**INTRODUCTION TO NETWORK SIMULATION**

**NETWORK**

Network is an arrangement of intersecting horizontal and vertical lines. Computer networks allow computers to exchange data. In computer networks, networked computing devices pass data to each other along data connections. The connection between nodes is established using either cabled media or wireless media. The best known computer network is the Internet.

Network computer devices that originate, route and terminate the data are called Network nodes. Nodes can include host such as computers, mobile phones and servers as well as networking hardware. Two such devices are said to be networked together when one device is able to exchange information with the other device, whether or not they have a direct connection to each other. Computer networks support applications such as access to the World Wide Web, shared use of application and storage servers, printers, and fax machines, and use of email and instant messaging applications. Computer networks differ in the physical media used to transmit their signals, the communications protocols to organize network traffic, the network's size, topology and organizational intent.

**NETWORK SIMULATION**

In network research, 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/packets, etc.) using mathematical formulas, or actually capturing and playing back observations from a production network. The behavior of the network and the various applications and services it supports can then be observed in a test lab; various attributes of the environment can also be modified in a controlled manner to assess how the network would behave under different conditions. When a simulation program is used in conjunction with live applications and services in order to observe end-to-end performance to the user desktop, this technique is also referred to as **network emulation.**

**NETWORK SIMULATOR**

A network simulator is software or hardware that predicts the behaviour of a computer network without an actual network being present. In simulators, the computer network is typically modelled with devices, traffic etc. and the performance is analysed. Typically, users can then customize the simulator to fulfil their specific analysis needs. Simulators typically come with support for the most popular protocols and networks in use today, such as WLAN, Wi-Max, TCP, WSN, cognitive radio etc.

**SIMULATION SCENARIO**

Most of the commercial simulators are GUI driven, while some network simulators are 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.

Some network simulation problems, notably those relying on queuing theory, are well suited to Markov chain simulations, in which no list of future events is maintained and the simulation consists of transiting between different system "states" in a memory less fashion. **Markov chain simulation is typically faster but less accurate and flexible than detailed discrete event simulation.**

Simulation of networks is a very complex task. For example, if congestion is high, then estimation of the average occupancy is challenging because of high variance. To estimate the likelihood of a buffer overflow in a network, the time required for an accurate answer can be extremely large. Specialized techniques such as "control variates" and "importance sampling" have been developed to speed simulation.

**USES OF NETWORK SIMULATOR**

Network simulators serve a variety of needs. Compared to the cost and time involved in setting up an entire test bed containing multiple  networked computers, routers and data links, network simulators are relatively fast and inexpensive. They allow engineers, researchers to test scenarios that might be particularly difficult or expensive to emulate using real hardware - for instance, simulating a scenario with several nodes or experimenting with a new protocol in the network.

Network simulators are particularly useful in allowing researchers to test new networking protocols or changes to existing protocols in a controlled and reproducible environment. A typical network simulator encompasses a wide range of networking technologies and can help the users to build complex networks from basic building blocks such as a variety of nodes and links. With the help of simulators, one can design hierarchical networks using various types of nodes like computers, hubs, bridges, routers, switches, links, mobile units etc.

Various types of Wide Area Network **(WAN) technologies** like *TCP, ATM, IP* etc. and Local Area Network **(LAN) technologies** like *Ethernet, token rings* etc., can all be simulated with a typical simulator and the user can test, analyse various standard results apart from devising some novel protocol or strategy for routing etc. Network simulators are also widely used to simulate battlefield networks in Network-centric warfare.

There are a wide variety of network simulators, ranging from the very simple to the very complex. Minimally, a network simulator must enable a user to represent a network topology, specifying the nodes on the network, the links between those nodes and the traffic between the nodes.

* More complicated systems may allow the user to specify everything about the protocols used to handle traffic in a network.
* Graphical applications allow users to easily visualize the workings of their simulated environment.
* Text-based applications may provide a less intuitive interface, but may permit more advanced forms of customization.

**SIMULATION TOOLS**

**1. Traffic**

Developed by: E. Software

Available at: <http://www.members.iinet.net.au/~clark/>

Traffic version 2 is a simulation product designed to solve complex call-centre modelling problems, where traditional Erlang equations fail, but it can also be applied to any other queuing problem. It has an easy to use graphical interface and it runs under Windows.

**2. PhySim**

Developed by: Tetcos

Available at: <http://www.tetcos.com/physim.html>

PhySim allows for creation of Baseband, RF, Channel and Receiver chain and comes with inbuilt Oscilloscope, Spectrum Analyzer and Polar Plot. Techniques covered include ASK, FSK, PSK, DPSK, QPSK, PAM, PWM, PPM, TDM - PAM, AM-DSB-SC, SSB - SC , PCM, DM, Transmission Line, IF, RF, Dipole and AWGN. A free download of PhySim demo version is available at www.tetcos.com.

