International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 2, Issue 3, pp: (209-225), Month: July - September 2014, Available at: www.researchpublish.com

Implementation of Wired and Wireless Networks, Analysis Simulation and Result Comparison Using Ns2

[1]S.Jeneeth Subashini, [2]D. Guna Shekar, [3]C.Harinath Reddy,
[4]M. Manikanta

[1,2,3,4] ECE, Final year students, Saveetha school of Engineering, Saveetha University-602105, India

I. INTRODUCTION

Communication refers to exchange of information between two entities. Communication networks in general divide into two main categories:

1. Wired networks.
2. Wireless networks.

Wired networks exist between a number of devices connected to each other using connecting media, such as cables and routers. Wired networks can be applied within an area limited by the cables and routers that allow for sending and receiving of data. Wireless networks, on the other hand, are free of such space limitations and are more easily able to connect different devices to each other. Wireless nodes can play the roles of both hosts and routers, which forward the packets to neighboring nodes. In recent days, wireless networks are mostly used than wired networks. The reasons for using wireless networks are cost effectiveness of network deployment and its applicability to environments where wiring is not possible or it is preferable solution compared with wired networks. When designing wireless networks and/or studying their behaviour under various conditions, software simulation tools are often used.

Whether it is a wired or wireless network, it should have a network topology. A communication network topology refers to the schematic representation of switching elements, routers, transmission links and other peripherals. It acts as a layout for communication network which exists in real time scenario. Network topologies can be classified into two types based on the layouts:

1. Physical network layout
2. Logical network layout

Physical network layout is the actual layout of computer cables, routers and other network devices. Logical network layout is the way in which network appears to the devices in use. Commonly used topologies are bus, ring and star. The analysis of a sophisticated network becomes difficult as there is a hectic calculation involved at each and every node spaced at a 100-200 km interval.

An infrastructure wireless network consists of an access point or a base station. In this type of network the access point acts like a hub, providing connectivity for the wireless computers. It can connect or bridge the wireless LAN to a wired LAN, allowing wireless computer access to LAN resources, such as file servers or existing Internet Connectivity.

There are four basic types of transmissions standards for wireless networking. These types are produced by the Institute of Electrical and Electronic Engineers (IEEE). These standards define all aspects of radio frequency wireless networking. They have established four transmission standards; 802.11, 802.11a, 802.11b, 802.11g.

The basic differences between these four types are connection speed and radio frequency. 802.11 and 802.11b are the slowest at 1 or 2 Mbps and 5.5 and 11Mbps respectively. They both operate off of the 2.4 GHz radio frequency. 802.11a operates off of a 5 GHz frequency and can transmit up to 54 Mbps and the 802.11g operates off of the 2.4 GHz frequency and can transmit up to 54 Mbps. Actual transmission speeds vary depending on such factors as the number and size of the physical barriers within the network and any interference in the radio transmissions.

Wireless networks are reliable, but when interfered with it can reduce the range and the quality of the signal. Interference can be caused by oher devices operating on the same radio frequency and it is very hard to control the addition of new devices on the same frequency. Usually if your wireless range is compromised considerably, more than likely, interference is to blame.

A major cause of interference with any radio signals are the materials in your surroundings, especially metallic substances, which have a tendency to reflect radio signals. Needless to say, the potential sources of metal around a home are numerous--things like metal studs, nails, building insulation with a foil backing and even lead paint can all possibly reduce the quality of the wireless radio signal. Materials with a high density, like concrete, tend to be harder for radio signals to penetrate, absorbing more of the energy. Other devices utilizing the same frequency can also result in interference with your wireless. For example, the 2.4GHz frequency used by 802.11b-based wireless products to communicate with each other. Wireless devices don't have this frequency all to themselves. In a business environment, other devices that use the 2.4GHz band include microwave ovens and certain cordless phones. On the other hand, many wireless networks can increase the range of the signal by using many different types of hardware devices. A wireless extender can be used to relay the radio frequency from one point to another without losing signal strength. Even though this device extends the range of a wireless signal it has some drawbacks. One drawback is that it extends the signal, but the transmission speed will be slowed.

There are many benefits to a wireless network. The most important one is the option to expand your current wired network to other areas of your organization where it would otherwise not be cost effective or practical to do so. An organization can also install a wireless network without physically disrupting the current workplace or wired network. Wireless networks are far easier to move than a wired network and adding users to an existing wireless network is easy. Organizations opt for a wireless network in conference rooms, lobbies and offices where adding to the existing wired network may be too expensive to do so.

Need For Simulation

The cost of building test-bed or actual systems for performance analysis is sometimes not effective or even not feasible. The intensity of the problem increases, as more and more real world applications deploying mobile agents are proposed and each need different configuration parameters for performance studies. For these reasons, it is necessary to build a simulation model of the existing network topology and study it as a surrogate for an actual system.

A network simulator is a piece of software that predicts the behavior of network, without an actual network being present. In Communication and Computer network, network simulation is a technique where a program models the behavior of a network either by calculating the interaction between the different network entities (hosts, nodes) using mathematical modeling or actually observing the traffic flow between the nodes in the network. Most of the commercial simulators are Graphical User Interface (GUI) driven, while some network simulators are Command Line Interface (CLI) driven. The network model / configuration describe the state of the network (nodes, routers, switches and links) and the events (data transmissions, packet error etc.). An important output of simulations is the trace files. Trace files log every packet, every event that occurred in the simulation and are used for analysis. Network simulators can also provide other tools to facilitate visual analysis of trends and potential trouble spots. Most network simulators use discrete event simulation, in which a list of pending events is stored, and those events are processed in order, with some events triggering future events such as the event of the arrival of a packet at one node triggering the event of the arrival of that packet at a downstream node. Considering the above facts, a proper simulation tool should be selected in order to simulate the network model. The following are some of the factors:

Stress Test of the Network:

Stress testing means ratcheting utilization levels up over 90 percent. Simulating every packet flowing through a congested network leads to very long simulations. However, it would be better to select some analytical simulation tools that blithely report link utilization above an impossible 100 percent or tools that do not accurately reflect how protocols like TCP respond to packet loss at high loads. It is better to select such a tool that can distinguish between the load offered to the network and the load actually carried by the network.

Balance Power and Ease of Use:

It is desired that the simulation tool should have balanced power utilization feature. Balance power utilization comes from the ability to precisely configure or customize device and protocol behavior. Some tools offer both full-featured designer versions and less-expensive runtime versions.

Confirm That Devices Are Connected Correctly:

Most tools ensure that devices and links are correctly matched and give a warning if, for example, a router is not compatible with the optical link it's connected. But few tools confirm that LANs are within their maximum lengths and that the network correctly compensates for delays over long-distance WANs.

Provide Statistically Useful Data:

Simulation tools should be selected in such a way that it should provide desirable results. Most tools simply compute average delay and utilization. But it is desired to calculate the delay at heavy traffic. The simulation tools should have a flexibility to provide such results.

In addition to the above factors delay equalization, throughput optimization and simulation of each and every node in the network are also included.

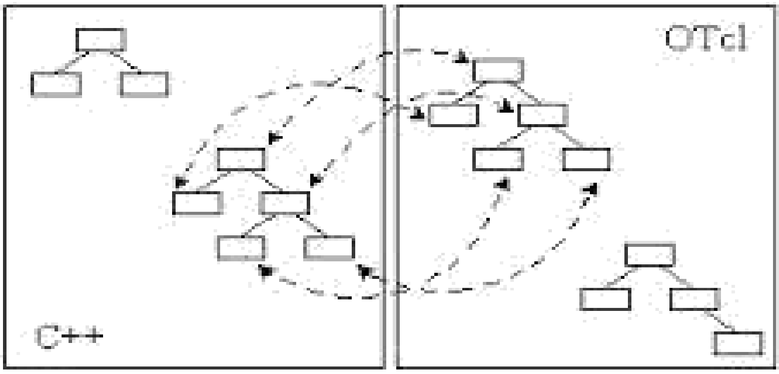
Network Simulator2 (NS2)

NS2 is an object-oriented, discrete event-driven network simulator developed at University of California Berkeley written in C++ and OTcl. NS2 is very useful for developing and investigating variety of protocols. They mainly include protocols regarding TCP behavior, router queuing policies, multicasting, multimedia, wireless networking and application-level protocols.

