Communication Networks
The University of Kansas EECS 780
Network Layer

James P.G. Sterbenz
Department of Electrical Engineering & Computer Science
Information Technology & Telecommunications Research Center
The University of Kansas

jgps@eecs.ku.edu

http://www.ittc.ku.edu/~jgps/courses/nets
Network Layer

Outline

NL.1 Network layer functions and services
NL.2 Network signalling paradigms
NL.3 Switches and packet structure
NL.4 Examples
   NL.4.1 PSTN and X.21
   NL.4.2 X.25 CONS
   NL.4.3 ISDN and Frame Relay
   NL.4.4 B-ISDN and ATM
   NL.4.5 MPLS
   NL.4.6 Internet: DNS, IP, ICMP, and IPv6
NL.5 Fast datagram routers
Network Layer

NL.1 Functions and Services

NL.1 Network layer functions and services
NL.2 Network signalling paradigms
NL.3 Switches and packet structure
NL.4 Examples
NL.5 Fast datagram routers
Layer 3: path control in control plane and forwarding in data plane

Interaction with management plane particularly important
Network Layer

Definitions

- **Forwards packets** along a *path* (or *route*) through the network to a destination *address*: either as individual *datagrams*, as a *flow* of datagrams, or on a *connection* across a *circuit* or *virtual circuit*. 
Network Layer

Sublayer History

• Network layer may be subdivided into sublayers
  – ad hoc designations without compete agreement

• History
  – IP originally *internetworking layer* above *subnetnetwork* layer
    • OSI model also contained this [ISO 8648]
  – IP evolved from internetworking (~3.5) layer...
  – ...to global *network layer*
Network Layer
Sublayers

• Network layer may be subdivided into sublayers
  – ad hoc designations without compete agreement

• Internetworking over IP (possibly L3.5)
  – research proposals such as PoMo (Postmodern Internet)

• Network L3: routing and forwarding
  – generally IP in the global internet

• Subnetwork: network technologies under IP
  – e.g. MANETs (mobile ad hoc nets), WSNs (sensor nets)

• Virtual link or topology (commonly L2.5)
  – underlays that provide a topology or virtual link, e.g. MPLS
Layer 2.5: topology in control plane and forwarding in data plane.

Interaction with management plane particularly important.

Virtual Link Layer

Hybrid Layer/Plane Cube

- management plane
- data plane
- control plane

L8
- social
L7
- application
L5
- session
L4
- transport
L3
- network
L2.5
- virtual link
L2
- link
L1.5
- MAC
Network Layer
Network Protocols and Devices

- **Network protocol**
  - responsible for determining on *which* link frame transmitted
  - moves packets on path through the network between *nodes*
Network Layer
Service and Interfaces

• Network layer 3 is above link layer 2
  – **addressing**: network layer identifier for end systems (hosts)
  – **forwarding**: transfers packets hop-by-hop
    • using link layer services
    • network layer responsible for determining *which* next hop
  – **routing**: determination of path to forward packets *Lect NR*
  – **signalling**: messages to control network layer behaviour
  – **traffic management**: management of traffic and congestion *Lecture TQ*

• Network layer service to transport layer (L4)
  – deliver TPDUs to destination transport entity
Network Layer
Forwarding vs. Routing

• **Forwarding** transfers packets at each hop
  – each switch (router) makes decision on which link to send
  – forwarding table (generally) used to make decision
  – forwarding is *per packet* decision
    
    *analogy: determining which exits to take on a drive*

• **Routing** determines the path to take
  – routing algorithm independent of forwarding
  – forwarding table entries populated by routing
  – routing is (generally) not done per packet
    
    *analogy: planning trip from source to destination*

*Forwarding and routing are very different*
Network Layer
Service and Interfaces

- Network layer *packet* encapsulates *TPDU*
  - *packet* = *header* + *TPDU* + *opt. trailer* (protocol dependent)
Network Layer Service

Service Models: Best Effort

- **Best effort**
  - network attempts to:
    - deliver most packets
    - most of the time
    - eventually
  - network *may* attempt to be fair among users
  - this is the Internet service model
  - contrast with best-effort applications

- **Differentiated service**

- **Guaranteed service**
Network Layer Service

Service Models: Differentiated Service

- Best effort
- Differentiated service
  - some users or traffic types receive preferential treatment
  - may provide statistical targets for service performance
  - this model has been proposed as DiffServ for the Internet
- Guaranteed service

*Lecture TQ*
Network Layer Service

Service Models: Guaranteed Service

- Best effort
- Differentiated Service
- Guaranteed service  \textit{Lecture TQ}
  - some (perhaps statistical) guarantees on service
    - delivery (reliability)
    - performance (delay, bandwidth, jitter, etc)
  - this model is provided by the PSTN and ATM networks
  - this model has been proposed by IntServ for the Internet
Network Layer Service

Granularity

- Granularity of service parameters
  - per packet: applies to individual packets
  - per flow: applies to sequence of packets between hosts
Network Layer Service

Service Models: Reliable Delivery

- Reliable delivery
  - all packets *eventually* reach destination
    - with very high probability
    - assuming no major network outage
  - recall E2E arguments
    - link vs. network vs. transport vs. app
  - recall sources of errors *Lecture TL*
Network Layer Service

Service Models: Statistical Reliability

- Reliable delivery
- Statistical reliability
  - packets delivered with probability $p$
Network Layer Service
Service Models: Unreliable

- Reliable delivery
- Statistical reliability
- Unreliable: packet may or may not reach destination
  - if necessary, reliability provided by higher layer
    - end-to-end transport
    - application-to-application
Network Layer Service
Service Models: Ordered Delivery

- Ordered: packets delivered in order
Network Layer Service
Service Models: Unordered Delivery

- Ordered
- Unordered: packets may be misordered
  why is this ok?
Network Layer Service
Service Models: Unordered Delivery

- Ordered
- Unordered: packets may be misordered
  - transport layer will reorder
  - application will reorder
  - application doesn’t care about order
Network Architecture

Characteristics

• Network establishes paths between end systems
  – all applications (that need to) must be able to communicate

• Heterogeneity of
  – underlying links and LAN technologies
  – overlying applications
  – service providers

• Requires common addressing mechanism
  – and compatible routing and signalling
Network Architecture

Hourglass Principle

- Internet is “waist of the hourglass”
  - common addressing and forwarding (IP)
  - compatible routing (BGP) and signalling (ICMP)

- The network layer is the hardest to replace or evolve
  - even to new versions (e.g., IPv4 → IPv6)

- IP won over alternatives
  - e.g., X.25 CONS, CLNP, SNA, XNS, DECNET, ATM, ...
  - but ideas in these architectures still important
Network Layer

NL.2  Network Signalling Paradigms

NL.1  Network layer functions and services
NL.2  Network signalling paradigms
NL.3  Switches and packet structure
NL.4  Examples
NL.5  Fast datagram routers
Network Layer Service

Service Models: Signalling Paradigms

• Circuit network service
• Connection-oriented network service
  – virtual circuit
• Connectionless network service
Network Circuits
State Management

- Circuits
  - physical path established
  - circuit state to establish and maintains path

*Examples?*
Network Circuits

State Management

- Circuits
  - physical path established
  - circuit state to establish and maintains path

- Examples
  - early PSTN
  - X.21 circuit switched networks
  - optical WDM lightpaths  *Lecture LL*
Network Circuits
Signalling and Data Transfer

• Circuit signalling

characteristics?
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency

Diagram showing signals and data transfer between nodes S, 1, 2, and R.
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency: RTT *before* data transfer
  - circuit seized *implication?*
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency: RTT before data transfer
  - circuit seized
    - no multiplexing efficiency
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency: RTT before data transfer
  - circuit seized
    - no multiplexing efficiency
Network Circuits
Signalling and Data Transfer

• Circuit signalling characteristics
  – setup latency: RTT *before* data transfer
  – circuit seized
    • no multiplexing efficiency
    • silence unusable by others
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency: RTT before data transfer
  - no multiplexing efficiency
  - negligible switch latency
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency: RTT *before* data transfer
  - no multiplexing efficiency
  + negligible switch latency
  - resources must be released
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency: RTT before data transfer
  - no multiplexing efficiency
  + negligible switch latency
  - resources must be released
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency: RTT before data transfer
  - no multiplexing efficiency
  + negligible switch latency
  - resources must be released
Network Circuits
Signalling and Data Transfer

- Circuit signalling characteristics
  - setup latency: RTT before data transfer
  - no multiplexing efficiency
  + negligible switch latency
  - resources must be released
Network Connections

Motivation

- Combine benefits of datagram and circuits
  - statistical multiplexing gains of datagrams
  - forwarding performance circuits
    - eliminate store-and-forward
    - high-performance switch design
  - provision of QOS 
    - admission control
    - resource reservation per connection

Lecture TQ
Network Connections

State Management

- Connection-oriented
  - connection state *required*
  - performance optimisations possible to reduce setup latency
    - fast reservations
    - optimistic connection establishment

*Examples?*
Network Connections
State Management

- Connection-oriented or *virtual circuit*
  - connection state *required*
  - performance optimisations possible to reduce setup latency
    - fast reservations
    - optimistic connection establishment
- Examples
  - CONS (connection-oriented network service)
  - PSPDNs (packet-switched public data networks)
    - [ISO/IEC 8878+8208 / ITU X.25]
  - ATM and MPLS
  - modern PSTN (wired and wireless)
Network Connections
Virtual-Circuit Signalling

- Connections (virtual circuits) characteristics?
Network Connections
Virtual-Circuit Signalling

- Connections (virtual circuits)
  - establish state once to reduce per packet processing
  + amortised for long flows
  - expensive for transactions
  - RTT delay before data transfer
  + high throughput possible
  + per hop messages reduce latency
Network Connections
Virtual-Circuit Forwarding

- Each packet contains a *connection identifier*
- Each switch does a *lookup* in a *connection table*
  - outgoing port = lookup (connection id)
- Each switch hop does a *label swap*
  - new label = lookup (connection id)
  - prevents the need for global connection id allocation
- Table lookup very efficient
  - connection id is index into simple table
  - can be done in fast hardware at line rate
Network Connections
Virtual-Circuit Label Swapping

- Connection id is index into table
Network Connections
Virtual-Circuit Label Swapping

- Connection id is index into table
- Table entry gives egress link and next hop id
Network Connections
Virtual-Circuit Label Swapping

- Connection id is index into table
- Table entry gives egress link and next hop id
  - port may be prepended for self-routing fabrics
  - examples: ATM, MPLS
Network Connections
Connection State Establishment

- **SETUP** message arrives at switch
Network Connections
Connection State Establishment

- **SETUP** message arrives at switch
  - connection routing
  - resolve dest. address $a_d$ to output port $p_{out}$ from topo. database
Network Connections
Connection State Establishment

- **SETUP** message arrives at switch
  - connection routing
    - resolve dest. address $a_d$ to output port $P_{out}$ from topo. database
  - check for available resources $r$ in switch and egress link
    - if available temporarily reserve
Network Connections
Connection State Establishment

- **SETUP** message arrives at switch
  - connection routing
    - resolve dest. address $a_d$ to output port $P_{out}$ from topo. database
    - check for available resources $r$ in switch and egress link
      - if available temporarily reserve
    - choose unused connection id $c_{out}$
      - temporarily put output port $P_{out}$ in connection forwarding table
Network Connections
Connection State Establishment

- **SETUP** message arrives at switch
  - connection routing
    - resolve *dest. address* $a_d$ to output port $P_{out}$ from topo. database
  - check for available resources $r$ in switch and egress link
    - if available temporarily reserve
  - choose unused connection id $c_{out}$
    - temporarily put output port $P_{out}$ in connection forwarding table
  - pass **SETUP** to next hop
    - save **SETUP** association ($a_s$, $a_d$, id) to match with **CONNECT**
Network Connections
Connection State Establishment

- **SETUP** message arrives at switch
  - connection routing
    - resolve **dest. address** $a_d$ to output port $P_{out}$ from topo. database
  - check for available resources $r$ in switch and egress link
    - if available temporarily reserve
  - choose unused connection id $c_{out}$
    - temporarily put output port $P_{out}$ in connection forwarding table
  - pass **SETUP** to next hop
    - save **SETUP** association $(a_s, a_d, id)$ to match with **CONNECT**
  - release all state if:
    - corresponding **RELEASE** message received
    - corresponding **CONNECT** not received within timeout interval
Network Connections
Connection State Establishment

- **SETUP** message arrives at switch
  - connection routing
  - check for and temporarily reserve resources
  - choose and insert unused connection id
  - pass **SETUP** to next hop

- **CONNECT** message returned to switch
Network Connections
Connection State Establishment

- **SETUP** message arrives at switch
  - connection routing
  - check for and temporarily reserve resources
  - choose and insert unused connection id
  - pass **SETUP** to next hop
- **CONNECT** message returned to switch
  - commit switch and egress link resources
Network Connections
Connection State Establishment

- **SETUP** message arrives at switch
  - connection routing
  - check for and temporarily reserve resources
  - choose and insert unused connection id
  - pass **SETUP** to next hop

- **CONNECT** message returned to switch
  - commit switch and egress link resources
  - pass **CONNECT** message back through ingress link
Network Connection Forwarding
Label Swapping Example
Network Connection Forwarding
Label Swapping Example
Network Connection Forwarding
Label Swapping Example
Network Connection Forwarding
Label Swapping Example

\[\begin{array}{ccc}
\text{0} & \text{2} \\
\text{5} & \text{2} \\
\text{7} & \text{2} \\
\text{3} & \text{0} \\
\text{1} & \text{2} \\
\end{array}\]
Network Connection Forwarding
Label Swapping Example
Network Connection Forwarding
Label Swapping Example
Network Connection Forwarding

Label Swapping Example

Network Connection Forwarding

Label Swapping Example
Network Connection Forwarding
Label Swapping Example
Network Connection Forwarding
Label Swapping Example

Network Connection Forwarding
Label Swapping Example
Connectionless Network

State Management

- Connectionless
  - no per flow state \textit{required} to forward information
  - but there still is state
    \textit{what?}
Connectionless Network
State Management

• Connectionless
  – no per flow state *required* to forward information
  – but there still is state
    • forwarding tables
  – other state may be used to improve performance
    • per-flow queueing *Lecture TQ*
    • soft state flow identification to improve performance

*Examples?*
Connectionless Network
State Management

• Connectionless
  – no per flow state \textit{required} to forward information
  – but there still is state
    • forwarding tables
  – other state may be used to improve performance
    • per-flow queueing \textit{Lecture TQ}
    • soft state flow identification to improve performance

• Examples
  – Internet IP
  – CLNP (connectionless layer network protocol)
    [ISO/IEC 8473 / ITU X.223]
Connectionless Network
Datagram Forwarding