**3. NetSim**

Developed by: tetcos

Available at: <http://www.tetcos.com/software.html>

NetSim is developed by Tetcos and it is used for network lab experimentation and research. The protocols covered in simulation are aloha, slotted aloha, Ethernet - CSMA / CD, Fast Ethernet, Gigabit Ethernet, Token Ring, Token Bus, W Lan, X.25 Frame Relay, ATM, TCP, IP -Routing RIP, OSPF, BGP, MPLS, Wi-Max, Wireless Sensor Networks and Zigbee 802.15.4. The standard version features protocol source codes in C which can be modified and linked via the development environment. A free demo version can be downloaded from the website.

**4. Shunra Virtual Enterprise**

Developed by: Shunra

Available at: <http://www.shunra.com/network_simulation_products.aspx>

Shunra VE network simulation is a tool for pre-deployment testing and capacity planning and can be used for a wide variety of applications and projects. Shunra VE website claims its use by over 1700 customers within the financial services, retail, manufacturing, energy, technology and telecommunications, healthcare and pharmaceutical, travel, media and hospitality industries as well as government agencies and militaries. Shunra Virtual Enterprise (Shunra VE) is a powerful network simulation solution that creates an exact model of any production environment. This includes the network, remote locations, and the number and distribution of local and remote end-users. With Shunra VE, you can test the functionality, performance, scalability and robustness of any application or network infrastructure under current and future real-world conditions.

**5. OPNET**

Developed by: O. Technologies

Available at: <http://www.opnet.com/>

OPNET's suite of products combine predictive modelling and a comprehensive understanding of networking technologies to enable customers to design, deploy, and manage network infrastructure, network equipment, and networked applications. In particular OPNET Modeller is a development environment, allowing you to design and study communication networks, devices, protocols, and applications.

**6. GloMoSim**

Developed by: UCLA

Available at: <http://pcl.cs.ucla.edu/projects/glomosim/>

GloMoSim is a scalable simulation environment for wireless and wired network systems. It employs the parallel discrete-event simulation  capability provided by Parsec. GloMoSim currently supports protocols for a purely wireless network. In the future, we anticipate adding functionality to simulate a wired as well as a hybrid network with both wired and wireless capabilities. GloMoSim source and binary code can be downloaded only by academic institutions for research purposed. Commercial users must use QualNet, the commercial version of GloMoSim.

**7. CNet**

Developed by: C. M. U. of Western Australia

Available at: <http://www.csse.uwa.edu.au/cnet/>

CNET is a simulator of computer networks. This simulator is not really focused on industrial simulation 'per-se', but to an aspect of it. It is a discrete-event network simulator enabling experimentation with various data-link layer, network layer, routing and transport layer networking protocols. It has been specifically developed for, and used in, undergraduate computer networking courses taken by thousands of students worldwide.

**8. OptSim**

Developed by: R. D. Group

Available at: <http://www.rsoftdesign.com/products.php?sub=System+and+Network&itm=OptSim>

OptSim 4.7 is the result of the merge between the former OptSim 3.6 sample mode engine, implementing simulation in the time domain sample by sample using a linear convolution algorithm, and the LinkSIM block mode engine, implementing the Frequency Domain Split Step. It is an intuitive modeling and simulation environment supporting the design and the performance evaluation of the transmission level of optical communication systems.

**9. GTNetS**

Developed by: GeorgiaTech

Available at: <http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/>

The Georgia Tech Network Simulator (GTNetS), developed by Dr George Riley, is a full-featured network simulation environment that allows researchers in computer networks to study the behavior of moderate to large scale networks, under a variety of conditions. The design philosophy of GTNetS is to create a simulation environment that is structured much like actual networks are structured. For example, in GTNetS, there is clear and distinct separation of protocol stack layers. It is availabe for download from the website.

**10. The Network Simulator - ns2**

Developed by: U. of Southern California

Available at: <http://www.isi.edu/nsnam/ns/>

Ns2 is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Ns is developed by ISI, the Information Sciences Institute at the USC school of engineering. The full source code of ns 2 can be downloaded and it can be compiled of multiple platform, including most popular Unix flavours and Windows.

**11. JiST / SWANS**

Developed by: R. B. C. University

Available at: <http://jist.ece.cornell.edu/>

SWANS is a scalable wireless network simulator built atop the JiST platorm. SWANS is organized as independent software components that can be composed to form complete wireless network or sensor network configurations. Its capabilities are similar to ns2 and GloMoSim, but is able to simulate much larger networks. SWANS leverages the JiST design to acheive high simulation throughput, save memory, and run standard Java network applications over simulated networks. In addition, SWANS implements a data structure, called hierarchical binning, for efficient computation of signal propagation.

**12. OMNeT++**

Developed by: O. C. Site

Available at: <http://www.omnetpp.org/>

OMNeT++ is a component-based, modular and open-architecture simulation environment with strong GUI support and an embeddable simulation kernel. The simulator can be used for modelling: communication protocols, computer networks and traffic modelling, multi-processors and distributed systems, etc. OMNeT++ also supports animation and interactive execution. It is freely distributed under an academic public license.

