Communication Networks
The University of Kansas EECS 780
Preliminaries and Foundations

James P.G. Sterbenz

Department of Electrical Engineering & Computer Science
Information Technology & Telecommunications Research Center
The University of Kansas

jpgs@eecs.ku.edu

https://www.ittc.ku.edu/~jpgs/courses/nets

Comm. Network Preliminaries

PR.1  Network Architecture and Topology

PR.1  Network architecture and topology
PR.2  Performance metrics and style
PR.3  Theoretical foundations and network science
PR.4  Scope of communication
PR.5  Protocols and layering
PR.6  Communication flow diagrams
Network Architecture and Topology

The Network

- Collection *nodes* or *intermediate systems* (IS)
  - switches, routers, bridges, etc.
- Interconnected by *links* that
- Provide connectivity among *end systems* (ES) or *hosts* or *terminals*
  - desktops, laptops, servers, telephone handsets, etc.
  - note: in some networks nodes may be both ES and IS
- To support distributed *applications*
  - e.g. email, Web browsing, peer-to-peer file sharing
Heterogeneous Networks

- Disparate networks are interconnected by gateways
  - translate data packet formats
  - interoperate signalling and control

Application Relationships

- Client/server
  - e.g. Web browsing
  - data streams with embedded synchronisation

- Peer-to-peer
  - e.g. telepresence (video-conferencing)
Network Architecture and Topology

Group Communication Topologies

- Group communication
  - communication among participants in a group of nodes
- Topologies
  - unicast
  - anycast
  - k-cast
  - multicast
  - broadcast

Group Communication Topologies

Unicast

- Unicast
  - point-to-point
- Anycast
  - point-to-any in group
- k-cast
  - point-to-k receivers in group
- Multicast
  - point-to-multipoint
  - multipoint-to-multipoint
  - multipoint-to-point (reverse multicast or Concast)
- Broadcast
  - point-to-all
  - broadcast and select multicast
Group Communication Topologies

Anycast

- Unicast
  - point-to-point
- Anycast
  - point-to-any in group
- $k$-cast
  - point-to-$k$ receivers in group
- Multicast
  - point-to-multipoint
  - multipoint-to-multipoint
  - multipoint-to-point (reverse multicast or Concast)
- Broadcast
  - point-to-all
  - broadcast and select multicast

$k = 3$
Group Communication Topologies

**Multicast: Point-to-Multipoint**

- **Unicast**
  - point-to-point

- **Anycast**
  - point-to-any in group

- **k-cast**
  - point-to-k receivers in group

- **Multicast**
  - point-to-multipoint
  - multipoint-to-multipoint
  - multipoint-to-point (reverse multicast or Concast)

- **Broadcast**
  - point-to-all
  - broadcast and select multicast

---

**Multicast: Multipoint-to-Multipoint**

- **Unicast**
  - point-to-point

- **Anycast**
  - point-to-any in group

- **k-cast**
  - point-to-k receivers in group

- **Multicast**
  - point-to-multipoint
  - multipoint-to-multipoint
  - multipoint-to-point (reverse multicast or Concast)

- **Broadcast**
  - point-to-all
  - broadcast and select multicast
Group Communication Topologies

**Concast: Multipoint-to-Point**

- **Unicast**
  - point-to-point

- **Anycast**
  - point-to-any in group

- **k-cast**
  - point-to-\( k \) receivers in group

- **Multicast**
  - point-to-multipoint
  - multipoint-to-multipoint
  - multipoint-to-point (reverse multicast or Concast)

- **Broadcast**
  - point-to-all
  - broadcast and select multicast

---

Group Communication Topologies

**Broadcast**

- **Unicast**
  - point-to-point

- **Anycast**
  - point-to-any in group

- **k-cast**
  - point-to-\( k \) receivers in group

- **Multicast**
  - point-to-multipoint
  - multipoint-to-multipoint
  - multipoint-to-point (reverse multicast or Concast)

- **Broadcast**
  - point-to-all
  - broadcast and select multicast
Group Communication Topologies

**Multicast: Broadcast and Select**

- **Unicast**
  - point-to-point
- **Anycast**
  - point-to-any in group
- **k-cast**
  - point-to-k receivers in group
- **Multicast**
  - point-to-multipoint
  - multipoint-to-multipoint
  - multipoint-to-point (reverse multicast or Concast)
- **Broadcast**
  - point-to-all
  - broadcast and select multicast

Network Architecture and Topology

**Star vs. Mesh Topologies**

- **Star hierarchy**
- **Centralised control**
- **Examples**
  - PSTN
  - early enterprise nets (SNA)
    - later became meshes
- **Mesh**
- **Fully distributed control**
- **Examples**
  - ARPANET, Internet
  - DECnet
- **Spanning tree may be overlaid**
Performance Metrics

