Basics of Computer Networking

Internet Technologies and Applications

Contents

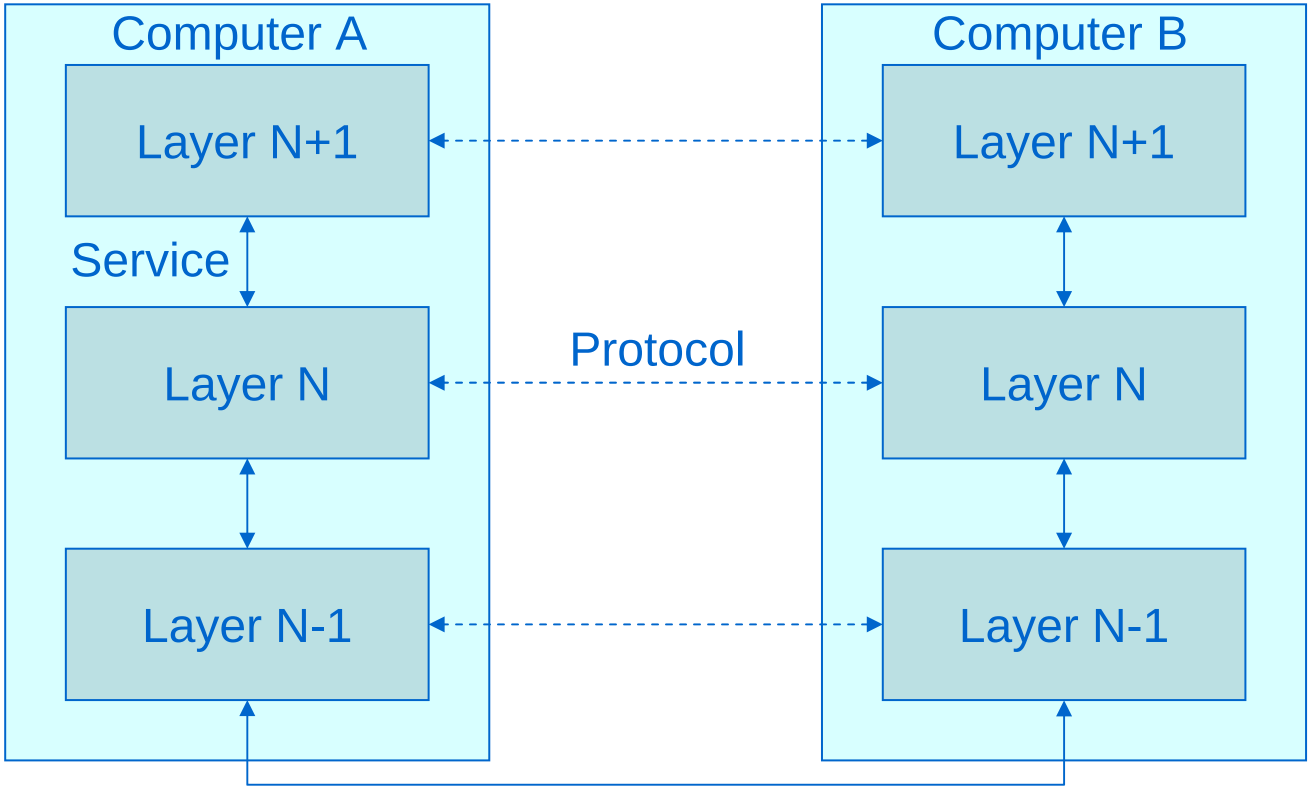
- Layered Architectures
- Network Organisation
- Internet Architecture
- Internet Protocols and Applications
- Network Analysis Tools

This is a refresher!

- Prerequisite courses should cover basics of networking and the Internet
- You should already know this material!
  - (If not, please tell me)
- This is not a comprehensive coverage of networking
  - Only mention concepts/protocols that are important for course
- Consult the reading resources for details

Layered Architectures

- Communication networking is hard! Therefore, divide-and-conquer
- In networking, this is done in layers



Layered Architectures

- Concepts:
  - Each node (e.g. computer) has a layered protocol stack
  - Actual communication goes from top to bottom layer on sending node and then bottom to top on receiving node
  - Standard interfaces (services) are defined between layers
  - Conceptually, a layer N protocol entity on sending node “talks to” layer N protocol entity on receiving node
  - Therefore, we can consider protocols at each layer separately
  - A protocol *provides a service* to the higher layer and *uses the service* of the lower layer
- Examples:
  - Open Systems Interconnection Reference Model (OSI RM) (formal)
  - Model of the Internet (informal)

