Planned Activities on Computer Networks for Sunday Academy

Jun Liu, Jason Waskiewicz, Allen Nash
Time Schedule

• 11:00-11:15 Cultural Connection
• 11:15-11:30 General Introduction
• 11:30-12:00 Activity 1: Network Configuration.
• 12:00-1:00 Activity 2: Wireless Sensor Network.
• 1:00-1:30 Lunch
• 1:30-2:00 Activity 3: Network Simulations.
• 2:00-2:30 Activity 4: Cloud Computing.
  – Back-up activity: Running a web server.
• 2:30-3:00 Wrap-up and Evaluation
Overview of the Internet

• A network is the interconnection of a set of devices capable of communication.
  – A device can be a host such as a large computer, desktop, laptop, workstation, cellular phone.
  – A device can also be a connecting device such as a router, a switch, a modem, and so on.

• The Internet is composed of thousands of inter-connected networks.

• The Internet allows any user to become part of it. The user, however, needs to be physically connected to an Internet Service Provider (ISP).
The Internet today

Customer network

Provider network

Customer network

Backbones

Provider network

Customer network

Peering point

Customer network

Provider network

Customer network

Customer network

Customer network

Customer network

Customer network

Customer network

Customer network

Customer network

Customer network

Customer network

Customer network

Customer network

Customer network

Customer network
Basic Introduction (1)

• Network
  – It consists of nodes and links, like a graph.
• Node
  – It is a device that can be connected with other devices in a network, like a vertex in a graph.
• Link
  – It is a network cable or a wireless connection between two network nodes, like an edge in a graph.
• Packets
  – They are the information unit for message exchange between network nodes.
A heterogeneous network made of various types of switching devices
Basic Introduction (2)

- **Wide Area Network (WAN)**
  - A network that covers a broad area.
Wireless Mesh Networks
Typical Network Services

• World Wide Web (WWW) Service
  – The hypertext documents distribution service.
• File Transfer Service (FTP)
  – It hosts files on file servers and allows the clients to access the files on the servers.
• Domain Name Service (DNS)
  – It translates user-friendly host names to the IP address of the host.
  – Example: The URL www.google.com is the user-friendly host name of the WWW google gateway server.
  – The DNS translates www.google.com to the IP address of the WWW server, which is 74.125.225.180.
Planned Activities

• Four activities have been planned:
  – Activity 1: Hands-on practice of network emulations.
  – Activity 2: Hands-on practice of emulating a wireless sensor network.
  – Activity 3: Hands-on practice of emulating the message transmissions in two network scenarios.
  – Activity 4: Hands-on practice of using an existing public platform to do cloud computing.
    • Back-up activity if the resource for Activity 4 is unavailable: Hands-on practice of running a web server that can be visited by the clients.
Activity 1: Emulate A Network (1)

• Emulate the operation of a computer network through making students to act as the communication devices.
• Students are divided into groups with each group representing an emulated network.
• This activity includes three scenarios.
  – A simple network with two students talking over a short distance.
  – A simple network with two students talking over a longer distance.
  – A network with two pairs of students talking to their peers by sharing a common link within the network.
• Goals: It is to show the main functions of switching devices in the network, which are
  – to restore the signal power over a longer distance;
  – to relay messages.
Activity 1: Emulate A Network (2)

- Scenario 1: A simple network with two students talking over a short distance.
  - A network link is emulated using a 3.25" x 60 - 120" 2-piece Adjustable Kraft Mailing Tubes with End Caps.
  - Two students act as a pair of sender and receiver.
    - The sender talks to the receiver over the tube.
    - Since the length of the tube is short, the receiver can successfully pick up the voice of the sender.
Activity 1: Emulate A Network (3)

Node 1 — Link — Node 2

Short Distance

User 1 — Tube — User 2

Voice is clear

Voice is clear
Activity 1: Emulate A Network (4)

Longer Distance

End Node 1 \[\text{Link}\] End Node 2

Voice is not so clear over extended tube

User 1 \[\text{Tube}\] \[\text{Tube}\] \[\text{Tube}\] User 2

Relay Messages

End Node 1 \[\text{Link 1}\] Repeater Node \[\text{Link 2}\] End Node 2

User 1 \[\text{Tube 1}\] User 3 \[\text{Tube 2}\] User 2
Activity 1: Emulate A Network (5)

