

TCP Performance through Simulation and Testbed in Multi-Hop Mobile Ad hoc Network

*Assistant Prof.Chandra Kanta Samal, AND College, University of Delhi, New Delhi, India
cksamal@gmail.com

ABSTRACT

When a packet is sent through wireless network it can be lost due to either environmental noise or due to congestion in the network. Traditional TCP protocols do not perform well in the wireless network because they were designed for wired networks with the assumption that the transmission medium is quite reliable. This often results in TCP's failure in distinguishing the cause of the packet loss as most of these problems hinder the throughput of the TCP connections in wireless networks. The work has been carried out to test the window size, packet size and path length of Multi-Hop Ad hoc Network, the TCP performance in a straight line, multi hop ad-hoc networks in two ways: the first one network simulator ns-2 and second one practical testbed, and the results that were obtained from both the models were compared and analyzed.

Keywords: *Wireless network, Throughput, Interference, Multi-Hop.*

I. INTRODUCTION

Ad Hoc Networks are complex distributed systems that consist of wireless mobile or static nodes that can freely and dynamically self-organize. In this way they form arbitrary and temporary "Ad hoc" networks topologies, allowing devices to seamlessly interconnect in areas with no pre-existing infrastructure. Recently, the introduction of new protocols such as Bluetooth [1], IEEE 802.11 [2] and Hyperlan [3] are making possible the deployment of Ad hoc networks for commercial purposes. As a result, considerable research efforts have been put on this new challenging wireless environment.

The TCP (Transmission Control Protocol) [4] was designed to provide reliable end-to-end delivery of data over unreliable networks. In theory, TCP should be independent of the technology of the underlying infrastructure. In particular, TCP should not care whether the Internet Protocol (IP) is running over wired or wireless connections. In practice, it does matter because most TCP deployments have been carefully designed based on assumptions that are specific to wired networks. Ignoring the properties of

wireless transmission can lead to TCP implementations with poor performance. Indeed, in mobile or static Ad hoc networks losses are not always due to network congestion, as it is mostly the case in wired networks.

In Ad hoc networks, the principal problem of TCP lies in performing congestion control in case of losses that are not induced by network congestion. Since bit error rates are very low in wired networks, nearly all TCP versions nowadays assume that packets losses are due to congestion. Consequently, when a packet is detected to be lost, either by timeout or by multiple duplicated ACKs, TCP slows down the sending rate by adjusting its congestion window. Unfortunately, wireless networks suffer from several types of losses that are not related to congestion, making TCP not adapted to this environment. Numerous enhancements and optimizations have been proposed over the last few years to improve TCP performance over one-hop wireless (not necessarily Ad hoc) networks. These improvements include infrastructure based WLANs [5, 6, 7, 8], mobile cellular networking environments [9, 10], and satellite networks [11, 12]. Ad hoc networks inherit several features of these networks, in particular high bit error rates and path asymmetry, and add new problems that come from mobility and multi-hop communications, such as network partitions, route failures, hidden and exposed terminals. We note that the following TCP versions: Tahoe, Reno, Newreno, and Vegas perform differently in Ad hoc networks [13]. However, all these versions suffer from the same problem of inability to distinguish between packet losses due to congestion from losses, due to the Hidden, Exposed stations and network partition of Ad hoc networks are correlated with the transmission range. By increasing the transmission range, the hidden terminal problem occurs less frequently. On the other hand, the exposed terminal problem becomes more important as the transmission range identifies the area affected by a single transmission.

II. PROBLEM STATEMENT

Ad hoc networks are complex distributed systems that consist of wireless mobile or static nodes that can freely and dynamically self-organize. Terminals that communicate with each other by forming a multi hop radio network and maintaining connectivity in a decentralized manner. In this way they form arbitrary, and temporary “ad hoc” networks topologies, allowing devices to seamlessly interconnect in areas with no pre-existing infrastructure. Since the nodes communicate over wireless links, they have to contend

with the effects of radio communication, such as noise, fading, and interference. Each node in a wireless ad hoc network functions as both a host and a router, and the control of the network is distributed among the nodes.

The focus of this study is on the performance of TCP operating in ad hoc networks. Since TCP/IP is the standard network protocol stack on the Internet, its use over ad hoc networks is a certainty. Not only does it leverage a large number of applications, but its use also allows seamless integration with the Internet, where available. Reliable data transfer and congestion control are key requirements for any computer network. TCP, which fulfills both of these requirements, is the most widely used reliable transport protocol in today's Internet and has demonstrated its viability with respect to Internet connectivity. Furthermore, TCP is used to transport a significant portion of Internet traffic such as e-mail (SMTP), file transfers (FTP), and WWW (HTTP). Thus, the use of TCP in mobile ad hoc networks is clearly advantageous. But the characteristics of ad hoc networks face a problem for the TCP protocol, which was primarily designed for wire line networks, which their topology is much less dynamic, and therefore have more predictable nature.

Main problems that commonly associated with performance analyses of TCP over wireless ad hoc networks are lack of specialization/specification and completeness. Most of the research on performance evaluations of TCP over ad hoc networks were not specific, and are generalized. Most available performance evaluations are usually analyze the performance of a TCP over different routing protocols available [12,14,15] (DSR, DSDV, TORA, AODV); or else considering all the TCP versions over a specific routing protocol with general scenarios - random mobility patterns and few studies have proposed improvements. But no such analysis has been done based upon the string topology with static natured nodes. Because of above reasons this research is mainly focused upon **"the performance of TCP in multi hop Mobile ad hoc networks"**. In this topology every node in one region communicates with node in other region. Many experts have done an extensive research on the performance of the TCP over ad hoc networks [12, 14] with simulation and have suggested many improvements but very less research has been done on experimental or practical approach for evaluation of TCP performance throughput over ad hoc network with a string topology, no research have taken place in both environment which help in more accurate analysis of TCP over Ad hoc networks, so research in testbed or practical approach is needed.

Here the experiments were done in both environments (simulation and testbed) the results obtained from simulation, testbed were compared and analyzed. It will observe whether there is a co-relation between results that were obtained, whether there is really an advantage is there with simulations or not, critics of both methods (simulation and testbed) will be analyzed, the reasons for less performance in static nature were observed.

