

## Wireless Networks and Mobile Systems

### Pre-work material Fundamentals: Technology Overview

## Lecture Objectives

---

- Discuss the course structure
- Provide a high-level overview of topics in
  - Data networking
  - Addressing
  - Protocols in the IP architecture
- Introduce basic network performance monitoring tools

## Agenda

---

- Course structure
- Fundamentals of data networks
- IP protocol suite
- Introduction to addressing
- Some basic tools for performance monitoring

## Course Structure

- Learning objectives
- Prerequisites

## Course Structure

---

- Multi-disciplinary (Computer Science, Computer Engineering, and Electrical Engineering)
- Considers aspects of wireless and mobile systems, including:
  - Wireless link and mobile network protocols
  - Mobile networking including support for the Internet Protocol suite

## Major Learning Objectives (1)

---

- Having successfully completed this course, you will be able to:
  - Describe the characteristics and operation of contemporary wireless network technologies such as the IEEE 802.11 wireless local area network and Bluetooth wireless personal area network.
  - Describe the operation of the TCP/IP protocol suite in a mobile environment, including the operation of Mobile IP and a mobile ad hoc routing protocol.
  - Suggest enhancements to protocols in the IP architecture to improve performance in a wireless environment, implement, test and evaluate the modified protocol.

## Major Learning Objectives (2)

- Having successfully completed this course, you will be able to:
  - Design, implement, and test a prototype mobile application.
  - Measure and characterize the performance a wireless local area network, mobile routing protocol, and mobile application.
  - Monitor the operation of mobile network protocols and applications using standard tools.

## Course Prerequisites

Basic communication engineering knowledge  
(TCP/IP, Modulation, Coding, etc.)

AND

Network  
Application Design

OR

Computer and  
Network  
Architectures

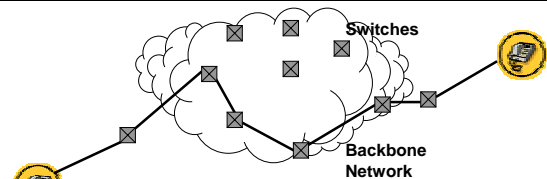
OR

Network  
Architecture and  
Programming

## Fundamentals of Data Networks

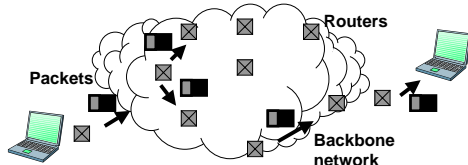
- Circuit and packet switching
- Protocols and layered architecture
- The OSI model

## Circuit Switching



- Stream of bits follows a path established during call set-up
- Resources reserved for the duration of the call
- Inefficient for exchange of data
- Example: traditional telephone network

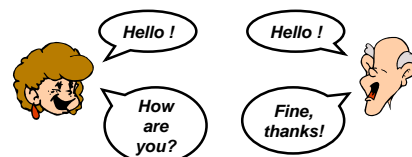
## Packet Switching



- Data are sent in blocks: data + control information = a "packet"
- Resources not necessarily reserved in advance
- Increased efficiency through statistical multiplexing
- Example: the Internet

## Protocols

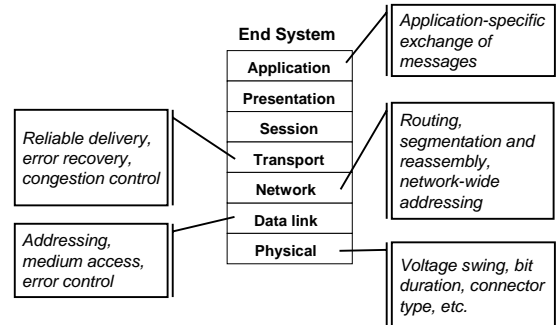
- Define the format and order of messages exchanged between two entities in the network
- Define the actions to be taken upon transmission or arrival of messages or some other event
- Examples: IP, HTTP, DHCP, etc.