Software Architecture

NS software promotes extensions by users. It provides a rich infrastructure for developing new protocols. Also, instead of using a single programming language that defines a monolithic simulation, NS uses the split-programming model in which the implementation of the model is distributed between two languages. The goal is to provide adequate flexibility without losing performance. In particular, tasks such as low-level event processing or packet forwarding through simulated router require high performance and are not modified frequently once put into place. Hence, they can be best implemented in compiled language like C++. On the other hand, tasks such as the dynamic configuration of protocol objects and exploring a number of different scenarios undergo frequent changes as the simulation proceeds. Hence, they can be best implemented in a flexible and interactive scripting language like OTcl. Thus, C++ implements the core set of high performance primitives and the OTcl scripting language express the definition, configuration and control of the simulation.

C++ - OTcl Linkage



NS supports a compiled class hierarchy in C++ and also similar interpreted class hierarchy in OTcl. From the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and a class in the compiled hierarchy. The root of this class hierarchy is the class Tcl Object. Users create new simulator objects through the interpreter. These objects are instantiated within the interpreter and are closely mirrored by a corresponding object in the compiled hierarchy. The interpreted class hierarchy is automatically established through methods defined in class Tcl Class while user instantiated objects are mirrored through methods defined in class Tcl Object.

Fig 1 C++ & OTcl linkage the following classes are mainly responsible for maintaining C++ and OTcl linkage.

i. **Class Tcl**: This class encapsulates the actual instance of the OTcl interpreter, and provides methods to access and communicate with that interpreter. It provides methods for obtaining a reference to Tcl instance, invoking OTcl procedures through the interpreter, getting or passing the results to the interpreter, storing and looking up "Tcl Objects" *etc*.

ii. **Class Tcl Object**: It is the base class for most of the other classes in the interpreted and compiled hierarchies. Every object in the class Tcl Object is created by the user from within the interpreter and an equivalent shadow object is created in the compiled hierarchy. The class Tcl Class performs this shadowing.

iii. **Class Tcl Class**: This is a pure virtual compiled class. Classes that are derived from this base class provide two functions: constructing the interpreted class hierarchy to mirror the compiled class hierarchy- and providing methods to instantiate new Tcl Objects.

iv. **Class Embedded Tcl**: The objects in this class are responsible for loading and evaluating some NS scripts that are required at initialization.

v. **Class InstVar**: This class defines the methods and mechanisms to bind a C++ member variable in the compiled shadow object to a specified OTcl instance variable in the equivalent interpreted object. This binding allows setting or accessing the variable from within the interpreter or compiled code at all times.

NS2

NS2 is a publicly available common simulator with support for simulations of large number of protocols. It provides a very rich infrastructure for developing new protocols. It also provides the opportunity to study large-scale protocol interaction in a controlled environment. Moreover, NS software really promotes extension by users. The fundamental abstraction the software architecture provides is "programmable composition". This model expresses simulation configuration as a program rather than as a static configuration. The compiled C++ hierarchy allows us to achieve efficiency and faster execution times. This is in particular useful for the detailed definition and operation of protocols. The otcl makes use of objects compiled in C++ through the otcl linkage discussed above, which creates a matching otcl object for each of the C++.NS also has certain disadvantages. It is a large system with a relatively steep learning curve. NS's dual language implementation is proving to be a barrier to some developers. But increasing awareness among the researchers along with the other tools like tutorials, manuals and mailing lists have improved the situation.

II. LITERATURE REVIEW

Almargni Ezreik and Abdalla Gherayani, 2012 proposed the method to use simulation for design and analysis of wireless networks. In order to achieve this defined task, they studied various networks and briefly described about basic wireless network categories. Using the NS2 as a simulation tool, they calculated the throughput, packet loss and the average end to end delay of the packets. In the proposed paper wireless telecommunications networks are generally implemented with some type of remote data transmission system that uses EM waves for the carrier and this usually takes place at the Physical layer. This may be a one way communication as in broadcasting systems, (such as radio and TV) or a two way communication (e.g. mobile phones). Wireless networks transfer data such as e-mail support and files, but advancements in the performance of wireless networks is enabling support for voice and video communications. The author's perspective of wireless network performance depends mainly on the average throughput and end to end delay. Real time applications such as Voice over Internet Protocol (VoIP) are highly sensitive to delay but they will function satisfactorily with a little bandwidth. In this paper, the simulation scenario is aimed at simulating the network performance through network throughput, packet drop rate and average packets end to end delay.

For determining the above factors the authors opted the following process. In order to determine throughput for each node, calculate the bytes received by each node using NS2 simulation tool.

i. In order to determine the packet losses, using NS2 we should calculate the bytes that are transmitted and not received by any node.

ii. Determining the packets end to end delay, NS2 should calculate the difference time of last packet received and number of all packets received

With the above calculations it has been concluded that wireless network simulators provide full control to researchers in investigating traffic flow behaviour, but not always reflect real world scenarios impeccably.

Johanna Antila (2002) proposed the performance of TCP model by relatively comparing the throughput of the two analytical models (TAHOE and RENO). For this he gave emphasis on TCP's congestion control and performance.

The structure of the study in the first part, instructions for the simulations were given which consisted of theory explaining TCP's congestion control algorithms and the analytical models for its throughput apart from the main features of the NS2 simulator used. Finally in the second part of the study, simulation results scenarios were presented and analysed. In the implementation of the process, the TCP's congestion control has the following overview.

The most important aspect of the TCP model is that it works or implements a window based control mechanism. A window based protocol means that a current window size in actual defines a strict upper bound on the amount of unacknowledged data that can be in transit between a given sender and a receiver. The system or protocol in general cannot be 100% efficient, which implies that even in TCP model/protocol there were no efficient utilization of resources. Hence to prevent the TCP senders from overrunning the resources of the network a special congestion control algorithm was released.

In TAHOE (1988) has provided three congestion control algorithms: slow start, congestion avoidance and fast re-transmit. Now the basic three parameters that form the basis of the TAHOE or RENO model are viz., receiver's advertised window (awnd), TCP's congestion window (cwnd) and the slow start threshold (ssthresh).

The window size of the sender 'w' for any three congestion algorithm supported by TAHOE is defined by:

w = min (cwnd,awnd) which certainly is not equal to awnd as opposed to contemporary logic.

The simple model as described by the author comprises of an upper bound on TCP's average sending rate that applies to any conformant TCP. A conformant TCP is defined in as a TCP connection where the TCP sender adheres to the two essential components of TCP's congestion control: First, whenever a packet drop occurs in a window of data, the TCP sender interprets this as a signal of congestion and responds by cutting the congestion window at least in half. Second, in the congestion avoidance phase where there is currently no congestion, the TCP sender increases the congestion window by at the most one packet per window of data. Thus, this behaviour corresponds to TCP RENO in the presence of only triple duplicate loss indications. Now for further insight a steady state model is assumed. It is also assumed for the purpose of the analysis that a packet is dropped from a TCP connection if and only if the congestion window has increased to W packets. Because of the steady state model the average packet drop rate, p, is assumed to be non bursty. The TCP sender follows the two components of TCP's congestion control as mentioned above. When a packet is dropped, the congestion window becomes halved. After the drop, the TCP sender increases linearly its congestion window until the congestion window has reached its old value W and another packet drops.

Behdad Jamshidi and Victor Mateescu (2012) proposed that VoIP network is a real time application of NS2 in which there are different protocols that are used to send voice information between two terminals. They are User Data Gram (UDP) protocol, Transmission Control Protocol (TCP) and Real Time Transport Protocol(RTP).