- Each datagram contains a *destination address*
- Each hop does a *lookup* in a *forwarding table*
  - outgoing port = lookup (destination address)
- Table lookup efficiency depends on:
  - address structure (e.g. class-based IP vs. CIDR)
  - address length
  - tables length (# destinations per switch/router)
- Example: IP
  - note: IP lookup could not be done at line rate in 1980s
    more later
Connectionless Network
Datagram Forwarding

- Connectionless signalling
  characteristics?
Connectionless Network
Datagram Forwarding

- Characteristics
  - + no setup latency
Connectionless Network

Datagram Forwarding

• Connectionless signalling
  + no setup latency
  – store-and-foreword lookup delay
Connectionless Network
Datagram Forwarding

- Connectionless signalling
  + no setup latency
  - store-and-foreword lookup delay
Connectionless Network
Datagram Forwarding

- Connectionless signalling
  + no setup latency
  - store-and-foreword lookup delay
  + multiplexing efficiency
Connectionless Network
Datagram Forwarding

- Connectionless signalling
  - no setup latency
  - store-and-forward lookup delay
  - multiplexing efficiency
- Large messages broken into packets

Diagram showing the flow of packets from source (S) to destination (R) through intermediate nodes 1, 2, 3, and 4.
Connectionless Network  
Datagram Forwarding

- **Connectionless signalling**
  - + no setup latency
  - – store-and-foreword lookup delay
  - + multiplexing efficiency
- **large messages broken into packets**
Connectionless Network
Datagram Forwarding

- Connectionless signalling
  + no setup latency
  - store-and-foreword lookup delay
  + multiplexing efficiency
- large messages broken into packets
  + other flows interleave
Connectionless Network
Datagram Forwarding

- Connectionless signalling
  + no setup latency
  - store-and-forward lookup delay
  + multiplexing efficiency
- large messages broken into packets
  + other flows interleave
Connectionless Network
Datagram Forwarding

- Connectionless signalling
  + no setup latency
  - store-and-foreword lookup delay
  + multiplexing efficiency
- large messages broken into *packets*
  + other flows interleave
Connectionless Network
Datagram Forwarding

- Connectionless signalling
  + no setup latency
  - store-and-foreword lookup delay
  + multiplexing efficiency
- large messages broken into *packets*
  + other flows interleave
Connectionless Network

Datagram Forwarding

- Connectionless signalling
  + no setup latency
    - store-and-foreword lookup delay
  + multiplexing efficiency
- large messages broken into packets
  + other flows interleave
Connectionless Network
Datagram Forwarding

- Connectionless signalling
  + no setup latency
  - store-and-forward lookup delay
  + multiplexing efficiency
- large messages broken into *packets*
  + other flows interleave
- $d_r > d_t$
Network-Layer Service
Comparison of Signalling Paradigms

- Connectionless vs. connection-oriented networks
  - major debate in 1980s and 1990s
- Comparison of characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Connectionless</th>
<th>Connection-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup latency</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Forwarding latency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forwarding information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switch state</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resilience to failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QOS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Network-Layer Service
Comparison of Signalling Paradigms

• Connectionless vs. connection-oriented networks
  – major debate in 1980s and 1990s

• Comparison of characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Connectionless</th>
<th>Connection-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup latency</td>
<td>↓ none</td>
<td>↑ round trip</td>
</tr>
<tr>
<td>Forwarding latency</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Forwarding information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switch state</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resilience to failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QOS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Network-Layer Service
Comparison of Signalling Paradigms

- Connectionless vs. connection-oriented networks
  - major debate in 1980s and 1990s
- Comparison of characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Connectionless</th>
<th>Connection-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup latency</td>
<td>↓ none</td>
<td>↑ round trip</td>
</tr>
<tr>
<td>Forwarding latency</td>
<td>↑ address lookup</td>
<td>↓ VC index</td>
</tr>
<tr>
<td>Forwarding information</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Switch state</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resilience to failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QoS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Network-Layer Service
Comparison of Signalling Paradigms

- Connectionless vs. connection-oriented networks
  - major debate in 1980s and 1990s
- Comparison of characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Connectionless</th>
<th>Connection-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup latency</td>
<td>↓ none</td>
<td>↑ round trip</td>
</tr>
<tr>
<td>Forwarding latency</td>
<td>↑ address lookup</td>
<td>↓ VC index</td>
</tr>
<tr>
<td>Forwarding information</td>
<td>↑ address per packet</td>
<td>↓ VC id per packet</td>
</tr>
<tr>
<td>Switch state</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Resilience to failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QOS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Network-Layer Service

#### Comparison of Signalling Paradigms

- **Connectionless vs. connection-oriented networks**
  - major debate in 1980s and 1990s

- **Comparison of characteristics**

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Connectionless</th>
<th>Connection-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup latency</td>
<td>↓ none</td>
<td>↑ round trip</td>
</tr>
<tr>
<td>Forwarding latency</td>
<td>↑ address lookup</td>
<td>↓ VC index</td>
</tr>
<tr>
<td>Forwarding information</td>
<td>↑ address per packet</td>
<td>↓ VC id per packet</td>
</tr>
<tr>
<td>Switch state</td>
<td>↑ forwarding tables</td>
<td>↓ connection id tables</td>
</tr>
<tr>
<td>Resilience to failure</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>QOS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Network-Layer Service
Comparison of Signalling Paradigms

- Connectionless vs. connection-oriented networks
  - major debate in 1980s and 1990s
- Comparison of characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Connectionless</th>
<th>Connection-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup latency</td>
<td>↓ none</td>
<td>↑ round trip</td>
</tr>
<tr>
<td>Forwarding latency</td>
<td>↑ address lookup</td>
<td>↓ VC index</td>
</tr>
<tr>
<td>Forwarding information</td>
<td>↑ address per packet</td>
<td>↓ VC id per packet</td>
</tr>
<tr>
<td>Switch state</td>
<td>↑ forwarding tables</td>
<td>↓ connection id tables</td>
</tr>
<tr>
<td>Resilience to failure</td>
<td>datagrams lost</td>
<td>connections terminated</td>
</tr>
<tr>
<td>QOS</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Network-Layer Service
Comparison of Signalling Paradigms

- Connectionless vs. connection-oriented networks
  - major debate in 1980s and 1990s
- Comparison of characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Connectionless</th>
<th>Connection-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup latency</td>
<td>↓ none</td>
<td>↑ round trip</td>
</tr>
<tr>
<td>Forwarding latency</td>
<td>↑ address lookup</td>
<td>↓ VC index</td>
</tr>
<tr>
<td>Forwarding information</td>
<td>↑ address per packet</td>
<td>↓ VC id per packet</td>
</tr>
<tr>
<td>Switch state</td>
<td>↑ forwarding tables</td>
<td>↓ connection id tables</td>
</tr>
<tr>
<td>Resilience to failure</td>
<td>datagrams lost</td>
<td>connections terminated</td>
</tr>
<tr>
<td>QoS</td>
<td>difficult</td>
<td>connection reservation</td>
</tr>
</tbody>
</table>
Network Layer

NL.3 Switches and Packet Structure

NL.1 Network layer functions and services
NL.2 Network signalling paradigms
NL.3 Switches and packet structure
NL.4 Examples
NL.6 Fast datagram routers
Switches
Overview

• **Switch**
  – intermediate system
  – switches packets from ingress to egress port
  – in Internet has become synonymous with non-IP switch

• **Router**
  – switch that operates on IP datagrams

• **We will use the term generically for now**
  – any network type (Internet, PSTN)
  – any technology
Switches
Functions: Overview

- **Routing / signalling**
  - per flow or longer
- **Transfer control**
  - per packet control
- **Data manipulation**
  - per byte or packet

![Diagram showing switch functions: routing and signalling, transfer control, data manipulation, input processing, link layer decapsulation, output processing, link layer framing, packet buffers, and link layer fabric.](image)
Switches

Functions: Routing and Signalling

- Routing / signalling
  - per flow or longer
- Management
- Signalling
- Topology database
- Routing algorithm
- Traffic management
Switches
Functions: Transfer Control

- **Transfer control**
  - per packet control
- **Input control**
  - classification tables
  - forwarding tables
- **Congestion control**
- **Fabric control**
  - e.g. set crosspoints
- **Output control**
  - link scheduling
**Switches**

**Functions: Data Manipulation**

- **Data manipulation**
  - per byte or packet
  - performance critical
- **Input processing**
  - link layer
  - packet classification
  - output lookup
- **Switch fabric**
- **Packet buffers**
- **Output processing**
  - packet scheduling
  - link layer
Store-and-Forward Switches
Second Generation: NSFNET Routers

- Second generation (1980s)
  - network interfaces share bus
  - general purpose CPU

*problems?
Store-and-Forward Switches
Second Generation: NSFNET Routers

- Second generation (1980s)
  - shared bus interconnect
  - packets traverse bus twice
  - severely limits # of ports
  - DMA transfers help
Store-and-Forward Switches
Second Generation: NSFNET Routers

- Second generation (1980s)
  - shared bus interconnect
    - packets traverse bus twice
    - severely limits # of ports
    - DMA transfers help
  - buffering
    - in general purpose memory
    - contention for memory
  - delays
    - store-and-forward
Store-and-Forward Switches
Second Generation: NSFNET Routers

• Second generation (1980s)
  – shared bus interconnect
    • packets traverse bus twice
    • severely limits # of ports
    • DMA transfers help
  – buffering
    • in general purpose memory
    • contention for memory
  – delays
    • store-and-forward
    • contention for CPU
    • non-trivial header processing

*alternative*?
Store-and-Forward Routers

Third Generation: NSFNET Routers

- Third generation (1990s)
  - NI (network interface)
    - packet processing
    - packet buffers
Store-and-Forward Routers
Third Generation: NSFNET Routers

- Third generation (1990s)
  - NI (network interface)
    - packet processing
    - packet buffers
  - third party bus transfer
    - bus is switch fabric
    - single transfer per packet
      - still significant bottleneck

![Diagram of network layer architecture](image)
Router Architecture
Example: Cisco IOS

- IOS (Internetwork Operating System)
  - OS for Cisco routers (IP switches)
  - proprietary embedded OS
Router Architecture
Example: Cisco IOS Architecture

- Processes
  - routing protocols
  - software forwarding
- Kernel
  - memory management
  - process scheduling
- Packet buffers
- Device drivers
  - network interface drivers
- Fast switching software
  - optimised packet switching
Router Architecture
Open Source Software Based

- Open source software routers
  - primarily intended for network research

- Click Modular Router
  - Morris, et al. (MIT)
  - runs as Linux kernel thread
  - modular elements: classifiers, forwarding, lookup, etc.

- XORP (Extensible Open Router Platform)
  - Handley, et al. (ICSI)
  - Unix-based software router
  - routing protocols run as user-space Unix processes
  - may run over Click
Fast Packet Switching

Motivation

- Allow network switching at line rate
  - 155 Mb/s (OC-3) in mid 1980s
- Eliminate store-and-forward processing bottlenecks
- Eliminate blocking in switch
- Provide support for QOS
- Solution:
  - virtual connection service
Fast Packet Switch Architecture

- Connection state
  - simple per packet processing

- Switch fabric
  - eliminate contention
  - no store-and-forward
Fast Packet Switch
Connectionless vs. Connection Tradeoff

• Connection-oriented fast packet switching
  – requires round trip connection setup latency
  – achieved higher data rate due to simple label swap
  • IP lookup was a bottleneck in 1980s
Packet Size and Structure

Variability

• Fixed vs. variable size packets

*tradeoffs?*
Packet Size and Structure

Variability

- Fixed size (cells)
  - + easier to design switches
  - - difficult to predetermine the best size

- Variable size
  - - more difficult to design switches
  - + no need for agreement on size
  - + less need for fragmentation/segmentation

- Discrete sizes: advantages of both fixed and variable
  - + integral multiples, e.g. 64B, 128B, 192B...
  - + power-of-2 scaling with data rate
    e.g. 128B @ OC-3, 256B @ OC-12, 512B @ OC-48
Packet Size and Structure

Size

- Small vs. large packets
  
  *tradeoffs?*
Packet Size and Structure

Size

• Small packets
  + efficient statistical multiplexing
  – high header/payload overhead
  – short interarrival time challenge per packet processing
    • note: this is one major reason ATM failed

• Large packets
  + significantly easier per packet processing
  – less efficient statistical multiplexing
  – larger queueing delays
  + efficient transport of large data blocks
  – inefficient transport of signalling and control messages
    e.g. TCP ACKs
Packet Size and Structure

Granularity

• Important to match granularity to packet processing

*why?*
Packet Size and Structure

Granularity

- Important to match granularity to packet processing
  - byte / octet (8 bits)
    - control fields should align to 8-bit boundaries
  - word (typically 32 bits)
    - most end-system processing at word granularity
    - payload should align to 32-bit boundaries
  - end system data unit
    - system buffers and memory structures
    - power-of-2 size will likely be integral fraction
  - commodity memory components
    - power-of-2 size
    - note: ATM cell was *none* of these
Packet Size and Structure

Packet Format

- **Header**
  - fields that *determine* packet processing
- **Payload**
  - TPDU transport protocol data unit
- **Trailer**
  - fields that are *dependent* on packet processing
    - e.g. checksum to allow cut-through
Packet Size and Structure
Packet Format: Header

- **Header**
  - fields that *determine* packet processing
    - protocol version
    - packet type: control, data, etc.
    - connection id or protocol demux
    - QoS and authentication if needed
    - header check
      - processing even if data corrupted

- **Payload**

- **Trailer**
Packet Size and Structure
Packet Format: Payload

- Header
- Payload
  - TPDU transport protocol data unit
- Trailer

### Diagram

- Header
  - version
  - type
- Payload
  - connection id
  - QoS
  - authentication
  - payload length
  - check
- Trailer
  - check
Packet Size and Structure

Packet Format: Trailer

- Header
- Payload
- **Trailer**
  - fields that are *dependent* on packet processing
    - checksum to allow cut-through
      - compute and compare

<table>
<thead>
<tr>
<th>version</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>connection id</td>
</tr>
<tr>
<td></td>
<td>QoS</td>
</tr>
<tr>
<td></td>
<td>authentication</td>
</tr>
<tr>
<td></td>
<td>payload length</td>
</tr>
<tr>
<td></td>
<td>check</td>
</tr>
</tbody>
</table>

**TPDU**

check
Packet Size and Structure

Control Fields

- Control field structure and encoding is critical
  - simple encoding (bit vectors vs. code points)
  - byte/octet granularity and alignment
  - field length
    - fixed when possible
    - variable length prepended with length (skip vs. hunt)
Packet Structure
Example: ATM Cells

