the domain of a base station, it could get overwhelmed with the processing of traffic associated with each connection. When a mobile node moves from the domain of one BS to another, the entire "state" of the connection (including any data that was buffered for retransmissions) needs to be handed over to the new base station. This can cause significant amount of overhead and might lead to the loss of some packets and the sender dropping congestion window, which would overcome the original purpose behind the whole endeavor (Kaspar, Evensen, Hansen, Engelstad, Halvorsen, & Griwodz, 2009) and (Qian, Sen, & Spatscheck, 2013).

To solve the problems listed above, this paper proposes a method that can be categorized to the third group (TCP modification), in which changes should be made at mobile node only, without interfering into the sender and intermediate elements of network.

## IV. Proposed Method

To solve the problems listed above, this paper proposes a method that can be categorized to the third group (TCP modification), in which changes should be made at mobile node only, without interfering into the sender and intermediate elements of network.

The proposed method is based on identifying the cause of packet loss. In this case, the link layer messages of wireless network protocol stack are used, as well as information from headers of transmitting packets to monitor network state. The proposed method combines CC method (that predicts beginning of handoff event) and Freeze TCP (Goff, Phatak, & Gupta, 2000) (that controls data transfer during handoff), to determine and predict the handoff event of a mobile node.

The first step of hand off takes place in Link Layer. Since this Layer is closer to the Physical Layer, it can check the movement of the node and also can predict the beginning of the Handoff. The time of Handoff spent in Link Layer depends on the technologies used in Lower Layer (Physical Layer). But usually, the Handoff spends less time in Link Layer than in IP Layer. The difference between the amount of time spent in these two layers leads to lack of coincidences of IP and MAC addresses. As a result, ACKs which are being sent from the MH to the sender are lost. In this case, TCP should send ZWA to the sender, before the start of Handoff in Link Layer. Figure 4 shows diagram of new mechanism and the prediction of the moving mobile node.

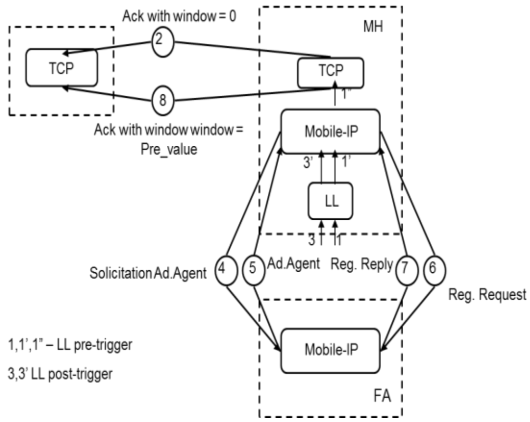



Fig. 4 New mechanism and the prediction of the moving mobile node

At the beginning of handoff, TCP sender stops data transfer, fixes RTO value, floating window size and threshold values for transition between TCP's algorithms, by sending zero-size ACK packets. During the process of determining the handoff event at link layer, MH sends a message (LL-trigger) to upper layers of protocol stack to inform them that handoff is coming. TCP sends one or few ACK packets of zero-size (ZWA) to inform the sender that receiver is not ready to receive data. In which case, the sender turns to waiting mode.

As mentioned above the first stage of handoff event occurs at link layer. This process takes less time compared to handoff at IP layer as illustrated in Figure 5. The process start with a message from the node 1 then a re-registration request is sent back 2 and finally re-registration acknowledgement is received 3.