Style and Usage: Variables

- Proper style in formulæ is important
  - sloppy style can lead to ambiguity
  - guidelines at physics.nist.gov/cuu/Units/
- Variables, indices, constants, and parameters (scalar)
  - italic serif font if single character
    - $\pi = 3.14159...$
    - $c \approx 3 \times 10^8$ m/s speed of light
    - $r_i$ rate of the $i$th link
    - $r_{ij}$ rate of the $(i,j)$th link
    - multiplication is implicit: $ab = a \times b = a \cdot b$
  - never use * for multiplication in typeset documents
Performance Metrics

Style and Usage: Variables

- Variables, indices, constants, and parameters (scalar)
  - italic serif font if single character
  - Roman serif font if multiple characters
    - RTT round trip time
      - warning: $RTT = RTT = RT^2 
eq RT$
    - queuelen queue length
    - adjacent multiplication must be explicit, e.g.: $a \times RTT$
    - in LaTeX math mode declare using \texttt{...}
    - in MathType (MS Equation Editor) change style to "Text"

- cwnd TCP congestion window
- init_val initialisation value
- consolas (Windows) or Monaco (Mac) are better than courier
- multiplication must be explicit
## Performance Metrics

### Greek Letters

<table>
<thead>
<tr>
<th>Name</th>
<th>Lower</th>
<th>Capital</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>α</td>
<td>Α</td>
</tr>
<tr>
<td>beta</td>
<td>β</td>
<td>Β</td>
</tr>
<tr>
<td>gamma</td>
<td>γ</td>
<td>Γ</td>
</tr>
<tr>
<td>delta</td>
<td>δ</td>
<td>Δ</td>
</tr>
<tr>
<td>epsilon</td>
<td>η</td>
<td>Η</td>
</tr>
<tr>
<td>zeta</td>
<td>ξ</td>
<td>ξ</td>
</tr>
<tr>
<td>eta</td>
<td>ζ</td>
<td>Ζ</td>
</tr>
<tr>
<td>theta</td>
<td>θ</td>
<td>Θ</td>
</tr>
<tr>
<td>iota</td>
<td>ι</td>
<td>Ι</td>
</tr>
<tr>
<td>kappa</td>
<td>κ</td>
<td>Κ</td>
</tr>
<tr>
<td>lambda</td>
<td>λ</td>
<td>Λ</td>
</tr>
<tr>
<td>mu</td>
<td>μ</td>
<td>Μ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Lower</th>
<th>Capital</th>
</tr>
</thead>
<tbody>
<tr>
<td>nu</td>
<td>ν</td>
<td>Ν</td>
</tr>
<tr>
<td>xi</td>
<td>χ</td>
<td>Χ</td>
</tr>
<tr>
<td>omicron</td>
<td>ο</td>
<td>Ω</td>
</tr>
<tr>
<td>pi</td>
<td>Π</td>
<td>Π</td>
</tr>
<tr>
<td>rho</td>
<td>ρ</td>
<td>Ρ</td>
</tr>
<tr>
<td>sigma</td>
<td>σ</td>
<td>Σ</td>
</tr>
<tr>
<td>upsilon</td>
<td>υ</td>
<td>Υ</td>
</tr>
<tr>
<td>phi</td>
<td>φ</td>
<td>Φ</td>
</tr>
<tr>
<td>chi</td>
<td>χ</td>
<td>χ</td>
</tr>
<tr>
<td>psi</td>
<td>ψ</td>
<td>ψ</td>
</tr>
<tr>
<td>omega</td>
<td>ω</td>
<td>ω</td>
</tr>
<tr>
<td>digamma</td>
<td>ρ</td>
<td>Ρ</td>
</tr>
</tbody>
</table>

1, 0, b, capitals identical to Latin rarely used

### Style and Usage: Attributes

- **Attributes**
  - Roman font even if single character
    - \( d_q \) delay due to queuing: note \( q \) is **not** an index
    - \( d^j \) delay due to queuing: note \( q \) is **not** an exponent
    - explicitly declare in LaTeX math mode or MathType
Performance Metrics

Style and Usage: Num., Delimiters, Operators

• Numerals, delimiters, operators, numerals in Roman
  – \textit{never} italic
  – e.g. $z = 4 (x + y)$ \textit{not} $z = 4 (x + y)$
  – e.g. $\sum_i d_i$ \textit{not} $\sum_i d_i$
  – LaTeX math mode and MathType do it properly
  – be selective when selecting inline PowerPoint or Word style

• Do not use hyphen for subtraction
  – en-dash approximation but does not perfectly vertically align
  – e.g.: $a = b - c$ \textit{not} $a = b - c$

Performance Metrics

Style and Usage: Functions

• Functions: similar rules to variables
  – \textit{italic} font if single character
    • $f(x)$ \textit{function} \textit{f} applied to $x$
  – \textit{roman} font if multiple characters
    • $\max(x)$ maximum value of $x$
### Performance Metrics