In the paper the authors refereed that VoIP has following advantages:

i. Low cost.

ii. Integrate different web services with VoIP.

iii. Possibility for greater bandwidth and efficiency.

They compared the voice information over three types of protocols and the study is sectioned into three parts: Voice over TCP, Voice over UDP and Voice over RTP.

Since IP is a best – effort protocol, it does not prevent errors such as packet loss from occurring. The Transmission Control Protocol (TCP) serves to mitigate this flaw by detecting errors, retransmitting lost packets and ensuring data is ordered properly in order to provide reliable service. TCP is a connection oriented protocol that lies in the Transport layer. The connection is formed via a handshake between two hosts with connection requests and acknowledgements. Once the connection is formed, the data being transmitted is broken into segments. Before the segment is transmitted, a header is attached which contains a sequence number. The receiver will respond to the arriving packet with an acknowledgement if no errors are found. If no acknowledgement arrives at the original sender after a certain timeout period, the sender will re-transmits the packet.

Despite its reliable service, the main drawbacks of TCP are the long delays inherent in taking such preventive measures. One of the main challenges of VoIP is the delay restriction in a real –time phone call. According to the ITU-T Recommendation G.114, one – way delay should be no more than 400 milliseconds for international calls. However, maintaining constant high quality audio during the phone call is also an important aspect i.e. one that is challenged by packet loss and errors during transmission. In general, large delays are more undesirable than data loss with regards to voice communications due to the “real – time“ aspect.

With reference to the voice over UDP, it has been observed that UDP is a simple protocol that passes data along from the application layer to IP to be transmitted. It performs none of the error checks that TCP does and is therefore unreliable. An UDP header merely consists of an optical source part, destination port, the length of the datagram, and a checksum. As previously mentioned, the main reason for UDP over TCP in VoIP applications is the reduced delay. In general, the sporadic loss of packets in a conversation will not be able as disruptive as excessively long delay times. In fact, a packet loss of about 5% is said to be tolerable depending on how the losses are distributed. We will investigate how well a UDP based VoIP network performs in contrast to its TCP counterpart.

Coming to voice over RTP, the RTP is an application layer protocol that attaches itself to UDP to provide added benefits for real time applications. An RTP header includes a sequence number to help preserve the order of transmitted packets. It also includes a timestamp, which is meant to provide information to the destination application so that it may compensate for problems such as delay or jitter if they arise. The optional companion protocol, RTCP (specified in RFC 3550), is used as a means of exchanging information on session quality, which can include the number of lost packets or the average delay time. RTP strikes a balance between UDP and TCP for VoIP applications. It is designed to operate rapidly like UDP and it provides the receiver with valuable information pertaining to the VoIP session. The receiving application can then use this information to alleviate problems caused by out-of-sequence packets and jitter, thus improving the quality of the session. RTP is the protocol of choice for streaming media over the internet and is widely used in VoIP applications.

III. NETWORK MODEL

The author referred to a standard circuit switched network. In their implementation, the authors assumed the commonly used G.711 codec, which transmits the information at a rate of 64 kbps. In a standard circuit- switching network, an analog voice signal must be sampled at twice it maximum frequency at 8 bits per sample. Standard human speeches reaches about 4000 kHz, thus a bandwidth of 64kbps is required. The advancement of codec technology has improved bandwidth efficiency in telephony by only transmitting information when a person is talking. Therefore, they deduced that a variable bit rate (VBR) as opposed to constant bit rate (CBR) on each end is required to accurately simulate a VoIP call. While there is low to moderate background traffic present in the network, There seems to be no loss in performance unless the network is under a maximal load, at which large drops in throughput are observed from the both ends.

In their implementation, the authors considered a circuit switched network as shown below:

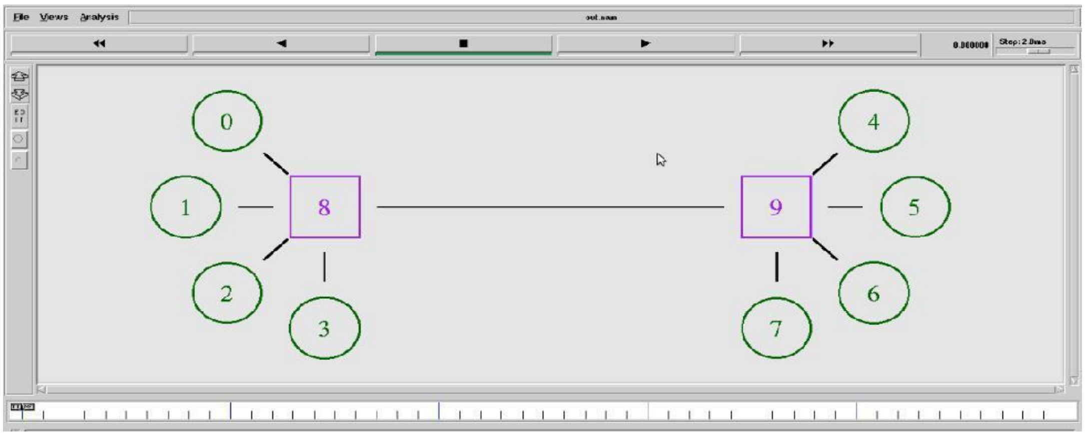


Figure 2: NS2 implementation of a VoIP network

The scale of the simulation was chosen to be nation – wide, thus the link between routers represented by node 8 and 9 was chosen to be an Optical Carrier Level-1 (OC-1) line, which was a bandwidth of 51.84 Mbps. The link used in the simulation is a duplex link and thus has its bandwidth set to 25.92 Mbps to properly mimic the OC-1 and a delay of 35ms.

The background traffic of the network is supplied by Nodes 1, 2, 3,5,6 and 7at constant bit rates. As the simulation begins, Nodes 1 and 5 create background traffic at a rate of 25.89 Mbps each (while all other background sources are turned off), providing a sub-maximal load for duplex link between the two routers. Then from 20s to 40s, Nodes 2 and 6 are turned on (while Nodes 1 and 5 turn off) to provide background traffic of 25.91 Mbps each.

Now all the three parameters were calculated viz. throughput, packet loss and end to end delay were calculated and with the results obtained, the author concluded that RTP is a reliable protocol for using VoIP.

The higher the RTT, the fewer throughputs. The report investigates how a higher RTT can deliver a higher throughput, in contrast with normal TCP calculation rules.

The author described a phenomenon called network phase effect which means TCP traffic shows a strong periodicity at bottleneck links, as a result of most packets carrying maximum payload. Traffic sources that tune their RTT to be in phase with the transmissions at that bottleneck are known to obtain a higher throughput at that link. In today’s network topology, anot-uncommon scenario has two or more TCP flows sharing a common bottleneck link, e.g. multiple users on a LAN, sharing a DSL link. When competing for bandwidth on that bottleneck, the roundtrip time (RTT) is a determining factor for the throughput obtained by these flows. In general, a higher RTT means a lower throughput, under general conditions, flows share the available bandwidth by the inverse of their RTT ratio.