Layered Architectures

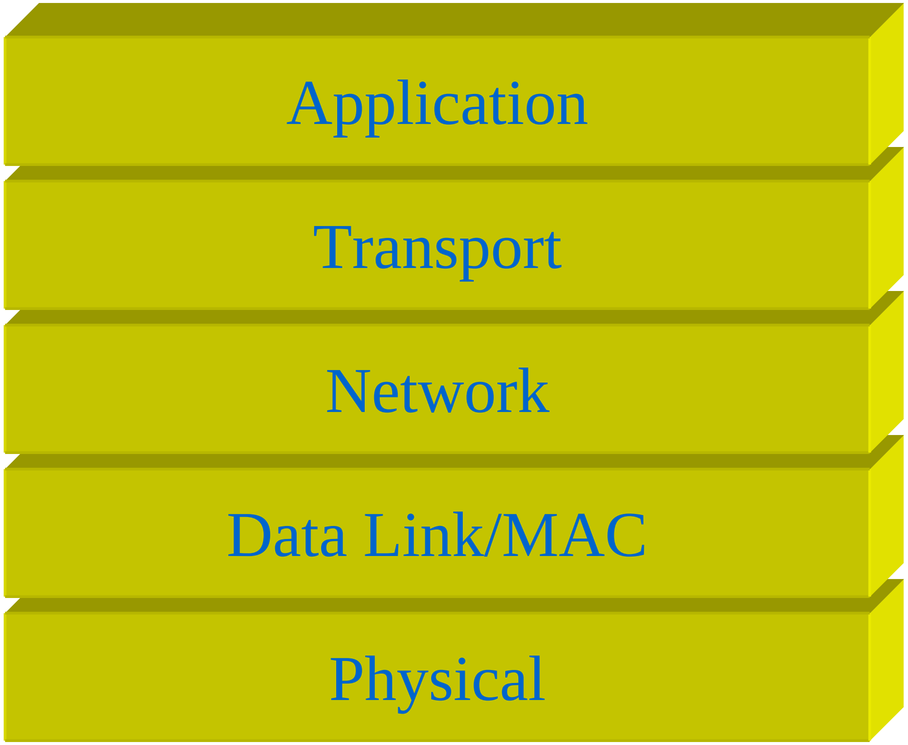
- Advantages
  - Break a large problem into several smaller problems
  - Can design and implement protocols without knowing details of other layers
  - Networks can be built by mix-and-matching different implementations, e.g. use Motorola for layer 1 and 2, Cisco for layer 3 and Microsoft for layer 4 and 5
  - Maintenance and upgrades are easier (only have to change one layer)
- Disadvantages
  - Overheads of layer to layer communications (e.g. headers)
  - Inflexible when optimisations are needed (e.g. for security, performance)

OSI Reference Model

- 7-layer protocol stack
- Developed by ISO as a standard for Open Systems Interconnection
- Protocols developed by ISO for each layer
- No longer significant commercial technologies

Internet Protocol Stack

- 5-layer protocol stack
- Based on what is actually in use in the Internet
- Less formal than OSI
- We will use this as conceptual model in course



Physical Layer (Layer 1)

- Transmission of bits over physical media
- Media include: twisted pair copper wire, optical fibres, radio frequencies, …
- Some tasks of Physical layer:
  - How to modulate bits to be sent as analog signals (e.g. volts, waves). And the reverse demodulation.
  - How to encode information efficiently so as little is sent as possible, but the original information can be correctly decoded at received
- Examples:
  - SDH/SONET, E1/T1, Optical fibre, copper, …
  - IEEE 802 standards (Ethernet, Wireless LAN);
  - Wireless transmission systems (GSM, 3G);

Data Link Layer (Layer 2)

- Transmission of frames over links
  - Provide reliable link to network layer
  - Error detection and correction
  - Retransmission schemes, such as automatic repeat request (ARQ)
    - Stop-and-wait, Go-back-N, Selective Repeat
  - Broadcast networks: how to share the channel amongst users
- Examples:
  - IEEE 802.2 Logical Link Control
  - IEEE 802 MAC layers (Ethernet, WLAN, Bluetooth, …)
  - HDLC, PPP, ATM, Frame Relay, FDDI, …
  - SDH/SONET, T1/E1, …

Network Layer (Layer 3)

- End-to-end packet delivery
  - Sometimes reliable delivery, for example provide some level of quality of service
- Interconnect different links/networks
- Routing: how to find a path from source to destination
- Flow control, error control, segmentation and reassembly, addressing, …
- Examples:
  - IP, IPX, X.25, CNLP, …

Transport Layer (Layer 4)

- Data delivery between endpoints (hosts, processes)
- May include:
  - Connection-orientation
  - Reliability
  - Flow and congestion control
  - Stream or message oriented
- Multiplexing: sending data from multiple processes over one link/network
- Examples:
  - TCP, UDP, OSI TP, SCTP, …

Application Layer (Layer 5)