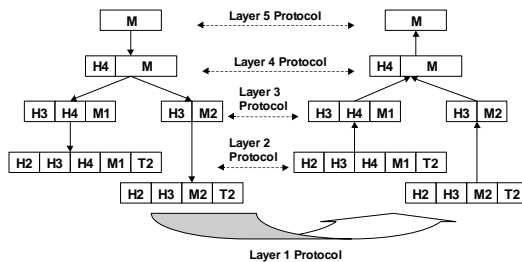
## Layering

- Start with services provided by the hardware, then add a sequence of layers, each providing services to the layer just above it
- Why?
  - Decomposes the very complex problem of providing networked communications into more manageable pieces
  - More modular design (easier to add a new service or to modify the functionality of a layer)
- Example of protocol layering
  - HTTP (for web browsing) uses services from TCP (for instance, reliable delivery of packets), which uses services provided by IP (for instance, globally unique addressing)

## OSI Model



## Encapsulation



## IP Protocol Suite

- IP stack
- Basic characteristics and reasons for ubiquity of IP
- ICMP

## Why is IP so successful?

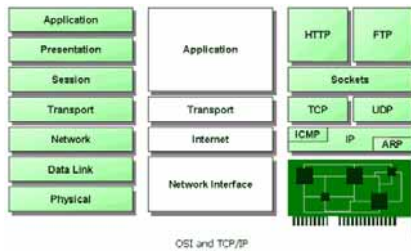
- Hourglass shape of the protocol stack
  - Many protocols run "over" IP
  - IP runs "over everything"
- Architectural principles
  - Minimalism, autonomy
  - Best effort service
  - Stateless routers
  - Decentralized control



## IP Protocol Stack

Application	e.g. TELNET, FTP, SNMP, DNS, HTTP, etc.
Transport	TCP, UDP
Internet	IP
Physical + Data Link	e.g. Ethernet, 802.11, SONET, ATM, etc.

## OSI and the IP suite



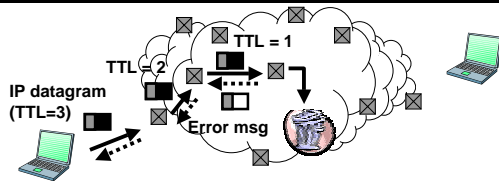
OSI and TCP/IP

Source: "Introducing TCP/IP," by FindTutorials.com

## Essential Characteristics of IP

- **Connectionless**
  - Each IP **datagram** is treated independently and may follow a different path
- **Best effort**
  - No guarantees of timely delivery, ordering, or even delivery
- **Globally unique 32-bit addresses**
  - Usually expressed in dot-decimal notation: 128.17.75.0
  - Each interface has its own IP address
  - Later, we will see that there are ways to use non-unique addresses
- **Typical IP datagram contains payload + a 20-byte header with control information (addressing, redundant bits for error detection, etc.)**

## Time to Live (TTL)



- IP datagram headers contain a TTL field
  - At each router, this field is decremented; if it reaches 0, datagram is discarded and an error message is generated
- **Original purpose was to prevent datagrams from endlessly circulating within the network**

## ICMP

- **Internet Control Message Protocol (ICMP)**
  - Used by hosts, routers and gateways to communicate network layer information to each other
  - Typically used for error reporting
- **Uses the services of IP**
  - ICMP messages are carried as IP payload
- **ICMP messages have a type and code and contain the first 8 bytes of the IP datagram that caused the ICMP message to be generated**
- **Many of the utilities we will use in this course (ping, traceroute, etc.) are implemented by processing ICMP messages**

## Introduction to Addressing

- IP addresses
- MAC addresses
- Address translation: DNS and ARP

## IP Addresses

- **32-bit addresses**

```
01001000 11000001 00000001 00001001
```
- Usually expressed in dot-decimal notation for convenience

↓                      ↓                      ↓                      ↓  
72 . 193 . 1 . 9

## IP Address Assignment

### Fixed IP address



OR

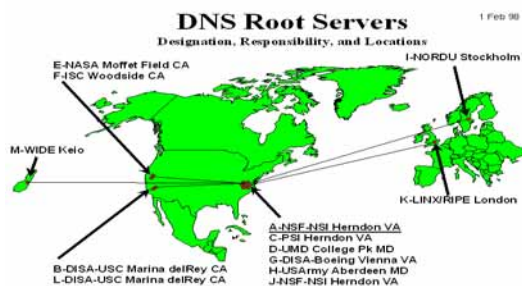
### Dynamically-assigned IP address (using DHCP)



## Address Translation: DNS

- From a domain name or URL (application layer) to an IP address (network layer)
  - Use Domain Name System (DNS)
  - Root and authoritative name servers provide the translation between any possible domain name and an IP address
  - Translation is cached locally