III METHODOLOGY

The work consists of finding TCP performance over ad hoc networks on specified conditions with specified straight-line topology, on both simulator and test bed environments and also consists of studying ns-2 simulator, implementation of simulation, and implementation of testbed and finally the comparison of both the results that were obtained

3.1 TOPOLOGY

Ad hoc network is a collection of wireless mobile nodes forming a temporary network without the use of any existing networks infrastructure or centralized administration. Each node in a wireless ad hoc network functions as both a host and a router, and the control of the network is distributed among the nodes. Ad hoc networks can be single-hop or multi-hop networks. The most basic Single-hop ad hoc network consists of two stations which communicate directly with each other. An example for a single-hop can be found in Figure 3.1 and for a multi-hop ad hoc network in Figure 3.2.

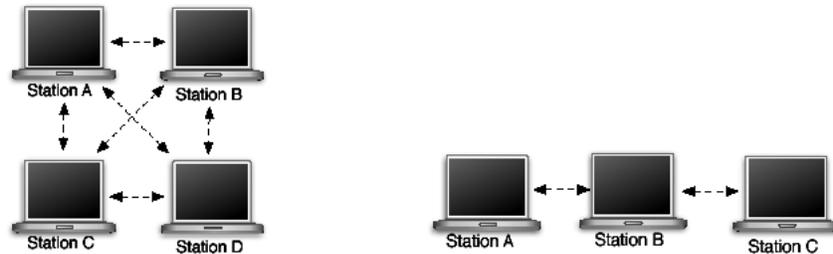


Figure 3.1: Single-hop ad hoc network Figure 3.2: Two hop/Multi-hop ad hoc network

In our research we focus on the straight line, multi hop topology which looks like as given figure 3.3.

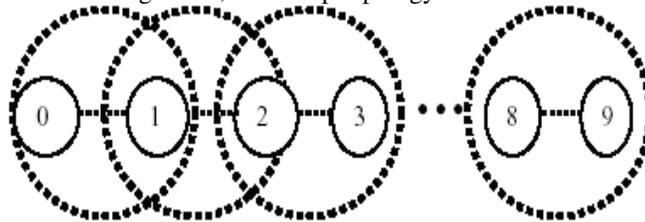


Figure 3.3: Topology

In this topology nodes may communicate directly if they are within the transmission range of each other. The dotted lines denote transmission range. Since we are trying to isolate the effect of hop count. Nodes are not mobile and they are static. For example the nodes 1 and three communicate through node 2 only, assuming if node 2 is not present then there is no communication at all in between node 1 and all other nodes, this is called as a straight-line communication.

IV RESULTS

Testbed experiments were carried out under MS Windows XP (with SP2), and Simulation experiments were carried under Red Hat Linux (kernel 2.4.0) environment, using NS2-2.28 simulator.

4.1 SIMULATION RESULTS

After running ns-2 simulations, it mainly generates two files, trace file and nam files of sizes 30 Mb each approximately, and the formats of files are as follows

Sample formats of NAM file data

```
+ -t 0.000000000 -s 0 -d -1 -p tcp -e 40 -c 2 -a 0 -i 0 -k AGT
- -t 0.004411734 -s 0 -d -1 -p DSR -e 32 -c 2 -a 0 -i 1 -k RTR
h -t 0.004646734 -s 0 -d -1 -p DSR -e 84 -c 2 -a 0 -i 1 -k MAC
r -t 0.005319563 -s 1 -d -1 -p DSR -e 32 -c 2 -a 0 -i 1 -k MAC
```

Sample formats of trace file data

```
r 82.565144931 _1_ RTR --- 1 DSR 24 [0 ffffffff 2 800] ----- [2:255 3:255 32 0] 1 [1 1] [0 1 0 0->0] [0 0 0 0->0]
r 82.626314559 _5_ RTR --- 0 tcp 544 [a2 5 2 800] ----- [2:0 3:1 32 5] [0] 1 2
f 82.626314559 _5_ RTR --- 0 tcp 544 [a2 5 2 800] ----- [2:0 3:1 32 3] [0] 1 2
```

When a nam file was given as input to NAM, it will generate NAM visualizations, as shown in figure 4.1 and 4.2. Figure 4.1 gives information about NAM developers, and NAM source. Figure 4.2 is an example of a NAM screen shot of 2 nodes when they are in communication.

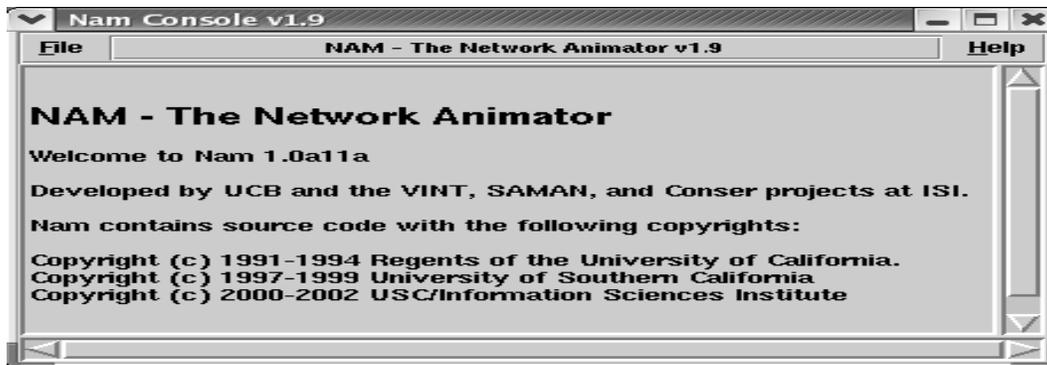


Figure 4.1: NAM source information

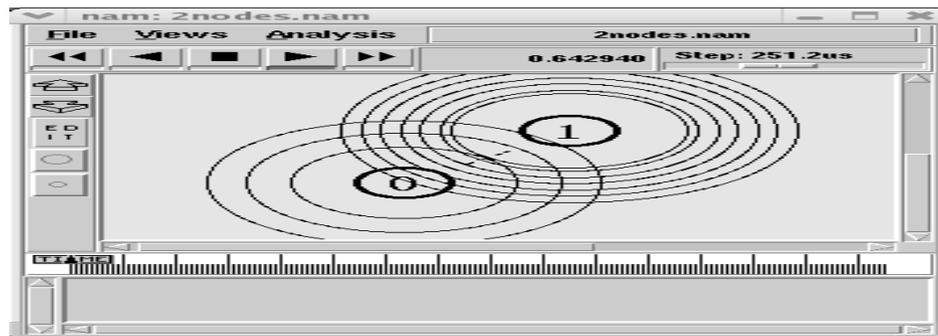


Figure 4.2: NAM screen shot

The given figure 4.2 was the simple screen shot of NAM. When two nodes are in communication range of each other and are able to send packets between each other. In which we can observe the communication range, packet forwarding, packet dropping, etc