- Provide common services to application processes or other application protocols
  - File transfer, mail/message delivery, directory and naming services, remote access/login, …
- Examples:
  - HTTP, FTP, Instant messaging protocols, SNMP, DNS, Telnet, DHCP, POP3, …

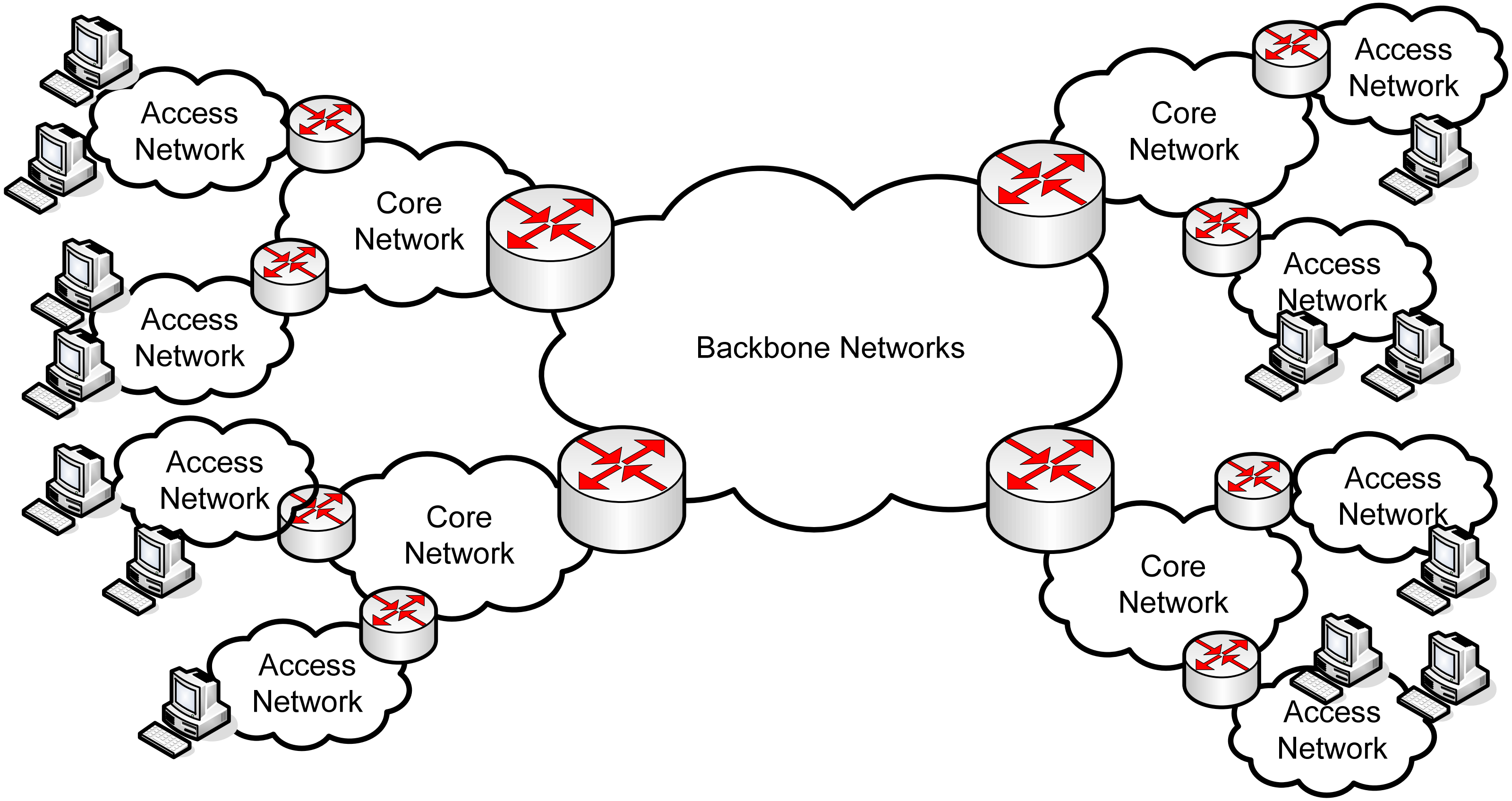
Circuit vs Packet Switching

- Circuit Switching
  - Establish a physical circuit between end-points
  - Example: Traditional telephone network
  - Requires a connection-setup process
  - Guarantees set resource (bandwidth) for duration of connection
  - May not use entire bandwidth all the time
- Packet Switching
  - Break data into packets and multiplex packets from various sources (computers) onto circuit
  - Example: Internet
  - Packets must carry destination information
  - No guarantee of bandwidth, e.g. if more packets are sent than the circuit can handle, then some packets will be dropped
  - More efficient use of bandwidth – statistical multiplexing
- Computer networks use packet switching for cost and performance advantages