• Scenario 2: A simple network with two students talking over a longer distance.
  – 2 or 3 tubes are attached together to form a link over a longer distance.
  – 2 students communicate over the extended link.
  – Due to the extended distance, the receiver can not clearly pick up the voice of the sender.
  – In order to make the pair of sender and receiver to normally communicate again, a longer link has to be broken down into pieces of short distance.
  – A student has to act as the repeater between two tubes to relay the voice for the sender.
Activity 1: Emulate A Network (6)

Two voices interfere
With each other.
Activity 1: Emulate A Network (8)

• Scenario 3: A network with two pairs of students talking to their peers by sharing a common link within the network.
  – The voices from the 2 connections interfere to each other over the shared link.
  – A student acts as the switcher to relay voices to the individual connections, such that each connection only hears its own voice.
Activity 1: Emulate A Network (7)

Node 1

Switching Node

Node 2

Node 3

Node 4

User 1

Intermediate User

User 2

User 3

User 4

No interference.
Activity 1: Network Connectivity (1)

• Demonstration of the network connectivity.
  – To show how a computer is connected to the network through a network interface card.
  – To show the interaction between a pair of sender and receiver using program ping.
  – To show the intermediate switching devices sitting between a pair of sender and receiver using program traceroute.
Activity 1: Network Connectivity (2)

- Network card configuration
  - ifconfig command in Linux and Mac OS
  - Display configuration information
    
    ```
    # ifconfig eth0
    ```
  - Attach and detach a network card
    
    ```
    # ifconfig eth0 up
    # ifconfig eth0 down
    ```
  - Assigning IP address
    
    ```
    # ifconfig eth0 192.168.2.2
    ```
Activity 1: Network Configuration (3)

• Network connectivity configuration
  – Test connectivity
    • ping command
    `# ping -n 5 -l 1500 www.google.com`
    Pinging www.google.com [74.125.224.82] with 1500 bytes of data:
    Reply from 74.125.224.82: bytes=1500 time=68ms TTL=52
    Reply from 74.125.224.82: bytes=1500 time=68ms TTL=52
    Reply from 74.125.224.82: bytes=1500 time=65ms TTL=52
    Reply from 74.125.224.82: bytes=1500 time=66ms TTL=52
    Reply from 74.125.224.82: bytes=1500 time=70ms TTL=52
Activity 1: Network Configuration (4)

• Show the intermediate switching devices in a connection.
  – traceroute command

    # traceroute www.google.com

    traceroute: Warning: www.google.com has multiple addresses; using 74.125.225.82

    traceroute to www.l.google.com (74.125.225.82), 64 hops max, 52 byte packets
    1  gw1-216 (134.129.216.1)  0.579 ms  0.436 ms  0.351 ms
    2  134.129.223.10 (134.129.223.10)  0.613 ms  0.638 ms  0.575 ms
    3  165.234.23.1 (165.234.23.1)  1.223 ms  1.169 ms  1.217 ms
    4  165.234.22.65 (165.234.22.65)  1.484 ms  1.043 ms  0.825 ms
    5  fargo-i1-3592.itd.nd.gov (165.234.98.4)  6.861 ms  4.034 ms  3.910 ms
    6  165.234.119.18 (165.234.119.18)  4.109 ms  6.084 ms  4.019 ms
Activity 2: Emulated Wireless Sensor Network (1)

• Goal: it is to reveal that WSN has the ability of forming a dynamic ad hoc network without the need of network configuration.
  – A wireless sensor is a tiny device which is capable of communicating with others over a wireless channel.
Activity 2: Emulated Wireless Sensor Network (2)

• Method:
  – Forward path:
    • All students randomly scatter in a classroom.
    • All students form groups of 2 students without overlapping.
    • Each student emulates a wireless sensor.
    • Each student can only talk to his/her direct neighbors within a distance of one arm.
    • Each student asks the help from direct neighbors to find a path to reach his/her peer.
    • The neighbors relay the help messages received to farther neighbors.
      – A help message will eventually reach the targeted peer.
      – Each student also keeps a rap sheet to record that “I forwarded a help message originated by A to reach a peer B from my neighbor C.”
Activity 2: Emulated Wireless Sensor Network (3)