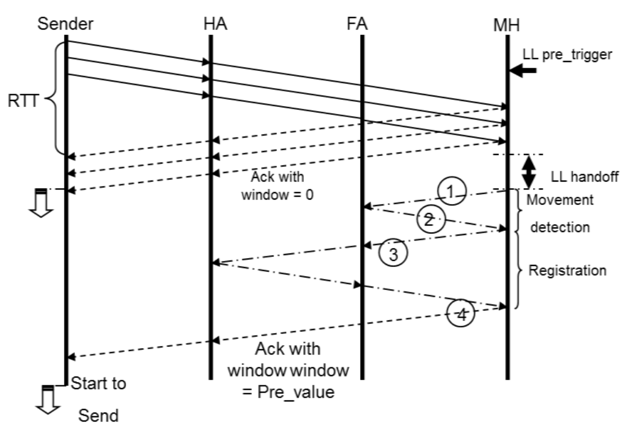
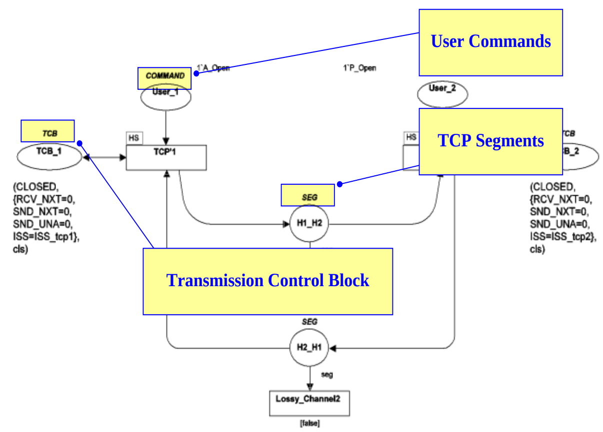


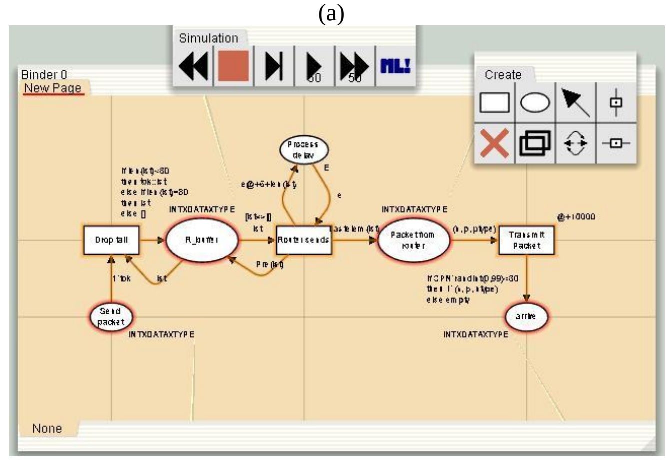

Fig 5 Process of Hand off in Link layer

## V. Simulation Results and Discussion

In order to test the proposed method, the CPN (Color Petri Network) (Reisig, 2013) tool has been used as a simulator to simulate the model and protocol. CPN Tools is a tool for editing, simulating, and analyzing Colored Petri nets. The tool features incremental syntax checking and code generation, which take place while a net is being constructed. A fast simulator efficiently handles untimed and timed nets. Full and partial state spaces can be generated and analyzed, and a standard state space report contains information, such as boundless properties and aliveness properties. Figure 6 illustrates transport layer protocol simulated in Color Petri network simulation.




(a)




(b)

Fig. 6 CPN-Simulation Setup for proposed protocol (a) Block diagram, (b) Petri Net program setup.

The proposed method performance during hand off is tested against some resent and old methods. The methods are SNOOP, TCP & MCP, M-TCP and Delay & Dupacks (Vaidya, Mehta, Perkins, & Montenegro, 2002). The comparative analyses and summary of the characteristics of these solutions against the proposed method is showed in Table 1.

TABLE I
CHARACTERISTICS OF VARIOUS MOBILE TCP SOLUTIONS (BER REFERS TO BIT-ERROR- RATE).

| | SNOOP | I-TCP&M-TCP | M-TCP | Delayed& Dupacks | Proposed method |
|---|---|---|---|---|---|
| Requires intermediate node TCP mods? | yes | Yes | yes | No | No |
| End-to-end TCP semantics | yes | No | no | Yes | yes |
| Handle long Disconnections | no | | yes | No | yes |
| Frequent Disconnections | no | | | No | yes |
| Handle high BER | yes | Yes | yes | No | yes |

To understand the effects of mobility on the performance of TCP, mobile IP is used. TCP Reno is used like a protocol of transport layer and the traffic generators available are FTP which creates FTP/GENERIC connection from nodes to MH. The mobile node is moving between two cells. The obtained results from our Petri net model also demonstrate that during simulation the Handoff occurred twice. The speed of MH is 5m/sec and the time spent in each cell is 10 sec.