Connection-oriented vs Connectionless

- Connection-oriented
  - End-systems (hosts or software processes) perform an “handshake” to setup a connection *before* they transfer data
    - Once setup, often the sender/receiver can control the connection to control the rate and reduce errors
    - But extra delay (and complexity) due to setup
  - Example: TCP
- Connectionless
  - Send the data immediately
    - No delay of connection setup
    - But recipient cannot expect packets, hence hard to provide reliability
  - Example: UDP

Network Organisation



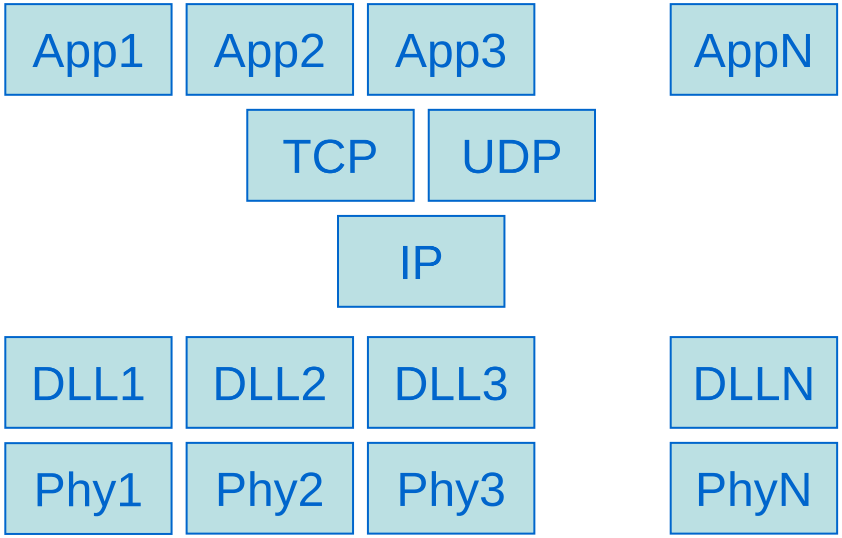
Note: this is a simplistic view; in reality, much more complex, for example, multiple layers in the hierarchy

Access, Core and Backbone Networks

- Access Networks
  - Provide user-level access
    - Home, office, campus, building, …
    - IEEE 802 networks, e.g. 100Mb/s or 1Gb/s Ethernet or wireless LAN
    - Dial-up and ADSL for residential access
    - GSM/GPRS/3G for mobile access
- Core Networks
  - Single provider network interconnecting access networks and backbone
    - Telecommunications provider (e.g. AIS, TOT) network or ISP
    - Large organisations (e.g. company or university with several sites)
    - Use ATM, MPLS, SDH/SONET and IP
- Backbone Networks
  - Bulk data transfer over cities, countries, continents
    - Telecommunication providers or ISPs
    - Optical or satellite links, SDH/SONET, ATM
- We may refer to core and backbone together as *transport networks*

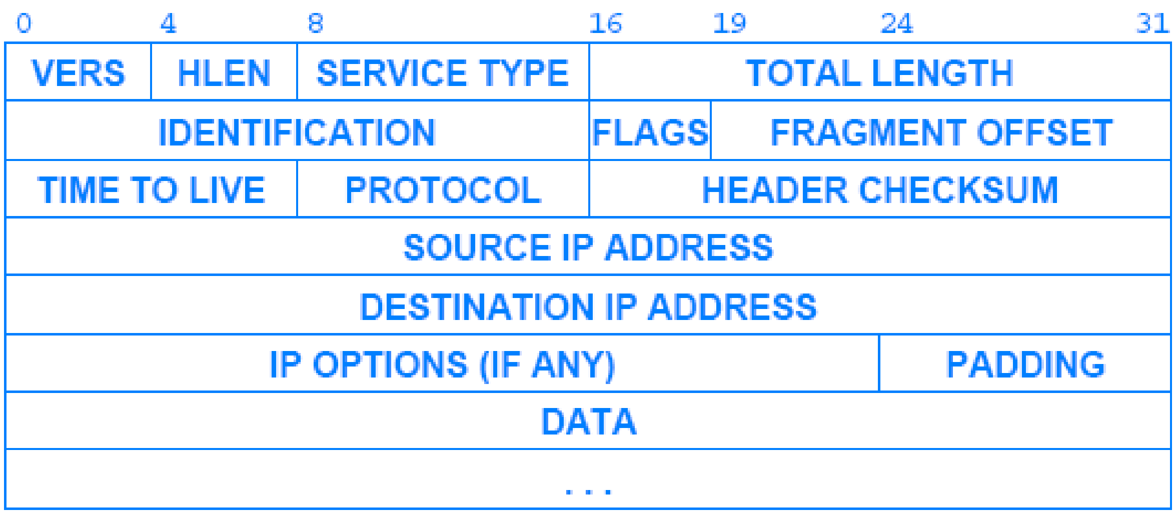
Internet Architecture

- Observations about communication networks:
  - No single network hardware technology can meet requirements of all applications
  - Users want to be able to connect to anyone
- Internets hide the network technology and allow many different networks to interconnect
  - Interconnection through routers (network layer)