Temp Forwarding Table

<table>
<thead>
<tr>
<th>Source</th>
<th>Sink</th>
<th>Upstream Neighbor</th>
<th>Downstream Neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Activity 2: Emulated Wireless Sensor Network (4)

• Method:
  – Backward path:
    • When a student receives the help message sent by his/her designated peer, he/she sends an acknowledgement back to the designated peer to the neighbor from whom the help message was received.
    • When a student receives an acknowledgement, he/she checks the rap sheet to find a neighbor to whom the acknowledgement is forwarded along the reverse path to the designated peer.
      – Each student also logs that “I can relay messages for a pair of peers (A,B) through my neighbors C (for downstream) and D (for upstream).”
Activity 2: Emulated Wireless Sensor Network (5)

Routing Table at A

<table>
<thead>
<tr>
<th>Source</th>
<th>Sink</th>
<th>Upstream Neighbor</th>
<th>Downstream Neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>-</td>
<td>U</td>
</tr>
</tbody>
</table>

Routing Table at U

<table>
<thead>
<tr>
<th>Source</th>
<th>Sink</th>
<th>Upstream Neighbor</th>
<th>Downstream Neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>A</td>
<td>X</td>
</tr>
</tbody>
</table>

Routing Table at B

<table>
<thead>
<tr>
<th>Source</th>
<th>Sink</th>
<th>Upstream Neighbor</th>
<th>Downstream Neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>X</td>
<td>-</td>
</tr>
</tbody>
</table>

Routing Table at X

<table>
<thead>
<tr>
<th>Source</th>
<th>Sink</th>
<th>Upstream Neighbor</th>
<th>Downstream Neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>U</td>
<td>B</td>
</tr>
</tbody>
</table>
Activity 2: Emulated Wireless Sensor Network (6)

Put it together: finding a route
Activity 3: Network Emulations (1)

• **Emulate two network scenarios.**
• **Goal**
  – It is to demonstrate the internal activities within the network when data flows are delivered across the network.
  – Demonstrating the general process of packet delivery between two end nodes.
    • The process of packet delivery is not viewable in our daily use of the network. This demonstration can allow students to visualize the process of end-to-end packet delivery.
  – Demonstrating the hidden terminal problem in wireless networks.
• **Scenarios:**
  – Stop-and-wait delivery.
  – Back-off in wireless networks to resolve the hidden terminal problem.
Activity 3: Network Emulations (2)

• Scenario 1: Stop-and-wait delivery.
  – Students form groups of three students: a pair of sender and receiver, and a middleman.
  – A sender sends a sequence of messages to the receiver through the middleman.
    • The 3 entities are separated by two curtains such that the pair of sender and receiver cannot see each other.
    • The sender and receiver only interact with the middleman.
  – A number of stickers are used to represent a sequence of messages (for example, 10 messages).
    • Each sticker is labeled with a message ID from 1 to 10.
  – A sender can only send one message at a time before the message is acknowledged by the receiver.
  – A receiver has to acknowledge each received message by sending back a sticker which is labeled with “Ack to message x” (x is the message ID).
Activity 3: Network Emulations (3)

• Scenario 1: Stop-and-wait delivery.
  – The middleman randomly drops the messages (or acknowledgements) by tossing a coin.
    • Head: the middleman will forward a message or acknowledgement.
    • Tail: the middleman drops a message or acknowledgement.
  – A sender has to send a message again if the corresponding acknowledgement is not received in 10 seconds since a message is sent.
    • A sender keeps sending the same message until the message is acknowledged.
  – A sequence of 10 messages is fully transmitted after the sender receives the acknowledgements for the 10 messages.
Activity 3: Network Emulations (4)

• Scenario 1: Stop-and-wait delivery.
Activity 3: Network Emulations (5)

• Scenario 1: Stop-and-wait delivery.
  – Running a network simulation to see what happens in a stop-and-wait delivery in network.
Activity 3: Network Emulations (6)

- Visualizing the packet flows in the network.
Activity 3: Network Emulations (7)