**13. QualNet Developer**

Developed by: S. N. Technologies

Available at: <http://www.scalable-networks.com/>

QualNet is a modelling tool for wireless and wired network,. The QualNet suite is composed of

QualNet Simulator, which claims to be the fastest for real-time traffic modelling. QualNet Animator allows to graphically design the network model (using a wide library of components) and it displays the results of simulation runs. QualNet Designer allos to create Finite State Automata to describe the behaviour of your network, while with QualNet Analyzer and Designer you can interpret and make sense of simulation results. Windows and Linux versions are available. A demo can be downloaded on request.

**14. PARSEC**

Developed by: U. P. C. Laboratory

Available at: <http://pcl.cs.ucla.edu/projects/parsec/>

Parsec is a C-based simulation language, developed by the Parallel Computing Laboratory at UCLA, for sequential and parallel execution of discrete-event simulation models. It can also be used as a parallel programming language. It is available in binary form only for academic institutions. Commercial users are directed to its commercial implementation QualNet, marketed by Scalable Networks technologies (http://www.scalable-networks.com/)

**15. NCTuns**

Developed by: U. of Twente

Available at: <http://nsl10.csie.nctu.edu.tw/>

The NCTUns is a high-fidelity and extensible network simulator and emulator capable of simulating various protocols used in both wired and wireless IP networks. Its core technology is based on the novel kernel re-entering methodology. NCTUns can be used as an emulator, it directly uses the Linux TCP/IP protocol stack to generate high-fidelity simulation results, and it has many other interesting qualities. It is commercialised by SimReal Inc.

**16. Performance PROPHET**

Developed by: U. of Vienna

Available at: <http://www.dps.uibk.ac.at/projects/prophet/>

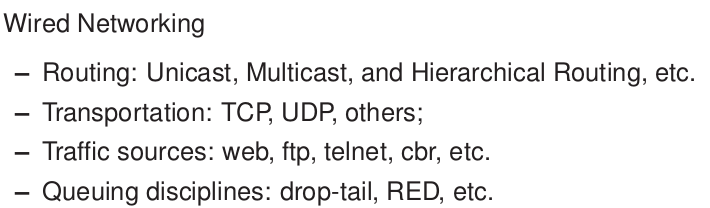
The Performance Prophet is a tool for modeling and simulation of high performance computing systems based on CSIM (Mesquite Software). Based on a UML model of an application and a simulator for a target architecture, the tool can predict the execution behavior of the application model on cluster and grid architectures. Performance Prophet has been deveoped at the Distributed and Parallel Systems Group of the University of Innsbruck and it has been supported by the Austrian Science Fund.

### PROTOCOLS AND MODELS SUPPORTED BY NS2 IN WIRED NETWORKS

NS2 MAINLY SUPPORT THREE MODES ON NETWORKING  
  
**1. WIRED NETWORKS**  
**2. ADHOC/MOBILE IP**  
**3. WIRELESS SENSOR NETWORKS**

**WIRED NETWORKS**

A wired network connects devices to the Internet or other network using cables. The most common wired networks use cables connected to Ethernet ports on the network router on one end and to a computer or other device on the cable's opposite end.  
  
Ethernet and wireless networks each have advantages and disadvantages; depending on your needs, one may serve you better than the other. Wired networks provide users with plenty of security and the ability to move lots of data very quickly. Wired networks are typically faster than wireless networks, and they can be very affordable. However, the cost of Ethernet cable can add up - the more computers on your network and the farther apart they are, the more expensive your network will be. In addition, unless you're building a new house and installing Ethernet cable in the walls, you'll be able to see the cables running from place to place around your home, and wires can greatly limit your mobility.  
  
*For more click here* :                 [WIRED VS WIRELESS](https://drive.google.com/file/d/0B6aQ8IUEyp5NNURid1QzdVE5ZzA/edit?usp=sharing)  
  
The following list shows the models and protocols supported by ns2 in wired configuration;

[](http://www.google.com/url?q=http%3A%2F%2F4.bp.blogspot.com%2F-3JBgzVjz9A0%2FU0S22KE8viI%2FAAAAAAAAAKo%2Fs2P9cBzoHwM%2Fs1600%2FScreenshot%2Bfrom%2B2014-04-09%2B08%3A26%3A05.png&sa=D&sntz=1&usg=AFrqEzewUSbJsqIkG-UaC3D-jFR28a-Wcw)

**ROUTING IN WIRED NETWORKS**

**UNICAST ROUTING**  
  
In computer networking, Unicast transmission is the sending of messages to a single network destination identified by a unique address. Unicast routing is the process of forwarding unicasted traffic from a source to a destination on an internetwork. Unicasted traffic is destined for a unique address.  
  
If an IP Unicast packet passes through a switch that does not know the location of the associated MAC Address, the packet will be broadcast to all ports on the switch. This failure of Unicast to 'cast to a single device' is called a*Unicast flood*.  
  