## DNS Root Servers

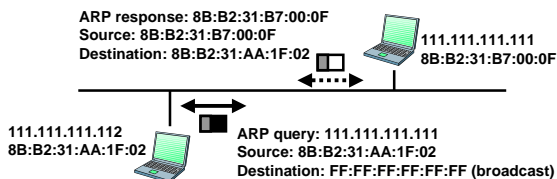


## MAC addresses

- LAN adaptors have hard-coded Medium Access Control (MAC) addresses
- These are 6-byte globally unique addresses
  - First 3 bytes identify the vendor
  - Expressed as hexadecimals separated by ":"
- Example:
  - 02 : 60 : 8C : E4 : B1 : 02
  - 3COM

## Address Translation: ARP

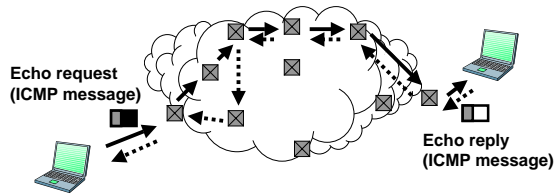
- From an IP address (network layer) to a MAC address (link layer)
  - Use the Address Resolution Protocol (ARP)
  - Results from an ARP query are kept locally in an ARP cache



## Some Basic Tools for Performance Monitoring

- Ping
- Traceroute
- Ethereal

## Ping



- Measures the round-trip time (RTT) between two nodes
- Source node generates echo request(s), destination node responds with echo reply (replies)

## Ping Example

```

Microsoft Windows [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

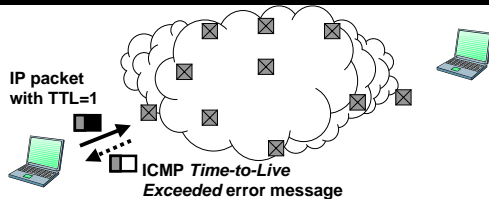
C:\>ping www.ut.edu

Pinging www.ut.edu [198.82.162.70] with 32 bytes of data:

Reply from 198.82.162.70: bytes=32 time=10ms TTL=249
Reply from 198.82.162.70: bytes=32 time=10ms TTL=249
Reply from 198.82.162.70: bytes=32 time=10ms TTL=249
Reply from 198.82.162.70: bytes=32 time=10ms TTL=249

Ping statistics for 198.82.162.70:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    approximate round trip times in milliseconds:
        Minimum = 10ms, Maximum = 10ms, Average = 10ms
    >>>
    
```

## Traceroute



- Lists all routers between source and destination
- Send consecutive IP datagrams with TTL = 1, 2, ...
  - Each of these will "die" at one of the intermediate routers, which will respond with an ICMP error message
  - Source will learn the identity of every router on the path

## Traceroute Example

```

Microsoft Windows [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>tracert www.ut.edu

Tracing route to www.ut.edu [198.82.162.70]
over a maximum of 30 hops:
  0  10 ms  10 ms  10 ms  208.17.194.204
  1  10 ms  10 ms  10 ms  WTR1-07freesandria@research.institute.networkvirgin
  ia.net [65.162.50.149]
  2  10 ms  10 ms  10 ms  65.162.89.38
  3  10 ms  10 ms  10 ms  lab-7507-1.sds.ut.edu [162.70.167.193]
  4  10 ms  10 ms  10 ms  lab-6000-1a.cit11.cnu.ut.edu [129.172.0.89]
  5  11 ms  10 ms  10 ms  lab-6000-1a.u171b.cnu.ut.edu [129.172.0.81]
  6  10 ms  10 ms  10 ms  www.ut.edu [198.82.162.70]

Trace complete.
    >>>
    
```

## Ethereal

- A "GUI protocol analyzer" that display, organizes and filters the results of packet sniffing
- A wide variety of packet types and protocols are supported by Ethereal
  - ATM, ARP, BGP, DNS, FTP, HTTP, IP, POP, TCP, UDP, and many others (even Quake...)
- Each packet is shown with source, destination, protocol type, and comments
  - A HEX dump shows you exactly what the packet looked like as it went over the wire
- Many more features to be explored in the homework
  - For more info, go to [www.ethereal.com](http://www.ethereal.com)

## Ethereal Example