• ATM cell format
  – fast packet switching
  – fine-grained statistical multiplexing

• Size determined by ITU committee compromise
  – 48B = avg(32, 64)
    • 64 from US = min of proposals for data (and voice)
    • 32 from European PTTs to avoid voice echo cancellers

• Problems:
  – header tiny to keep overhead low (5B); no room for seq #
  – nothing a power of 2
  – 48B + 5B = 53B; not even a multiple of 8
Switch Fabric Architecture

Introduction

- Switch fabric determines input → output
- Critical issues
  - blocking
  - contention
- Designs
  - many choices

Quick overview

depth in EECS 881
Switch Fabric Architecture

Blocking

• Blocking (among different outputs)
• Goal: nonblocking switch fabric
  – input–output path \( i_j \rightarrow o_m \) will not block a different path \( i_k \rightarrow o_n \)
• Some switch designs are mostly nonblocking
  – strictly nonblocking: under all conditions
  – wide-sense nonblocking: if particular algorithm is used
  – rearrangeably nonblocking: if existing paths are rearranged
  – virtually nonblocking: with extremely low probability
Switch Fabric Architecture

Contestation and Buffering

- Contestation (burst collisions) in a non-blocking fabric
  - occurs when traffic destined for *same* output
  - requires buffering even for well-behaved traffic
Switch Fabric Architecture

Contention and Buffering

- Unbuffered fabric
  - optical lightpath switches
  - packet switches with buffered line cards

- Internal buffers
Switch Fabric Architecture

Contestation and Buffering

- Input queueing

*problem?
Switch Fabric Architecture

Contention and Buffering

- Input queueing
  - suffers from *head-of-line blocking*
Switch Fabric Architecture

Contention and Buffering

• Input queueing
  – suffers from *head-of-line blocking*
Switch Fabric Architecture

Contention and Buffering

- Input queueing
  - suffers from *head-of-line blocking*
Switch Fabric Architecture

Contention and Buffering

- Input queueing
  - suffers from *head-of-line blocking*
Switch Fabric Architecture

Contestation and Buffering

- Input queueing
  - suffers from *head-of-line blocking*
Switch Fabric Architecture

Contention and Buffering

- Input queueing
  - suffers from *head-of-line blocking*
Switch Fabric Architecture

Contention and Buffering

- Input queueing
  - suffers from *head-of-line blocking*
Switch Fabric Architecture

Contestation and Buffering

- Input queueing
  - suffers from head-of-line blocking

- Output queueing
  - problem?
Switch Fabric Architecture

Contention and Buffering

- Input queueing
  - suffers from head-of-line blocking

- Output queueing
  - requires either:
    - internal speedup
    - internal expansion
Switch Fabric Architecture

Virtual Output Queueing

- Virtual output queueing
  - parallel buffers
  - non-FIFO buffers
Switch Fabric Architecture

Single Stage: Bus as a Switch

- Simple design shared medium bus
  - point of blocking: only one input active at a time
    - 2nd/3rd generation routers
      - suitable for small switches

- Multicast
  - inherent broadcast
Switch Fabric Architecture

Single Stage: Shared Memory Switch

- Simple design
  - packets written by input
  - packets read by output

- Shared memory
  - point of contention
  - speedup necessary
    - but access times not scaling with Moore’s

- Multicast
  - multiple writes or
  - multicast output demux
Switch Fabric Architecture
Single Stage: Basic 2×2 Switch Element

- **States**
  - point-to-point
    - straight
    - cross
  - multicast
- **Types**
  - buffered or unbuffered
  - self routing or externally controlled
Switch Fabric Architecture

Single Stage: Crossbar Switch

- **Crosspoint switch element**
  - electronic
    - multicast possible
  - optical MEMS
    - rotating mirror
Switch Fabric Architecture
Single Stage: Crossbar Switch

- **Crossbar** fabric
  - square array of crosspoint elements
  - $O(n^2)$ growth complexity
  - reasonable for moderate $n$
Crossbar Switch
Path Selection

- Crossbar fabric
  - simple path routing
  - element \( (o,i) \) turns
Crossbar Switch
Path Selection

- Crossbar fabric
  - simple path routing
  - element \((o, i)\) turns
  - \(i_3 \rightarrow o_4\)
Crossbar Switch
Path Selection

- Crossbar fabric
  - simple path routing
    - element \((o, i)\) turns
    - \(i_3 \rightarrow o_4\)
Crossbar Switch
Path Selection

- Crossbar fabric
  - simple path routing
  - element $(o, i)$ turns
  - $i_3 \rightarrow o_4$
Crossbar Switch
Strictly Nonblocking

- Crossbar fabric
  - simple path routing
    - element \((o, i)\) turns
    - \(i_3 \rightarrow o_4\)
  - strictly nonblocking
    - \(i_j \rightarrow o_n\) noblock \(i_k \rightarrow o_m\)
    - \(\forall j, k, n, m: i \neq j, n \neq m\)
    - \(i_1 \rightarrow o_1\)
Crossbar Switch
Strictly Nonblocking

- Crossbar fabric
  - simple path routing
    - element \((o, i)\) turns
    - \(i_3 \rightarrow o_4\)
  - strictly nonblocking
    - \(i_j \rightarrow o_n\) noblock \(i_k \rightarrow o_m\)
      \(\forall j, k, n, m: i \neq j, n \neq m\)
    - \(i_1 \rightarrow o_1\)
Crossbar Switch
Strictly Nonblocking

- Crossbar fabric
  - simple path routing
    - element \((o,i)\) turns
    - \(i_3 \rightarrow o_4\)
  - strictly nonblocking
    - \(i_j \rightarrow o_n\) noblock \(i_k \rightarrow o_m\)
    - \(\forall j,k,n,m: i \neq j, n \neq m\)
    - \(i_1 \rightarrow o_1\)
Switch Fabrics
Multistage Switches

- Large switches built from single stage elements
  - 2×2 elements or $n \times n$ crossbars
  - $O(n \log n)$ growth complexity
Multistage Switch Fabrics
Delta Fabric Construction and Scalability

- Delta fabric
  - $O(n \log n)$
    - $n/2$ rows
    - $\log_2 n$ stages
    - $n = 2$
    - $2/2 \log_2 2 = 1$
Multistage Switch Fabrics

Delta Fabric Construction and Scalability

- Delta fabric
  - $O(n \log n)$
    - $n/2$ rows
    - $\log_2 n$ stages
    - $n = 4$
    - $4/2 \log_2 4 = 4$
Multistage Switch Fabrics
Delta Fabric Construction and Scalability

- Delta fabric
  - $O(n \log n)$
  - $n/2$ rows
  - $\log_2 n$ stages
  - $n = 8$
  - $8/2 \log_2 8 = 12$
Multistage Switch Fabrics
Delta Fabric Construction and Scalability

- Delta fabric
  - $O(n \log n)$
    - $n/2$ rows
    - $\log_2 n$ stages
    - $n = 16$
    - $16/2 \log_2 16 = 32$
- **Delta fabric**
  - *self-routing*
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
  - $i_2 \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - \(i^{th}\) bit of \(p_{\text{out}}\) used to make routing decision in \(i^{th}\) stage
  - \(i_2 \rightarrow o_{10}\)
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
  - $i_2 \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
  - $i_2 \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - \(i^{th}\) bit of \(p_{\text{out}}\) used to make routing decision in \(i^{th}\) stage
  - \(i_2 \rightarrow o_{10}\)
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
  - $i_2 \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
  - $i_2 \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
  - $i_2 \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i$th bit of $p_{out}$ used to make routing decision in $i$th stage
  - $i_2 \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- **Delta fabric**
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
  - $i_2 \rightarrow o_{10}$
Multistage Switch Fabrics

Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
    - $i_2 \rightarrow o_{10}$
    - $i_{13} \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
    - $i_2 \rightarrow o_{10}$
    - $i_{13} \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
    - $i_2 \rightarrow o_{10}$
    - $i_{13} \rightarrow o_{10}$

![Diagram of Delta Fabric Construction Self-Routing](image)
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^\text{th}$ bit of $p_{\text{out}}$ used to make routing decision in $i^\text{th}$ stage
  - $i_2 \rightarrow o_{10}$
  - $i_{13} \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
  - $i_2 \rightarrow o_{10}$
  - $i_{13} \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
    - $i_2 \rightarrow o_{10}$
    - $i_{13} \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
    - $i_2 \rightarrow o_{10}$
    - $i_{13} \rightarrow o_{10}$
Multistage Switch Fabrics
Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - $i^{th}$ bit of $p_{out}$ used to make routing decision in $i^{th}$ stage
    - $i_2 \to o_{10}$
    - $i_{13} \to o_{10}$
Multistage Switch Fabrics

Delta Fabric Construction Self-Routing

- Delta fabric
  - self-routing
  - \( i^{th} \) bit of \( p_{\text{out}} \) used to make routing decision in \( i^{th} \) stage
  - \( i_2 \rightarrow o_{10} \)
  - \( i_{13} \rightarrow o_{10} \)

\[ \begin{align*}
  &\begin{array}{cccccccccccccccc}
    i_0 & i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 & i_8 & i_9 & i_{10} & i_{11} & i_{12} & i_{13} & i_{14} & i_{15} \\
    o_0 & o_1 & o_2 & o_3 & o_4 & o_5 & o_6 & o_7 & o_8 & o_9 & o_{10} & o_{11} & o_{12} & o_{13} & o_{14} & o_{15} \\
  \end{array}
  \\
  & \begin{array}{cccccccc}
    s_0 & s_1 & s_2 & s_3 \\
    1010 & 1010 & 1010 & 1010 \\
  \end{array}
\]
Network Layer

NL.4  Examples

NL.1  Network layer functions and services
NL.2  Network signalling paradigms
NL.3  Switches and packet structure
NL.4  Examples
   NL.4.1  PSTN
   NL.4.2  X.25 CONS
   NL.4.3  ISDN and Frame Relay
   NL.4.4  B-ISDN and ATM
   NL.4.5  MPLS
   NL.4.6  IP, ICMP, and IPv6
NL.5  Fast datagram routers
Network Layer

NL.4 Examples: PSTN

NL.1 Network layer functions and services
NL.2 Network signalling paradigms
NL.3 Switches and packet structure
NL.4 Examples
  NL.4.1 PSTN
  NL.4.2 X.25 CONS
  NL.4.3 ISDN and Frame Relay
  NL.4.4 B-ISDN and ATM
  NL.4.5 MPLS
  NL.4.6 IP, ICMP, and IPv6

NL.5 Fast datagram routers
PSTN
Functions and Protocols

- Addressing
- Forwarding
- Routing
- Signalling
- Traffic Management
PSTN
Functions and Protocols: Addressing

- **Addressing**: telephone number [ITU E.164]
- Forwarding
- Routing
- Signalling
- Traffic Management
PSTN
Functions and Protocols: Forwarding

- **Addressing**: E.164

- **Forwarding**:
  - traditional circuit switch: physical connection
  - modern virtual circuit
    - TDM mux/demux
    - ATM label-swap switching
  - emerging: VoIP using IP forwarding

- **Routing**

- **Signalling**

- **Traffic Management**
PSTN
Functions and Protocols: Routing

- Addressing: E.164
- Forwarding: telephone switch
- **Routing**: Lecture NR
  - traditional: HIER
    - static hierarchy based on telephone number
  - network engineering to provide required service
    - blocking probability
  - modern: dynamic routing (DNHR, RTNR)

- Signalling
- Traffic Management
PSTN
Functions and Protocols: Signalling

- Addressing: IP
- Forwarding: IP
- Routing: HIER, DNHR, RTNR, ...
- Signalling:
  - traditional: in-band audio
  - modern: out-of-band (common channel signalling) SS7
- Traffic Management
PSTN

Functions and Protocols: Traffic Management

- Addressing: IP
- Forwarding: IP
- Routing: HIER, DNHR, RTNR, ...
- Signalling: SS7
- Traffic management:
  - network engineering to provide required service
    - blocking probability
  - modern: ATM and MPLS traffic engineering
PSTN Addressing

History

• Each telco devised its own numbering plan
  – US small towns frequently had 4- or 5-digits
• Bell System standardised on 7 digits
  – 3 digit exchange represented as exchange name + number
    • intended to make numbers easier to remember
    • std. names: http://www.ourwebhome.com/TENP/Recommended.html
  – 4 digit subscriber line id
  – example evolution in Lawrence
    • UNiversity 4-7890
    • UN 4-7890
    • 864-7890
PSTN Addressing
Addressing Notation

- **Notation** [ITU E.123] symbol & icons [ITU E.121]
  - country-specific international access denoted by +
  - followed by grouped digits (no hyphens, dots)
    - grouping based on each country's numbering plan, e.g.
      +1 785 864 7890
      +1 508 944 3067
      +44 1524 510302
      +41 44 632 70 01
PSTN Addressing
Address Format

- Telephone number format (≤ 15 digits) [ITU E.164]
  \langle country-code \rangle \langle national-destination-code \rangle \langle subscriber-number \rangle
- country code (1–3 digits) assigned by ITU
  *List of ITU-T Recommendation E.164 Assigned Country Codes*
- NDC: national destination code (city code or area code)
  - may be fixed length (e.g. US) or variable length (e.g. UK)
  - may be structured (e.g.) Germany or unstructured (e.g. US)
- SN: subscriber number
  - may be structured as in US
  \langle subscriber-number \rangle = \langle central-office-exch. \rangle \langle subscriber-line-id \rangle
PSTN: Addressing
Global Addressing

- Bell System IDDD
  - international direct distance dialing
- ITU Country codes
  - early numbering in 1960 ITU Red Book
  - current scheme defined 1963 ITU Blue Book
- Grouped into 10 zones
- 1 – 3 digits
  - variable length code

<table>
<thead>
<tr>
<th>zone</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>US, Canada, Caribbean</td>
</tr>
<tr>
<td>2</td>
<td>Africa</td>
</tr>
<tr>
<td>3</td>
<td>Europe</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Mexico, Central and S. America</td>
</tr>
<tr>
<td>6</td>
<td>South Pacific</td>
</tr>
<tr>
<td>7</td>
<td>Russia (former Soviet Union)</td>
</tr>
<tr>
<td>8</td>
<td>East Asia and special services</td>
</tr>
<tr>
<td>9</td>
<td>West and South Asia</td>
</tr>
<tr>
<td>0</td>
<td>spare</td>
</tr>
</tbody>
</table>
# PSTN: Addressing Zone 1 Country Codes (NANP)