Unicast messaging is used for all network processes in which a private or unique resource is requested.  
Certain network applications which are mass-distributed are too costly to be conducted with unicast transmission since each network connection consumes computing resources on the sending host and requires its own separate network bandwidth for transmission. Such applications include streaming media of many forms. Internet radio stations using unicast connections may have high bandwidth costs.

|  |
| --- |
| [http://1.bp.blogspot.com/-W8KxZn6MRSM/U0QgEDQMScI/AAAAAAAAAJ8/mcP5cIgMyFk/s1600/Screenshot+from+2014-04-08+21:42:38.png](http://www.google.com/url?q=http%3A%2F%2F1.bp.blogspot.com%2F-W8KxZn6MRSM%2FU0QgEDQMScI%2FAAAAAAAAAJ8%2FmcP5cIgMyFk%2Fs1600%2FScreenshot%2Bfrom%2B2014-04-08%2B21%3A42%3A38.png&sa=D&sntz=1&usg=AFrqEzd4npmYiviD18lZFee0gnZqsdJEGQ) |
| UNICAST ROUTING |

*For more click here:              >>*[URP SECTION 1](https://drive.google.com/file/d/0B6aQ8IUEyp5NRW9GcGxhR3dzNEk/edit?usp=sharing)  
*>>*[URP SECTION 2](https://drive.google.com/file/d/0B6aQ8IUEyp5NVk1RT2FmVWhxS0U/edit?usp=sharing)  
 **MULTICAST ROUTING**  
  
In computer networking, multicast is the delivery of a message or information to a group of destination computers simultaneously in a single transmission from the source. Copies are automatically created in other network elements, such as routers, but only when the topology of the network requires it.  
  
Multicast is most commonly implemented in IP multicast, which is often employed in Internet Protocol (IP) applications of streaming media and Internet television. In IP multicast the implementation of the multicast concept occurs at the IP routing level, where routers create optimal distribution paths for datagrams sent to a multicast destination address.  
  
At the Data Link Layer, multicast describes one-to-many distribution such as Ethernet multicast addressing, Asynchronous Transfer Mode (ATM) point-to-multipoint virtual circuits (P2MP) or Infiniband multicast.

|  |
| --- |
| [http://4.bp.blogspot.com/-oF_LRfexHqM/U0QjCR2ZeKI/AAAAAAAAAKI/y0IgIKE6goI/s1600/Screenshot+from+2014-04-08+21:55:16.png](http://www.google.com/url?q=http%3A%2F%2F4.bp.blogspot.com%2F-oF_LRfexHqM%2FU0QjCR2ZeKI%2FAAAAAAAAAKI%2Fy0IgIKE6goI%2Fs1600%2FScreenshot%2Bfrom%2B2014-04-08%2B21%3A55%3A16.png&sa=D&sntz=1&usg=AFrqEzdZd3S4nTVDrBC7Mq0IwbUqtViFOQ) |
| MULTICAST ROUTING |

**IP MULTICAST**  
  
IP multicast is a technique for one-to-many and many-to-many real-time communication over an IP infrastructure in a network. It scales to a larger receiver population by requiring neither prior knowledge of a receiver's identity nor prior knowledge of the number of receivers. Multicast uses network infrastructure efficiently by requiring the source to send a packet only once, even if it needs to be delivered to a large number of receivers. The nodes in the network (typically network switches and routers) take care of replicating the packet to reach multiple receivers such that messages are sent over each link of the network only once. The most common low-level protocol to use multicast addressing is User Datagram Protocol (UDP). By its nature, UDP is not reliable—messages may be lost or delivered out of order. Reliable multicast protocols such as Pragmatic General Multicast (PGM) have been developed to add loss detection and retransmission on top of IP multicast.  
  
Key concepts in IP multicast include an IP multicast group address, a multicast distribution tree and receiver driven tree creation.  
  
An IP multicast group address is used by sources and the receivers to send and receive multicast messages. Sources use the group address as the IP destination address in their data packets. Receivers use this group address to inform the network that they are interested in receiving packets sent to that group. For example, if some content is associated with group 239.1.1.1, the source will send data packets destined to 239.1.1.1. Receivers for that content will inform the network that they are interested in receiving data packets sent to the group 239.1.1.1. The receiver joins 239.1.1.1. The protocol typically used by receivers to join a group is called the Internet Group Management Protocol (IGMP).  
  
With routing protocols based on shared trees, once the receivers join a particular IP multicast group, a multicast distribution tree is constructed for that group. The protocol most widely used for this is Protocol Independent Multicast (PIM). It sets up multicast distribution trees such that data packets from senders to a multicast group reach all receivers which have joined the group. For example, all data packets sent to the group 239.1.1.1 are received by receivers who joined 239.1.1.1. There are variations of PIM implementations: Sparse Mode (SM), Dense Mode (DM), Source Specific Mode (SSM) and Bidirectional Mode (Bidir, or Sparse-Dense Mode, SDM). Of these, PIM-SM is the most widely deployed as of 2006;[citation needed] SSM and Bidir are simpler and scalable variations developed more recently and are gaining in popularity.  
  