- Scenario 2: Back-off in wireless networks to resolve the hidden terminal problem.
  - A number of imaginary wireless devices compete to transmit messages through a common wireless channel.
    - A message is emulated by a student.
    - The common wireless channel is emulated by a door.
    - The fact that multiple wireless devices send their messages to the common channel is emulated by making students to go through a common door.
    - The door can only pass one student at a time.
    - Students move toward the door at a constant speed with eyes closed.
    - When multiple students reach the door at the same, a collision happens.
      - When multiple messages collide to each other, neither message can be successfully decoded by the receiver.
    - The successful transmission of a message by a wireless device corresponds to the case that only one student passes the door without colliding with other students at the door.
Activity 3: Network Emulations (8)

• Scenario 2: Back-off in wireless networks to resolve the hidden terminal problem.
  – Back-off mechanism
    • When a collision happens, the students have to step back and wait for a randomly chosen time duration before making the next attempt of passing the door.
    • The collision can happen repeatedly.
    • Since each student chooses the back-off duration at random without telling others, it is possible that a student with the shortest back-off time to start the next attempt while others still back off.
  – Successful transmission of a message
    • When only one student passes the door without collision, the message (represented by the student) is successfully transmitted.
Activity 3: Network Emulations (9)

• Visualization of wireless transmissions
Activity 4: Cloud Computing (1)

- Hands-on practice of using an existing public platform to do cloud computing.

- Purpose
  - It is to get a sense of doing cloud computing using the public resources.

- Method
  - Running a peer-to-peer network application on a public resource.

- Goals
  - Demonstrating the process of acquiring the public resource.
  - Demonstrating the process of running a peer-to-peer network on public resource.
  - Demonstrating a set of typical operations in the peer-to-peer network.
Activity 4: Cloud Computing (2)

• Tools
  – The FLACK user interface for accessing the cloud resource.
  – The Chord package written in Java for running a peer-to-peer network in the cloud computing platform.

• To Do’s for students
  – Try to insert, query, or delete the data items on the nodes in a pre-established peer-to-peer network.
  – Try to understand the mechanism in the peer-to-peer network.
Activity 4: Cloud Computing (3)
Activity 5: Running A Web Server (1)

• Hands-on practice of running a web server that can be visited by the clients.

• Purpose
  – It is to demonstrate the operations in web service.

• Method
  – An Apache web server program is run on a computer. A set of sample files are loaded under the directory tree of the web server.

• Goals
  – Demonstrating the method of specifying URLs to access files in various locations in the directory tree.
  – Demonstrating the exchange of messages between a pair of web server and client in the process that the client accesses files on the server.
Activity 5: Running A Web Server (2)

• System requirements
  – Operating systems: ubuntu Linux OS
  – Web server application: the Apache HTTP Web server

• To Do’s by students:
  – To start and stop a web server;
  – To make a web server to operate at a non-default port;
  – To add files in the directory tree of the web server;
  – To write a simple portal for the web server;
  – To access files on the server through the simple portal.
Activity 5: Running A Web Server (3)

• Enable Apache
  – To start it:
    sudo apachectl start
  – To stop it:
    sudo apachectl stop
  – To restart it:
    sudo apachectl restart

• Once Apache is running, http://localhost should display the default “It works!” page.

• The configuration file is found in /etc/apache2/httpd.conf
Activity 5: Running A Web Server (4)

• Make web server to run on a non-default port 8000
  – The default port is 80.
  – Change to new port: Listen 8000

• The default document root is found in /Library/WebServer/Documents/
  – DocumentRoot "/Library/WebServer/Documents"
  – Change to a new document root by specifying a new directory name.
  – The default portal page is: index.html.en
    • <html><body><h1>It works!</h1></body></html>
    • Index.html.en prevents browsing the content of the document root.
Activity 5: Running A Web Server (5)

• Write a new portal for the web site.
  – To make a link to a page in the portal:
    • `<a href="pagename.html">page name or whatever</a>`
  – To point to another site:
    • `<a href="yoururl.com">URL name or whatever</a>`
  – New index.html.en

```html
<html>
<body>
<a href="file1.html">file1</a><br>
<a href="file2.html">file2</a><br>
<a href="old_index.html.en">Original index.html</a><br>
<a href="http://www.google.com">www.googl.com</a><br>
</body>
</html>
```
Handouts for Activity 3
Network Simulations
Activity 3: Network Simulations (1)