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1NXX</td>
<td>.us</td>
<td>United States</td>
<td>United States</td>
</tr>
<tr>
<td>+1NXX</td>
<td>.ca</td>
<td>Canada</td>
<td>Canada</td>
</tr>
<tr>
<td>+1706</td>
<td>.mx</td>
<td>Mexico</td>
<td>NW Mexico</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mexico City</td>
</tr>
<tr>
<td>+1NXX</td>
<td></td>
<td>(Caribbean Nations)</td>
<td>now +52 part of Mexico was accessible in NANP before 1991</td>
</tr>
</tbody>
</table>
# PSTN: Addressing

## Zone 2 Country Codes (2-Digit)

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+20</td>
<td>.eg</td>
<td>مصر (Misr)</td>
<td>Egypt</td>
</tr>
<tr>
<td>+21X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+22X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+23X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+24X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+25X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+26X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+27</td>
<td>.za</td>
<td>iNingizimu Afrika Suid-Afrika South Africa</td>
<td>South Africa</td>
</tr>
<tr>
<td>+28X</td>
<td></td>
<td></td>
<td>unassigned</td>
</tr>
<tr>
<td>+29X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
</tbody>
</table>
## PSTN: Addressing Zone 3 Country Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+30</td>
<td>.gr</td>
<td>Ελλάς</td>
<td>Greece</td>
</tr>
<tr>
<td>+31</td>
<td>.nl</td>
<td>Nederland</td>
<td>Netherlands</td>
</tr>
<tr>
<td>+32</td>
<td>.be</td>
<td>België Belgique Belgien</td>
<td>Belgium</td>
</tr>
<tr>
<td>+33</td>
<td>.fr</td>
<td>France</td>
<td>France</td>
</tr>
<tr>
<td>+34</td>
<td>.es</td>
<td>España Spain Españaia Espanha</td>
<td>Spain</td>
</tr>
<tr>
<td>+36</td>
<td>.hu</td>
<td>Magyarország</td>
<td>Hungary</td>
</tr>
<tr>
<td>+37</td>
<td>(dd)</td>
<td>Deutsche Dem. Rep.</td>
<td>East Germany now 3-digit codes</td>
</tr>
<tr>
<td>+38</td>
<td>.yu</td>
<td>Jugoslavija / Југославија</td>
<td>Yugoslavia now 3-digit codes</td>
</tr>
<tr>
<td>+39</td>
<td>.it</td>
<td>Italia</td>
<td>Italy</td>
</tr>
</tbody>
</table>
## PSTN: Addressing

### Zone 3 Country Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+350</td>
<td>.gi</td>
<td>Gibraltar</td>
<td>Gibraltar</td>
</tr>
<tr>
<td>+351</td>
<td>.pt</td>
<td>Portugal</td>
<td>Portugal</td>
</tr>
<tr>
<td>+352</td>
<td>.lu</td>
<td>Lëtzebuerg Luxembourg Luxembourg</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>+353</td>
<td>.ie</td>
<td>Éire / Ireland</td>
<td>Ireland</td>
</tr>
<tr>
<td>+354</td>
<td>.is</td>
<td>Ísland</td>
<td>Iceland</td>
</tr>
<tr>
<td>+355</td>
<td>.al</td>
<td>Shqipëria</td>
<td>Albania</td>
</tr>
<tr>
<td>+356</td>
<td>.mt</td>
<td>Malta</td>
<td>Malta</td>
</tr>
<tr>
<td>+357</td>
<td>.cy</td>
<td>Κύπρος Kýbris</td>
<td>Cyprus</td>
</tr>
<tr>
<td>+358</td>
<td>.fi</td>
<td>Suomi Finlands</td>
<td>Finland</td>
</tr>
<tr>
<td>+359</td>
<td>.bg .bg</td>
<td>България</td>
<td>Bulgaria</td>
</tr>
</tbody>
</table>
# PSTN: Addressing

## Zone 3 Country Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+370</td>
<td>.lt</td>
<td>Lietuva</td>
<td>Lithuania</td>
</tr>
<tr>
<td>+371</td>
<td>.lv</td>
<td>Latvija</td>
<td>Latvia</td>
</tr>
<tr>
<td>+372</td>
<td>.ee</td>
<td>Eesti</td>
<td>Estonia</td>
</tr>
<tr>
<td>+373</td>
<td>.md</td>
<td>Moldova</td>
<td>Moldova</td>
</tr>
<tr>
<td>+374</td>
<td>.am</td>
<td>Հայաստան (Hayastan)</td>
<td>Armenia</td>
</tr>
<tr>
<td>+375</td>
<td>.by</td>
<td>Беларусь Бieлaruś</td>
<td>Belarus</td>
</tr>
<tr>
<td>+376</td>
<td>.ad</td>
<td>Andorra</td>
<td>Andorra</td>
</tr>
<tr>
<td>+377</td>
<td>.mc</td>
<td>Monaco Munegu</td>
<td>Monaco</td>
</tr>
<tr>
<td>+378</td>
<td>.sm</td>
<td>San Marino</td>
<td>San Marino</td>
</tr>
<tr>
<td>+379</td>
<td>.va</td>
<td>Civitatis Vaticanæ</td>
<td>Vatican City  actually uses +39</td>
</tr>
</tbody>
</table>
### PSTN: Addressing

#### Zone 3 Country Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+380</td>
<td>.ua</td>
<td>Україна</td>
<td>Ukraine</td>
</tr>
<tr>
<td>+381</td>
<td>.rs</td>
<td>Србија и Црна Гора Republika Srbija</td>
<td>Republic of Serbia formerly .yu .cs never used</td>
</tr>
<tr>
<td>+381</td>
<td></td>
<td>Republika e Kosovës Република Косово</td>
<td>Republic of Kosovo</td>
</tr>
<tr>
<td>+382</td>
<td>.me</td>
<td>Република Црна Гора Republika Crna Gora</td>
<td>Republic of Montenegro formerly .yu</td>
</tr>
<tr>
<td>+383</td>
<td></td>
<td></td>
<td>unassigned</td>
</tr>
<tr>
<td>+384</td>
<td></td>
<td></td>
<td>unassigned</td>
</tr>
<tr>
<td>+385</td>
<td>.hr</td>
<td>Hrvatska</td>
<td>Croatia</td>
</tr>
<tr>
<td>+386</td>
<td>.si</td>
<td>Slovenija</td>
<td>Slovenia</td>
</tr>
<tr>
<td>+387</td>
<td>.ba</td>
<td>Bosna i Hercegovina Босна и Херцеговина</td>
<td>Bosnia &amp; Herzegovina</td>
</tr>
<tr>
<td>+388</td>
<td>.eu</td>
<td></td>
<td>Europe European telephony</td>
</tr>
<tr>
<td>+389</td>
<td>.mk</td>
<td>Македонија</td>
<td>Macedonia</td>
</tr>
</tbody>
</table>
## PSTN: Addressing
### Zone 4 Country Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+40</td>
<td>.ro</td>
<td>Romania</td>
<td>Romania</td>
</tr>
<tr>
<td>+41</td>
<td>.ch</td>
<td>Schweiz / Suisse Svizzera / Svizra</td>
<td>Switzerland</td>
</tr>
<tr>
<td>+42</td>
<td>.cs</td>
<td>Československo</td>
<td>Czechoslovakia before 1998</td>
</tr>
<tr>
<td>+420</td>
<td>.cz</td>
<td>Česká Republika</td>
<td>Czech Republic was +42</td>
</tr>
<tr>
<td>+421</td>
<td>.sk</td>
<td>Slovensko</td>
<td>Slovakia was +42</td>
</tr>
<tr>
<td>+423</td>
<td>.li</td>
<td>Liechtenstein</td>
<td>Liechtenstein was +41 75</td>
</tr>
<tr>
<td>+43</td>
<td>.at</td>
<td>Österreich</td>
<td>Austria</td>
</tr>
<tr>
<td>+44</td>
<td>.uk</td>
<td>United Kingdom</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>+45</td>
<td>.dk</td>
<td>Danmark</td>
<td>Denmark</td>
</tr>
<tr>
<td>+46</td>
<td>.se</td>
<td>Sverige</td>
<td>Sweden</td>
</tr>
<tr>
<td>+47</td>
<td>.no</td>
<td>Norge</td>
<td>Norway</td>
</tr>
<tr>
<td>+48</td>
<td>.pl</td>
<td>Polska</td>
<td>Poland</td>
</tr>
<tr>
<td>+49</td>
<td>.de</td>
<td>Deutschland</td>
<td>Germany</td>
</tr>
</tbody>
</table>
# PSTN: Addressing
## Zone 5 Country Codes (2-Digit)

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+50X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+51</td>
<td>.pe</td>
<td>Perú Piruw</td>
<td>Peru</td>
</tr>
<tr>
<td>+52</td>
<td>.mx</td>
<td>México</td>
<td>Mexico</td>
</tr>
<tr>
<td>+53</td>
<td>.cu</td>
<td>Cuba</td>
<td>Cuba</td>
</tr>
<tr>
<td>+54</td>
<td>.ar</td>
<td>Argentina</td>
<td>Argentina</td>
</tr>
<tr>
<td>+55</td>
<td>.br</td>
<td>Brasil</td>
<td>Brazil</td>
</tr>
<tr>
<td>+56</td>
<td>.cl</td>
<td>Chile</td>
<td>Chile</td>
</tr>
<tr>
<td>+57</td>
<td>.co</td>
<td>Colombia</td>
<td>Colombia</td>
</tr>
<tr>
<td>+58</td>
<td>.ve</td>
<td>Venezuela</td>
<td>Venezuela</td>
</tr>
<tr>
<td>+59X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PSTN: Addressing
### Zone 6 Country Codes (2-Digit)

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+60</td>
<td>.my</td>
<td>Malaysia</td>
<td>Malaysia</td>
</tr>
<tr>
<td>+61</td>
<td>.au</td>
<td>Australia</td>
<td>Australia</td>
</tr>
<tr>
<td>+62</td>
<td>.id</td>
<td>Indonesia</td>
<td>Indonesia</td>
</tr>
<tr>
<td>+63</td>
<td>.ph</td>
<td>Pilipinas</td>
<td>Philippines</td>
</tr>
<tr>
<td>+64</td>
<td>.nz</td>
<td>New Zealand</td>
<td>New Zealand</td>
</tr>
<tr>
<td>+65</td>
<td>.sg</td>
<td>Singapura</td>
<td>Singapore</td>
</tr>
<tr>
<td>+66</td>
<td>.th</td>
<td>ไทย (Thai)</td>
<td>Thailand</td>
</tr>
<tr>
<td>+67X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+68X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+69X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
</tbody>
</table>
# PSTN: Addressing Zone 7 Country Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+7</td>
<td>.ru</td>
<td>России́я</td>
<td>Russia</td>
</tr>
<tr>
<td></td>
<td>.ро́ф</td>
<td></td>
<td>formerly USSR .su</td>
</tr>
<tr>
<td></td>
<td>.су</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+7</td>
<td>.кz</td>
<td>Казакстán</td>
<td>Kazakhstan</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>other former Soviet republics now in</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+37X Baltic states and Belarus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+38X Ukraine</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+99X Asian states</td>
</tr>
</tbody>
</table>
# PSTN: Addressing Zone 8 Country Codes (2-Digit)

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+81</td>
<td>.jp</td>
<td>日本 (Nihon)</td>
<td>Japan</td>
</tr>
<tr>
<td>+82</td>
<td>.kr</td>
<td>대한민국</td>
<td>大韓民國 (Daehan Minguk)</td>
</tr>
<tr>
<td>+84</td>
<td>.vn</td>
<td>Việt Nam</td>
<td>Vietnam</td>
</tr>
<tr>
<td>+86</td>
<td>.cn</td>
<td>中国 / 中國 (Zhōngguó)</td>
<td>China</td>
</tr>
<tr>
<td>+89X</td>
<td></td>
<td></td>
<td>unasigned</td>
</tr>
</tbody>
</table>
### PSTN: Addressing
#### Zone 8 Country Codes (3-Digit)

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+850</td>
<td>.kp</td>
<td>조선민주주의인민공화국 (Chosŏn Minjujuŭi Inmin Konghwaguk)</td>
<td>North Korea</td>
</tr>
<tr>
<td>+852</td>
<td>.hk</td>
<td>香港 (Hèung Góng</td>
<td>Xiānggǎng)</td>
</tr>
<tr>
<td>+853</td>
<td>.mo</td>
<td>澳門 / Macau (Jyutping / Àomén)</td>
<td>Macao</td>
</tr>
<tr>
<td>+855</td>
<td>.kh</td>
<td>កមពជា (Kâmpŭchea)</td>
<td>Cambodia</td>
</tr>
<tr>
<td>+856</td>
<td>.la</td>
<td>ນລາວ (Muang Lao)</td>
<td>Laos</td>
</tr>
<tr>
<td>+880</td>
<td>.bd</td>
<td>বাংলাদেশ</td>
<td>Bangladesh</td>
</tr>
<tr>
<td>+886</td>
<td>.tw</td>
<td>中華民國 (JhōngHuá MínGuó)</td>
<td>Taiwan ROC</td>
</tr>
</tbody>
</table>

© James P.G. Sterbenz
## PSTN: Addressing Zone 8 Service Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Service</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+800</td>
<td></td>
<td>international free phone</td>
<td>Internat. free phone</td>
</tr>
<tr>
<td>+83</td>
<td></td>
<td></td>
<td>unassigned</td>
</tr>
<tr>
<td>+87</td>
<td></td>
<td>Inmarsat</td>
<td>+870 – +874</td>
</tr>
<tr>
<td>+87</td>
<td></td>
<td>maritime mobile</td>
<td>+875 – +877</td>
</tr>
<tr>
<td>+878</td>
<td></td>
<td>universal personal telecommunications</td>
<td></td>
</tr>
<tr>
<td>+881</td>
<td></td>
<td>global mobile satellite system</td>
<td></td>
</tr>
<tr>
<td>+882</td>
<td>.int</td>
<td>international networks</td>
<td></td>
</tr>
<tr>
<td>+883</td>
<td>.int</td>
<td>international networks</td>
<td></td>
</tr>
<tr>
<td>+888</td>
<td>.un.org</td>
<td></td>
<td>UN OCHA disaster relief</td>
</tr>
<tr>
<td>+89</td>
<td></td>
<td></td>
<td>unassigned</td>
</tr>
</tbody>
</table>
# PSTN: Addressing