The IP Datagram

- Most common size (no options) is:
  - 20 byte header
  - Up to 65,535 bytes of payload
- Header includes:
  - IP address of source host and destination host
  - Field that identifies the protocol carried in payload (e.g. TCP)
  - …



IP Addressing

- 32-bit addresses to identify hosts
- Address is conceptually split in two:
  1. Network address: identifying the network
  2. Host address: identifying the host on that network
- Convenient dotted-decimal form:
  - E.g. 172.16.0.1, 66.249.89.104, 203.131.209.77

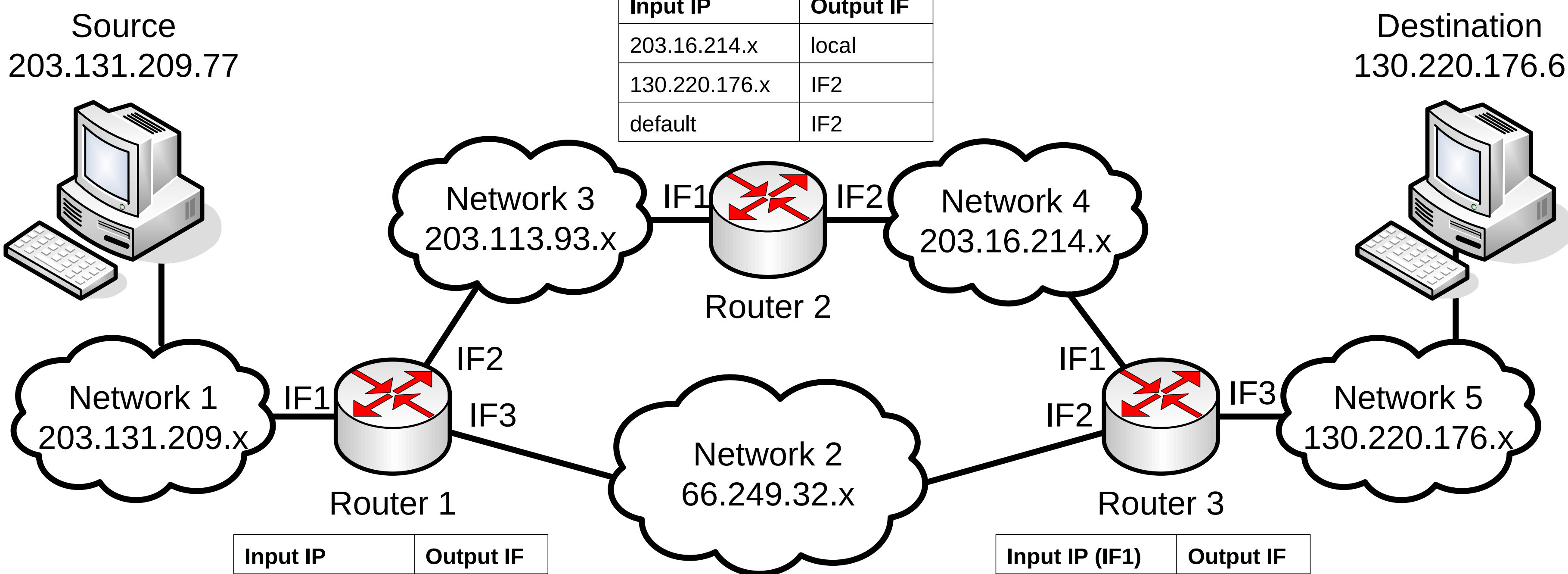
- Other addresses:
  - Address Resolution Protocol (ARP) is used to map IP addresses to hardware address:
    - Example: IEEE 802 devices have unique 48-bit address
  - Domain Name System (DNS) maps human readable domain names to IP addresses

IP Routing

- Routers interconnect networks
- IP datagrams are forwarded by routers based on destination IP address
- When router receives IP datagram:
  - If destination network address same as router, then send datagram directly to destination host (e.g. via Ethernet)
  - Otherwise, lookup forwarding (routing) table to determine the next network to reach destination. Send datagram to next router
- Routing protocols
  - Distribute topology information throughout the internet
  - Populate the forwarding tables so router knows next router/network to use to reach destination