IP multicast operation does not require an active source to know about the receivers of the group. The multicast tree construction is receiver driven and is initiated by network nodes which are close to the receivers. IP multicast scales to a large receiver population. The IP multicast model has been described by Internet architect Dave Clark as, "You put packets in at one end, and the network conspires to deliver them to anyone who asks."  
  
IP multicast creates state information per multicast distribution tree in the network. If a router is part of 1000 multicast trees, it has 1000 multicast routing and forwarding entries. On the other hand, a multicast router does not need to know how to reach all other multicast trees in the Internet. It only needs to know about multicast trees for which it has downstream receivers. This is key to scaling multicast-addressed services. It is very unlikely that core Internet routers would need to keep state for all multicast distribution trees,[citation needed] they only need to keep state for trees with downstream membership. In contrast, a unicast router needs to know how to reach all other unicast addresses in the Internet, even if it does this using just a default route. For this reason, aggregation is key to scaling unicast routing. Also, there are core routers that carry routes in the hundreds of thousands because they contain the Internet routing table.

**ROUTING**

Each host (and in fact each application on the host) that wants to be a receiving member of a multicast group (i.e. receive data corresponding to a particular multicast address) must use the Internet Group Management Protocol (IGMP) to join. Adjacent routers also use this protocol to communicate.

In unicast routing, each router examines the destination address of an incoming packet and looks up the destination in a table to determine which interface to use in order for that packet to get closer to its destination. The source address is irrelevant to the router. However, in multicast routing, the source address (which is a simple unicast address) is used to determine data stream direction. The source of the multicast traffic is considered upstream. The router determines which downstream interfaces are destinations for this multicast group (the destination address), and sends the packet out through the appropriate interfaces. The term reverse path forwarding is used to describe this concept of routing packets away from the source, rather than towards the destination.

A number of errors can happen if packets intended for unicast are accidentally sent to a multicast address; in particular, sending ICMP packets to a multicast address has been used in the context of DoS attacks as a way of achieving packet amplification.

On the local network, multicast delivery is controlled by IGMP (on IPv4 network) and MLD (on IPv6 network); inside a routing domain, PIM or MOSPF are used; between routing domains, one uses inter-domain multicast routing protocols, such as MBGP.

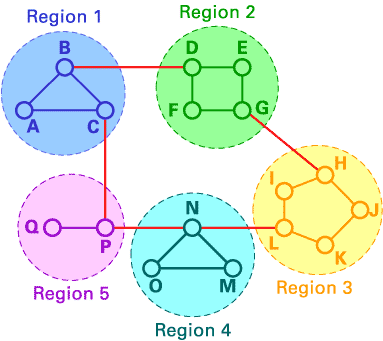
The following are some common delivery and routing protocols used for multicast distribution:

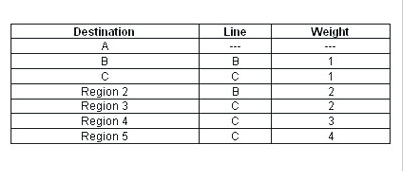
1. Internet Group Management Protocol (IGMP)
2. Protocol Independent Multicast (PIM)
3. Distance Vector Multicast Routing Protocol (DVMRP)
4. Multicast Open Shortest Path First (MOSPF)
5. Multicast BGP (MBGP)
6. Multicast Source Discovery Protocol (MSDP)
7. Multicast Listener Discovery (MLD)
8. GARP Multicast Registration Protocol (GMRP)

*For more click here:                   >>*[MULTICAST ROUTING](https://drive.google.com/file/d/0B6aQ8IUEyp5Nc2pQUlFoNmdscDA/edit?usp=sharing)   
*>>*[INTRODUCTION TO IP MULTICAST](https://drive.google.com/file/d/0B6aQ8IUEyp5NZkhuMUUzQk95c1k/edit?usp=sharing)  
                                                    >>   [MULTICAST ROUTING PROTOCOLS](https://drive.google.com/file/d/0B6aQ8IUEyp5Na1BlQW9vZno0am8/edit?usp=sharing)

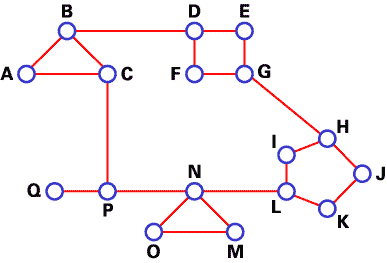
**HIERARCHICAL ROUTING**

Hierarchical routing is a method of routing in networks that is based on hierarchical addressing. Hierarchical routing is the procedure of arranging routers in a hierarchical manner. A good example would be to consider a corporate intranet. Most corporate intranets consist of a high speed backbone network. Connected to this backbone are routers which are in turn connected to a particular work group. These work groups occupy a unique LAN. The reason this is a good arrangement is because even though there might be dozens of different work groups, the span (maximum hop count to get from one host to any other host on the network) is 2. Even if the work groups divided their LAN network into smaller partitions, the span could only increase to 4 in this particular example.  
  