Simulation results for 2 nodes

As it mentioned above, after running simulation scripts on ns-2, ns-2 generates trace file, on

giving the trace file as input to trace graph, Trace graph generates different types of outputs like trace information (figure 4.3), network information (figure 4.4), and graphs of two and three dimensions (figure 4.5, figure 4.6). Trace graph generates trace information as shown in figure 4.3. and it gives information about packet type, packet size, source trace level, destination trace level sent packet type, acknowledge type, simulation start time, simulation end time, number of node information. The information generated for the our simulations were packet type was TCP, source trace level was AGT, packet size starts from 28, sent packet type, ACK packet types were given as TCP, simulation time was given as 300 seconds. As the simulations run were to observe TCP information.

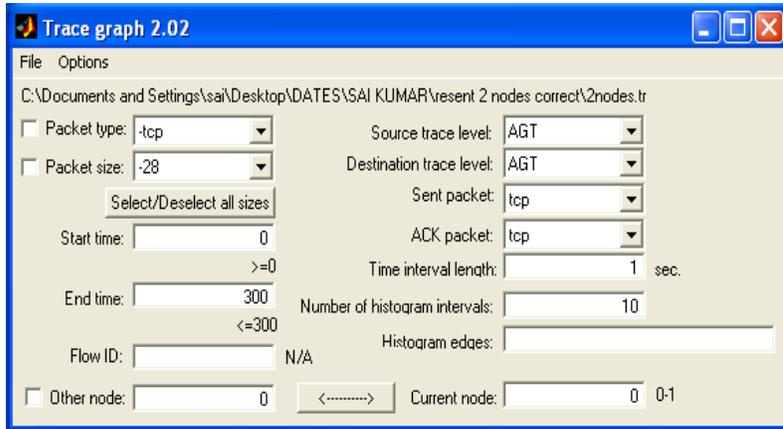


Figure 4.3: Two nodes Trace graph information

Network information generated by Trace graph was given in figure 4.4. the information generated was about simulation time, total number of nodes, number of sending nodes, number of receiving nodes, no of generated packets, no of sent packets, no of forwarded packets, no of dropped packets, minimum packet size, maximum packet size, average packet size, no of sent bytes, no of forwarded bytes, no of dropped bytes,

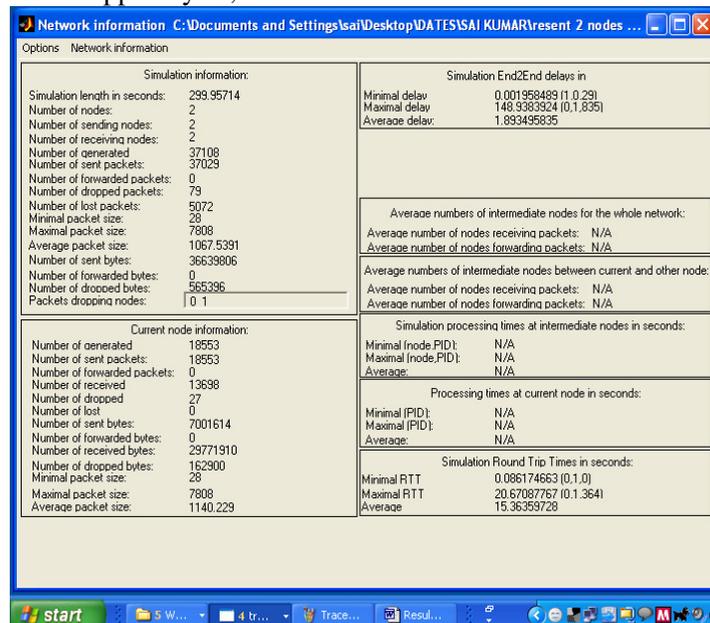


Figure 4.4: Two nodes Network information

number of dropped nodes, and the above mentioned all parameters information was also given for current node, and also maximum delay, minimum delay, average delay, minimum RTT, maximum RTT, average RTT and etc, were given as shown in figure4.4. In figure 4.4 network information is for 2 nodes were given , as it was for 2 nodes, the sending nodes were 2, receiving nodes were 2, and dropped nodes were also 2 because the dropping may takes place at source side or destination side.

In the simulations the first set of results were to observe the cumulative sum of generated bytes, cumulative sum of send bytes, cumulative sum of received bytes, cumulative sum of dropped bytes, and cumulative sum of forwarded bytes corresponds to the simulation time, as shown in figure 4.5.

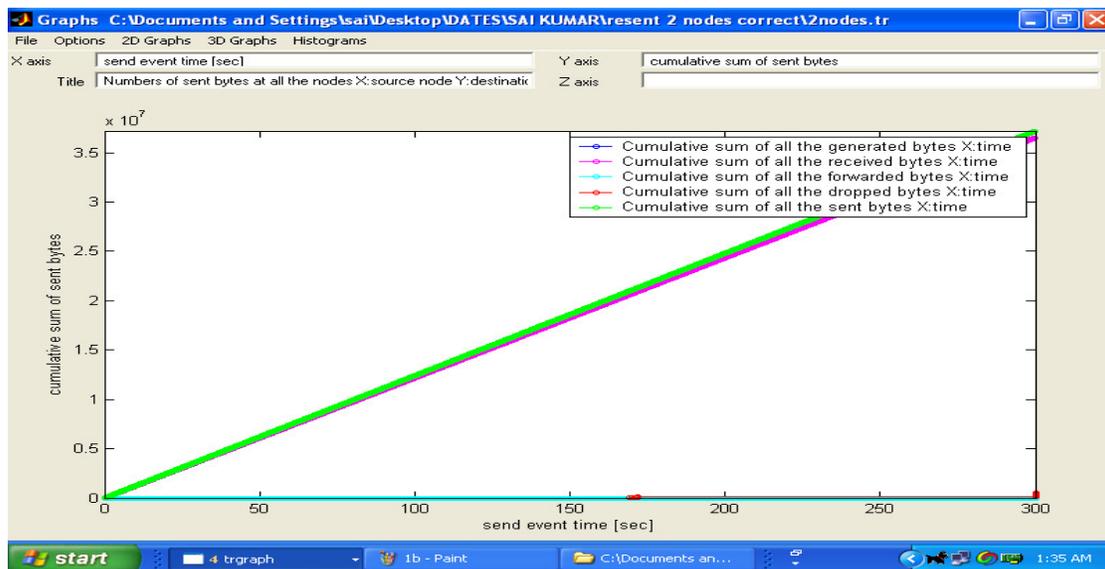


Figure 4.5: Cumulative sum of diff. Types vs. Simulation time

the second set of results were to observe the throughput of generated bytes, throughput of send bytes, throughput of received bytes, throughput of dropped bytes, throughput of forwarded bytes corresponds to the simulation time as shown in figure 4.6