IP Routing Example

Input IP	Output IF
203.16.214.x	local
130.220.176.x	IF2
default	IF2



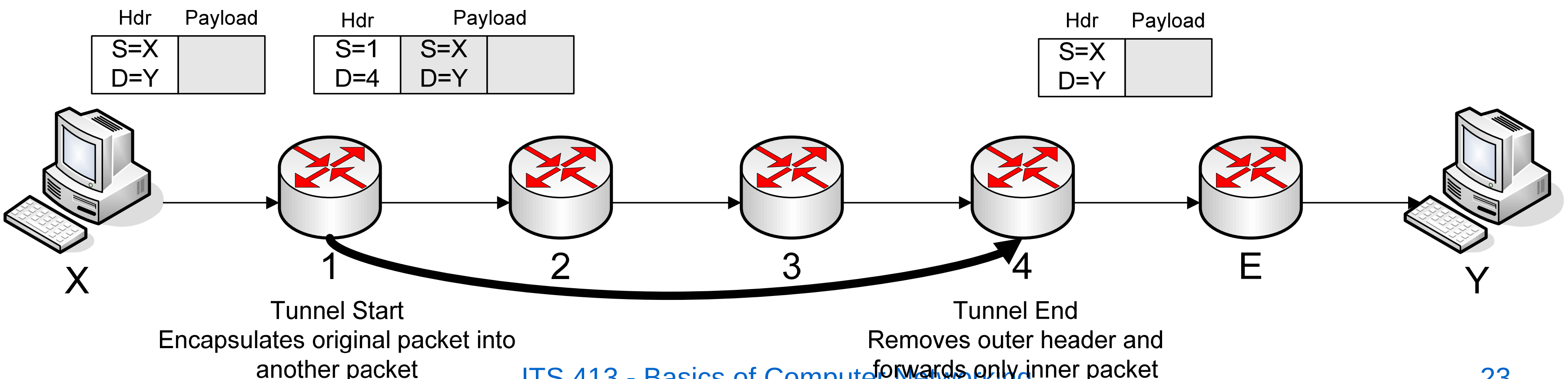
Input IP	Output IF
203.113.93.x	IF2
203.16.214.x	IF2
66.249.32.x	IF3
default	IF2

IF = Interface

Input IP (IF1)	Output IF
130.220.176.x	local
203.131.209.x	IF1
default	IF2

Tunnelling and Encapsulation

- Tunnelling: generic network mechanism for sending a packet inside another packet of the same type
  - Example: IP datagram inside another IP datagram
  - Encapsulate the packet inside another packet at one end-point of tunnel, and de-capsulate at other end-point of tunnel
- Used often for security, mobility, private networks, …
- Both tunnel end-points must be aware tunnelling is being used



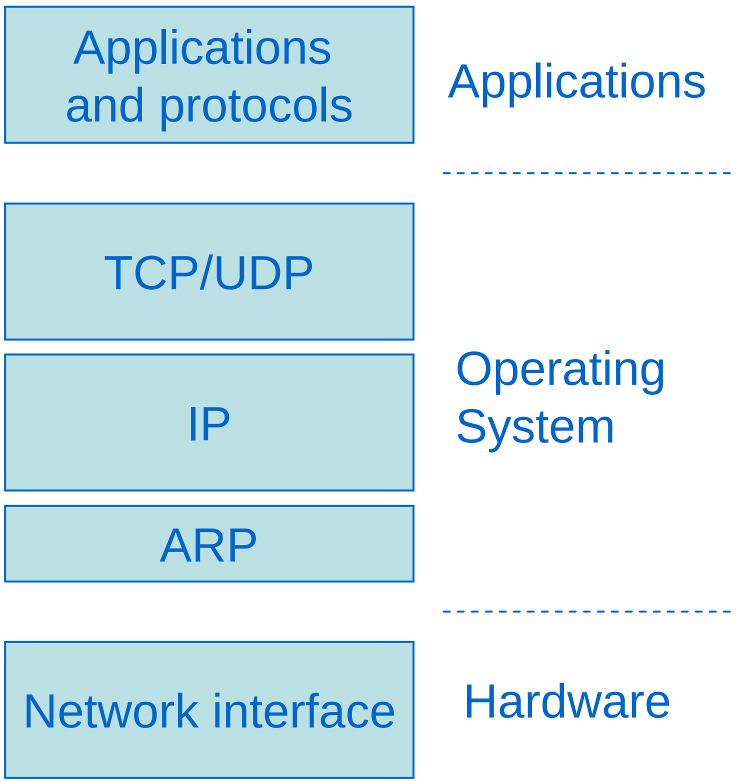
IPv6

- Aims to replace current version, IPv4, to:
  - Increase number of addresses (4 billion in v4, 3×10^{38} in v6!)
  - Update features, remove unused features
  - Incorporate security
- IPv6 has been defined and tested
  - Some experimental v6 networks are 10 years old
  - Operating systems and routers support IPv6
- Still not commonly used!
  - Minor issues with transition from IPv4 to IPv6
- We will look at some IPv6 mechanisms in later topic