## Zone 9 Country Codes (2-Digit)

<table>
<thead>
<tr>
<th>Code</th>
<th>TLD</th>
<th>Country</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>+90</td>
<td>.tr</td>
<td>Türkiye</td>
<td>Turkey</td>
</tr>
<tr>
<td>+91</td>
<td>.in</td>
<td>भारत (Bharat)</td>
<td>India</td>
</tr>
<tr>
<td>+92</td>
<td>.pk</td>
<td>اسلامی Pakistan</td>
<td>Pakistan</td>
</tr>
<tr>
<td>+93</td>
<td>.af</td>
<td>افغانستان</td>
<td>Afghanistan</td>
</tr>
<tr>
<td>+94</td>
<td>.lk</td>
<td>දේශයාත්මක උප්ප්‍රදේශය</td>
<td>Sri Lanka</td>
</tr>
<tr>
<td>+95</td>
<td>.mm</td>
<td>မြန်းပို့</td>
<td>Myanmar (Burma)</td>
</tr>
<tr>
<td>+96X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+97X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+98</td>
<td>.ir</td>
<td>ایران</td>
<td>Iran</td>
</tr>
<tr>
<td>+99X</td>
<td></td>
<td></td>
<td>3-digit codes</td>
</tr>
<tr>
<td>+999</td>
<td></td>
<td></td>
<td>reserved</td>
</tr>
</tbody>
</table>
PSTN Addressing

NANP (WZ1) Origins

• Long distance originally required operator assistance
• NANP (North American numbering plan) in 1947
  – DDD (direct distance dialing) began in 1951
PSTN Addressing
NANP (WZ1) Administration

- NANPA (NANP Administrator) www.nanpa.org
- CNA (Canadian Number Administrator) www.cnac.ca
- Regulation by FCC in US
  - NANC (North American Numbering Council)
    www.fcc.gov/wcb/tapd/Nanc
- Guidance from
  - ATIS INC (Industry Numbering Committee)
    www.atis.org/inc/docs.asp
  - CISC CSCN (Canadian Steering Committee on Numbering)
    www.crtc.gc.ca/cisc/eng/cisf3f.htm
PSTN Addressing
NANP (WZ1) Traditional Hierarchical Structure

final trunk group
(to class 4 long distance switching)  913

+1

tandem trunk group

tandem office

tandem trunk group

local (class 5) switches

direct trunk group

tandem office

local office 897
direct trunk group

local loops

direct trunk group

8538
PSTN Addressing
NANP (WZ1) Nomenclature and Format

- Nomenclature: \( N = \{2 \ldots 9\} \); \( X = \{0 \ldots 9\} \); \( 0/1 = \{0|1\} \)
- 1947: 86 NPAs (numbering plan areas or area codes)
  - entire state codes of form \( N0X \)
  - split state codes of form \( N1X \)
  - high-population codes generally low \( N/X \)

*why?*
PSTN Addressing
NANP (WZ1) Nomenclature and Format

• Nomenclature: \( N = \{2...9\}; X = \{0...9\}; ^0/1 = \{0|1\} \)
  – entire state codes of form \( N0X \)
  – split state codes of form \( N1X \)
  – high-population codes generally low \( N/X \)
    • reduce pulse delay
    • reduce dial pullback
PSTN Addressing

NANP (WZ1) Nomenclature and Format

- Nomenclature: \( N = \{2\ldots9\}; \ X = \{0\ldots9\}; \ \theta/1 = \{0|1\} \)
  - entire state codes of form \( N\theta X \)
  - split state codes of form \( N1X \)
  - high-population codes generally low \( N/X \)
    - reduce pulse delay
    - reduce dial pullback

- NANP is ITU E.164 compliant
  \( \langle \text{national-destination-code} \rangle = \text{NPA (area code)} \)
  \( \langle \text{subscriber number} \rangle = \langle \text{central-office-exch.} \rangle \langle \text{sub.-line-id} \rangle \)
  - e.g. +1 913 897 8538
    US KC OP SLID(KU)
PSTN: Addressing
NANP (WZ1) 1947: Original Bell System

- US and Canada
  - N0X dedicated
  - N1X split state
- 86 NPAs assigned
  - 152 N0/1X poss.
  - N00 / N11 reserv.
PSTN: Addressing
NANP Address Capacity

- **Address space fields**
  - each CO code has 10,000 SLIDs (subscriber line ID)
  - each NPA can have 640 NNX CO codes
  - 152 N\(^0/1\) X NPAs

- **Total address space**
  - \(152 \times 640 \times 10000 = 972,800,000 \approx 10^{10}\)

  *isn’t this plenty?*
PSTN: Addressing
NANP Address Capacity

- Address space fields
  - each CO code has 10,000 SLIDs (subscriber line ID)
  - each NPA can have 640 NNX CO codes
  - 152 N^6/1 X NPAs

- Total address space
  - \(152 \times 640 \times 10,000 = 972,800,000 \approx 10^{10}\)
    - order of magnitude less than \(10^{11}\) possible with 10 digits
    - NPA and CO geography determines distribution
    - usable number much smaller

*How to accommodate growth?*
PSTN: Addressing
NANP Address Capacity

- Options to accommodate growth
  - add SLIDs to approach 10,000 per CO
    - some spares needed for churn
    - reduce redirect period
PSTN: Addressing
NANP Address Capacity

• Options to accommodate growth
  – add SLIDs
  – add CO codes until 640 per NPA
    • adding SLIDs and CO codes is relatively easy
    • add and expand CO switches and trunks
PSTN: Addressing
NANP Address Capacity

• Options to accommodate growth
  – add SLIDs
  – add CO codes until 640 per NPA
  – add NPAs to 152 maximum
    • more difficult
      why?
PSTN: Addressing
NANP NPA Growth and Capacity

- NPA capacity process:
  - NANP forecasts need for new NPAs
  - balancing act
    - too aggressive causes unnecessary number changes
    - too conservative prevents new number assignments
  - jeopardy: demand for new numbers exceed forecasts
    - new CO code assignments are restricted until relief
PSTN: Addressing
NANP NPA Growth and Capacity

• Options for relief
  – NPA split
  – NPA repartition
  – NPA overlay

• Geographic partition required for hierarchical routing
  
  Lecture NR
  
  – requires user numbers to change: disruptive and expensive
PSTN: Addressing

NANP (WZ1): 1960s and 1970s

- US and Canada
- NPAs
  - near capacity
PSTN: Addressing NANP NPA Growth and Capacity

- Many NPAs near capacity in 1970s
  - \( N^{\theta/1} \times \) NPAs nearly exhausted

Solution?
PSTN: Addressing
NANP NPA Growth and Capacity

- Many NPAs near capacity in 1970s
  - $N^0/\_1X$ NPAs nearly exhausted
- Option
  - underused NPA realignment would cause *massive* disruption
PSTN: Addressing
NANP NPA Growth and Capacity

• 1st step solution
  – adjust NANP addressing within current framework
  – institute *interchangeable* NPA/CO codes
  – CO codes NNX → NXX
    • CO address space increases from 640 → 792
    • $152 \times 792 \times 10000 = 1203840000 \approx 10^{10}$ total numbers

*implication?
PSTN: Addressing
NANP NPA Growth and Capacity

• 1st step solution
  – adjust NANP addressing within current framework
  – institute *interchangeable* NPA/CO codes
  – CO codes NNX → NXX
    • address space increases from 640 → 792
    • $152 \times 792 \times 10000 = 1203840000 \approx 10^{10}$ total numbers
  – implication: 10-digit local numbers
    • NPAs no longer distinguishable from CO codes
PSTN: Addressing
NANP (WZ1)

- **Nomenclature:** \( N = \{2 \ldots 9\}; \quad X = \{0 \ldots 9\}; \quad Y = \{0 \ldots 8\} \quad \theta/1 = \{0|1\} \)

<table>
<thead>
<tr>
<th></th>
<th>1947</th>
<th>1974</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CC</strong></td>
<td></td>
<td>( N^+ )</td>
</tr>
<tr>
<td><strong>NPA</strong></td>
<td>( N^\theta/1X \ [152] )</td>
<td>( N^\theta/1X \ [152] )</td>
</tr>
<tr>
<td><strong>Central office</strong></td>
<td>( NNX \ [640] )</td>
<td>( NXX \ [792] )</td>
</tr>
<tr>
<td><strong>SLID</strong></td>
<td>( XXXX \ [10000] )</td>
<td>( XXXX \ [10000] )</td>
</tr>
<tr>
<td><strong>free</strong></td>
<td></td>
<td>( 800 \ [1] )</td>
</tr>
<tr>
<td><strong>IDDD +</strong></td>
<td></td>
<td>( 011 )</td>
</tr>
<tr>
<td><strong>Service code</strong></td>
<td>( N11 \ [8] )</td>
<td>( N11 )</td>
</tr>
<tr>
<td><strong>Toll center</strong></td>
<td>( 0XX ) ( \theta/1XX \ [200] )</td>
<td>( 0XX ) ( \theta/1XX \ [200] )</td>
</tr>
<tr>
<td><strong>System code</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PSTN: Addressing
NANP NPA Growth and Capacity

• 1st step solution
  – adjust NANP addressing within current framework
  – institute *interchangeable* NPA/CO codes
  – CO codes NNX → NXX
  – implication: 10-digit local numbers
  – still insufficient to meet demand explosion in late 1980s
    • fax machines
    • 2nd residential lines for home computer modems
    • mobile telephones

*solution?*
PSTN: Addressing
NANP NPA Growth and Capacity

• 2nd step solution
  – NPA $N^0/X \rightarrow NYX$: NPA space increased from 152 $\rightarrow$ ~900
    • $Y = \{0...8\}$
    • usually NYX is still written NXX with a footnote
  – address space now $\approx 712 \times 792 \times 10000 = 6272640000$
    • note: some NPAs and CO code reserved for special use

issues?
PSTN: Addressing
NANP NPA Growth and Capacity

• Massive growth in 1980s and 1990s
  – ~500 new NYX NPAs
  – but frequent NPA splits very disruptive for users and carriers

Solution?
PSTN: Addressing
NANP NPA Growth and Capacity

• Massive growth in 1980s and 1990s
  – ~500 new NYX NPAs
  – but frequent NPA splits very disruptive for users and carriers

• 3rd step solution
  – remove restriction that NPAs be geographically unique
  – enabled by more flexible switch routing software
    • nonhierarchical routing *Lecture NR*
    – *overlay* NPAs: multiple NPAs shared in a given area
    – initially resisted by some PUCs (public utility commissions)
PSTN: Addressing
NANP (WZ1)

- **Nomenclature:** \( N = \{2...9\}; \ X = \{0...9\}; \ Y = \{0...8\} \ \ \ \ 0/1 = \{0|1\} \)

<table>
<thead>
<tr>
<th></th>
<th>1947</th>
<th>1974</th>
<th>1995</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CC</strong></td>
<td></td>
<td>( N^+ )</td>
<td>( N^+ )</td>
</tr>
<tr>
<td><strong>NPA</strong></td>
<td>( N_0/1 X \ [152] )</td>
<td>( N_0/1 X \ [152] )</td>
<td>( NYX \ [712] )</td>
</tr>
<tr>
<td><strong>Central office</strong></td>
<td>( NNX \ [640] )</td>
<td>( NXX \ [792] )</td>
<td>( NXX \ [792] )</td>
</tr>
<tr>
<td><strong>SLID</strong></td>
<td>( XXXX \ [10000] )</td>
<td>( XXXX \ [10000] )</td>
<td>( XXXX \ [10000] )</td>
</tr>
<tr>
<td><strong>free</strong></td>
<td></td>
<td>( 800 \ [1] )</td>
<td>( 8xx \ [8] )</td>
</tr>
<tr>
<td><strong>IDD+</strong></td>
<td></td>
<td>( 011 )</td>
<td>( 011 )</td>
</tr>
<tr>
<td><strong>Service code</strong></td>
<td>( N11 \ [8] )</td>
<td>( N11 )</td>
<td>( N11 )</td>
</tr>
<tr>
<td><strong>Toll center</strong></td>
<td>( 0XX )</td>
<td>( 0XX )</td>
<td>( 0XX )</td>
</tr>
<tr>
<td><strong>System code</strong></td>
<td>( 0/1 XX \ [200] )</td>
<td>( 0/1 XX \ [200] )</td>
<td>( 0/1 XX \ [200] )</td>
</tr>
</tbody>
</table>
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

- **1947**: original NPAs [original MO boundary approximate]
  - KS: two N1X codes split between Kansas City and Wichita
  - MO: two N1X codes split between St. Louis and Kansas City
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

- **1950**: geographic split
  - MO: Kansas City keeps 816; St. Louis keeps 314
  - MO: southwest replaced with 417 including Springfield
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

• 1995: geographic split
  – MO: St. Louis and suburbs keep 314
  – MO: rest of east replaced by 573
### PSTN: Addressing

**NANP NPA Evolution Example: KS + MO**

- **1996: jeopardy**
  - **MO:** 816 NPA in danger of CO code exhaustion before relief
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

- 1997: geographic split
  - MO: Kansas City suburbs keep 816
  - MO: rest of northwest replaced by 660
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

- 1997: jeopardy
  - KS: 913 NPA in danger of CO code exhaustion before relief
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

• 1997: geographic split
  − KS: Kansas City suburbs keep 913
  − KS: rest of north replaced by 785 including Lawrence
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

- 1998: jeopardy
  - MO: 314 NPA in danger of CO code exhaustion before relief
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

• 1999: geographic split
  – MO: St. Louis city and inner suburbs keeps 314
  – MO: St. Louis suburbs code replaced with 636
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

- 1999: interchangeable codes
  - KS/MO: Kansas City 10 digit local 913/816 dialing
  - note that 913 and 816 are in the same LATA
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

- 2001: geographic split
  - KS: Wichita keeps 316
  - KS: rest of south replaced by 620

KS: Wichita
KS: rest of south
MO: St. Louis
MO: Jefferson City
PSTN: Addressing
NANP NPA Evolution Example: KS + MO

- 2001: first overlays planned
  - 975 for KC (1.1M metro pop.); 557 for St. Louis (2.0M pop.)
  - initial plans suspended but codes still reserved
PSTN: Addressing
NANP NPA Growth and Capacity

• NYX NXX XXXX addresses
  – still in danger of exhaustion within next decade
  – repartition of existing geographical boundaries impractical

• Expansion requires additional digits
  – ITU E.164 allows 15

• Options
  – use reserved N9X for more NPA digits
  – increase number of SLID digits

• Fixed vs. variable length codes
  – much of world used variable length
  – US reluctant to change fixed-length tradition
### PSTN: Addressing NANP (WZ1)

- **Nomenclature**: \( N = \{2\ldots9\} \); \( X = \{0\ldots9\} \); \( Y = \{0\ldots8\} \); \( 0/1 = \{0|1\} \)