Considering alternative solutions with every router connected to every other router, or if every router was connected to 2 routers, shows the convenience of hierarchical routing. It decreases the complexity of network topology, increases routing efficiency, and causes much less congestion because of fewer routing advertisements. With hierarchical routing, only core routers connected to the backbone are aware of all routes. Routers that lie within a LAN only know about routes in the LAN. Unrecognized destinations are passed to the default route.  
  
Most Transmission Control Protocol/Internet Protocol (TCP/IP) routing is based on a two-level hierarchical routing in which an IP address is divided into a network portion and a host portion. Gateways use only the network portion until an IP datagram reaches a gateway that can deliver it directly. Additional levels of hierarchical routing are introduced by the addition of subnetworks.  
  
In hierarchical routing, routers are classified in groups known as regions. Each router has only the information about the routers in its own region and has no information about routers in other regions. So routers just save one record in their table for every other region. In this example, we have classified our network into five regions (see below).

[](http://www.google.com/url?q=http%3A%2F%2Fstatic.ddmcdn.com%2Fgif%2Frouting-algorithm10.gif&sa=D&sntz=1&usg=AFrqEzf28Xh0Zh_ilXIy8rxNuXC-E77b9A)

[](http://www.google.com/url?q=http%3A%2F%2F1.bp.blogspot.com%2F-HU5lNrhqZNg%2FU0S1R5d5BvI%2FAAAAAAAAAKY%2FFfp_1F0Is58%2Fs1600%2FScreenshot%2Bfrom%2B2014-04-09%2B08%3A18%3A31.png&sa=D&sntz=1&usg=AFrqEzck_UzG3mTds4Rewmx57fpCOzHszA)

If A wants to send packets to any router in region 2 (D, E, F or G), it sends them to B, and so on. As you can see, in this type of routing, the tables can be summarized, so network efficiency improves. The above example shows two-level hierarchical routing. We can also use three- or four-level hierarchical routing.  
  
In three-level hierarchical routing, the network is classified into a number of clusters. Each cluster is made up of a number of regions, and each region contains a number or routers. Hierarchical routing is widely used in Internet routing and makes use of several routing protocols.  
  
In here, DV algorithms are used to find best routes between nodes. In the situation depicted below, every node of the network has to save a routing table with 17 records. Here is a typical graph and routing table for A:

[](http://www.google.com/url?q=http%3A%2F%2Fstatic.ddmcdn.com%2Fgif%2Frouting-algorithm9.gif&sa=D&sntz=1&usg=AFrqEzeTfAd3Uz_Hsr4QvP3za3-gRiZvng)

|  |
| --- |
| [http://1.bp.blogspot.com/-KArbMYEUZKU/U0S1blIMnyI/AAAAAAAAAKg/ujQtwt5Me4Q/s1600/Screenshot+from+2014-04-09+08:19:00.png](http://www.google.com/url?q=http%3A%2F%2F1.bp.blogspot.com%2F-KArbMYEUZKU%2FU0S1blIMnyI%2FAAAAAAAAAKg%2FujQtwt5Me4Q%2Fs1600%2FScreenshot%2Bfrom%2B2014-04-09%2B08%3A19%3A00.png&sa=D&sntz=1&usg=AFrqEze8vmEzd-s6aEwIhU3SWrIYti-aww) |
| Network Graph and Routing Table of A |

*For more click here:*[HIERARCHIAL ROUTING](https://drive.google.com/file/d/0B6aQ8IUEyp5NS1pzLUNZZjBFSHM/edit?usp=sharing)

**TRANSPORTATION AGENT**

**TRANSMISSION CONTROL PROTOCOL [TCP]**

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called TCP/IP. TCP provides reliable, ordered and error-checked delivery of a stream of octets between programs running on computers connected to a local area network, intranet or the public Internet. It resides at the transport layer.

Web browsers use TCP when they connect to servers on the World Wide Web, and it is used to deliver email and transfer files from one location to another. HTTP, HTTPS, SMTP, POP3, IMAP, SSH, FTP, Telnet and a variety of other protocols are typically encapsulated in TCP.

While IP takes care of handling the actual delivery of the data, TCP takes care of keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

For example, when an HTML file is sent to you from a Web server, the Transmission Control Protocol (TCP) program layer in that server divides the file into one or more packets, numbers the packets, and then forwards them individually to the IP program layer. Although each packet has the same destination IP address, it may get routed differently through the network. At the other end (the client program in your computer), TCP reassembles the individual packets and waits until they have arrived to forward them to you as a single file.

TCP is known as a connection-oriented protocol, which means that a connection is established and maintained until such time as the message or messages to be exchanged by the application programs at each end have been exchanged. TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end. In the Open Systems Interconnection (OSI) communication model, TCP is in layer 4, the Transport Layer.