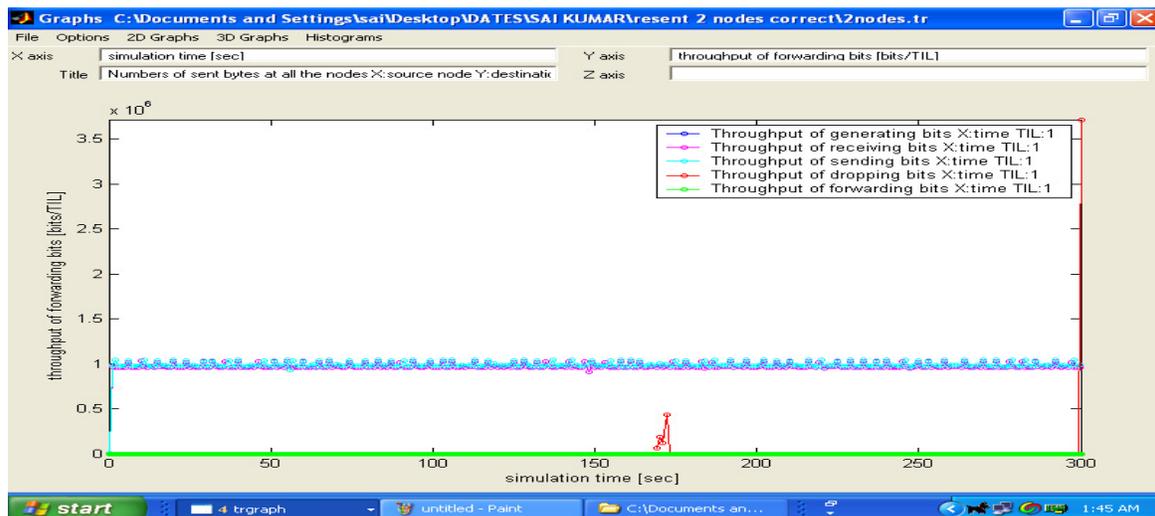


Figure 4.6: Throughput of diff. Types vs. Simulation time

Simulation results for 3 nodes

Three nodes trace graph information was given in figure 4.7, the information that was given by Trace graph: packet type was TCP, source trace level was AGT, packet size

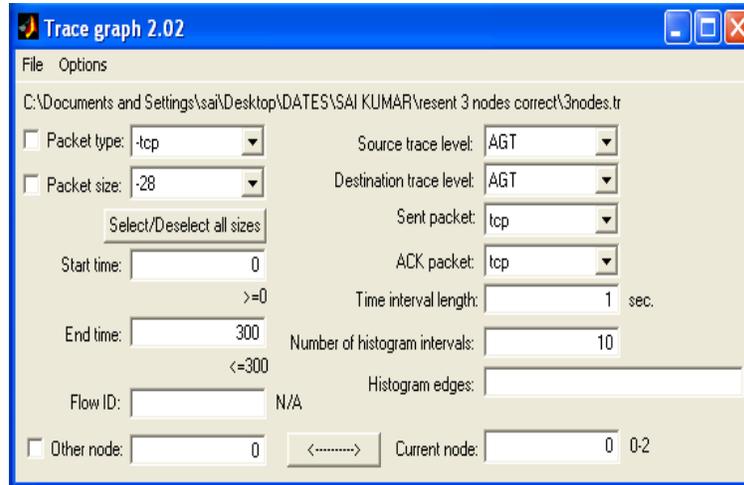


Figure 4.7: Three nodes Trace graph information

starts from 28, sent and ACK packet types were TCP, current node is 0, and nodes participated were 0, 1, and 2.

The network information for three nodes was given in figure 4.8, as it's for 3 nodes its given that the sending nodes were 3, receiving nodes were 3, and dropped nodes were also 3 because the dropping may be at source node or destination node or at forwarding node, number of generated packets were given as 36794, and the number of sent packets were also 36794 generally these should be same, number of forwarded packets were 4516, as the number of forwarded packets were very less as compared with sent packets cause less throughput, and the packet size varies from 28 bytes to 7812 where as the average packet size is 1040.2847, and we can also observe sent bytes, forwarded bytes, dropped bytes and so on.