<table>
<thead>
<tr>
<th></th>
<th>1947</th>
<th>1974</th>
<th>1995</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NPA</strong></td>
<td>( N^0/1X ) [152]</td>
<td>( N^0/1X ) [152]</td>
<td>( NYX ) [712]</td>
<td>( N9XX ) [7920]</td>
</tr>
<tr>
<td><strong>Central office</strong></td>
<td><strong>NNX</strong> [640]</td>
<td><strong>NXX</strong> [792]</td>
<td><strong>NXX</strong> [792]</td>
<td></td>
</tr>
<tr>
<td><strong>SLID</strong></td>
<td>( XXXX ) [10000]</td>
<td>( XXXX ) [10000]</td>
<td>( XXXX ) [10000]</td>
<td>( XXXX^+ ) [10^n]</td>
</tr>
<tr>
<td><strong>free</strong></td>
<td></td>
<td>( 800 ) [1]</td>
<td>( 8xx ) [8]</td>
<td>?</td>
</tr>
<tr>
<td><strong>IDD +</strong></td>
<td></td>
<td>( 011 )</td>
<td>( 011 )</td>
<td>( 011 )</td>
</tr>
<tr>
<td><strong>Service code</strong></td>
<td>( N11 ) [8]</td>
<td>( N11 )</td>
<td>( N11 )</td>
<td>?</td>
</tr>
<tr>
<td><strong>Toll center</strong></td>
<td>( 0XX )</td>
<td>( 0XX )</td>
<td>( 0XX )</td>
<td></td>
</tr>
<tr>
<td><strong>System code</strong></td>
<td>( 0/1XX ) [200]</td>
<td>( 0/1XX ) [200]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PSTN: Addressing
NANP (WZ1) $^0/_1$ Call Type Codes

- Trunk access and operator codes

<table>
<thead>
<tr>
<th>$^0/_1X$</th>
<th>Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>local operator</td>
<td>requires digit timeout</td>
</tr>
<tr>
<td>00</td>
<td>toll operator</td>
<td></td>
</tr>
<tr>
<td>011</td>
<td>IDDD access</td>
<td></td>
</tr>
<tr>
<td>1N...</td>
<td>non-local (toll) call</td>
<td>8/11 digit number</td>
</tr>
<tr>
<td>1N11</td>
<td></td>
<td>1+3 digit service code</td>
</tr>
<tr>
<td>N...</td>
<td>local call</td>
<td>7/10 digit number</td>
</tr>
<tr>
<td>N11</td>
<td></td>
<td>3 digit service code</td>
</tr>
</tbody>
</table>
PSTN: Addressing
NANP (WZ1) Service Codes

- N11 codes administered by FCC

<table>
<thead>
<tr>
<th>N11</th>
<th>Use</th>
<th>Basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>211</td>
<td>community information &amp; referral services</td>
<td>FCC</td>
</tr>
<tr>
<td>311</td>
<td>non-emergency police &amp; other government services</td>
<td>FCC</td>
</tr>
<tr>
<td>411</td>
<td>local directory assistance</td>
<td>traditional</td>
</tr>
<tr>
<td>511</td>
<td>traffic and transportation info (US)</td>
<td>FCC</td>
</tr>
<tr>
<td>611</td>
<td>repair service</td>
<td>traditional</td>
</tr>
<tr>
<td>711</td>
<td>TRS (telecommunications relay service)</td>
<td>FCC</td>
</tr>
<tr>
<td>811</td>
<td>dig safe</td>
<td>FCC</td>
</tr>
<tr>
<td>911</td>
<td>emergency</td>
<td>FCC</td>
</tr>
</tbody>
</table>
### PSTN: Addressing
NANP (WZ1) Reserved NPAs

- NPAs reserved for special use

<table>
<thead>
<tr>
<th>NXX</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nxx</td>
<td>ERC: easily recognisable codes (x = 2nd = 3rd digit)</td>
</tr>
<tr>
<td>37X</td>
<td>reserved for future contiguous block use</td>
</tr>
<tr>
<td>96X</td>
<td>reserved for future contiguous block use</td>
</tr>
<tr>
<td>N9X</td>
<td>reserved for future expansion to 4-digit NPA</td>
</tr>
</tbody>
</table>
PSTN: Addressing
NANP (WZ1) Non-Geographic NPAs

- NPAs not tied to geographical region

<table>
<thead>
<tr>
<th>NXX</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>456</td>
<td>inbound international carrier identification</td>
</tr>
<tr>
<td>5XX</td>
<td>personal communication service (“follow-me”)</td>
</tr>
<tr>
<td>600</td>
<td>Canadian services</td>
</tr>
<tr>
<td>700</td>
<td>IXC (interexchange) carrier services</td>
</tr>
<tr>
<td>710</td>
<td>GETS: US government emergency telecom service</td>
</tr>
<tr>
<td>8xx</td>
<td>TRS (telecommunications relay service)</td>
</tr>
<tr>
<td>900</td>
<td>premium services (additional billing)</td>
</tr>
</tbody>
</table>
PSTN: Addressing
NANP (WZ1) Special CO Codes

- Restricted and special use CO codes

<table>
<thead>
<tr>
<th>NXX-XXXX</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>555-XXXX</td>
<td>NANP-wide assignment</td>
</tr>
<tr>
<td>555-01XX</td>
<td>fictitious use (media and advertising)</td>
</tr>
<tr>
<td>555-1212</td>
<td>directory assistance</td>
</tr>
<tr>
<td>555-1313</td>
<td>Bell Canada</td>
</tr>
<tr>
<td>700-4141</td>
<td>intraLATA carrier verification</td>
</tr>
<tr>
<td>950-XXXX</td>
<td>CAC: carrier access code (CIC xxxx)</td>
</tr>
<tr>
<td>958-XXXX</td>
<td>test code</td>
</tr>
<tr>
<td>959-XXXX</td>
<td>test code</td>
</tr>
<tr>
<td>976-XXXX</td>
<td>information delivery services (additional billing)</td>
</tr>
</tbody>
</table>
PSTN: Addressing
NANP (WZ1) Local Number Portability

• Problem: users must change phone number
  – when they move
  – when they change providers (ILEC and new CLECs)
  – when they convert to mobile telephones for home use
  – when they change mobile providers

Solution?
PSTN: Addressing
NANP (WZ1) Local Number Portability

• Problem: users must change phone number
• Solution: local number portability (LNP)
  – strongly resisted by providers
    • ILEC monopolistic customer base
    • mobile providers want to avoid customer churn
      – alternative: signup deals with long contracts and heavy penalties

• Mandated by FCC
  – 2003
    • LNP for wireline providers within an NPA
    • WLNP (wireless LNP) 100 largest MSAs (metro statistical areas)
  – 2004 WLNP in all areas (NPA not required to be portable)
PSTN Addressing
UK NTNP (WZ3) Administration

- National Telephone Numbering Plan Administration
  - formerly GPO (General Post Office)
  - 1981: BT (British Telecom) split from GPO
  - 1984: Oftel (Office of Telecommunications) created
  - 2003: Oftel merged into Ofcom (Office of Communications)
  - www.ofcom.org.uk
PSTN Addressing
UK NTNP (WZ3) Structure

• Overall structure (ITU E.164 compliant)
  – country code +44
  – NDC: area code (formerly STD code)
    • variable length: 2–5 digits
    • geographic area code: beginning with 1–3
    • non-geographic code: beginning with 4–9
  – SN: subscriber number (assigned since 1995)
    • fixed length within each area code
    • 5–8 digits = 10 digits – NDC
    • typically begin with N (2–9) but not 99
  – SN: subscriber number (assigned before 1995)
    • SN+SDC variable number of digits
PSTN: Addressing
UK NTNP (WZ3) Structure: Special Codes

- Trunk access, operator, and service codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STD</td>
<td>long distance</td>
</tr>
<tr>
<td>00</td>
<td>international access</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>operator</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>directory enquiries</td>
<td>European standard</td>
</tr>
<tr>
<td>195</td>
<td>directory enquires for disabled</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>speaking clock</td>
<td></td>
</tr>
<tr>
<td>14X</td>
<td>call features</td>
<td>CLID, trace, screen, etc.</td>
</tr>
<tr>
<td>190X</td>
<td>VTN (voice text network)</td>
<td>services for disabled</td>
</tr>
<tr>
<td>999</td>
<td>emergency</td>
<td></td>
</tr>
</tbody>
</table>
PSTN Addressing
UK NTNP (WZ3) Structure: Geographic Codes

- Geographic codes

<table>
<thead>
<tr>
<th>Code</th>
<th># Digits</th>
<th>Subscriber # Digits</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>1X1</td>
<td>3</td>
<td>7</td>
<td>large cities</td>
</tr>
<tr>
<td>11X</td>
<td>3</td>
<td>7</td>
<td>small cities</td>
</tr>
<tr>
<td>1XXX</td>
<td>4</td>
<td>6</td>
<td>medium cities</td>
</tr>
<tr>
<td>1XXXX</td>
<td>5</td>
<td>5</td>
<td>small cities</td>
</tr>
<tr>
<td>1XXX</td>
<td>4</td>
<td>5</td>
<td>legacy codes</td>
</tr>
<tr>
<td>1XXXX</td>
<td>5</td>
<td>4</td>
<td>legacy codes</td>
</tr>
<tr>
<td>2X</td>
<td>2</td>
<td>8</td>
<td>wide area code</td>
</tr>
<tr>
<td>3X</td>
<td></td>
<td></td>
<td>reserved for expansion</td>
</tr>
</tbody>
</table>
PSTN Addressing

UK NTNP (WZ3) Structure: Nongeographic Codes

- Nongeographic codes

<table>
<thead>
<tr>
<th>Code</th>
<th># Digits</th>
<th>Subscriber # Digits</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>unassigned</td>
</tr>
<tr>
<td>5X</td>
<td>2</td>
<td>8</td>
<td>corporate numbering</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>unassigned</td>
</tr>
<tr>
<td>7XXX</td>
<td>4</td>
<td>6</td>
<td>mobile and personal</td>
</tr>
<tr>
<td>8XX</td>
<td>3</td>
<td>7</td>
<td>special services</td>
</tr>
<tr>
<td>800</td>
<td>3</td>
<td>7</td>
<td>freephone</td>
</tr>
<tr>
<td>808</td>
<td>3</td>
<td>7</td>
<td>freephone</td>
</tr>
<tr>
<td>82X</td>
<td>3</td>
<td>7</td>
<td>Internet for schools</td>
</tr>
<tr>
<td>9XXX</td>
<td>4</td>
<td>6</td>
<td>premium rate</td>
</tr>
</tbody>
</table>
### PSTN Addressing

**UK NTNP (WZ3) Structure: Geographic Codes**

- Wide area geographical codes

<table>
<thead>
<tr>
<th>Code</th>
<th># Digits</th>
<th>Subscriber # Digits</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>2</td>
<td>8</td>
<td>London</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>8</td>
<td>South Hampshire</td>
</tr>
<tr>
<td>24</td>
<td>3</td>
<td>8</td>
<td>Coventry</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>8</td>
<td>Northern Ireland</td>
</tr>
<tr>
<td>29</td>
<td>5</td>
<td>8</td>
<td>Cardiff, planned for Wales</td>
</tr>
</tbody>
</table>
PSTN Addressing
UK NTNP (WZ3) Growth History

• Selected highlights of UK NTNP history
  – 2000: 2X codes introduced
    • London
      171 XXX XXXX → 20 7XXX XXXX (outer)
      181 XXX XXXX → 20 8XXX XXXX (inner)
  – Oftel publishes new NTNP: NDC+SN = 10 digits
  – 1994–95: phONE day
    • most geographic codes prepended by 1
    • e.g. London 71 → 171; Lancaster 524 → 1524
  – 1990: London splits
    • inner London: from 1 → 171 XXX XXXX
  – 1958: STD (subscriber trunk dialing) direct long distance
PSTN Signalling
Overview

Signalling needed for?
PSTN Signalling
Overview

• Signalling needed for
  – call setup and routing
  – call modification
  – call release

• Signalling between
  – telephone and central office (local loop)
  – inter-office (trunks)
PSTN Signalling

In-Channel Signalling

• In-channel (per-trunk) signalling
  – same transmission links for voice and signalling
  – in-band: uses voice frequencies
    • SF (single frequency): 2600 Hz tone
    • MF (multifrequency): interoffice
    • DTMF (dual-tone multifrequency)
      – row \{697|770|852|941\} Hz + column \{1209|1336|1477\} Hz
  – out-of-band: non-voice frequencies
    • DC -48V for off-hook
    • dial pulses from rotary telephone (10 pulse/sec)
    • 20Hz ringing voltage from central office

Advantages?
PSTN Signalling
In-Channel Signalling

- In-channel (per-trunk) signalling
  - same transmission links for voice and signalling
  - in-band: uses voice frequencies
  - out-of-band: non-voice frequencies

- Advantages
  - does not require distinct infrastructure
  - compatible with simple POTS telephones

Disadvantages?
PSTN Signalling

In-Channel Signalling

- In-channel (per-trunk) signalling
  - same transmission links for voice and signalling
  - in-band: uses voice frequencies
  - out-of-band: non-voice frequencies

- Advantages
  - does not require distinct infrastructure
  - compatible with simple POTS telephones

- Disadvantages
  - user can hack into network
    - blue boxes, etc.
  - security by obscurity
PSTN Signalling
Common Channel Signalling

• Common channel signalling
  – distinct data network for signalling (control plane)
    • *out-of-band* signalling
  – connects control processing of switches
  – associated signalling
    • signalling overlay network matches trunk topology
  – dissociated signalling
    • signalling network topology optimised for signalling
PSTN Common Channel Signalling
SS7 Overview

- SS7: signalling system no. 7
  - evolution of SS6 for capabilities beyond voice telephony
    - support for ISDN (integrated services data network)
    - variable message lengths and higher link rates
  - deployed beginning in the 1980s
    - initially for 800 freephone number translation in the US
PSTN Common Channel Signalling
SS7 Network Architecture

- SS7 signalling network components and architecture
  - telephone switch
  - SSP: service switching point
    - adjunct signalling processor
  - STP: signalling transfer point
    - signalling message switch
  - SCP: service control points
    - interface to databases
    - e.g. 800 number translation

Diagram:
- SSP
- STP
- SCP
- 800
- non-associated signalling links
- redundant paths
- associated signalling channel
- voice trunks
PSTN Common Channel Signalling
SS7 Network Protocol Stack

- SS7 protocol stack
  - ITU-T Q.700
    - TUP: telephone user part
    - ISUP: ISDN user part
    - TC: transaction capabilities
    - SSCP: signalling connection control part
    - MTP: message transfer part
      - level 3
      - level 2
      - level 1
PSTN Common Channel Signalling

SS7 Protocols: MTP

- **MTP**: message transfer part
  - ITU Q.701–705  ANSI T1.111
    - signalling data link
    - level 1: link characteristics
    - level 2: message transfer
    - level 3: common functions
      - user part message handling
      - signalling network management
        - configuration
        - routing
        - restoration
PSTN Common Channel Signalling