- Running various network simulations using an open-source network simulator.
- Purpose
  - It is to demonstrate the internal activities within the network when data flows are delivered across the network.
  - Demonstrating the general process of packet delivery between two end nodes.
    - The process of packet delivery is not viewable in our daily use of the network. This demonstration can allow students to visualize the process of end-to-end packet delivery.
- Method
  - A set of interesting network operations are simulated using the open-source network simulator (ns-2).
  - The simulator can record the sequence of packet events throughout a simulation.
  - Then, the recorded packet trace can be played back by an animator to reveal the internal activities in chronological order.
- Goals
  - Visualizing the packet losses that are hidden from the view of end users.
  - Demonstrating the operations in wireless networks.
Activity 3: Network Simulations (2)

- Task 1: Bandwidth Share of TCP and UDP Traffic
  - Network configuration
    - a network with 4 nodes,
    - the link between node 3 and node 4 has a speed of 1 Mbps,
    - all other links have a speed of 2 Mbps,
    - all links have a propagation delay of 10 ms.
    - There is a TCP/FTP flow from node 1 to node 4, and a UDP/CBR (Constant Bit Rate) flow from node 2 to node 4.
    - The packet length for both flows is 1000 Bytes.
    - The rate of the UDP/CBR flow is 0.2 Mbps.
Activity 3: Network Simulations (3)

• Task 1: Bandwidth Share of TCP and UDP Traffic
  – Step 1, create the simulator object.
    • # create a simulator object
    • set ns [new Simulator]
  – Step 2: Set up the network topology
    • create four nodes
      set node1 [$ns node]
      set node2 [$ns node]
      set node3 [$ns node]
      set node4 [$ns node]
    • create links between the nodes
      $ns duplex-link $node1 $node3 2Mb 10ms DropTail
      $ns duplex-link $node2 $node3 2Mb 10ms DropTail
      $ns duplex-link $node3 $node4 1Mb 10ms DropTail
      $ns queue-limit $node3 $node4 4
Activity 3: Network Simulations (4)

- **Task 1: Bandwidth Share of TCP and UDP Traffic**
  - Step 3, Define traffic patterns by creating agents, applications and flows.
  - TCP traffic flow
    ```
    # create a TCP source agent and attach it to node node1
    set tcp [new Agent/TCP]
    $ns attach-agent $node1 $tcp
    # create a TCP sink agent and attach it to node node4
    set sink [new Agent/TCPSink]
    $ns attach-agent $node4 $sink  # connect both agents
    $ns connect $tcp $sink  # create an FTP source "application"
    set ftp [new Application/FTP]
    $ftp attach-agent $tcp
    ```
  - UDP traffic flow
    ```
    set udp [new Agent/UDP]
    $ns attach-agent $node2 $udp  # create a CBR traffic source and attach it to udp
    set cbr [new Application/Traffic/CBR]  $cbr set packetSize_ $packetSize  $cbr
    set rate_ 0.2Mb  $cbr attach-agent $udp  # create a Null agent (a traffic sink) and attach it
    set null [new Agent/Null]  $ns attach-agent $node4 $null  $ns
    connect $udp $null
    ```
Activity 3: Network Simulations (5)

- **Task 1: Bandwidth Share of TCP and UDP Traffic**
  - Step 4, define the trace files, and place monitors at places in the topology to collect information about packets flows.
    ```
    # open the nam trace file
    set nam_trace_fd [open tcp_tahoe.nam w]
    ns namtrace-all $nam_trace_fd
    set trace_fd [open tcp_tahoe.tr w]
    # Define a 'finish' procedure
    proc finish {} {
        global ns nam_trace_fd trace_fd
        # close the nam trace file
        $ns flush-trace
        close $nam_trace_fd
        # execute nam on the trace file
        $ns execute nam on $trace_fd
        exit 0
    }
    
    - Step 5, schedule the simulation by defining the start and stop of the simulation, traffic flows, tracing, and other events.
      ```
      ```
      # schedule events for all the flows
      ns at 0.1 "$ftp start"
      ns at 0.1 "$cbr start"
      ns at 5.0 "$cbr stop"
      ns at 5.0 "$ftp stop" # call the finish procedure after 6 seconds of simulation time
      ns at 6 "finish" # run the simulation
      ```
Activity 3: Network Simulations (6)