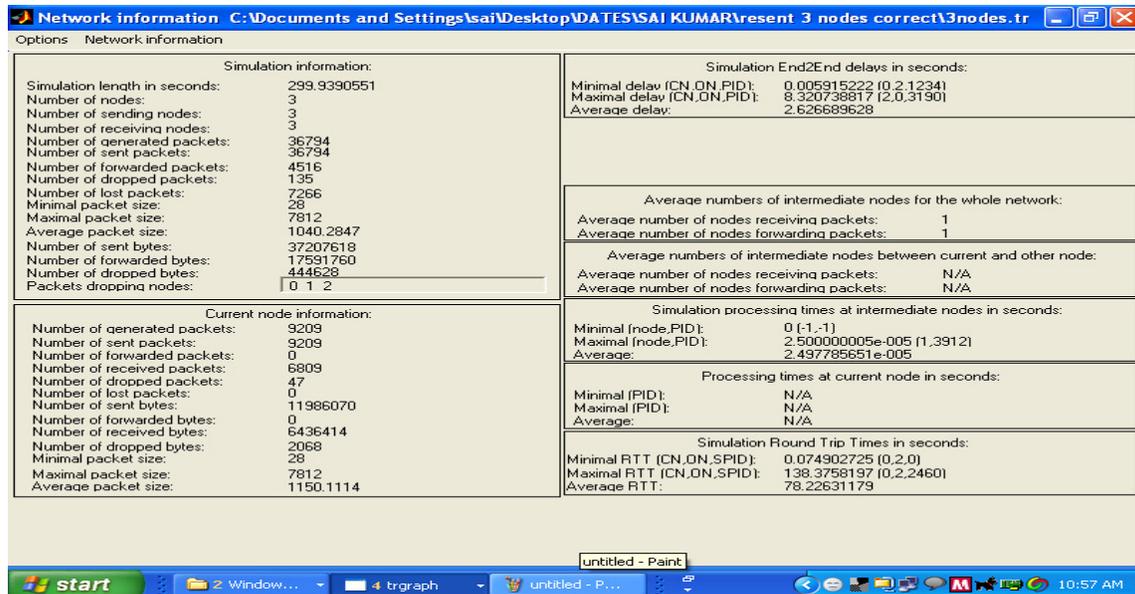


Figure 4.8: Three nodes Network information

for three nodes the first set of results were to observe the cumulative sum of generated bytes, cumulative sum of send bytes, cumulative sum of received bytes, cumulative sum of dropped bytes, cumulative sum of forwarded bytes corresponds to the simulation time. as shown in figure 4.9 it mainly useful for the comparison cumulative sum of generated bytes, cumulative sum of send bytes, cumulative sum of received bytes, cumulative sum of dropped bytes, cumulative sum of forwarded bytes according to the time.

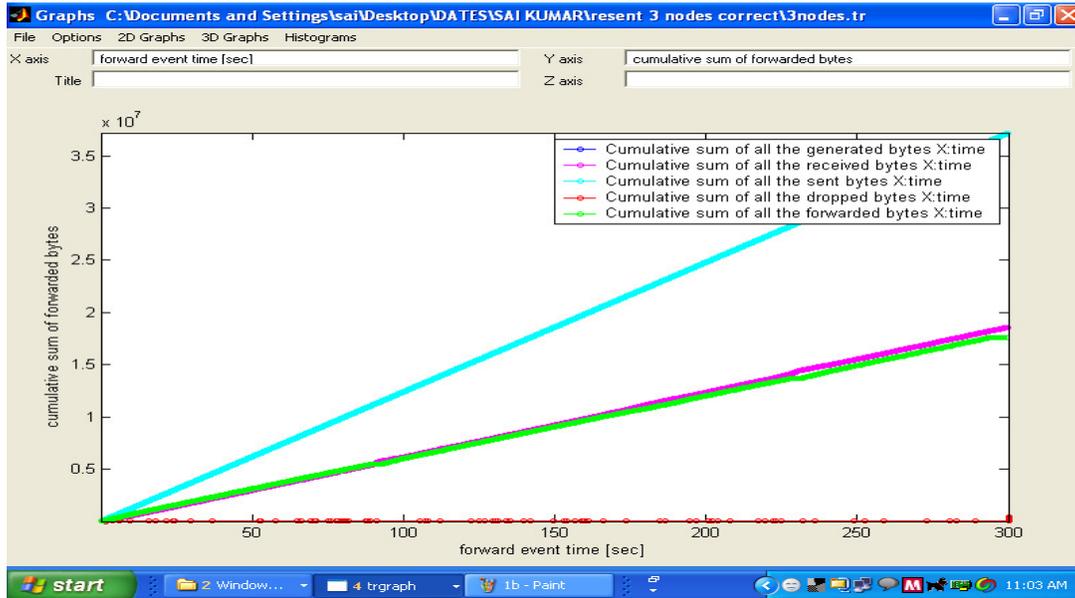


Figure 4.9: Cumulative sum of diff. Types vs. Simulation time

the second set of results were taken to observe the throughput of generated bytes, throughput of send bytes, throughput of received bytes, throughput of dropped bytes, throughput of forwarded bytes corresponds to the simulation time, as shown in figure 4.10. In which the comparison of throughput of generated bytes, throughput of send bytes, throughput of received bytes, throughput of dropped bytes, and throughput of forwarded bytes can be done easily.

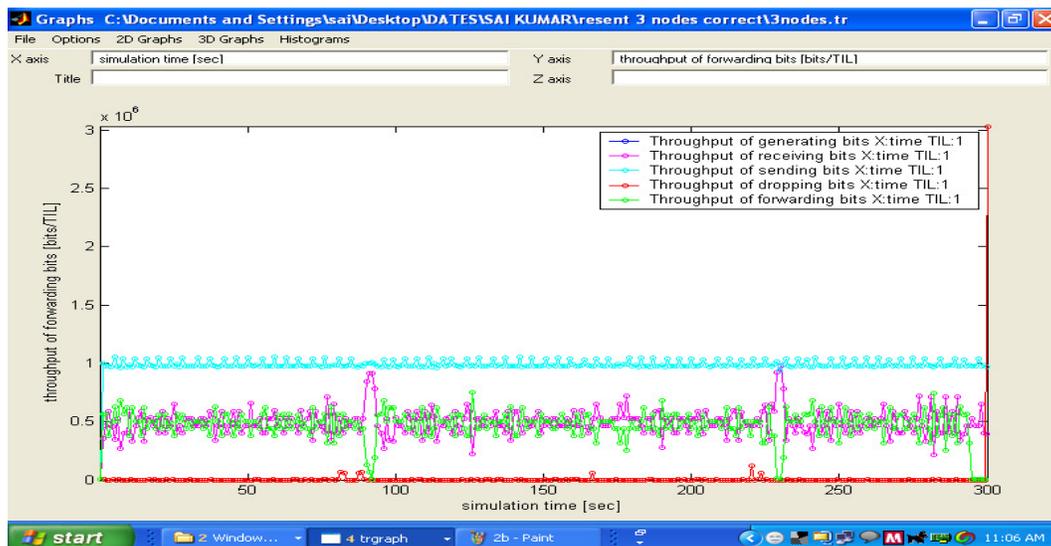


Figure 4.10: Throughput of diff. Types vs. Simulation time

The effect of Window size on TCP throughput

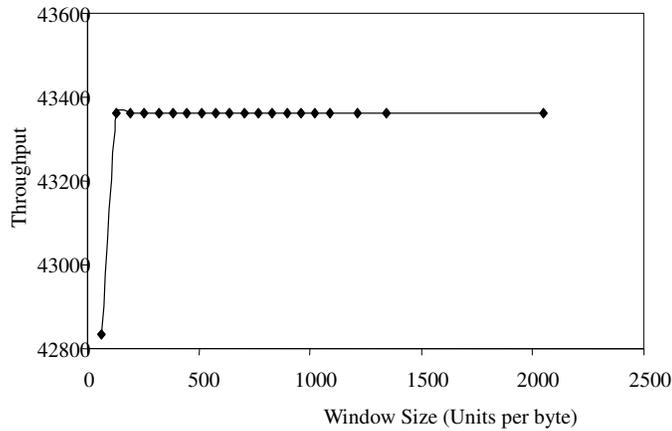


Figure 4.11: Throughput Vs Window Size

Here the experiments were conducted in such a manner to observe the effect of window size on throughput, in which the packet size was kept constant to 1024 bytes, varying window size from 64 bytes to 2048 bytes, throughput was taken as average of both the receiving nodes because its two way communication with 2 TCP flows, the graph was drawn between window size (bytes) Vs. throughput (AVG bytes per second) as shown in figure 4.11