SS7 Protocols: SSCP

- **SSCP**: signalling connection control part
  - ITU Q.711–716 ANSI T1.112
  - E2E signalling services over MTP
    - connection oriented
    - connectionless
    - used by MTP and TC
PSTN Common Channel Signalling

SS7 Protocols: TUP

• TUP: telephone user part
  ITU Q.721–725
  – telephone call control signalling
  – based on SS6
  – uses MTP for message transport
  – largely replaced by ISUP
PSTN Common Channel Signalling
SS7 Protocols: ISUP

- **ISUP**: ISDN user part
  - ITU Q.761–766  ANSI T1.113
  - signalling for ISDN: integrated services data network
    - data, voice, video
  - uses MTP for message transport
  - uses SSCP for E2E signalling
PSTN Common Channel Signalling
SS7 Protocols: ISUP Messages

- ISUP signalling message examples
  - call establishment
    - IAM: initial address message (call SETUP)
    - ACM: address complete message (call PROCEEDING)
    - ANM: answer message (call CONNECT)
  - call modification
  - call teardown
    - REL: release (call RELEASE)
    - RLC: release complete (call RELEASE ack)
PSTN Common Channel Signalling
SS7 Protocols: TCAP

- TCAP: transaction capabilities application part
  ITU Q.771–775 ANSI T1.114
  (also called TCAP: transaction capabilities application part)
  - signalling for transactional (non-circuit) communication
  - uses SSCP for E2E signalling
  - used by OAM and ASEs
PSTN Common Channel Signalling

SS7 Protocols: TCAP Users

- TCAP users
  - OAM: operations, administration, maintenance
  - ASE: application service elements
PSTN Common Channel Signalling
SS7 Protocols: Telephony Call Example

- Telephony signalling
PSTN Common Channel Signalling
SS7 Protocols: Telephony Call Example

- Telephony signalling
  - caller goes off-hook
PSTN Common Channel Signalling

SS7 Protocols: Telephony Call Example

- Telephony signalling
  - caller goes off-hook
  - caller receives dial tone
PSTN Common Channel Signalling

SS7 Protocols: Telephony Call Example

- Telephony signalling
  - caller goes off-hook
  - caller receives dial tone
  - caller enters digits
PSTN Common Channel Signalling
SS7 Protocols: Telephony Call Example

- Telephony signalling
  - caller goes off-hook
  - caller receives dial tone
  - caller enters digits
  - calling switch sends IAM
PSTN Common Channel Signalling
SS7 Protocols: Telephony Call Example

- **Telephony signalling**
  - caller goes off-hook
  - caller receives dial tone
  - caller enters digits
  - calling switch sends IAM
  - called switch rings and returns ACM
PSTN Common Channel Signalling
SS7 Protocols: Telephony Call Example

- Telephony signalling
  - caller goes off-hook
  - caller receives dial tone
  - caller enters digits
  - calling switch sends IAM
  - called switch rings and returns ACM
  - caller hears ring tone
PSTN Common Channel Signalling

SS7 Protocols: Telephony Call Example

• Telephony signalling
  – caller goes off-hook
  – caller receives dial tone
  – caller enters digits
  – calling switch sends IAM
  – called switch rings and returns ACM
  – caller hears ring tone
  – callee answers and goes off-hook
PSTN Common Channel Signalling
SS7 Protocols: Telephony Call Example

- Telephony signalling
  - caller goes off-hook
  - caller receives dial tone
  - caller enters digits
  - calling switch sends IAM
  - called switch rings and returns ACM
  - caller hears ring tone
  - callee answers and goes off-hook
  - called switch returns ANM; connected
PSTN Common Channel Signalling

SS7 Protocols: Telephony Call Example

- Telephony signalling
  - caller goes off-hook
  - caller receives dial tone
  - caller enters digits
  - calling switch sends IAM
  - called switch rings and returns ACM
  - caller hears ring tone
  - callee answers and goes off-hook
  - called switch returns ANM; connected
  - parties talk
PSTN Common Channel Signalling
SS7 Protocols: Telephony Call Example

- Telephony signalling
  - caller goes off-hook
  - caller receives dial tone
  - caller enters digits
  - calling switch sends IAM
  - called switch rings and returns ACM
  - caller hears ring tone
  - callee answers and goes off-hook
  - called switch returns ANM; connected
  - parties talk
  - parties hang up by going on-hook
PSTN Common Channel Signalling

SS7 Protocols: Telephony Call Example

• Telephony signalling
  – caller goes off-hook
  – caller receives dial tone
  – caller enters digits
  – calling switch sends IAM
  – called switch rings and returns ACM
  – caller hears ring tone
  – callee answers and goes off-hook
  – called switch returns ANM; connected
  – parties talk
  – parties hang up by going on-hook
  – REL and RLC messages terminate
PSTN Common Channel Signalling

IN Overview

- **IN**: Intelligent Network
  - ITU-T Q.1200 series
- **Network programmability and interfaces**
  - based on SS7
  - **BCSM**: basic call state model
    - provides triggers for altering SS7 state machine behaviour
  - interfaces to add programmable services
    - without altering core SS7/ISUP/B-ISUP code
    - e.g. new call features, VPNs, mobile networking
Network Layer

NL.4.2 Examples: X.25 CONS

NL.1 Network layer functions and services
NL.2 Network signalling paradigms
NL.3 Switches and packet structure
NL.4 Examples
  NL.4.1 PSTN
  NL.4.2 X.25 CONS (for reference only)
  NL.4.3 ISDN and Frame Relay
  NL.4.4 B-ISDN and ATM
  NL.4.5 MPLS
  NL.4.6 IP, ICMP and IPv6

NL.5 Fast datagram routers
X.25 CONS PSPDN
Functions and Protocols

• Recall: early packet-switched public data networks

• X.25 CONS PSPDN functions and protocols
  – addressing: international data number [ITU X.121]
    • alternatives: E.164, Telex, private
  – forwarding:
    • interfaces
      – DTE (end-system) – DCE (intermediate system)
      – DCE – DCE
    • X.25 packet switch
  – routing: standards say nothing about routing
    • PDN provider dependent
X.25 CONS PSPDN
Overview

- Packet-switched public data networks (PSPDN)
  - abundant before emergence of global Internet
- Interface specification
  - DTE (data terminal equipment) end system
  - DCE (data communication equipment) intermediate system
  - X.25 DTE–DCE and DTE–DTE
  - X.75 DCE–DCE and internetworking between X.25 PSPDNs
  - ISO CONS (connection-oriented network service)
    - [ISO/IEC 8878] (also [ITU X.213]) and [ISO/IEC 8208]
- Historical importance
  - illustrative of how to construct connection-oriented networks
X.25 CONS PSPDN

Addressing

• X.25 *addressing* needed to determine VC endpoints
• Options defined by *numbering plan identification*
  – IDN international data number [ITU X.121]
  – PSTN number [ITU E.164]
  – telex number [ITU F.69]
  – private numbering plan
X.25 CONS PSPDN
Routing

• X.25/X.75 does *not* specify *routing* mechanism
  – each network provider chooses algorithms and protocols

• Examples
  – TRANSPAC (France):
    • centralised least-cost (link utilisation) with local optimisations
    • similar to delta-routing [Rudin 1976]
X.25 CONS PSPDN
Service Models and Transfer Modes

- Permanent virtual circuits
  - virtual circuits provisioned in advance by network operator
  - no per VC signalling; packet forwarding required
- Switched virtual circuits
  - per virtual connection signalling
  - connection routing and packet forwarding required
- Fast select (transactions added in [X.25-1980])
  - CALL REQUEST packet can contain 128B of initial data
  - connection routing and packet forwarding required
- Datagram (added in [X.25-1980] following ANSI work)
  - datagram routing and forwarding required
# X.25 CONS PSPDN

## X.25 Signalling Messages

<table>
<thead>
<tr>
<th>Message Type</th>
<th>X.25 Origination</th>
<th>X.25 Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETUP</td>
<td>Call request</td>
<td>Incoming call</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Call connected</td>
<td>Call accepted</td>
</tr>
<tr>
<td>RESET</td>
<td>Reset request</td>
<td>Reset indication</td>
</tr>
<tr>
<td>RELEASE</td>
<td>Clear request</td>
<td>Clear indication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clear confirmation</td>
</tr>
</tbody>
</table>

todo: expand and diagram
X.25 CONS PSPDN

X.25 Generic Packet Format: Type

- **GFI**: general format indicator [4b]
  - describes packet format
- **LCN/LCGN** [12b]
- **Control data** [8b]
  - packet type identifier for control packets
  - control fields for data packets
  - C/D: 1 = control / 0 = data
- **Additional information**
  - control and datagram: additional header
  - data packet: user data field (payload)
### X.25 CONS PSPDN

**X.25 Generic Packet Format: Channels**

- **GFI**
- **Logical channel number** [12b]
  - LCGN logical channel group number [4b]
  - LCN logical channel number [8b]
- **Control data**
- **Additional information**

### Logical Channel Format

<table>
<thead>
<tr>
<th>GFI</th>
<th>LCGN</th>
</tr>
</thead>
<tbody>
<tr>
<td>logical channel #</td>
<td>control data</td>
</tr>
</tbody>
</table>

- **LCGN**: logical channel group number [4b]
X.25 CONS PSPDN
X.25 Control Packet Type

- GFI
- LCGN/LCN
- Packet type identifier
  - codepoint for signalling message
- Addressing information
- Facilities (options)
- User data field (payload)
X.25 CONS PSPDN
X.25 Control Packet Addressing

- GFI + LCGN/LCN
- Packet type identifier
- **Addressing information**
  - source and destination addr length
  - destination or source address
    - TOA type of address [4b]
    - NPI numbering plan id [4b]
    - address: BCD encoding
    - zero filled to even byte

- Facilities (options)
- User data field (payload)
X.25 CONS PSPDN

X.25 Control Packet Options

- GFI + LCGN/LCN
- Packet type identifier
- Addressing information
- Facilities (options)
  - facility length [6b]
  - facilities
- User data field (payload)

<table>
<thead>
<tr>
<th>GFI</th>
<th>LCGN</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCN</td>
<td></td>
</tr>
<tr>
<td>PTI</td>
<td>1</td>
</tr>
<tr>
<td>SAL</td>
<td>DAL</td>
</tr>
<tr>
<td>DTE address</td>
<td></td>
</tr>
<tr>
<td>Ø Ø facility length</td>
<td></td>
</tr>
<tr>
<td>facilities</td>
<td></td>
</tr>
<tr>
<td>call user data (0 – 16B)</td>
<td></td>
</tr>
</tbody>
</table>
X.25 CONS PSPDN
X.25 Control Packet Options

- GFI + LCGN/LCN
- Packet type identifier
- Addressing information
- Facilities (options)
  - facility length [6b]
  - facilities
- User data field (payload)

<table>
<thead>
<tr>
<th>GFI</th>
<th>LCGN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PTI</th>
<th>1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SAL</th>
<th>DAL</th>
</tr>
</thead>
</table>

DTE address

Ø Ø facility length

facilities

call user data (0 – 16B)
X.25 CONS PSPDN
X.25 Data Packet Format

• Header GFI
  – Q data qualifier bit (user defined)
  – D delivery confirmation (ACK) requested
• Logical group and channel #
• Control data
  – P(R) receive sequence no.
  – M more data 1 = control / 0 = data
  – P(S) send sequence no.
• User data field (payload)
X.25 CONS PSPDN
X.25 Datagram Format

• Header GFI
• Logical channel #
• Control data
  – sequence numbers (hop-by-hop)
• Addressing information
  – source and destination addr length
  – destination and source address
• Facilities
• User data (payload)
  – datagram ID (optional)
Network Layer

NL.4.3  Examples: ISDN and Frame Relay

NL.1  Network layer functions and services
NL.2  Network signalling paradigms
NL.3  Switches and packet structure
NL.4  Examples
   NL.4.1  PSTN
   NL.4.2  X.25 CONS
   NL.4.3  ISDN and Frame Relay
   NL.4.4  B-ISDN and ATM
   NL.4.5  MPLS
   NL.4.6  IP, ICMP and IPv6

NL.5  Fast datagram routers
ISDN Overview

- ISDN (integrated services digital network)
  - evolution of digital network services to *end user*
  - intended to support *integrated services*
    - voice
    - data
    - eventually video
  - supported by SS7 ISUP
- ISDN assumed that PSTN would evolve into the GII
  - but the TCP/IP-based Internet won
Frame Relay Overview

• Frame relay [ITU I.233]
  – protocol to relay variable length packets

• Simple and connection-oriented
  – based on X.25 heritage
  – generally implemented over PVC (permanent virtual circuits)

• Widely deployed as an Internet service
  – but diminishing in importance
Network Layer

NL.4.4 Examples: B-ISDN and ATM

NL.1 Network layer functions and services
NL.2 Network signalling paradigms
NL.3 Switches and packet structure
NL.4 Examples
  NL.4.1 PSTN
  NL.4.2 X.25 CONS
  NL.4.3 ISDN and Frame Relay
  NL.4.4 B-ISDN and ATM
  NL.4.5 MPLS
  NL.4.6 IP, ICMP and IPv6

NL.5 Fast datagram routers
ATM Overview

- Key ideas emerged from fast packet switching
  - connection-oriented for high-line-speed processing and QOS
  - needed for 155Mb/s to the desktop
- Standards bodies made some very bad decisions
  - ITU: 53B cell (5B header + 48B payload)
  - ATM Forum *tried* to produce simple quick standards
  - required *replacement* of IP-based Internet infrastructure
- Meanwhile:
  - Ethernet 100b/s standards wars finally ended
  - IP-based global Internet became irreplaceable
B-ISDN

Overview

• B-ISDN: *broadband* ISDN
  – extension of SS7 and ISDN for high-speed networks
    • \( \geq 155 \text{ Mb/s} \) (OC-3/STM-1 line rate)
• B-ISUP: broadband ISUP
  ITU-T Q.2761–2765
  – extension of ISUP signalling for NNI: network node interface
  – uses SSCP and MTP
  – UNI: user–network interface
    ITU-T Q.2931 (extension of ISDN UNI ITU-T Q.931)
• Fast-packet cell-switching for data transport
  – ATM: asynchronous transfer mode
ATM packet format requirements

- fast packet switching: *design implication?*
- fine-grained statistical multiplexing: *design implication?*
- simplify switch designs: *design implication?*
B-ISDN
ATM Cell Structure

- ATM packet format requirements
  - fast packet switching: connection id with label swapping
  - fine-grained statistical multiplexing: small packets
  - simplify switch designs: fixed size cell

Problem?
B-ISDN ATM Cell Structure

- ATM packet format requirements
  - fast packet switching: connection id with label swapping
  - fine-grained statistical multiplexing: small packets
  - simplify switch designs: fixed size cell