#### Unit Multipliers

<table>
<thead>
<tr>
<th>SI decimal</th>
<th>EIC binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>10⁻¹</td>
<td>d</td>
</tr>
<tr>
<td>10⁻²</td>
<td>c</td>
</tr>
<tr>
<td>10⁻³</td>
<td>m</td>
</tr>
<tr>
<td>10⁻⁶</td>
<td>µ</td>
</tr>
<tr>
<td>10⁻⁹</td>
<td>n</td>
</tr>
<tr>
<td>10⁻¹²</td>
<td>p</td>
</tr>
<tr>
<td>10⁻¹⁵</td>
<td>f</td>
</tr>
<tr>
<td>10⁻¹⁸</td>
<td>a</td>
</tr>
<tr>
<td>10⁻²¹</td>
<td>z</td>
</tr>
<tr>
<td>10⁻²⁴</td>
<td>y</td>
</tr>
</tbody>
</table>

- d: deci
- c: centi
- m: milli
- µ: micro
- n: nano
- p: pico
- f: femto
- a: atto
- z: zepto
- y: yocto
- da: deka
- h: hecto
- k: kilo
- M: Mega
- G: Giga
- T: Tera
- P: Peta
- E: Exa
- Z: Zetta
- Y: Yotta

- m, µ, n, k, M, G, T, P are essential for networking and must be memorised.

### Style and Usage: Units and Prefixes

- **Units and prefixes**
  - **Roman font**
    - µs: microseconds
  - always insert space between number and unit
    - e.g. 4 m/s not 4m/s
  - use solidus or exponent for division rather than “p” (“per”)
    - e.g. 2.4 Gb/s or 2.4 Gbs⁻¹ not 2.4 Gbps
  - optionally use square brackets for unit dimensions
    - e.g. 3.25 [m/s]
  - data unit conventions
    - b for bit, B for Byte
    - k prefix for 10³, K prefix for 10²⁴

- m, µ, n, k, M, G, T, P are essential for networking and must be memorised.
Performance Metrics

Style and Usage: Plain Text

- Plain text style
  - frequently necessary to put formulae in plain text, e.g. email
  - no standard, but generally combination of C and LaTeX

- Operators
  * for explicit multiplication, e.g. \( a \times b = a \times b \)
  ^ for exponentiation, e.g. \( e^x = e^x \)

- Variables
  _ for subscripts, e.g. \( x_i = x_i \)
  ^ for superscripts, e.g. \( x^i = x^i \)

- Scientific notation
  * and ^, e.g. \( 3 \times 10^{-5} = 3.5 \times 10^{-5} \)

- Greek letters
  - spell out, e.g. \( \pi \)
  - u for micro (\( \mu \)) prefix if unambiguous, e.g. \( 50 \ \mu s = 50 \ \mu s \)
Performance Metrics

Delay and Bandwidth

- Delay or latency
  - $D$: end-to-end delay
  - $d$: per hop delay
    - Jitter is delay variance
- Bandwidth or data rate
  - $B$: aggregate bandwidth
  - $b$: per flow bandwidth
    - Not channel capacity (bandwidth in EE sense)
- Bandwidth-$\times$-delay product
  - Number of bits in flight on a high-speed path
  - $b$ [bits/sec] $\times$ $d$ [sec] = [bits]

Network Path Latency

- Delays sum along a path
  - Benefit of optimising link directly proportional to contribution
Performance Metrics
Path Length and Network Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Channel</th>
<th>Distance</th>
<th>RTT</th>
<th>BW×delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAN personal</td>
<td>RF</td>
<td>10 m</td>
<td>100 ns</td>
<td>.05 b</td>
</tr>
<tr>
<td>SAN system</td>
<td>Cu/Fiber</td>
<td>100 m</td>
<td>1 μs</td>
<td>½ b</td>
</tr>
<tr>
<td>LAN local</td>
<td>Cu/Fiber/RF</td>
<td>1 km</td>
<td>10 μs</td>
<td>5 b</td>
</tr>
<tr>
<td>MAN metropolitan</td>
<td>Fiber/RF</td>
<td>100 km</td>
<td>1 ms</td>
<td>500 b</td>
</tr>
<tr>
<td>WAN transon. wide</td>
<td>Fiber</td>
<td>5 000 km</td>
<td>50 ms</td>
<td>25 kb</td>
</tr>
<tr>
<td>WAN global wide</td>
<td>Fiber</td>
<td>20 000 km</td>
<td>200 ms</td>
<td>100 kb</td>
</tr>
<tr>
<td>LEO* low earth</td>
<td>RF</td>
<td>2–1 000 km</td>
<td>25 ms</td>
<td>12 kb</td>
</tr>
<tr>
<td>GEO geosynchronous</td>
<td>RF/laser</td>
<td>2–36 000 km</td>
<td>480 ms</td>
<td>240 kb</td>
</tr>
<tr>
<td>DSN earth–moon</td>
<td>RF/laser</td>
<td>400 000 km</td>
<td>2.5 s</td>
<td>1.2 Mb</td>
</tr>
<tr>
<td>IPN♂ interplanetary</td>
<td>RF/laser</td>
<td>55–400×10⁶ km</td>
<td>6–45 min</td>
<td>1.3 Gb</td>
</tr>
<tr>
<td>IPN♀ interplanetary</td>
<td>RF/laser</td>
<td>10¹² km</td>
<td>2 hr</td>
<td>3.6 Gb</td>
</tr>
<tr>
<td>IPN♂ space interplanetary</td>
<td>RF/laser</td>
<td>10¹³ km</td>
<td>20 hr</td>
<td>36 Gb</td>
</tr>
</tbody>
</table>