*For more click here:*[TCP OVERVIEW](https://drive.google.com/file/d/0B6aQ8IUEyp5NaXFySjcxMTZycUE/edit?usp=sharing)

**USER DATAGRAM PROTOCOL [UDP]**

The User Datagram Protocol (UDP) is a transport layer protocol defined for use with the IP network layer protocol. It is defined by RFC 768 written by John Postel. It provides a best-effort datagram service to an End System (IP host).

The service provided by UDP is an unreliable service that provides no guarantees for delivery and no protection from duplication (e.g. if this arises due to software errors within an Intermediate System (IS)). The simplicity of UDP reduces the overhead from using the protocol and the services may be adequate in many cases.

UDP provides a minimal, unreliable, best-effort, message-passing transport to applications and upper-layer protocols. Compared to other transport protocols, UDP and its UDP-Lite variant are unique in that they do not establish end-to-end connections between communicating end systems. UDP communication consequently does not incur connection establishment and teardown overheads and there is minimal associated end system state. Because of these characteristics, UDP can offer a very efficient communication transport to some applications, but has no inherent congestion control or reliability. A second unique characteristic of UDP is that it provides no inherent On many platforms, applications can send UDP datagrams at the line rate of the link interface, which is often much greater than the available path capacity, and doing so would contribute to congestion along the path, applications therefore need to be designed responsibly [RFC 4505].

One increasingly popular use of UDP is as a tunneling protocol, where a tunnel endpoint encapsulates the packets of another protocol inside UDP datagrams and transmits them to another tunnel endpoint, which decapsulates the UDP datagrams and forwards the original packets contained in the payload. Tunnels establish virtual links that appear to directly connect locations that are distant in the physical Internet topology, and can be used to create virtual (private) networks. Using UDP as a tunneling protocol is attractive when the payload protocol is not supported by middleboxes that may exist along the path, because many middleboxes support UDP transmissions.

UDP does not provide any communications security. Applications that need to protect their communications against eavesdropping, tampering, or message forgery therefore need to separately provide security services using additional protocol mechanisms.

*For more click here:                       >>* [PART 1](https://drive.google.com/file/d/0B6aQ8IUEyp5NdW9oMEs5UlVOQWM/edit?usp=sharing" \t "_blank)

                                                       >>   [PART 2](https://drive.google.com/file/d/0B6aQ8IUEyp5NanU2WGRmV0JpRDA/edit?usp=sharing)

**TRAFFIC SOURCES**

**FILE TRANSFER PROTOCOL [FTP]**  
File Transfer Protocol (FTP) is a standard Internet protocol for transmitting files between computers on the Internet. Like the Hypertext Transfer Protocol (HTTP), which transfers displayable Web pages and related files, and the Simple Mail Transfer Protocol (SMTP), which transfers e-mail, FTP is an application protocol that uses the Internet's TCP/IP protocols. FTP is commonly used to transfer Web page files from their creator to the computer that acts as their server for everyone on the Internet. It's also commonly used to download programs and other files to your computer from other servers.  
  
As a user, you can use FTP with a simple command line interface (for example, from the Windows MS-DOS Prompt window) or with a commercial program that offers a graphical user interface. Your Web browser can also make FTP requests to download programs you select from a Web page. Using FTP, you can also update (delete, rename, move, and copy) files at a server. You need to logon to an FTP server. However, publicly available files are easily accessed using anonymous FTP.  
  
Basic FTP support is usually provided as part of a suite of programs that come with TCP/IP. However, any FTP client program with a graphical user interface usually must be downloaded from the company that makes it.

*For more click here:*[FTP](https://drive.google.com/file/d/0B6aQ8IUEyp5NYVJCOU5uUldfOFE/edit?usp=sharing)

**TELNET**

Telnet is a user command and an underlying TCP/IP protocol for accessing remote computers. Through Telnet, an administrator or another user can access someone else's computer remotely. On the Web, HTTP and FTP protocols allow you to request specific files from remote computers, but not to actually be logged on as a user of that computer. With Telnet, you log on as a regular user with whatever privileges you may have been granted to the specific application and data on that computer.  
  
A Telnet command request looks like this (the computer name is made-up):  
  
                                                 telnet the.libraryat.whatis.edu  
  
  
The result of this request would be an invitation to log on with a userid and a prompt for a password. If accepted, you would be logged on like any user who used this computer every day.  
  
Telnet is most likely to be used by program developers and anyone who has a need to use specific applications or data located at a particular host computer.  
  