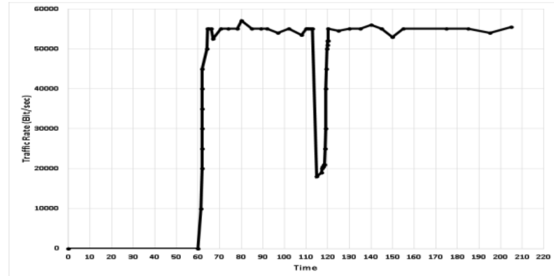
Table 2 shows the number of Timeout occurred for both hand off during the simulation. As see in CC approach, the number of Timeout increases by increasing the time of handoff but in the proposed method it is almost constant because of freezing data transfer, fixing RTO value, floating window size and threshold values for transition between TCP's algorithms. Hence, the number of time outs is reduced. The comparison of different version of TCP protocol is shown in Figure7.

TABLE II
NUMBER OF TIMEOUT DUE TO HAND OFF

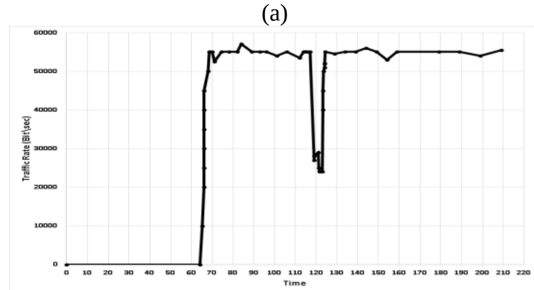
| Approach / Handoff Time(s) | CC | Proposed method |
|---|---|---|
| 0 | 1 | 1 |
| 1.5 | 4 | 2 |
| 4.3 | 8 | 2 |
| 5.2 | 6 | 4 |
| 8.9 | 9 | 4 |

Figure 7 show a comparison between previous methods and the proposed method with regards to the traffic rate during hands off. The comparison involved methods such as SNOOP (Figure 7 (a)), M-TCP(Figure 7 (b)), I-TCP & M-TCP (Figure 7 (c)),Delay and Dupacks (Figure 7 (d)) and proposed method(Figure 7 (e)). The simulation tests were carried away with traffic rate up to 56000 bit/sec for a period of 250 sec. The hand off was triggered for all at a time 115 sec to identify the change in both traffic drop rate and recovery time for reaching the top rate of 56000 bit/sec.

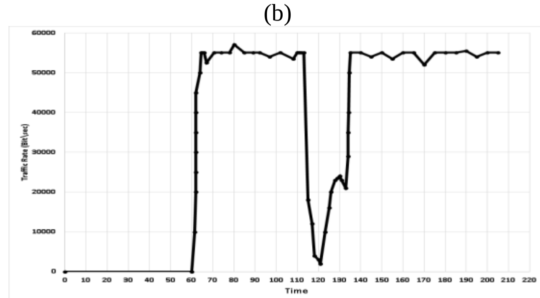
As illustrated in these graphs, during hands off, Delay and Dupacks method gives a traffic drop rate less than the other three methods (SNOOP, M-TCP and I-TCP & M-TCP). However, it takes much time to return or recover to the top average rate of 56000 bit/sec compared to SNOOP. On the other hand, I-TCP & M-TCP gives the worst performance of drop in traffic rate and traffic rate recovery between the other three methods. Comparing these methods to the proposed method, the modification in hand off algorithms gives the best performance with regard to both drop in traffic rate during hands off and recovery to the top rate of 56000 bit/sec. As a matter of fact, the graph in Figure 7 (e) shows the proposed method has a size of congestion windows no less than 50000 bit/sec in traffic compared to 23500 bit/sec for Delay and Dupacks method (Figure 7 (a)), 90 bit/sec for I-TCP & M-TCP (Figure 7 (b)), 24000 bit/sec for M-TCP and 18600 bit/sec for SNOOP (Figure 7 (c)). Moreover, the recovery time for the proposed method took only 8 sec compare to 15 sec for Delay and Dupack, 18 sec of I-TCP & M-TCP, and 6 sec for M-TCP. Only SNOOP achieved a better time of 4 sec.