- Maximum bandwidth limited by bottleneck link
  - there is no point in optimising a link that is not a bottleneck
Performance Metrics

Error and Loss Characteristics

- Error and loss characteristics
  - \( \text{Pr}[\text{bit-error}] \)
  - burst error (multibit)
  - channel fades (e.g. rain)
  - episodic link connectivity
  - link and node failures

*We’ll discuss why these happen later*

Comm. Network Preliminaries

PR.3 Theoretical Foundations

PR.1 Network architecture and topology
PR.2 Performance metrics and style
PR.3 Theoretical foundations and network science
  - PR.3.1 Probability, queueing theory, and performance
  - PR.3.2 Information theory
  - PR.3.3 Graph theory and network topology
  - PR.3.4 Game theory and tussle
  - PR.3.5 Control Theory
  - PR.3.6 Fault tolerance and dependability
  - PR.3.7 Complex systems
PR.4 Scope of communication
PR.5 Protocols and layering
PR.6 Communication flow diagrams
Science of Communication Nets

Introduction

- A number of disciplines are fundamental to comm nets
  - probability, queueing theory, and performance EECS 86x
  - information theory EECS 769
  - graph theory and network topology EECS 718
  - game theory and tussle
  - control theory EECS 444
  - fault tolerance and dependability EECS 983
  - complex systems

EECS 784 covers many of these

see also [Keshav 2012]

Network Science

Introduction

- Network science
  - “organized knowledge of networks based on their study using the scientific method” [NRC 2005]
  - theory of the interactions of entities organised as nets
  - network examples [Newman 2010]
    - technological networks (e.g. PSTN, Internet, grid, transport)
    - social networks (e.g. genealogy, friendship)
    - information networks (e.g. citation nets, Web)
    - biological networks (e.g. biochemical, ecological)

- We are primarily interested with
  - communication networks (including the Web and social)
Network Science
Science of Communication Networks

- Fundamental disciplines (e.g.)
  - graph theory and flows
  - game theory and tussle
  - control theory and automata
  - fault tolerance and dependability
  - queuing theory

Graph Theory
Graph Components

- Model network as a graph
- Vertex | node

- Edge | link
Graph Theory

Graph Components

- Model network as a graph
- Vertex | node: \( v \)
  - switch, Internet router, end system
  - represented as • •
- Edge | link
  - \( e_i = \{v_j, v_k\} \)
  - \( e_i \) is joins and is incident to \( v_j \) and \( v_k \)
  - \( v_j, \leftrightarrow v_k \) denotes neighbours such that \( v_j \) is adjacent to \( v_k \)
  - wired physical link, wireless link association, P2P relationship
  - generally undirected | bidirectional link
  - may be directed | unidirectional link
Graph Theory

Graph Definition

- **Graph** $G = (V, E)$ is a triple $E \subseteq V \times V$
  - set of vertices $V(G) = \{v_0, v_1, \ldots\}$ correspond to links
  - set of edges $E(G) = \{e_0, e_1, \ldots\}$ correspond to nodes
  - mapping of $\forall e_j \rightarrow$ endpoint pairs $\{v_j, v_k\}$ gives net topology
    - not the mathematical definition of topology but in common use

- **Example**
  Königsberg graph with multiedges + (self-)loop
  - $V = \{v_0, v_1, v_2, v_3\}$
  - $E = \{e_0, e_1, e_2, e_3, e_4, e_5, e_6\}$
  - $e_0,e_1 = \{v_0,v_1\}$, $e_2,e_3 = \{v_0,v_2\}$, $e_4 = \{v_0,v_3\}$,
    $e_5 = \{v_1,v_3\}$, $e_6 = \{v_2,v_3\}$, $e_7 = \{v_3,v_3\}$

Graph Theory

Graph Properties

- **Structural**
  - adjacency matrix
  - degree distribution

- **Connectivity**
  - how well connected, e.g. clustering coefficient

- **Centrality**
  - importance of a node or link, e.g. betweenness
Comm. Network Preliminaries

**PR.4 Scope of Communication**

PR.1 Network architecture and topology
PR.2 Performance metrics and style
PR.3 Theoretical foundations and network science
PR.4 Scope of communication
PR.5 Protocols and layering
PR.6 Communication flow diagrams