*For more click here:*[TELNET](https://drive.google.com/file/d/0B6aQ8IUEyp5Nb09wMU9QM3FXdjQ/edit?usp=sharing)  
  
**CONSTANT BIT RATE [CBR]**  
Constant bitrate (CBR) is a term used in telecommunications, relating to the quality of service. Compare with variable bitrate. When referring to codecs, constant bit rate encoding means that the rate at which a codec's output data should be consumed is constant. CBR is useful for streaming multimedia content on limited capacity channels since it is the maximum bit rate that matters, not the average, so CBR would be used to take advantage of all of the capacity. CBR would not be the optimal choice for storage as it would not allocate enough data for complex sections (resulting in degraded quality) while wasting data on simple sections.  
  
The problem of not allocating enough data for complex sections could be solved by choosing a high bitrate (e.g., 256 kbit/s or 320 kbit/s) to ensure that there will be enough bits for the entire encoding process, though the size of the file at the end would be proportionally larger.  
  
Most coding schemes such as Huffman coding or run-length encoding produce variable-length codes, making perfect CBR difficult to achieve. This is partly solved by varying the quantization (quality), and fully solved by the use of padding. (However, CBR is implied in a simple scheme like reducing all 16-bit audio samples to 8 bits.)  
  
  
In the case of streaming video as a CBR, the source could be under the CBR data rate target. So in order to complete the stream, it's necessary to add stuffing packets in the stream to reach the data rate wanted. These packets are totally neutral and don't affect the stream.  
  
*For more click here:*[CBR](http://www.google.com/url?q=http%3A%2F%2Fen.wikipedia.org%2Fwiki%2FConstant_bitrate&sa=D&sntz=1&usg=AFrqEzdCO7rPT6SkM31qpnxlSTC0Fa5MdA)

**QUEUING DISCIPLINES**  
 **DROP TAIL**

It is a simple queue mechanism that is used by the routers that when packets should to be drop. In this mechanism each packet is treated identically and when queue filled to its maximum capacity the newly incoming packets are dropped until queue have sufficient space to accept incoming traffic.

When a queue is filled the router start to discard all extra packets thus dropping the tail of mechanism. The loss of packets (datagram’s) causes the sender to enter slow start which decreases the throughput and thus increases its congestion window.

**FAIR QUEUING**

It is a queuing mechanism that is used to allow multiple packets flow to comparatively share the link capacity. Routers have multiple queues for each output line for every user. When a line as available as idle routers scans the queues through round robin and takes first packet to next queue. FQ also ensure about the maximum throughput of the network. For more efficiency weighted queue mechanism is also used.

**DEFICIT ROUND ROBIN**

It is a modified weighted round robin scheduling mechanism. It can handle packets of different size without having knowledge of their mean size. Deficit Round Robin keeps track of credits for each flow. It derives ideas from Fair Queuing and Stochastic FQ. It uses hashing to determine the queue to which a  flow  has  to be  assigned  and  collisions  automatically  reduce  the bandwidth guaranteed  to  the  flow.  Each queue is assigned a quantum and can send a packet of size that can fit in the available quantum. If not, the idle quantum gets added to this meticulous queue’s deficit and the packet can be sent in the next round.  The quantum size is a very vital parameter  in  the DRR  scheme,  determining  the  upper  bound  on  the  latency as well as  the throughput.

This queue mechanism used a well-designed idea to get better performance and can also be implemented in a cost effectiveness manner. It provides a generic framework to implement fair queuing efficiently.

Although DRR serves fine for throughput fairness, but when it comes to Latency bounds it performs rather poorly. Also it does not operate well for real time traffic. The queuing delays introduced by DRR can have exciting results on the congestion window sizes.

**RANDOM EARLY DETECTION [RED]**

Random Early Detection (RED) is a congestion avoidance queuing mechanism (as opposed to a congestion administration mechanism) that is potentially useful, particularly in high-speed transit networks. Sally Floyd and Van Jacobson projected it in various papers in the early 1990s.It is active queue management mechanism. It operates on the average queue size and drop packets on the basis of statistics information. If the buffer is empty all incoming packets are acknowledged. As the queue size increase the probability for discarding a packet also increase. When buffer is full probability becomes equal to 1 and all incoming packets are dropped.

RED is capable to evade global synchronization of TCP flows, preserve high throughput as well as a low delay and attains fairness over multiple TCP connections, etc. It is the most common mechanism to stop congestive collapses.

When the queue in the router starts to fill then a small percentage of packets are discarded. This is deliberate to start TCP sources to decrease their window sizes and hence suffocate back the data rate. This can cause low rates of packet loss in Voice over IP streams. There have been reported incidences in which a series of routers apply RED at the same time, resulting in bursts of packet loss.

**STOCHASTIC FAIR QUEUING [SFQ]**

This queuing mechanism is based on fair queuing algorithm and proposed by John Nagle in 1987. Because it is impractical to have one queue for each conversation SFQ uses a hashing algorithm which divides the traffic over a limited number of queues. It is not so efficient than other queues mechanisms but it also requires less calculation while being almost perfectly fair. It is called "Stochastic" due to the reason that it does not actually assign a queue for every session; it has an algorithm which divides traffic over a restricted number of queues using a hashing algorithm. SFQ assigns a pretty large number of FIFO queues.