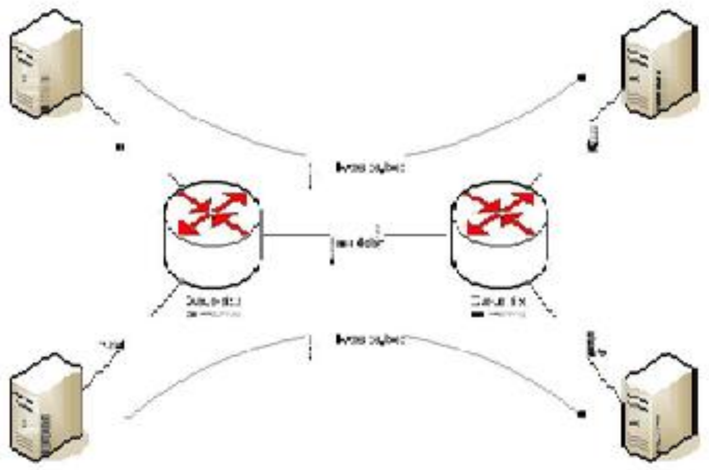


Figure 3: Lay-out of network in the ns2 simulation

The author reported that above network has been taken into consideration for his project work. The network consists of:

Two source nodes (1 and 2) connected to the first router via 100 Mbit/s links Two end nodes (5 and 6) connected to the second router via 100 Mbit/s links Two routers, interconnected via a 100 Kbit/s link Only two flows exist: node 1 sends to node 5, and node 2 to node 6, accompanied by their respective acknowledgment return flows. All links are lossless, but packets may be dropped at the routers when their queues fill up to their maximum of 10 packets.

All traffic is TCP (New Reno), packet size is set to 1,000 bytes; network protocol is

IPv4.The 100 Mbit/s Ethernet links for the 1to 5 flow are set to a fixed delay of 5 ms, while delays for the links for 2 to 6 are variable (and set between 5 and 25 ms). The first flow experiences a smaller RTT and is therefore expected to be the ‘faster’ connection. Given that the clear bottleneck for both flows is the 100 Kbit/s link between the routers, one may expect a quick fill of the queue at the first router, causing packets to be dropped and the packet windows of both flows being adjusted in classic TCP form (the ‘saw tooth’).

The simulation is performed using Ns2 and the simulation results are plotted which is shown below.

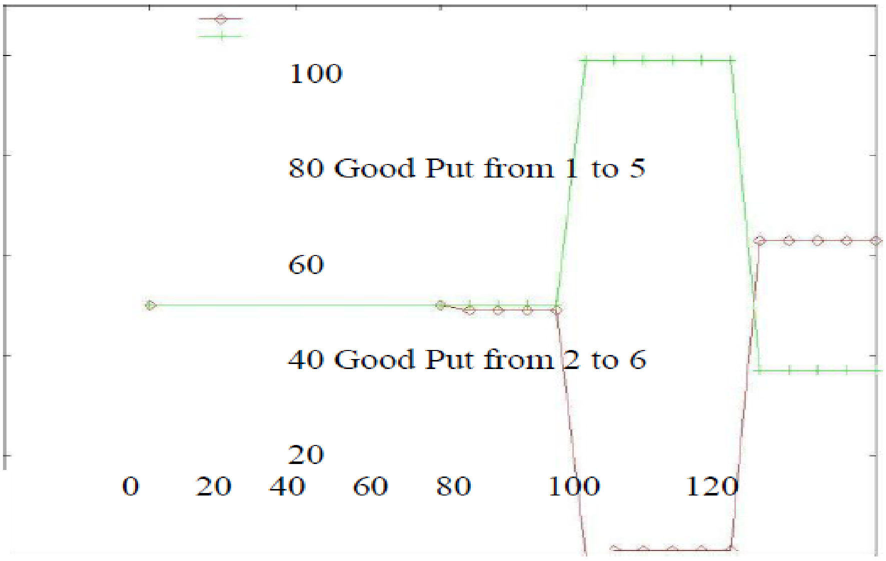


Figure 4: Graph showing the simulation results

For delays up to 19ms per link on the variable delay links the good put is shared equally between the two links, giving both flows around 50% of the capacity on the 100 Kbit/s link. This is shown in Figure 2.3. The TCP windows of both flows increase to 6 (sending 12 packets to the first router) which causes packets to be dropped since no more than 10 can be accommodated in the queue, the window decreased to half the size and thereafter increasing to 6 again. With this the author explained the network phase effect which can be seen as a phase effect in a traffic flow through a network. Here, phase denotes the time between the ends of transmission queue. Because all traffic in this setup is of the same size (i.e. 1000 bytes), all packet transmissions over the bottleneck link take the same time, making the RTT a multiple of the transmission time.

Adding delay in the amount of this transmission time to another part of the total connection can make a flow to a non-multiple of the transmission time makes that flow loose out over a greater period of time. End hosts can use this method to their advantage; especially if they 'know' that the major part of the RTT is caused at one particular part of the link, and provided almost all traffic is of same size.

In the physical interference model, a signal propagation model is used to determine transmission strength, which decays with distance. A transmission is received correctly if the signal to noise ratio of the transmitted signal at the receiver is above a specified threshold. In this model, one of two transmissions that interfere with each other can be properly received if its signal strength at the receiver is much greater than the competing signal's strength. The specification of the physical interference model is dependent on the details of the signal propagation model, for which there are several choices. Thus, for the physical interference model, high transmission powers do not negatively impact capacity and topology control will not improve capacity. However, in the protocol interference model, high transmission powers can negatively impact capacity and it is worth investigating the potential capacity and throughput benefits of topology control. Now another and one of the most crucial factor taken into consideration by the authors for the paper was the Impact of MAC approach. The MAC approach comprised of the following features: In this paper, the major focus on MAC protocols was based on CSMA/CA. This includes the MAC protocols of several standard wireless network technologies, e.g. 802.11 and 802.15.4. In CSMA/CA, when a node has a packet to send, it delays its transmission until it senses that the communication channel is free. Thus, the only way that two transmissions within range of each other can occur at the same time (thus interfering) is if the transmissions are started at the same moment so that both senders think that the channel is free. Random delays are inserted in CSMA/CA at each pending sender when the channel becomes free in order to reduce the probability of simultaneous start times. The effect of CSMA/CA is to produce network capacity characteristics that are more like those with protocol interference rather than physical interference. This is because concurrent transmissions that are within range to interfere with each other are prevented with high probability. Thus, CSMA/CA will prevent A and C from transmitting simultaneously in the example mentioned by the authors to support their research. Thus, CSMA/CA networks will potentially sacrifice network capacity unless they can reduce interference, possibly through the use of topology control. Now the simulation model considered by the authors laid emphasis on node densities. In this model, n nodes are independently and uniformly distributed over the deployment region (1 km x 1 km in the simulations). Each node sends constant bit rate (CBR) traffic to a random destination selected as follows. For each source node, a uniformly distributed point in the deployment region is selected. The destination for this source is then chosen as the node (other than itself) that is closest **About the NS Simulator.**

Simulator provides a set of interfaces for configuring a simulation and for choosing the type of event scheduler used to drive the simulation. A simulation script generally begins by creating an instance of this class and calling various methods to create nodes, topologies, and configure other aspects of the simulation. A subclass of Simulator called Old Sim which is used to support *ns* v1 backward compatibility. To create a simulator object, the command is as follows

```
set ns [new Simulator]
```

Now we open a file for writing code that is going to be used for the nam trace data.

```
set nf [open out.nam w]

$ns namtrace-all $nf
```

The first line opens the file 'out.nam' for writing and gives it the file handle 'nf'. In the second line we tell the simulator object that we created above to write all simulation data that is going to be relevant for nam into this file. The next step is to add a 'finish' procedure that closes the trace file and starts nam.

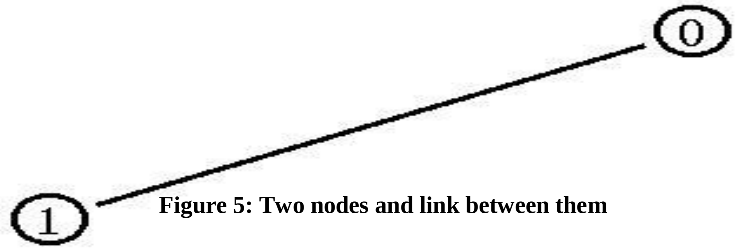


Figure 5: Two nodes and link between them

Agents and Applications

Having defined the topology (nodes and links) we should now make traffic through them. To that end, we need to define routing (in particular sources, destinations) the agents, protocols and applications that use them. In our network, we preferred to run a File transfer Protocol (FTP) application between node 0 and node 4, and a Constant Bit Rate (CBR) application between node 1 and node 5. The internet protocol used by FTP is TCP/IP, and one used by CBR is User Datagram Protocol (UDP).