---

Network Components

**ES vs. IS**

- End system (ES)
  
  *what is it?*
Network Components

ES vs. IS

- End system (ES)
  - computing system that runs applications (e.g. server)
  - may be used by a human user (e.g. client)

- Intermediate system (IS)
  what is it?
Network Components

**ES vs. IS**

- **End system (ES)**
  - computing system that runs applications (e.g. server)
  - may be used by a human user (e.g. client)
- **Intermediate system (IS)**
  - network component used to interconnected distributed ESs
  - examples: switches, routers, hubs

Some systems may be both ES and IS:

*examples?*
Network Components

ES vs. IS

- **End system (ES)**
  - computing system that runs applications (e.g. server)
  - may be used by a human user (e.g. client)
- **Intermediate system (IS)**
  - network component used to interconnected distributed ESs
  - examples: switches, routers, hubs
- **Some systems may be both ES and IS**
  - ES may also transit traffic in multihop MANET *Lecture MW*
  - IS may also run applications in active programmable net

Scope of Communication

E2E vs. HBH Definitions

- **Hop-by-hop (HBH)**
  - *what is it?*
Scope of Communication
E2E vs. HBH Definitions

• Hop-by-hop (HBH)
  – communication or link between directly attached nodes
  – typically IS – IS or ES – IS
  – may rarely by ES – ES (no network)

• Edge-to-edge (e2e)
  – communication or link between edges of a subnetwork
• Hop-by-hop (HBH)
  – communication or link between directly attached nodes
  – typically IS – IS or ES – IS
  – may rarely be ES – ES (no network)

• Edge-to-edge (e2e)
  – communication or link between edges of a subnetwork

• End-to-end (E2E)
  what is it?

• Hop-by-hop (HBH)
  – communication or link between directly attached nodes
  – typically IS – IS or ES – IS
  – may rarely be ES – ES (no network)

• Edge-to-edge (e2e)
  – communication or link between edges of a subnetwork

• End-to-end (E2E)
  – communication or path between end systems: ES – ES
  – typically involves multiple HBH segments
Scope of Communication

E2E vs. HBH Definitions

- Hop-by-hop (HBH)
  - communication or link between directly attached nodes
  - typically IS – IS or ES – IS
  - may rarely by ES – ES (no network)
- Edge-to-edge (e2e)
  - communication or link between edges of a subnetwork
- End-to-end (E2E)
  - communication or path between end systems: ES – ES
  - typically involves multiple HBH segments
- Application-to-application (A2A)
  - communication between applications (similar to E2E)

E2E vs. HBH Examples

- End system
- Intermediate system
  - edge or access switch
  - core or backbone switch
- multihomed
- HBH
- E2E
Comm. Network Preliminaries

PR.5  Protocols and Layering

PR.1  Network architecture and topology
PR.2  Performance metrics and style
PR.3  Theoretical foundations and network science
PR.4  End-to-end vs. hop-by-hop
PR.5  Protocols and layering
PR.6  Communication flow diagrams

Protocols and Services

Definition

What is a protocol?
Protocols and Services

Definition

- Protocol: rules for communication between entities
  - message format and sequence
    - information transfer (data plane)
    - signalling of control information (control plane)
    - monitoring and management (management plane)
  - definition of actions (state machine)
**Protocols and Services**

**Definition**

- **Protocol**: rules for communication between entities
  - message format and sequence
    - information transfer (data plane)
    - signalling of control information (control plane)
    - monitoring and management (management plane)
  - definition of actions (state machine)
- **Service**: functional primitives provided by layer
- **Interface**: service interface to layers above and below

*Proper design separates protocols from services*

---

**Protocol Layering**

**Protocol Layering**

- **Layering provides service abstraction**
  - isolate: protocols, components, technology
    - any transport layer over IP
    - IP over any link layer
    - commodity link layer chip evolution, e.g.
      - 10BASE-T → 100BASE-T → 1000BASE-X
      - → 802.11b → 802.11g

---

*24 August 2015 KU EECS 780 – Comm Nets – Preliminaries NET-PR-61*
Protocol Layering

Layering is useful abstraction
- thinking about networking system architecture
- organising protocols based on role
  2. link
  3. switch
  4. end system

ISO 7498: open systems interconnection

Attempt to formalise needed:
- protocol layers and their services
- interfaces between layers
ISO 7498: open systems interconnection
- protocol: rules for communication between entities

<table>
<thead>
<tr>
<th>Layer</th>
<th>Protocol Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>application</td>
</tr>
<tr>
<td>6</td>
<td>presentation</td>
</tr>
<tr>
<td>5</td>
<td>session</td>
</tr>
<tr>
<td>4</td>
<td>transport</td>
</tr>
<tr>
<td>3</td>
<td>network</td>
</tr>
<tr>
<td>2</td>
<td>link</td>
</tr>
<tr>
<td>1</td>
<td>physical</td>
</tr>
</tbody>
</table>