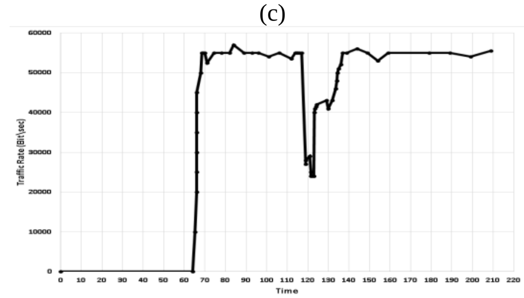

(a)



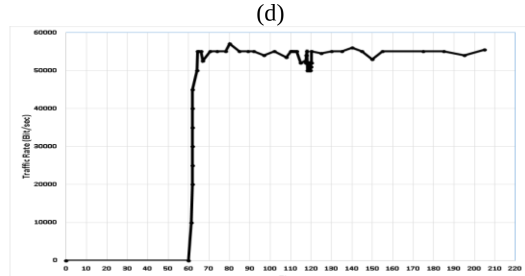

(b)




(c)




(d)




(e)

Fig 7 Comparison between throughputs of different methods, (a) SNOOP, (b) M-TCP, (c) I-TCP & M-TCP,(d) Delay and Dupacks (e) Proposed Method.

To understand the enhancement made by the proposed method, the number of data packet lost during handoff is reduced by decreasing the time required to detect mobile node movements between coverage areas. After handoff ends, mobile node turns the sender from waiting node by single ACK packet (with the last non-zero size of the window before handoff), and data transfer starts with the advertised window size.

Throughput is used as the quantity in which the efficiency of these methods is determined. Since SNOOP obtained a better performance from the previous methods (SNOOP, M-TCP, I-TCP & M-TCP and Delay and Dupacks), it is used to compare the throughput against the proposed method. It can be seen from Figure 7 that the throughput of the proposed protocol (blue line) is higher than the standard SNOOP protocol (red line). Each time the handoff time is increased the proposed method has a better throughput than SNOOP as illustrated in the figure.

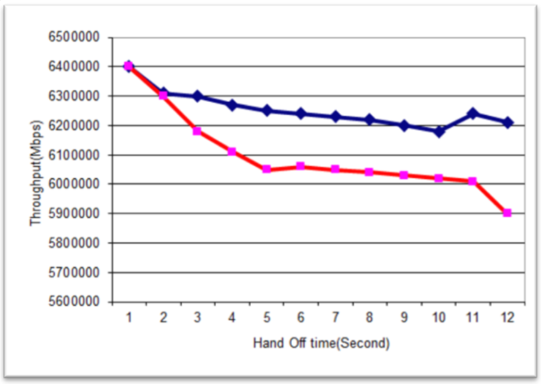



Fig 8 Throughput comparison against handoff time between SNOOP and proposed method

## VI. Conclusion

This paper presents a method for compensation of TCP throughput degrading during movement of mobile node between areas in wireless networks. The method determines the start of a handoff event using link layer messages then, it freezes the transport layer protocols parameters until the end of handoff. As a result, the modification will reduce the amount of lost data and improve the efficiency of TCP protocol. This is very important during long handoffs when the node moves from one to another non-overlapping coverage areas. The method requires changes at mobile node only, without interference between sender and intermediate elements of the network. Hence, it will work with encrypted network traffic as well.