FTP Over TCP

TCP is a dynamic, reliable congestion control protocol. It uses acknowledgements created by the destination to know whether the packets are well received, lost packets are interpreted as congestion signals, why TCP thus requires bidirectional links in order for the acknowledgements to return to source.

There are number of variants of the TCP protocol, such as Tahoe, Reno, Newreno, Vegas, type of agent appears in the first line:

```
set tcp [new agent/TCP]
```

This command also gives a pointer called “tcp” here to the TCP agent, which is an object in NS. The command $ns attach-agent $n0 $tcp defines the source node of the TCP connection. The command Set sink [new agent/TCP sink] defines the behavior of the destination node of TCP and assigns it to a pointer called sink. The command $ns attach-agent $n4 $sink defines the destination node. The command $ns connect $tcp $sink finally makes the TCP connections between the source and the destination nodes.

TCP has many parameters with initial fixed defaults values that can be changed if mentioned explicitly. For example, the default TCP packet size has a size of 1000 bytes. This can be changed to another value say 552 bytes, using the command $tcp set packetSize_552. Once the TCP connection is defined, the FTP application is defined over it. This is done in the last three lines of program coded below.

CBR over UDP

We define the UDP connection and the CBR application over it. A UDP source (Agent/UDP) and the destination (Agent/NULL) id defined in the similar way as in the case of TCP. For the CBR application that uses UDP, defines the source and transmission and packet size. Instead of defining the rate, in the command $CBR set rate _0.01 Mb, one can define the time interval between transmission of packets using the command $cbr set interval_0.005

Other characteristics of CBR are random, Which is a flag indicating whether or not to introduce random "noise" in the scheduled transmission times. It is off by default, and can be set to ON by typing

$cbr set random_1

The packet size can be set to some value (in bytes) using

$cbr set packet size_<packet size>

Scheduling Events

NS is a discrete event based simulation. The tcl script defines when event should occur. The initialization command set ns [new simulator] creates an event scheduler, and events are then scheduled using the format:

$ns at <time><event>

The scheduler is started when running ns, i.e., through the command $ns run.**Experiment with 6-Node Wired Network using NS2**

With a six node network chosen, definition of the links is as follows. In Fig 3.2, shows the network where the leaf nodes are duplex, Nodes 2-3 are having two simplex links.

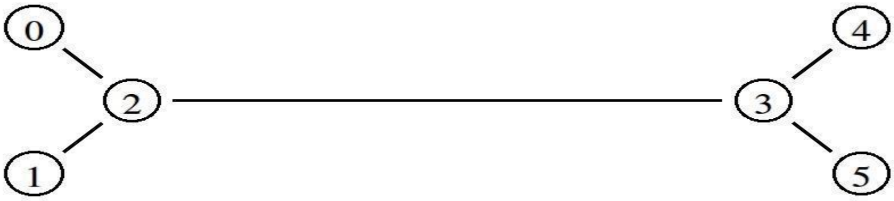


Fig 6: A six node network

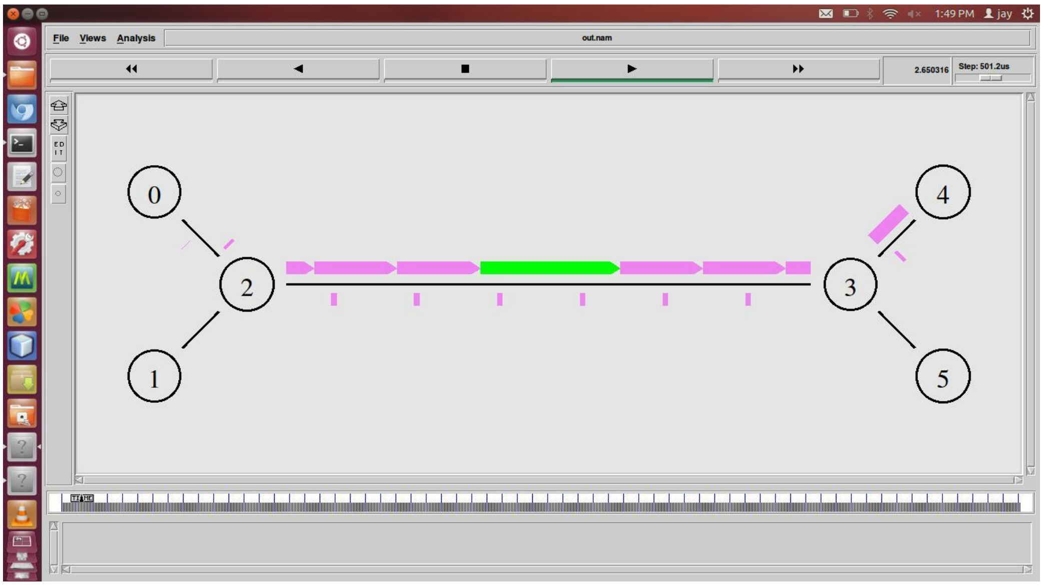


Fig. 7 The animation shows the packet flow in a 6 node network

IV. DESIGN OF AN ADHOC NETWORK WITH THREE NODES

Wireless Network is described as a network of devices which communicate by using Wireless technologies. They can be classified into diverse categories depending on the classification criteria. The following types of wireless networks satisfy diverse user requirements.

i. Wireless Personal Area Network (WPAN)

ii. Wireless Local Area Network (LAN)

iii. Wireless Metropolitan Area Network (MAN)

iv. Wireless Wide Area Network (WAN)

Wireless Personal Area Network (WPAN):

Personal Area Network includes Bluetooth devices which connect to a range of 10 meters. They support very minimal data rates and mostly used for data transfer only. Typical examples are wireless mouse connected with Bluetooth to a network.

Wireless Local Area Network (LAN):

They are typically privately owned data communication networks in which 10 to 100 devices share data resources. LAN generally supports two way communications and bit rate typically ranges from 10 to 100 Mbps. They are confined to small geographical area such as a building complex. It supports Bus, Star and Ring topologies. Office automation, factory automation, distributed computing, fire and security system, process control document distribution etc. are typical examples of LAN.

Wireless Metropolitan Area Network (MAN):

These are high speed networks which are usually designed to support entire city or cities. It covers a distance of 5-50 Kms of geographical area. It is generally privately owned. Generally, Broad band co-axial cables and optical fiber cables are employed. Data rates are of 50-150Mbps. Network structures are similar to LAN like star, bus and ring. It supports the transmission of both data and voice. Cable TV, Fiber Distributed Data Interface (FDDI), Asynchronous Transfer Mode (ATM) are few typical examples of this type.

Wireless Wide Area Networks (WAN):

These are the oldest type of data networks which provides relatively high speed and operates for a long distance transmission of data, voice and video information. Generally it covers a geographical area such as country or countries or entire continent. It supports a data rate of 1.5 Mbps-2.4 Gbps and supports a distance of more than 1000 kilometers. WANs are owned by public (or) corporate companies. Nearly, 1000-10000 workstations can be interconnected.

A Wireless Network performance depends mainly on the end to end throughput and average delay. Real time applications such as voice over IP are highly sensitive to delay but function satisfactorily with little bandwidth. At the other hand data transfer applications like FTP are insensitive to delay but require as much bandwidth as possible. For this, we have opted for a three wireless nodes which are connected by an ad hoc network. A wireless ad hoc network is a decentralized type of wireless network. The network is ad hoc because it does not rely on a pre existing infrastructure such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data for other nodes, so the determination of which nodes forward data is made dynamically on the basis of network connectivity. In addition to the classic routing, ad hoc networks can use flooding for forwarding data. An ad hoc network typically refers to any set of networks where all devices have equal status on a network and are free to associate with any other ad hoc network device in link range.