The effect of Packet size on TCP throughput

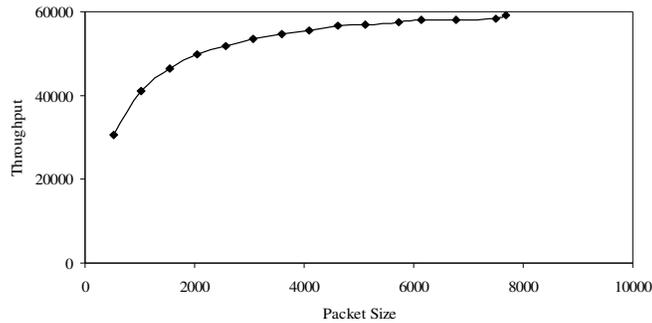


Figure 4.12 Throughput Vs Packet size

Here the experiments were conducted in such a manner to observe the effect of packet size on throughput, so window size was kept constant to 32 bytes, varying packet size from 512 bytes to 7680 bytes, throughput was taken as average of both the receiving nodes because its two way communication with 2 TCP flows, the graph was drawn between packet size (bytes) Vs. throughput (AVG bytes per second) as shown in figure 4.12.

The effect of Path length on TCP throughput (full duplex)

Here the experiments were conducted in such a manner to observe the effect of path length on throughput, so packet size and window were kept constant, packet size was taken as 7680 bytes,

window size taken as 128 bytes, the reason for taking 7680 bytes for packet size and 128 bytes for window size is that from the above experiments it is observed that throughput is maximum at 128 bytes of window size and there after increasing the window size haven't shown any change in throughput. And in the case of packet size also same, that the throughput is maximum at 7680 bytes of packet size, throughput was taken as average of both the receiving nodes because its two way communication with 2 TCP flows, the graph was drawn between no of hops Vs. throughput (AVG bytes per second), and is shown in figure 4.13 which is for full duplex communication link, and figure 4.14 was for half duplex link.

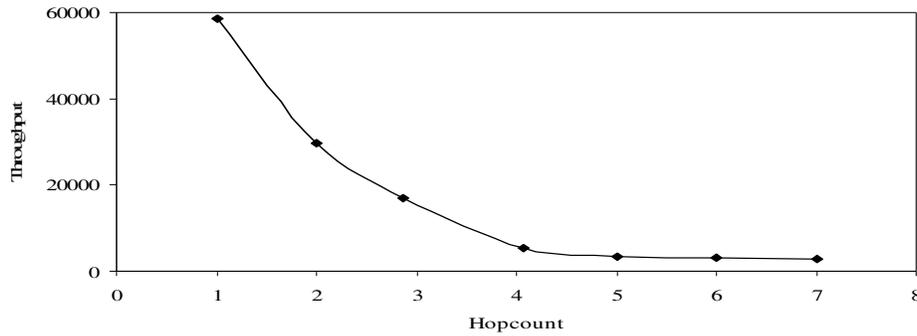


Figure 4.13: Throughput Vs Hop Count

The effect of Path length on TCP throughput (half duplex)

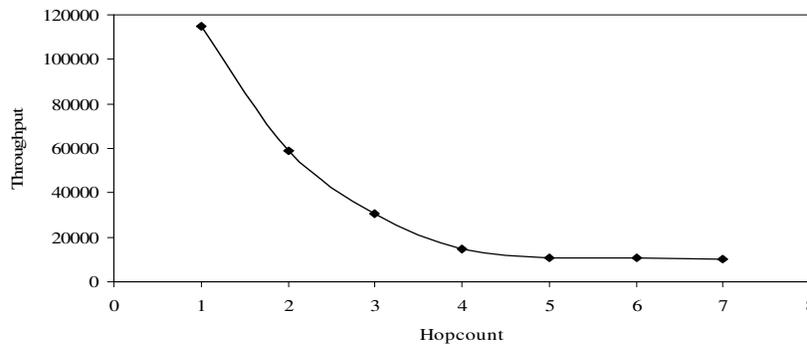


Figure 4.14: Throughput Vs Hop Count

4.2 Observation of unfairness

When two nodes were communicating with each other, with two different TCP half duplex connections in a same environment for a specified simulation time with the same parameters specified, then the received bytes at both the nodes should be same and TCP throughput at both nodes should be same, but its observed that the received bytes at the both nodes were not same and throughput at both the nodes were not same and it shows unfairness.

Unfairness when window size is varying

With 1024 bytes fixed packet size and varying the window size from 32 bytes to 2048 bytes the throughput was taken at both nodes, the blue line shows the throughput at node 0 and the pink line shows the throughput at node 1, the throughput was bytes per second and window size was bytes. The graph was to observe the throughput at node 0 and throughput at node 1. The figure 4.15 was the graph drawn between throughput (bytes per second) Vs window size (bytes), in which the comparison of throughput at the both the nodes can be easier.

Unfairness when packet size is varying

By keeping the window size 32 bytes as a constant and varying the packet size from 512 bytes to 7680 bytes the throughput was taken at both the nodes, and the blue line shows the throughput at

node 0 and the pink line shows the throughput at node 1, the throughput was bytes per second and packet size was bytes. The graph was to observe the throughput at node 0 and throughput at node 1. The figure 4.16 was the graph drawn between throughput (bytes per second) Vs window size (bytes), in which the comparison of throughput at the both the nodes can be easier.