Internet Protocols and Applications

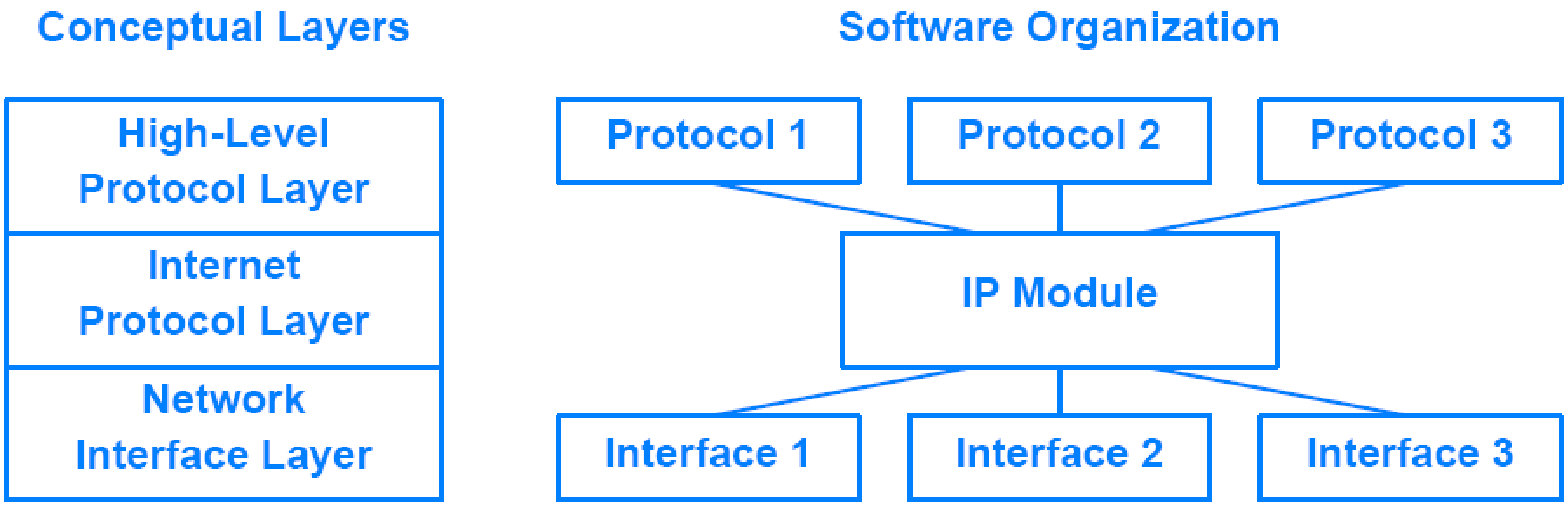
- Network (and Related) Protocols
  - IP
  - ICMP: error reporting and management
  - ARP: mapping IP addresses to hardware addresses
- Transport Protocols
  - Transmission Control Protocol (TCP): Reliable stream service
  - User Datagram Protocol (UDP): Unreliable, best effort protocol
  - Socket interfaces – programming Internet applications
- Application Protocols
  - HTTP: web browsing
  - FTP: file transfer
  - SMTP: email transfer
  - DNS: mapping domain names (www.google.com) to IP addresses (66.249.32.77)
- Many others!