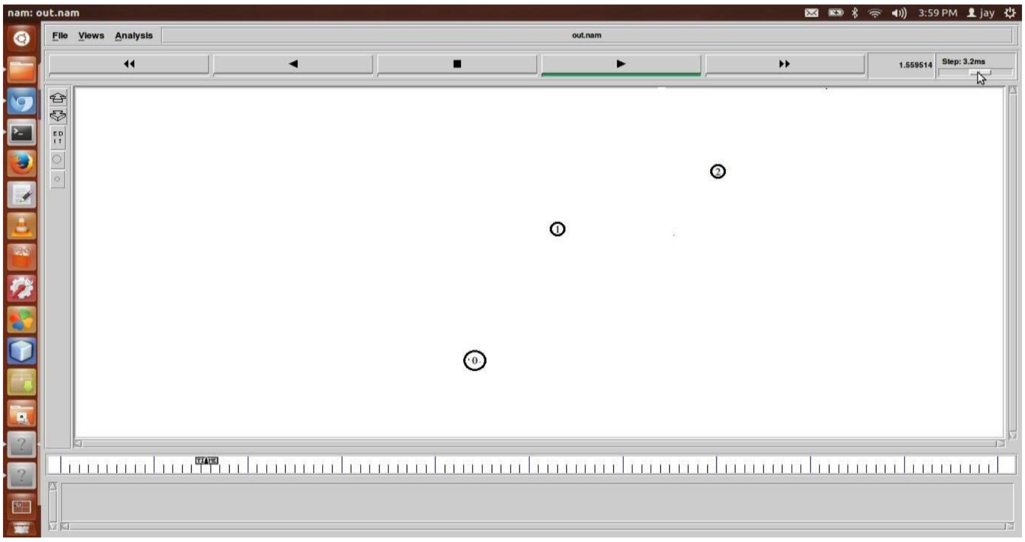
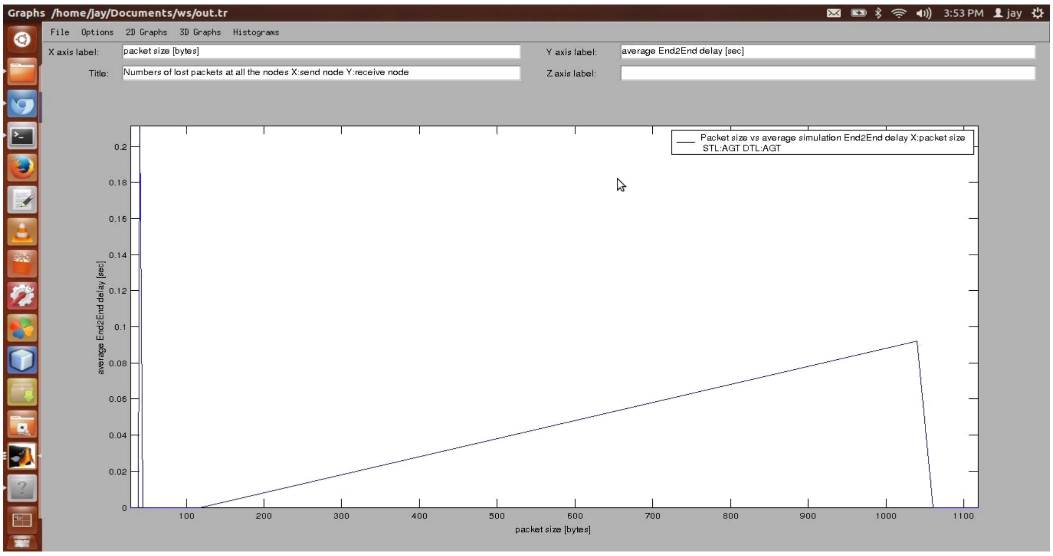


Fig. 8: NAM output showing three wireless nodes

In the Fig 4.1, wireless node ‘0’ is considered as source node and a TCP agent is attached to it. Wireless nodes ‘1’ and ‘2’ are receiver nodes or otherwise called as sink nodes.

Node ‘2’ acts as an intermediate node and relays the data packets sent by source node to the destination node. However, there is a drop in data packets when intermediate node moves towards the source node which results in loss of data. Employing a wireless scenario between the nodes is the major factor for the loss of packets. The performance of the ad hoc network designed will be evaluated by calculation of throughput which is defined as the number of successful transmission of packets in a predefined time slot and the end to end average delay between the nodes. The throughput and average delay for the above ad hoc network is calculated and the graphs are shown in respective figures Graph showing the through put for the ad hoc network.


Graphs /home/jay/Documents/ws/out.tr

X axis label: packet size [bytes]

Title: Numbers of lost packets at all the nodes X:send node Y:receive node

Y axis label: average End2End delay [sec]

Z axis label:

Fig. 9: Graph showing the end to end average delay of the ad hoc network.

From the above graphs it is incurred that the throughput for a given time period, initially increases as the intermediate node is fixed initially and hence there is no congestion. After some time, the intermediate node starts moving towards source node which results in loss of data packets between nodes ‘1’ and ‘2’ which results in decrease of through put. Similarly, the end to end delay also increases gradually as there is a movement in the intermediate node towards source node.

V. IMPLEMENTATION OF A SIXTEEN NODE WIRELESS NETWORK

Ad hoc On Demand Distance Vector Protocol (AODV)

The Ad-Hoc On-demand Distance Vector (AODV) routing protocol is one of several published routing protocols for mobile ad-hoc networking. Wireless ad-hoc routing protocols such as AODV are currently an area of much research among the networking community. Thus, tools for simulating these protocols are very important. Each AODV router is essentially a state machine that processes incoming requests from the network entity. When the network entity needs to send a message to another node, it calls upon AODV to determine the next-hop. Whenever an AODV router receives a request to send a message, it checks its routing table to see if a route exists. Each routing table entry consists of the following fields:

i. Destination address

ii. Next hop address

iii. Destination sequence number

iv. Hop count

If a route exists, the router simply forwards the message to the next hop. Otherwise, it saves the message in a queue, and then it initiates a route request to determine a route. Upon receipt of the routing information, it updates its routing table and sends the queued messages.

AODV nodes use four types of messages to communicate among each other. Route Request (RREQ) and Route Reply (RREP) messages are used for route discovery. Route Error (RERR) messages and HELLO messages are used for route maintenance. The following sections describe route determination and route maintenance in greater detail.

AODV Route Discovery

When a node needs to determine a route to a destination node, it floods the network with a Route Request (RREQ) message. The originating node broadcasts a RREQ message to its neighboring nodes, which broadcast the message to their neighbors. To prevent cycles, each node remembers recently forwarded route requests in a route request buffer. As these requests spread through the network, intermediate nodes store reverse routes back to the originating node. Since an intermediate node could have many reverse routes, it always picks the route with the smallest hop count. When a node receiving the request either knows of a “fresh enough” route to the destination, or is itself the destination, the node generates a Route Reply (RREP) message, and sends this message along the reverse path back towards the originating node. As the RREP message passes through intermediate nodes, these nodes update their routing tables, so that in the future, messages can be routed though these nodes to the destination. It is possible for the RREQ originator to receive a RREP message from more than one node. In this case, the RREQ originator will update its routing table with the most “recent” routing information; that is, it uses the route with the greatest destination sequence number.

Sequence Numbers

Each destination (node) maintains a monotonically increasing sequence number, which serves as a logical time at that node. Also, every route entry includes a destination sequence number, which indicates the “time” at the destination node when the route is created. The protocol uses sequence numbers to ensure that nodes only update routes with newer ones. Doing so, we also ensure loop- freedom for all routes to a destination.

All RREQ messages include the originator’s sequence number, and its (latest known) destination sequence number. Nodes receiving the RREQ add/update routes to the originator with the originator sequence number, assuming this new number is greater than that of any existing entry. If the node receives an identical RREQ message via another path, the originator sequence numbers would be the same, so in this case, the node would pick the route with the smaller hop count.