• ISO 7498: open systems interconnection
  - protocol: rules for communication between entities

Real implementations
- ISO model missed medium access control
- presentation layer
  - not sensible to standardise
  - not necessarily right layer of stack
- session layer
  - generally not needed for data
  - useful for control (e.g. SIP, H.323)
Protocol Layering

**OSI Model**

- **Send ES**
  - application
  - presentation
  - session
  - transport
  - network
  - link
  - physical

- **Receive ES**
  - application
  - presentation
  - session
  - transport
  - network
  - link
  - physical

**Performance Issues**

- Layered implementations may perform very poorly
- Inter-layer transfers involve non-trivial overhead
  - encapsulation/decapsulation of PDUs
  - inter-layer control transfer
    - context switching and data copying
  - effects of overlapping intra-layer control mechanisms
- Protocol layers should be designed with this in mind
  - antithesis of layering to isolate protocols and technology
Protocol Layering

Planes

- Data plane
  - role of data plane?

  8: social layer  social information exchange
  7: application layer  application-to-application exchange
  4: transport layer  end-to-end flow
  3: network layer  forwarded through switch/router
  2.5: virtual link layer  hop-by-hop over virtual link
      (concatenation of physical links)
  2: link layer  hop-by-hop over a link
  1: physical layer  bits as signals in a medium
Protocol Layering

Hybrid Layer/Plane Cube

<table>
<thead>
<tr>
<th>Layer</th>
<th>Plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>physical</td>
</tr>
<tr>
<td>L1.5</td>
<td>social</td>
</tr>
<tr>
<td>L2</td>
<td>link</td>
</tr>
<tr>
<td>L2.5</td>
<td>virtual link</td>
</tr>
<tr>
<td>L3</td>
<td>network</td>
</tr>
<tr>
<td>L4</td>
<td>transport</td>
</tr>
<tr>
<td>L5</td>
<td>application</td>
</tr>
<tr>
<td>L7</td>
<td>application</td>
</tr>
<tr>
<td>L8</td>
<td>data plane</td>
</tr>
</tbody>
</table>

Planes

- **Data plane**
  - information transfer
- **Control plane**
  - role of control plane?
Protocol Layering

**Planes**

- **Data plane**
  - information transfer

- **Control plane**
  - signalling to control information transfer, including:
    - flow or connection establishment/modification/termination
    - error control
    - flow and congestion control
  - control of network components and organisation, e.g.
    - network topology and connectivity
  - some control plane layers do not correspond to data plane

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>session layer</td>
</tr>
<tr>
<td>1.5</td>
<td>MAC layer</td>
</tr>
</tbody>
</table>

**Hybrid Layer/Plane Cube**

- **data plane**
- **control plane**

<table>
<thead>
<tr>
<th>Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>L8</td>
</tr>
<tr>
<td>L7</td>
</tr>
<tr>
<td>L5</td>
</tr>
<tr>
<td>L4</td>
</tr>
<tr>
<td>L3</td>
</tr>
<tr>
<td>L2.5</td>
</tr>
<tr>
<td>L2</td>
</tr>
<tr>
<td>L1.5</td>
</tr>
<tr>
<td>L1</td>
</tr>
</tbody>
</table>

- social
- application
- session
- transport
- network
- virtual link
- link
- MAC
- physical
Protocol Layering

Planes

- **Data plane**
  - information transfer
- **Control plane**
  - signalling to control information transfer
  - control of network components and organisation
- **Management plane**
  - role of management plane?

- monitoring and management of network and its elements
- cuts across all layers
Protocol Layering

Hybrid Layer/Plane Cube

- L8: social
- L7: application
- L5: session
- L4: transport
- L3: network
- L2.5: virtual link
- L2: link
- L1.5: MAC
- L1: physical

Protocol Layering

Overlays

- Sometimes reasons to run additional *overlay* layers
  - layer 3 over layer 3
    - e.g. IP over ATM
  - layer 3 over layer 7
    - e.g. P2P addressing and routing for file sharing

- Overlay definition
  - layer *n* over layer *m*, where *n* ≤ *m*
  - overlay is over an *underlay*
Protocol Layering

Internet Hourglass

- Internet “hourglass”
- Common network layer: IP
  - common addressing essential for seamless interworking
  - compatible routing & signalling

TCP | UDP | RTP | •••
Protocol Layering

**Internet Hourglass**

- Internet “hourglass”
- Common network layer: IP
- Any transport layer above
- Any link layer below
  - SONET, 802.n, ...