• Task 2: a simple wireless simulation
  – Network configuration
    • Containing 4 flows.
    • moving within 1000mX1000m flat topology
    • using DSDV ad hoc routing protocol
    • Random Waypoint mobility model
    • TCP and CBR traffic
Activity 3: Network Simulations (7)

• **Task 2:** a simple wireless simulation
  
  - Parameter specification:
    ```
    set cbr_size 500
    set cbr_interval 0.002
    set num_row 4
    set time_duration 100
    
    # channel type
    set val(chan) Channel/WirelessChannel
    # radio-propagation model
    set val(prop) Propagation/TwoRayGround
    # network interface type
    set val(netif) Phy/WirelessPhy
    # MAC type
    set val(mac) Mac/802_11
    # interface queue type
    set val(ifq) Queue/DropTail/PriQueue
    # link layer type
    set val(ll) LL
    # antenna model
    set val(ant) Antenna/OmniAntenna
    # max packet in ifq
    set val(ifqlen) 50
    # routing protocol
    set val(rp) DSDV
    ```
Activity 3: Network Simulations (8)

- **Task 2**: a simple wireless simulation
  - **Initialization:**
    ```
    set ns_ [new Simulator]
    set tracefd [open simple.tr w]
    $ns_ trace-all $tracefd
    
    # set up topography object
    set topo [new Topography]
    $topo load_flatgrid 1000 1000
    
    create-god [expr $num_row * $num_row ]
    
    $ns_ node-config -adhocRouting $val(rp) -llType $val(ll) \ 
    -macType $val(mac) -ifqType $val(ifq) \ 
    -ifqLen $val(ifqlen) -antType $val(ant) \ 
    -propType $val(prop) -phyType $val(netif) \ 
    -channel [new $val(chan)] -topoInstance $topo \ 
    -agentTrace ON -routerTrace OFF\ 
    -macTrace ON \ 
    -movementTrace OFF
    ```
Activity 3: Network Simulations (9)

- **Task 2:** a simple wireless simulation
  - Make nodes and positions:
    ```
    for {set i 0} {i < [expr $num_row*$num_row]} {incr i} {
        set node_(i) [$ns_node]
    }
    set k 0;
    while {$k < $num_row} {
        for {set i 0} {$i < $num_row} {incr i} {
            set m [expr $i+$k*$num_row];
            $node_(m) set X_ [expr $i*240];
            $node_(m) set Y_ [expr $k*240+20.0];
            $node_(m) set Z_ 0.0
        }
        incr k;
    }
    ```
Activity 3: Network Simulations (10)

- **Task 2**: a simple wireless simulation
  - Make traffic agents and applications:
    ```tcl
    for {set i 0} {$i < $num_row} {incr i} {
        set udp_($i) [new Agent/UDP]
        set null_($i) [new Agent/Null]
    }
    $ns_ attach-agent $node_(0) $udp_(0)
    $ns_ attach-agent $node_(7) $udp_(1)
    $ns_ attach-agent $node_(2) $udp_(2)
    $ns_ attach-agent $node_(7) $udp_(3)
    $ns_ attach-agent $node_(6) $null_(0)
    $ns_ attach-agent $node_(1) $null_(1)
    $ns_ attach-agent $node_(8) $null_(2)
    $ns_ attach-agent $node_(15) $null_(3)
    for {set i 0} {$i < $num_row} {incr i} {
        $ns_ connect $udp_($i) $null_($i)
    }
    for {set i 0} {$i < $num_row} {incr i} {
        set cbr_($i) [new Application/Traffic/CBR]
        $cbr_($i) set packetSize_ $cbr_size
        $cbr_($i) set interval_ 0.5
        $cbr_($i) attach-agent $udp_($i)
    }
    $ns_ at 11.0234 "$cbr_(0) start"
    $ns_ at 10.4578 "$cbr_(1) start"
    $ns_ at 12.7184 "$cbr_(2) start"
    $ns_ at 12.2456 "$cbr_(3) start"
    ```
Handouts for Basic Technical Terms
Basic Technical Terms (1)