Internet Protocol Implementation

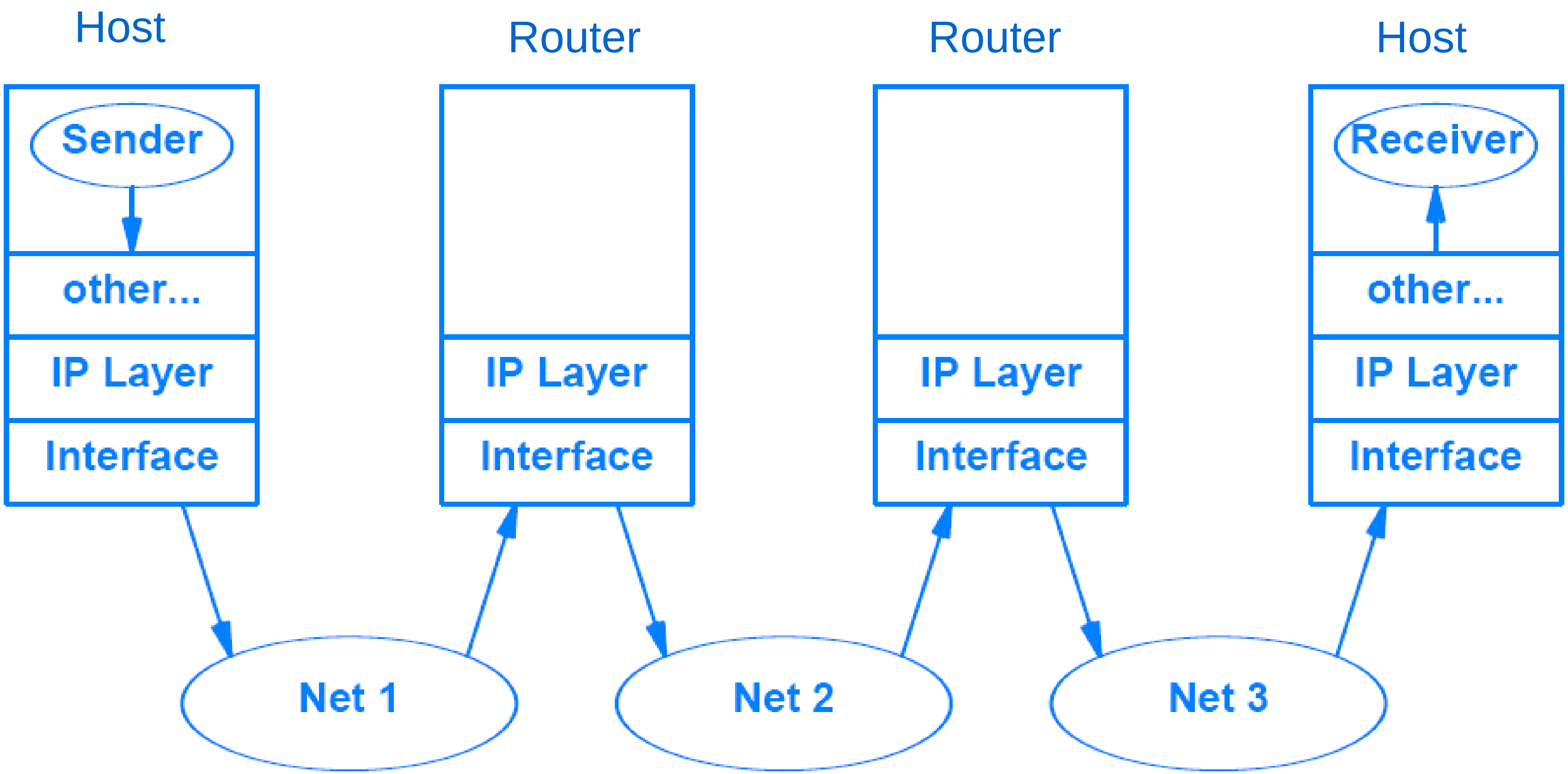


- Operating system implements TCP/IP
- Interface from application to OS is *sockets interface*
  - socket(), bind(), connect(), listen(), send(), recv(), …
- Interface from OS to hardware is *drivers*

Internet Protocol Implementation



Internet Protocol Implementation



Internet Application Architectures

- Client/Server Model (e.g. web, email, file transfer)
  - Requires an ‘always-on’ host as server, e.g. Web server
  - Server has well-known address (IP address, port)
  - Client sends request to server, and server responds
  - Reliable, manageable distributed system
- Peer-to-Peer Model (e.g. file sharing, telephony)
  - Hosts (peers) talk to each other; no central server
  - Peers are not required to be always-on, do not need well-known address
  - Highly scalable, but hard to manage
- Hybrids (e.g. file sharing, instant messaging)
  - Use server for indexing/searching, then peer-to-peer
- Implementations
  - P2P is still implemented with client/server processes; peer just runs both client and server

Network Analysis

- Software and hardware tools to analyse and understand network behaviour
- Ethereal (www.ethereal.com)
  - Open source software for capturing packets send by interface card
  - www.ethereal.com
- OPNET IT Guru (www.opnet.com)
  - Commercial network simulation and analysis software
- We will use Ethereal and OPNET IT Guru throughout course for examples and demonstration
- You should be familiar with basic operation of Ethereal
- Other useful (simple) tools
  - ping – test connectivity to a host
  - traceroute or tracert – identify route to host
  - nslookup – find match between domain name and IP address

OPNET IT Guru

- Free Academic Edition of IT Guru; limited capabilities
- Network modelling and analysis software. Basic steps of use:
  1. Create network topology: workstations, switches, routers, LANs/WANs, mobile devices, …
  2. Create application traffic: detailed models (e.g. file transfer); generic models (e.g. traffic from 10 LAN users); or real-captured traffic
  3. Define metrics/statistics of interest: network utilisation, throughput, delay, dropped packets, …
  4. Run simulation
  5. Analyse and plot results
- OPNET IT Guru used for planning new networks and analysing existing networks
  - What happens if we add 100 more users? What if we use protocol X instead of protocol Y?
- OPNET Modeller used for research of existing and new protocols
  - How will protocol X perform in this environment? How will my new protocol perform?