The proposed method was tested against standard TCP protocols (i.e, SNOOP, M-TCP, ITCP & MTCP and Delay & Dupacks) previously used to overcome this problem. The obtained results demonstrate that the proposed method yield a better recovery time and throughput when compared to these methods. However, this modification has the disadvantage of having less efficiency in heterogeneous network when the transport layer parameters are different.

## References

[1] Bakre, A., & Badrinath, B. (1995). I-TCP: indirect TCP for mobile hosts. Distributed Computing Systems, 1995., Proceedings of the 15th International Conference on, (pp. 136-143). http://dx.doi.org/10.1109/icdcs.1995.500012

[2] Balakrishnan, H., Seshan, S., & Katz, R. H. (1995). Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. Wireless Networks, 1(4), 469-481. http://dx.doi.org/10.1007/BF01985757

[3] Bonde, P., Jharia, B., & Shrivastav, A. K. (2010). Performance Evaluation of Exclusive Handover Message TCP in Mobile IP Based Networks During Handover. Proceedings of the 12th International Conference on Advanced Communication Technology (pp. 761-764). Gangwon-Do, South Korea: IEEE Press.

[4] Gerla, M., Ng, B. K., Sanadidi, M. Y., Valla, M., & Wang, R. (2004). TCP Westwood with adaptive bandwidth estimation to improve efficiency/friendliness tradeoffs. Computer Communications, 27(1), 41-58. http://dx.doi.org/10.1016/S0140-3664(03)00114-2

[5] Goff, T. J., Phatak, D. S., & Gupta, V. (2000). Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments. In Proceedings of IEEE INFOCOM'2000, (pp. 1537-1545). Tel Aviv. http://dx.doi.org/10.1109/infcom.2000.832552

[6] Kaspar, D., Evensen, K., Hansen, A., Engelstad, P., Halvorsen, P., & Griwodz, C. (2009). An analysis of the heterogeneity and IP packet reordering over multiple wireless networks. Computers and Communications, 2009. ISCC 2009. IEEE Symposium on, (pp. 637-642). http://dx.doi.org/10.1109/iscc.2009.5202249

[7] Own, K., & Singh, S. (October, 1997). M-TCP: TCP for Mobile Cellular Networks. SIGCOMM Comput. Commun. Rev., 27(5), 19-43. http://dx.doi.org/10.1145/269790.269794

[8] Perkins, C. (2002). IP Mobility Support for IPv4. RFC Editor.

[9] Postel, J. (1981). Transmission Control Protocol. Internet Engineering Task Force. Internet Engineering Task Force. http://dx.doi.org/10.17487/rfc0793

[10] Qian, F., Sen, S., & Spatscheck, O. (2013). Silent TCP Connection Closure for Cellular Networks. Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies (pp. 211-216). Santa Barbara, California, USA: ACM. http://dx.doi.org/10.1145/2535372.2535402

[11] Reisig, W. (2013). Understanding Petri Nets - Modeling Techniques, Analysis Methods,Case Studies. Springer. http://dx.doi.org/10.1007/978-3-642-33278-4

[12] Sasikumar, B., & Vasudevan, V. (2008). Improving the Performance of TCP/IP over Wireless Networks with a RETSINA Agent. Emerging Trends in Engineering & Technology, International Conference on (pp. 151-156). IEEE Computer Society. http://dx.doi.org/10.1109/icetet.2008.157

[13] Sun, W., Wen, T., & Guo, Q. (2008). A Novel Protocol for Mobile-Induced Packet Reordering in Mobile Ad Hoc NetWorks. Information Science and Engineering, 2008. ISISE '08. International Symposium on, (pp. 626-631). http://dx.doi.org/10.1109/ISISE.2008.204

[14] Vaidya, N. H., Mehta, M. N., Perkins, C. E., & Montenegro, G. (2002). Delayed duplicate acknowledgements: a TCP-Unaware approach to improve performance of TCP over wireless. Wireless Communications and Mobile Computing, 2(1), 59-70. http://dx.doi.org/10.1002/wcm.33