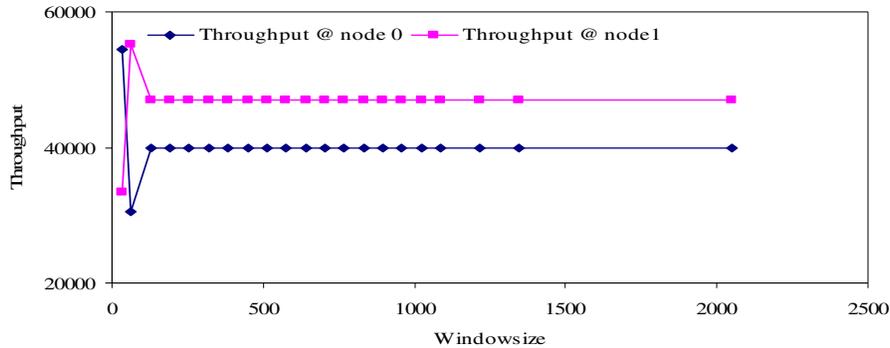


Figure 4.15: Comparison of Throughput at node 0 and at node 1

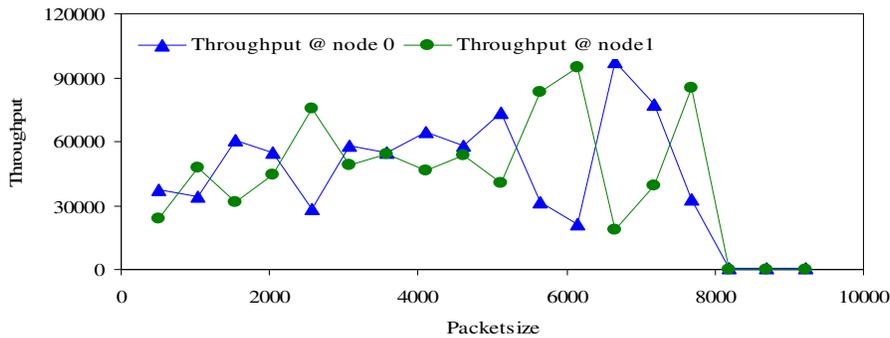


Figure 4.16: Comparison of Throughput at node 0 and at node 1

4.3 TESTBED RESULTS

Incase of Testbed model the experiments are performed in an indoor environment. Sending and receiving buffer sizes of 32768 bits were taken. A straight line topology is considered here in which one node communicate with another node in the other region, if they are with in the communicating range to each other. 200 x 200 meter network topography was used. To have a test bed setup we have taken three systems and were made to have a straight line topology. Laptops, desktops were used for our experiments with D-Link wireless NICs which provide data rates up to 11 Mbps.

HARDWARE AND SOFTWARE REQUIREMENTS:

- COMPAQ 1 Intel Pentium4, 1.8 GHz, 256 MB, D-Link DWL650+
- COMPAQ 2 Intel Pentium4, 1.8 GHz, 256 MB, D-Link DWL650+
- TOSHIBA Intel Pentium4, 1.6 GHz, 256 MB, Net Gear
- DELL Inspiron Intel Pentium4, 1.8GHz, 512 MB, Intel Pro2200g

On Desktops and Laptops, MS Windows XP (with SP2), VC++ IDE, Network Simulator (ns-2.28), Network Animator, Trace Graphs were used. Testbed experiments were carried out under MS Windows XP (with SP2). Combined source and sink program were developed to generate and transfer traffic from the source to destination. The sink listens to a port, receives packets,

accounts them, and drops them. Program was developed in such a manner that the same program will work as sender and receiver. Packet size, send and receiver buffer sizes can be specified and the TCP throughput can be calculated. The source can send packets of certain size as fast as it can but the actual rate would depend on the kernel protocol stack bandwidth. To study the effects on the throughput of TCP in testbed, through put of TCP was found between two communicating nodes using MS visual studio 6.0 as a programming language. Effect of packet size, Window size, and path length were examined on TCP throughput. The results that were obtained from both (simulation and testbed) were compared. It will observe whether there is a co-relation between results that were obtained, whether there is really an advantage is there with simulations or not, critics of both methods (simulation and testbed) will be analyzed, the reasons for less performance in static nature were observed.

The effect of Buffer size on TCP throughput

The experiment was conducted with sending, receiving buffer sizes varying from 1 kb to 64kb, packets with TCP payload of 1460 bytes and actual size on the air of 1500 bytes. TCP throughput was taken.

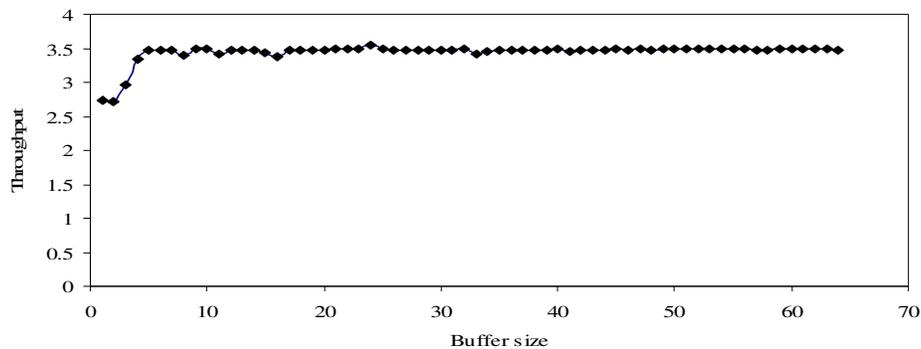


Figure 4.17: Throughput (Mbps) Vs buffer size (KB)

The effect of Packet size on TCP throughput

To observe the effect packet size on the TCP throughput, Packet size was varied gradually up to 1460 bytes only. As expected, the throughput increases with packet size as shown in Figure 4.18. As it explained above, including IP header of 20 bytes and TCP header of 20 bytes, with MTU of 1460 bytes, it attains a maximum TCP throughput at around 1500. So experiments were conducted using packets with TCP payload 1460 bytes and actual size on the air of 1500 bytes.