![Diagram of protocol layering]

### Internet Protocols

**Important Link and MAC Protocols**

<table>
<thead>
<tr>
<th>Common name</th>
<th>Standard</th>
<th>Scope</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet</td>
<td>IEEE 802.3</td>
<td>LAN/MAN</td>
<td>wire, fiber</td>
</tr>
<tr>
<td>Token ring</td>
<td>IEEE 802.5</td>
<td>LAN</td>
<td>wire</td>
</tr>
<tr>
<td>WirelessLAN WiFi</td>
<td>IEEE 802.11</td>
<td>LAN</td>
<td>RF, (IR)</td>
</tr>
<tr>
<td>WPAN</td>
<td>IEEE 802.15</td>
<td>PAN</td>
<td>RF</td>
</tr>
<tr>
<td>WirelessMAN WiMAX</td>
<td>IEEE 802.16</td>
<td>MAN</td>
<td>RF</td>
</tr>
<tr>
<td>SONET</td>
<td>ANSI T1.105</td>
<td>MAN/WAN</td>
<td>fiber electronic switch</td>
</tr>
<tr>
<td>OTN</td>
<td>ITU G.709</td>
<td>MAN/WAN</td>
<td>fiber optical switch</td>
</tr>
</tbody>
</table>

IEEE 802 network standards are available from standards.ieee.org/getieee802/portfolio.html
### Important Network Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Name</th>
<th>Function</th>
<th>Status</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Internet protocol</td>
<td>addressing datagram forwarding</td>
<td>standard</td>
<td>RFC 0791</td>
</tr>
<tr>
<td>ICMP</td>
<td>Internet control message protocol</td>
<td>signalling</td>
<td>standard</td>
<td>RFC 0792</td>
</tr>
<tr>
<td>IGMP</td>
<td>Internet group management protocol</td>
<td>multicast signalling</td>
<td>proposed</td>
<td>RFC 3376</td>
</tr>
<tr>
<td>BGP</td>
<td>border gateway protocol</td>
<td>interdomain routing</td>
<td>draft</td>
<td>RFC 1771</td>
</tr>
<tr>
<td>OSPF</td>
<td>open shortest path routing</td>
<td>intradomain routing</td>
<td>standard</td>
<td>RFC 2328</td>
</tr>
<tr>
<td>ISIS</td>
<td>intermediate system-intermediate system</td>
<td>intradomain routing</td>
<td>proposed</td>
<td>ISO10589</td>
</tr>
<tr>
<td>DNS</td>
<td>domain name system</td>
<td>domain name to IP address resolution</td>
<td>standard</td>
<td>RFC 1035</td>
</tr>
</tbody>
</table>

RFCs are available from [www.rfc-editor.org](http://www.rfc-editor.org)

---

### Important Transport Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Name</th>
<th>Function</th>
<th>Status</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>transmission control protocol</td>
<td>reliable data transfer with congestion control</td>
<td>standard</td>
<td>RFC 0793</td>
</tr>
<tr>
<td>UDP</td>
<td>user datagram protocol</td>
<td>socket access to unreliable IP datagrams</td>
<td>standard</td>
<td>RFC 0768</td>
</tr>
<tr>
<td>RTP</td>
<td>real-time protocol</td>
<td>streaming media (typically over UDP)</td>
<td>standards track</td>
<td>RFC 1889</td>
</tr>
<tr>
<td>T/TCP</td>
<td>TCP for transactions</td>
<td>remote login</td>
<td>experimental</td>
<td>RFC 1644</td>
</tr>
<tr>
<td>RDP</td>
<td>reliable data protocol</td>
<td>reliable data transfer with no congestion control</td>
<td>experimental</td>
<td>RFC 0908</td>
</tr>
<tr>
<td>SCTP</td>
<td>stream control transmission protocol</td>
<td>signalling</td>
<td>proposed</td>
<td>RFC 2960</td>
</tr>
</tbody>
</table>

---

24 August 2015

KU EECS 780 – Comm Nets – Preliminaries

NET-PR-83

NET-PR-84
## Internet Protocols

### Important “Application Layer” Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Name</th>
<th>Function/ Use</th>
<th>Status</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>hypertext transfer protocol</td>
<td>Web browsing</td>
<td>draft</td>
<td>RFC 2616</td>
</tr>
<tr>
<td>FTP</td>
<td>file transfer protocol</td>
<td>file and document transfer</td>
<td>standard</td>
<td>RFC 0959 STD 0009</td>
</tr>
<tr>
<td>Telnet</td>
<td>telnet</td>
<td>remote login</td>
<td>standard</td>
<td>RFC 0854 STD 0008</td>
</tr>
<tr>
<td>SMTP</td>
<td>simple mail transfer protocol</td>
<td>email relay and delivery</td>
<td>standard</td>
<td>RFC 0821 STD 0010</td>
</tr>
<tr>
<td>POP</td>
<td>post office protocol</td>
<td>server mail download</td>
<td>standard</td>
<td>RFC 1939 STD 0053</td>
</tr>
<tr>
<td>IMAP</td>
<td>internet message access protocol</td>
<td>server mail access</td>
<td>proposed</td>
<td>RFC 3501</td>
</tr>
<tr>
<td>NFS</td>
<td>network file system</td>
<td>remote access to files</td>
<td>proposed</td>
<td>RFC 3530</td>
</tr>
<tr>
<td>RTSP</td>
<td>real-time streaming protocol</td>
<td>control of multimedia streaming</td>
<td>proposed</td>
<td>RFC 2326</td>
</tr>
</tbody>
</table>
Communication Flow Diagrams