- Nodes are categorized into two classes:
  - Switching Nodes
    - They are the switching devices dedicated to relay packets.
    - Hub: the simplest and cheapest
      - It sends packets to all links that are connected by a hub.
    - Switch: more complicated than a hub but simpler than a backbone router
    - Backbone Router: with full-blown features, high speed, and expensive
  - End Nodes:
    - They generate and consume messages.
    - Generally, they have to be connected to switching device.
    - Two end nodes can also be directly connected to exchange messages.
Basic Technical Terms (2)

• **Point-to-Point Connectivity**
  - **Modem**
    - It converts digital information to transmittable electronic signal which can be transmitted over the telephone lines.
    - Its transmission speed is usually slow.
    - It cannot transmit with the telephone signals at the same time.
  - **ADSL (Asymmetric Digital Subscriber Line)**
    - It also converts digital information to transmittable electronic signal which can be transmitted over the phone line along with telephone signals.
      - It has two bands: the downstream and upstream bands and a voice band.
    - Its speed is much faster than regular modem, but slower than the Ethernet.
Basic Technical Terms (3)

• **Local Area Network (LAN)**
  – A network spans over a short distance, like in a school.
  – Ethernet is a technique for interconnecting nodes within a small area at faster speed.
• Computers are connected to each other by switches or hubs.
  – Cat-5 cables are used to connect computers (looks like fat phone cable)
  – A Cat-5 cable is terminated with a RJ45 connector which is plugged with the network card in a computer.
Basic Technical Terms (4)

- **Network Topology**
  - Bus topology
    - Usually used in Ethernet.
  - Star topology
    - Usually used in home networks.
  - Tree topology
    - Usually used to organize the Domain Name Service (DNS).
  - Mesh topology
    - Usually used in city-wide wireless networks.
Basic Technical Terms (5)

• **Communication Protocol** is a set of rules that governs the processing of network events.
  – A protocol defines the reactions that a network node takes in response to a packets.
  – Commonly used protocols
    • **Dynamic Host Configuration Protocol (DHCP)** assigns a dynamic IP address to a computer when the computer connects to the network.
    • **Internet Protocol (IP)** is the principal protocol for routing and relaying packets.
    • **User Datagram Protocol (UDP)** provides unrelaible transmission of messages.
    • **Transport Control Protocol (TCP)** provides reliable transmission of messages.
    • **HyperText Transfer Protocol (HTTP)** is the core protocol of the World Wide Web (WWW) service.
Basic Technical Terms (6)

• **Packet** is a unit that messages are organized and encapsulated.
  – The length of a packet is restrained by a maximum length, no matter how long a piece of user message is.
  – Long messages are broken into multiple packets which are individually transmitted.

• **IP Address** is a logical identifier which distinguishes network nodes
  – An IP address has 32 bits with the ability of denoting the geographical location of a network node, for example, 74.125.225.180.
  – An IP address is comparable to phone number (area code gives state, then city, then your number: 701-555-1234).

• **Units** in Networking
  – Bit: a single 1 or 0
  – Byte: a single character, 8 bits
  – Kilobyte: 1000 bytes (10³ bytes)
  – Megabytes: 1000 kilobytes (10⁶ bytes)
  – Gigabyte: 1000 megabytes (10⁹ bytes)
  – Terabyte: 1000 gigabytes (10¹² bytes)
Basic Technical Terms (7)

- **Bandwidth** is the number of bits transmitted per second.
  - bps: bits per second.
- **Transmission latency** is the overall time delay for a packet to be transmitted from a sender to a receiver. A transmission latency consists of three components: the propagation delay, the transmitting delay, and the queuing delay.
  - **Propagation delay** is the time delay for a signal to travel over a distance.
    - For example, it takes the light about 8 minutes to go from the Sun to the Earth. This is the propagation delay from the Sun to the Earth.
  - **Transmitting delay** is the duration to fully transmit a packet onto the wire.
    - For example, a pool of water of 8 gallons is drained by a pump with rate 1 gallon/second. It takes 8 seconds to fully drain the pool.
  - **Queuing delay** is the delay for a packet to wait in line before it gets the chance to be transmitted.