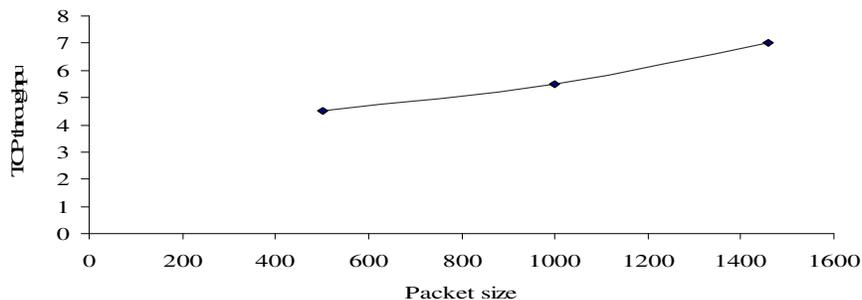


Figure 4.18: Throughput (Mbps) Vs Packet size (bytes)

The effect of Path length on TCP throughput

The experiment was stabilized with sending, receiving buffers of sizes 32768 bytes and packets with TCP payload of 1460 bytes and actual size on the air of 1500 bytes. The throughput was taken for 1 hop for 2 nodes, and for 2 hops for three nodes to observe the path length effect on TCP throughput. The effect of path length on TCP throughput was shown in figure 7.19.

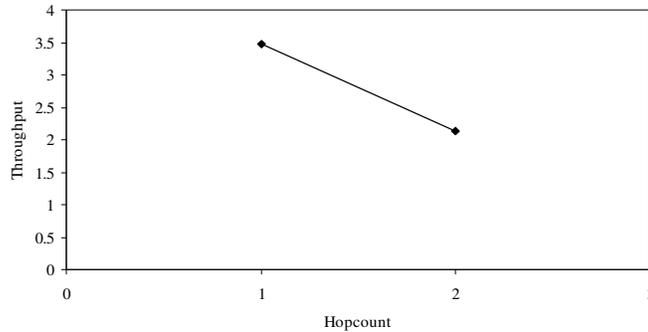


Figure 4.19: Throughput (Mbps) vs. Hop length

CONCLUSION

Advantages of Simulation

1. Simulations provide advanced simulation environments to test and debug any kind of networking protocols, including wireless applications.
2. To conduct different experiments or to study different parameters effect, there are lots of parameters which can be changed in C++ implemented code and as well as in TCL code which make to study different settings with different parameters, simulation study is more flexible as compared with experimental study, and its not always possible to change required parameters in testbed/experimental study
3. Conducting practical/testbed experiments were not always possible because when we take large number of nodes with mobility it's really difficult to conduct experiments.

Disadvantages of simulation

Some disadvantages of NS-2 from its open source nature. First, documentation is often limited and out of date with the current release of the simulator. Fortunately most problems may be solved by consulting the highly dynamic newsgroups and browsing the source code. Then code consistency is lacking at times in the code base and across releases. Finally, there is a lack of tools to describe simulation scenarios and analyze or visualize simulation trace files. These tools are often written with scripting languages. The lack of generalized analysis tools may lead to different people measuring different values for the same metric names, the learning curve for NS-2 is steep and debugging is difficult due to the dual C++/OTcl nature of the simulator. A more troublesome limitation of NS-2 is its large memory footprint and its lack of scalability as soon as simulations of a few hundred to a few thousand of nodes are undertaken

Advantages of testbed:

It is important to conduct real experiments in studying wireless multi hop ad hoc networking systems. Such experiments can provide a valuable baseline to identify important issues, provide reference values for simulation studies, and be used to set parameters of protocols appropriately to optimize performance. Since these systems are dynamic and difficult to model accurately, purely simulation studies could be quite removed from reality. Here the accuracy will be more and the required steps to improve performance can be performed at the observation point, and this leads to achieve best outputs.

Disadvantages of testbed:

It's not always possible to change required parameters in testbed/experimental study; Conducting practical/testbed experiments were not always possible because when we take large number of nodes with mobility it's really difficult to conduct experiments.

REFERENCES

- [1] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE Transactions on Networking*, vol. 5, no. 6, pp. 756–769, Dec. 1997.
- [2] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP performance over wireless networks," in *Proc. of ACM MOBIHOC*, Berkeley, CA, USA, 1995, pp. 2–11.
- [3] A. V. Bakre and B.R. Badrinath, "Implementation and performance evaluation of indirect TCP," *IEEE Transactions on Networking*, vol. 46, no. 3, pp. 260–278, Mar. 1997.
- [4] T. Henderson and R. Katz, "Transport protocols for Internet-compatible satellite networks," *IEEE JSAC*, vol. 17, no. 2, pp. 345–359, Feb. 1999.
- [5] S. Xu and T. Saadawi, "Performance evaluation of TCP algorithms in multi-hop wireless packet networks," *Journal of Wireless Communications and Mobile Computing*, vol. 2, no. 1, pp. 85–100, 2002.
- [6] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss'," in *Proc. of IEEE INFOCOM*, San Francisco, USA, Apr. 2003.
- [7] C. Jones, K. Sivalingam, P. Agarwal, and J. Chen, "A survey of energy efficient network protocols for wireless and mobile networks," *ACM Wireless Networks*, vol. 7, no. 4, pp. 343–358, 2001.
- [8] F. Klemm, S. Krishnamurthy, and S. Tripathi, "Alleviating effects of mobility on tcp performance in ad hoc networks using signal strength based link management," in *Proc. of the Personal Wireless Communications*, Venice, Italy, Sep. 2003, pp. 611–624.
- [9] M. Gerla, K. Tang, and R. Bagrodia. TCP Performance in Wireless Multi-hop Networks. In *Proceedings of IEEE WMCSA*, 1999.
- [10] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, October 1998.
- [11] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part ii - the hidden terminal problem in Carrier Sense Multiple-Access modes and the busy-tone solution," *IEEE Transactions on Networking*, vol. 23, no. 12, pp. 1417–1433, 1975.

- [12] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash. A Feedback-based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks. In *Proceedings of 18th. International Conference on Distributed Computing Systems*, pages 747–479, May 1998.
- [13] K. Chin, J. Judge, A. Williams, and R. Kermode, “Implementation experience with MANET routing protocols,” *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 5, pp. 49–59, Nov. 2002.
- [14] IEEE Computer Society LAN/MAN Standards Committee. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Standard 802.11-1999, 1999.
- [15] K. Xu, M.Gerla, and S. Bae, “Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks,” *Ad Hoc Networks Journal, Elsevier*, vol. 1, no. 1, pp. 107–123, Jul. 2003.

Author:



*B.Sc and M.C.A from Utkal University, Orissa, India, M.TECH (Computer Science),
PhD (Computer Science (MANET) (Pursuing)) Jawaharlal Nehru University, New Delhi, India.
Presently working as Assistant Professor AND College, University of Delhi, New Delhi, India.*