If a node receiving the RREQ message has a route to the desired destination, then we use sequence numbers to determine whether this route is “fresh enough” to use as a reply to the route request. To do this, we check if this node’s destination sequence number is at least as great as the maximum destination sequence number of all nodes through which the RREQ message has passed. If this is the case, then we can roughly guess that this route is not terribly out of-date, and we send a RREP back to the originator.

As with RREQ messages, RREP messages also include destination sequence numbers. This is so nodes along the route path can update their routing table entries with the latest destination sequence number.

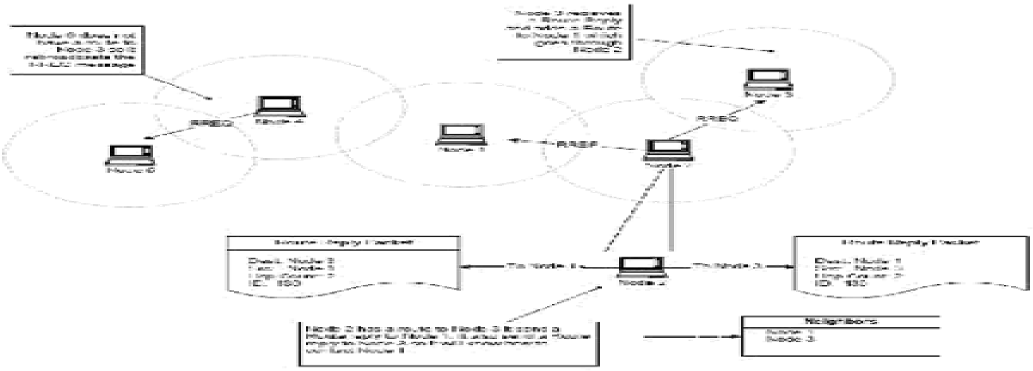


Fig 10: AODV routing mechanism

Dynamic Source Routing Protocol (DSR)

Dynamic Source Routing (DSR) is a routing protocol for wireless mesh networks. It is similar to AODV in that it forms a route on-demand when a transmitting computer requests one. However, it uses source routing instead of relying on the routing table at each intermediate device.

Determining source routes requires accumulating the address of each device between the source and destination during route discovery. The accumulated path information is cached by nodes processing the route discovery packets. The learned paths are used to route packets. To accomplish source routing, the routed packets contain the address of each device the packet will traverse. This may result in high overhead for long paths or large addresses, like IPv6. To avoid using source routing, DSR optionally defines a flow id option that allows packets to be forwarded on a hop-by-hop basis. This protocol is truly based on source routing whereby all the routing information is maintained (continually updated) at mobile nodes. It has only two major phases, which are Route Discovery and Route Maintenance. Route Reply would only be generated if the message has reached the intended destination node (route record which is initially contained in Route Request would be inserted into the Route Reply).To return the Route Reply, the destination node must have a route to the source node. If the route is in the Destination Node's route cache, the route would be used. Otherwise, the node will reverse the route based on the route record in the Route Request message header (this requires that all links are symmetric). In the event of fatal transmission, the Route Maintenance Phase is initiated whereby the Route Error packets are generated at a node. The erroneous hop will be removed from the node's route cache; all routes containing the hop are truncated at that point. Again, the Route Discovery Phase is initiated to determine the most viable route.Dynamic source routing protocol (DSR) is an on-demand protocol des

Implementation of sixteen node network using AODV routing protocol

A sixteen node ad hoc wireless network is defined and the AODV routing protocol is chosen in order to route the data packets from the source node to destination. The following figure shows the output of NAM for sixteen node ad hoc network.

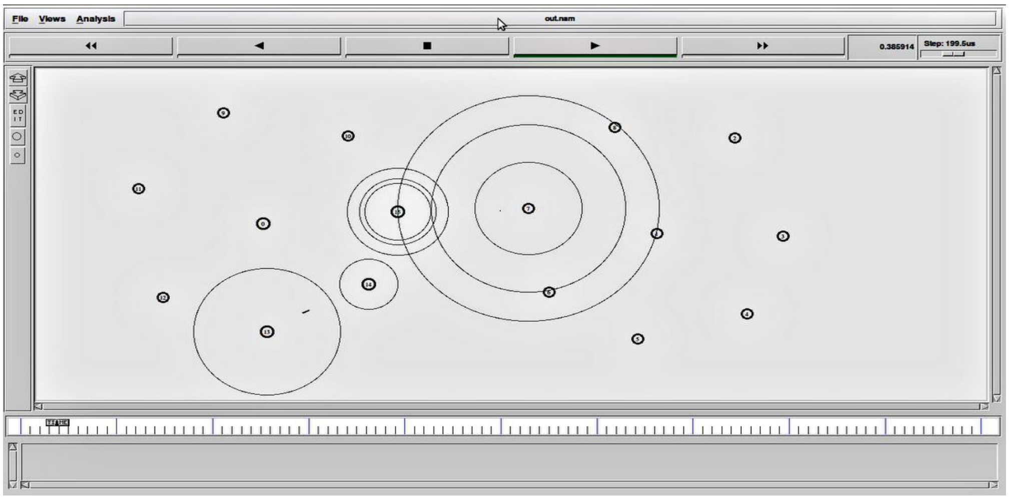
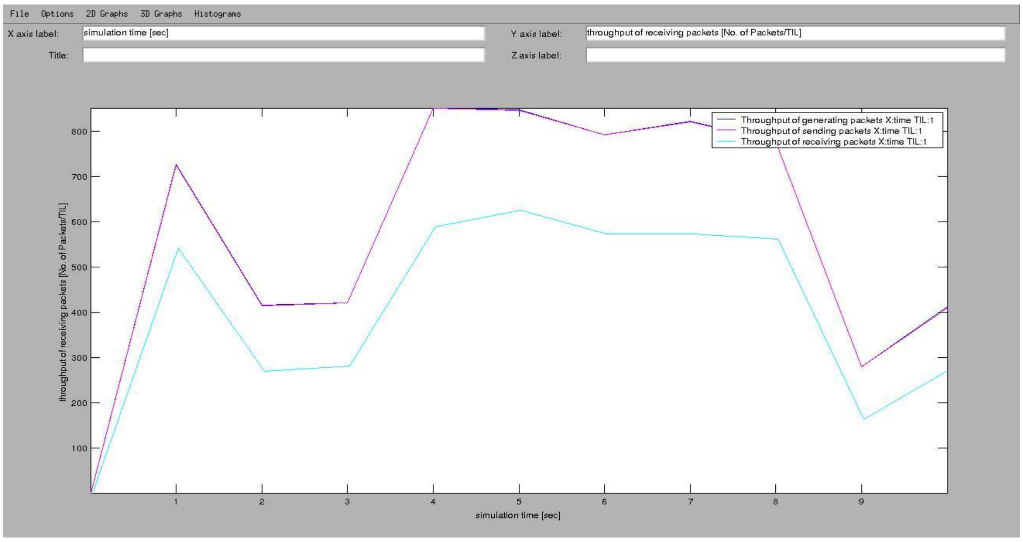


Fig 11: Output of NAM showing wireless network with sixteen nodes

For the above defined network, a source node and a destination node is defined. The AODV routing protocol chooses the shortest route in order to transmit the data packets efficiently. Once the route is identified, the source node transmits the data packets to destination. In our network, we chose node 13 as source and node 5 as destination. The source and destination node can be changed as per requirement. The node 13 generates an RREQ message to its neighboring nodes. The neighboring node with a smallest hop is determined and from the neighbor node the message will be transmitted to other nodes which have smallest hops from the current node. As these requests spread through the network, intermediate nodes store reverse routes back to the originating node. When the message reaches the destination node, the node generates a Route Reply (RREP) message, and sends this message along the reverse path back towards the originating node. As the RREP message passes through intermediate nodes, these nodes update their routing tables, so that in the future, messages can be routed though these nodes to the destination. For the wireless network, the performance is evaluated by calculating its parameters such as throughput. The following graphs show the throughput and average number of packets received.