Motivation

- Complex network interactions are hard to understand
- Visualisation technique: *communication flow diagram*

Concepts

- Time and space
Communication Flow Diagrams

Concepts: Distance

- Time and space
  - distance represented horizontally

Concepts: Time

- Time and space
  - distance represented horizontally
  - time represented vertically
Communication Flow Diagrams

Concepts: Bit Propagation

- Bit propagation
  - how represented?

- distance \( d_{12} \) from node 1 to node 2
- propagation delay \( t_p \)
  - how computed?

\[ \text{time} \]
\[ \text{time} \]
Communication Flow Diagrams

Concepts: Bit Propagation

- Bit propagation
  - distance $d_{12}$ from node 1 to node 2
  - propagation delay
    $t_p [s] = \frac{d_{12} [m]}{kc [m/s]}
    - k constant for particular medium
    - speed of light $c \approx 3 \times 10^8$ m/s

- slope of bit flow
  can this vary within a diagram?
Communication Flow Diagrams

Concepts: Bit Propagation

- Bit propagation
  - distance \( d_{12} \) from node 1 to node 2
  - propagation delay
    \[
    t_p [s] = \frac{d_{12} [m]}{kc [m/s]}
    \]
    - \( k \) constant for particular medium
    - speed of light \( c \approx 3 \times 10^8 \text{ m/s} \)
    - slope is velocity: only affected by \( k \) (\( k \approx 0.6 \) for fiber)
    - \( c \) constant unless you alter the laws of physics
    - *don’t sketch these with widely varying slopes!*

Communication Flow Diagrams

Concepts: Packet Transmission

- Packet transmission
  - sequence of bits transmitted on an interface
  - transmission delay \( t_b \)

\[ \text{how computed?} \]
Packet transmission
- sequence of bits transmitted on an interface
- transmission delay
  \[ t_b [s] = \frac{b [b]}{r [b/s]} \]
  - \( r \) rate in bits/sec
  - \( b \) number of bits

Packet propagation
- how to represent?
- what shape?
Communication Flow Diagrams

Concepts: Packet Propagation

- Packet propagation
  - packet is parallelogram
    - width is distance
    - thickness is packet size
Communication Flow Diagrams

Concepts: Packet Propagation

- Packet propagation
  - packet is parallelogram
    - width is distance
    - thickness is packet size
  - total HBH delay: \( t_b + t_p \)
    - \( t_b \) to transmit object
    - \( t_p \) for last bit to propagate

Communication Flow Diagrams

Concepts: Multihop Packet Propagation

- Multihop propagation
  - processing delay in switches and routers
  - forwarding delay
    - processing time \( t_i \)
**Communication Flow Diagrams**

**Concepts: Multihop Packet Propagation**

- Multihop propagation
  - processing delay in switches and routers
  - forwarding delay
    - processing time $t_i$
    - before retransmission
      - on next hop

**Communication Flow Diagrams**

**Concepts: Signalling Message**

- Packets are parallelograms
- Signalling messages are directed line segments
  *why?*
• Packets are parallelograms
• Signalling messages are directed line segments
  – generally short with respect to data (thin parallelogram)
  – makes diagrams easier to draw and read
  • labelled with message name and parameters

Communication Flow Diagrams

Concepts: Signalling Message

Summary

• Complex network interactions are hard to understand
• Visualisation technique: communication flow diagram
  – distance is represented horizontally
  – time is represented vertically
  – signalling messages are arcs with labels
  – packets are parallelograms
  – keep slope constant
• It is essential that you are very confident with these!
  – you will need to properly draw these on the exams
Communication Flow Diagrams

Example: Connection Establishment

- Signalling message exchange
  - connection SETUP
Communication Flow Diagrams

Example: Connection Establishment

- Signalling message exchange
  - connection **SETUP**
Communication Flow Diagrams

Example: Connection Establishment

- Signalling message exchange
  - connection **SETUP**
  - **CONNECT**ion established
Communication Flow Diagrams

Example: Connection Establishment

- Signalling message exchange
  - connection SETUP
  - CONNECTION established

- Data transfer
Comm. Network Preliminaries

Acknowledgements

Some material in these foils is based on the textbook

- Sterbenz and Touch,
  *High-Speed Networking, A Systematic Approach to High-Bandwidth Low Latency Communication*

Comm. Network Preliminaries

Further Reading

Some material in these foils is based on the textbook

- S. Keshav,
  *Mathematical Foundations of Computer Networking*