X axis label: simulation time [sec]
Y axis label: throughput of receiving packets [No. of Packets/TIL]
Title:
Z axis label:

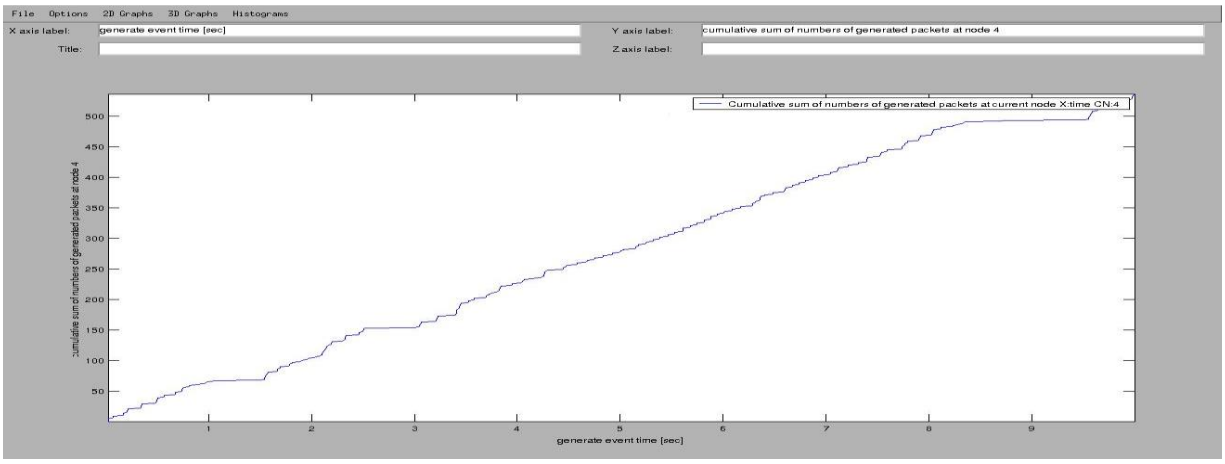
Fig 12 Graph Showing the throughput of wireless network at two different nodes

VI. CONCLUSION

In order to meet the complex specifications presented by modern sophisticated networks, network simulation provides a reliable alternative to calculate the performance of a given wired or wireless network. The NS2, which is a open source simulation software predicts the behavior of either a wired or wireless network which is wide spread over a range of kilometers. NS2 checks the functioning of any given network virtually and presents the performance characteristics of the network, which is a better alternative to tedious calculation of performance characteristics such as throughput, end to end delay and average number of packets received at each node.

In the project, It has been designed a six node wired network with TCL script as front end to the simulator. The TCL script uses compiled C++ hierarchy to achieve efficiency in simulation and faster execution time. After defining the topology using tool command language, agents such as FTP over TCP and CBR over UDP are attached to the respective nodes of the network. The traffic flow between the nodes allows us to calculate the performance of the network. The throughput of wired six node network has been calculated.

Extending these results to a wireless scenario, a three node ad hoc network has been defined. This is considered as an example of a Wireless Personal Area Network (WPAN). Compared to wired network, delay and packet drop rate is higher in a wireless network. For the ad hoc network, the throughput and the average end to end delay has been calculated. A wireless sixteen node network, which comes under


File
Options
2D Graphs
3D Graphs
Histograms
Cumulative sum of numbers of generated packets at current node X:time CN:4
cumulative sum of numbers of generated packets at node 4
500
450
400
350
300
250
200
150
100
50
1
2
3
4
5
6
7
8
9
generate event time [sec]

X axis label: generate event time [sec]
Title:
Y axis label: cumulative sum of numbers of generated packets at node 4
Z axis label:

Wireless Local Area Network (WLAN), has been designed and various routing protocols such as AODV, DSR and DSDV are investigated for the defined network. It is observed that AODV suits the best to the wireless network and we implemented AODV protocol. The throughput and other performance parameters are calculated for different nodes in the network as sources and destinations.

Future Scope of the Project

Using NS2, real time networks such as implementation of high speed LAN in a closed geographical area such as an apartment, office complex can be easily simulated and current implementations such as A Secure Intrusion Detection System for MANETs used in military applications, Fast And Secure Data Transmission In MANET for civilian applications and Self-Reconfigurable Wireless Mesh Networks used mainly in civil applications can be simulated which comes as further extension to our project.

REFERENCES

[1] Almargni Ezreik and Abdalla Gheryani , “ Design and Simulation of Wireless Network using NS-2,” 2nd (ICCSIT’2012) Singapore , April 2012

[2] Raid Alghamdi , John DeDourek and Przemyslaw Pochec , “ Evaluation and Improvements to Motion Generation in ns2 for Wireless Mobile Network Simulation,” (IJDIWC’2012)

[3] Yulei Wu , Keiqu Li and Bahman Javadi , “ Modelling and Analysis of Computer Networks in Multicluster Systems under Spatio Temporal Bursty Traffic,” IEEE Trans On Parallel and Distributed Systems , Vol 23 , NO. 5 , May 2012

[4] Mohammad Shahidul Hasan , Christopher Harding , Hongnian Yu and Alison Griffiths , “ Modelling Delay and Packet Drop in Networked Control Systems Using Network Simulator NS2,” (IJAC 2 ’2005) , 187-194

[5] Johanna Antila , “ TCP Performance Simulations using NS-2,” 2002

[6] Maarten Burghout , “ Experimental Analysis of the impact of RTT differences on the Throughput of two competitng TCP flows,” Individual Research Assignment , University Of Twente Netherlands, June 2008

[7] Behdad Jamshidi and Victor Mateescu , “ Voice over Internet Protocol (VoIP) in NS2 Simulation,” Ensc 427 Communication Networks Spring , 2012

[8] Kyu- Han Kim and Kang G Shin , “ Self- Reconfigurable Wireless Mesh Networks,” IEEE/ACM Trans On Networking , Vol 19, NO.2 , April 2011

[9] William Stallings , “Wireless Communications and Networking,” Prentice Hall, 2002

[10] T.S. Rappaport , “ Wireless Communications: Principles and Practices,” Prentice

[11] " The Network Simulator – NS2 , " [Online]. Available : http:// www.isi.edu/nsnam/ns /tutorial / , December 2010

[12] Eitan Altaman and Tania Jimenaz , " Network Simulator for Begginers," December 4, 2003

[13] antanu Santra and Pinaki Pratim Acharjya , " A Study and Analysis On Computer Network Topology For Data Communication,"(IJETAE), Volume 3 , Issue 1, January 2013

[14] Lalit Kishore Arora and Rajkumar , " Simulation and Analysis of Packet Loss in Mesh Interconnection Network with Source Routing ," (IJARCSSE), Volume 2 , Issue 6 , June 2012

[15] Ajay Singh and Dr . Pankaj Dashore , " Mobile Coverage Problem Analysis by using NS-2 ," (IJAIEM) , Volume 2 , Issue 8 , August 2013

[16] Douglas M Blough , Cyrus Harvest , Giovanni Resta , George Riley and Paolo Santi, " A Simulation Based Study on the Throughput Capacity of Topology in CSMA/CA Networks," a Research Paper supported By National Science Foundation , Atlanta ,2003

[17] Dr . Neeraj Bhargava , Dr . Ritu Bhargava , Anchal Kumawat and Bharat Kumar , " Performance of TCP – Throughput on NS2 by using different Simulation Parameters , (IJACR) , Volume 2 , Number 4 , Issue 6 , December 2012

[18] Transmission Control Protocol , http ://en.wikipedia.org/wiki/Transmission_ Control_ Protocol#cite_note -12