*Problem*: how to determine cell size?
B-ISDN
ATM Cell Size

• Size a balance between:
  – large enough for efficient data transport
  – small enough for statistical multiplexing

*Outcome?*
B-ISDN
ATM Cell Size

- Size determined by ITU committee compromise
  - 48B = avg(32, 64)
    - 64 from US = min of proposals for data (and voice)
    - 32 from European PTTs to avoid voice echo cancellers

*Problems?*
- header tiny to keep overhead low; no room for seq #
- nothing a power of 2
- 48B + 5B = 53B; not even a multiple of 8
- small cell
B-ISDN
ATM Cell Size

- Size determined by ITU committee compromise
  - 48B = avg(32, 64)
    - 64 from US = min of proposals for data (and voice)
    - 32 from European PTTs to avoid voice echo cancellers
- Problems:
  - header tiny (5B) to keep overhead low; no room for seq #
  - nothing a power of 2
  - 48B + 5B = 53B; not even a multiple of 8
  - small cell ⇒ short interarrival time
    - cell processing had to be done in expensive custom VLSI
    - Ethernet became cheaply available and killed ATM
B-ISDN
ATM AAL-5 Cells

- AAL 3/4 for data
  - 4B AAL header
    - useless seq#
  - 44B payload
    - not div 8

- AAL 5
  - no header
  - trailer last frag
  - forced on ITU

\[
\begin{align*}
\text{SAR-SDU} & = 48B \\
\text{payload} & = 40B \\
\text{trailer} & = 8B
\end{align*}
\]
B-ISDN and ATM

Current State

- ATM is (mostly) dead
  - some carrier infrastructure remains
  - some bits are used, e.g. for ADSL
- Fast packet switching technology is thriving
  - cell-based fast IP routers
  - MPLS as IP underlay for traffic engineering
    - uses label swapping

More in EECS 881
Network Layer

NL.4.5 Examples: MPLS

NL.1 Network layer functions and services
NL.2 Network signalling paradigms
NL.3 Switches and packet structure
NL.4 Examples
  NL.4.1 PSTN
  NL.4.2 X.25 CONS
  NL.4.3 ISDN and Frame Relay
  NL.4.4 B-ISDN and ATM
  NL.4.5 MPLS
  NL.4.6 Internet: DNS, IP, ICMP and IPv6

NL.5 Fast datagram routers
MPLS Overview

• MPLS (multiprotocol label switching) [RFC 3031]
  – intended as a label-swapping shim underlay to IP (L2.5)
  – originally intended to enable Internet fast packet switching
    • without using ATM under IP
  – fast datagram switching made this unnecessary
    • now used as a traffic engineering underlay by some ISPs

• RSVP-TE (RSVP for traffic engineering) [RFC 3209]
  – used for signalling label-switched paths
  – chosen over competing LDP [RFC 3036] proposal

• GMPLS: generalised MPLS [RFC 3741, 4238]
  – extensions for optical networks
MPLS
Packet Shim Format

- MPLS label shim
  - switches swap label
  - stacked labels
    - allows net hierarchy (ala VP/VC)
Network Layer

NL.4.6 Examples: Internet

NL.1 Network layer functions and services
NL.2 Network signalling paradigms
NL.3 Switches and packet structure
NL.4 Examples
   NL.4.1 PSTN
   NL.4.2 X.25 CONS
   NL.4.3 ISDN and Frame Relay
   NL.4.4 B-ISDN and ATM
   NL.4.5 MPLS
   NL.4.6 Internet: DNS, IP, ICMP and IPv6

NL.5 Fast datagram routers
Internet
Functions and Protocols

- Addressing
- Forwarding
- Routing
- Signalling
- Traffic Management
Internet
Functions and Protocols

- Addressing?
- Forwarding
- Routing
- Signalling
- Traffic Management
Internet
Functions and Protocols: Addressing

- **Addressing**: IP
  - IPv4 [RFC 0791 / STD 0005]
  - subnetting [RFC 0950 / STD 0005]
  - CIDR [RFC 1519]
  - MAC layer address resolution: ARP [RFC 0826 / STD 0037]
  - IPv6 [RFC 2460, 3513]

- **Forwarding**
- Routing
- Signalling
- Traffic Management
Internet

Functions and Protocols: Forwarding

- Addressing: IP
- *Forwarding*: IP
  - IP address lookup in routers
- *Routing*
- Signalling
- Traffic Management
Internet
Functions and Protocols: Routing

- Addressing: IP
- Forwarding: IP
- **Routing**: Lecture NR
  - interdomain EGP (exterior gateway protocol): BGP
  - intradomain IGP (interior gateway protocol): RIP, OSPF, ...
- **Signalling?**
- Traffic Management
Internet

Functions and Protocols: Signalling

- Addressing: IP
- Forwarding: IP
- Routing: BGP + IGPs
- **Signalling**: ICMP
  - ICMPv4 [RFC 0792 / STD 0005]
  - ICMPv6 [RFC 2463]
- **Traffic Management?**
Internet Functions and Protocols: Traffic Management

- Addressing: IP
- Forwarding: IP
- Routing: BGP + IGPs
- Signalling: ICMP

- **Traffic management**: Lecture TQ
  - congestion avoidance and control (e.g. RED)
  - fair queuing
  - DiffServ
  - IntServ
Internet Architecture

Overview

- Interconnection of service provider networks
  - tier 1 service providers,
  - lower tier (2 and 3) service provider networks
  - access networks
  - enterprise, campus, home
- No organised structure
  - since end of NSFNET
- Service model
  - best effort: no performance guarantees
  - providers may offer SLAs (service level agreements)
    - generally by over-provisioning of network infrastructure
Internet Architecture
Overview: Regulation

• Minimal regulation
  – ICANN [www.icann.org]
    Internet Corporation for Assigned Names and Numbers
    • administers DNS TLDs (top level domains) & IP address blocks
    • registrars administer second level domains and subnets
  – IANA [www.iana.org]
    Internet Assigned Numbers Authority
    • administers various number- and name-spaces

• Repeated attempts to regulate
  – by government and traditional PSTN carriers
  – largely unsuccessful so far
  – ultimately futile due to global scope of Internet
Internet Architecture
Overview: Standards

- **IAB** (Internet Architecture Board) [www.iab.org]
  - advisory role to ISoc (Internet Society) [www.isoc.org]
  - committee of Internet Engineering Task Force (IETF)
  - Internet Research Task Force (IRTF) [www.irtf.org]

- **IETF protocol standards** [www.ietf.org]
  - but vendors implement what they wish...
    subject to ...
  - service providers who offer what they wish...
    subject to ...
  - customer demand
Internet Architecture

Design Principles

• ARPANET design principles
  – simple (relatively) stateless core for resiliency
  – most functionality at end systems
  – end-to-end addressing transparency
  – hourglass model (actually a bit later than original design)
    • any transport protocol over IP over any link layer

• End-to-end arguments
  – what functionality *must* be located on end systems
  – what functionality *should* be in the network for performance

*Recall: these two are not the same thing*
Names and Addresses

Overview

• **Address:** identifier of a node
  - may only be machine readable (binary address)
    • e.g. 10000001 11101101 01010111 00010010
  - may be represented by human readable number
    • e.g. 129.237.87.18 or 148.88.3.47
  - may be indirection by human friendly form (e.g. DNS name)
    • e.g. www.eecs.ku.edu or www.comp.lancs.ac.uk

• **Name:** *global persistent* identifier of an entity
  - e.g. James Philip Guenther Sterbenz

• Unfortunately “name” is commonly used for both
Network Layer

NL.4.6 Examples: DNS

NL.1 Network layer functions and services
NL.2 Network signalling paradigms
NL.3 Switches and packet structure
NL.4 Examples
   NL.4.1 PSTN
   NL.4.2 X.25 CONS
   NL.4.3 ISDN and Frame Relay
   NL.4.4 B-ISDN and ATM
   NL.4.5 MPLS
   NL.4.6 Internet: DNS, IP, ICMP and IPv6

NL.5 Fast datagram routers
Domain Name System
Overview

• DNS: domain name system [RFC 1034/1035 / STD 0013]
  – directory service for the Internet
  – resolves *hostnames* to IP addresses
    • hostname is merely a human friendly address redirection

• DNS is
  – technically an application layer protocol
    • runs over TCP or UDP
  – an essential network infrastructure
    *why?*
Domain Name System
Overview

- DNS: domain name system [RFC 1034/1035 / STD 0013]
  - directory service for the Internet
  - resolves *hostnames* to IP addresses
    - hostname is merely a human friendly address redirection

- DNS is
  - technically an application layer protocol
    - runs over TCP or UDP
  - in reality an essential network infrastructure service
    http://www.eecs.ku.edu
    rather than
    http://129.237.87.18
Domain Name System

Services

• Hostname to IP addresses *resolution*
  – e.g. www.eecs.ku.edu to 129.237.87.18
Domain Name System Services

- Hostname to IP addresses \textit{resolution}
- Hostname aliasing to canonical name
  - allows stable DNS names for Web and mail servers, e.g.
    - \texttt{www.sterbenz.org} to \texttt{abell.lunarpages.com:80}
    - \texttt{www.ku.edu} to \texttt{raven.cc.ku.edu:80}
    - \texttt{mail.ittc.ku.edu} to \texttt{stephens.ittc.ku.edu:25}
  - allows proper default behavior on incoming port 80
    - \texttt{http://example.com} to \texttt{http://www.example.com}
    - many servers not properly configured to do this
Domain Name System

Services

- Hostname to IP addresses \textit{resolution}
- Hostname aliasing to canonical name
- Load distribution
  - set of IP addresses for one canonical name
  - typically used for replicated Web servers
  - e.g. www.cnn.com to 264.236.{255|224|226}.n
Domain Name System
Implementation

- Distributed database implemented in a hierarchy
  - many *name servers*
  - *no* relationship to IP addressing structure!
- Distributed implementation
  - improves scalability
  - decentralises administration
Domain Name System

Name Structure

- Fully qualified domain name (FQDN): ... . ⟨SLD⟩ . ⟨TLD⟩
- TLD: top level domain
  - originally assigned by IANA
  - now assigned by ICANN [www.icann.org/tlds](http://www.icann.org/tlds)
  - gTLD: generic TLD
  - ccTLD: country-code TLD
- SLD: second level domain
  - may be defined by TLD policy
  - may be assigned by TLD registrar to domain owner
- nLD: nth level domain
  - may be defined by TLD or SLD policy
  - may be assigned by registrar/sub-registrar to domain owner
Domain Name System
Name Structure: Generic TLDs

• gTLD: generic top level domain
  – assigned by ICANN www.icann.org/tlds
  – gTLD types
    • reserved
    • infrastructure
    • unrestricted and restricted
    • unsponsored and sponsored
    • unrestricted
    • pseudo

• SLD: second level domain
  – policy based on each TLD
  – direct indicates that SLD assigned to domain owner
Domain Name System
Name Structure: Reserved gTLDs

• DNS names reserved for special use  [RFC 2606]

<table>
<thead>
<tr>
<th>TLD</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>.test</td>
<td>reserved for DNS code testing</td>
</tr>
<tr>
<td>.example</td>
<td>online and documentation examples for valid DNS name</td>
</tr>
<tr>
<td>.invalid</td>
<td>online and documentation examples for invalid DNS name</td>
</tr>
<tr>
<td>.localhost</td>
<td>resolves to loopback address (typically 127.0.0.1)</td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: Infrastructure gTLDs

- Special names used for DNS infrastructure
- `.arpa` originally used for transition to DNS
  - ARPANET host tables to DNS databases
- Remains in use [RFC 3172]
  - reverse lookup IP address → DNS: `in-addr.arpa` `ip6.arpa`
  - service mapping: `e.164.arpa` `uri.arpa` `urn.arpa`
- `.root` used by Verisign root servers
  - nonstandard but apparently used for end of zone file

<table>
<thead>
<tr>
<th>TLD</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>.arpa</code></td>
<td>reverse IP and service lookup</td>
</tr>
<tr>
<td><code>.root</code></td>
<td>used by Verisign root DNS servers</td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: Unrestricted gTLDs

- Among original seven 1980 TLDs  [RFC 0920]
- Originally partitioned among
  - commercial entities and businesses
  - network service providers
  - non-profit organisations
- NSI mismanaged assignments & ignored IANA intent
  - registered to *anyone* willing to pay annual fee
  - distinction is now almost meaningless

<table>
<thead>
<tr>
<th>TLD</th>
<th>Original Use</th>
<th>Registrar</th>
</tr>
</thead>
<tbody>
<tr>
<td>.com</td>
<td>commercial</td>
<td>many</td>
</tr>
<tr>
<td>.net</td>
<td>network service provider</td>
<td>many</td>
</tr>
<tr>
<td>.org</td>
<td>non-profit organisation</td>
<td>ISOC Public Interest Reg.</td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: Restricted gTLDs

- Among original
  - .edu .gov .mil among seven 1980 TLDs [RFC 0920]
  - .nato created in late 1980s but replaced by .int
  - .int created in late 1980s for international use

- Registrar sets policy and restricts use

<table>
<thead>
<tr>
<th>TLD</th>
<th>Use</th>
<th>Registrar</th>
</tr>
</thead>
<tbody>
<tr>
<td>.edu</td>
<td>higher educational institution</td>
<td>EDUCAUSE</td>
</tr>
<tr>
<td>.gov</td>
<td>US government</td>
<td>US GSA</td>
</tr>
<tr>
<td>.mil</td>
<td>US DOD (military)</td>
<td>US DOD DISA</td>
</tr>
<tr>
<td>.int</td>
<td>international</td>
<td>IANA</td>
</tr>
<tr>
<td>.nato</td>
<td>NATO originally</td>
<td>replaced by .int</td>
</tr>
</tbody>
</table>
# Domain Name System

## Name Structure: Un-sponsored gTLDs

- Among second batch of seven new TLDs in 2000
- ICANN responsible for TLD policy
  - registry delegated under ICANN contract
  - .biz and .info use and policies loosely applied, at best

<table>
<thead>
<tr>
<th>TLD</th>
<th>SLD</th>
<th>Use</th>
<th>Registrar</th>
</tr>
</thead>
<tbody>
<tr>
<td>.biz</td>
<td>direct</td>
<td>business</td>
<td>Neulevel (Neustar)</td>
</tr>
<tr>
<td>.info</td>
<td>direct</td>
<td>information</td>
<td>Afilias</td>
</tr>
<tr>
<td>.name</td>
<td>(first). (last)</td>
<td>individuals</td>
<td>Global Name Registry</td>
</tr>
<tr>
<td>.pro</td>
<td>3-letter-prof-code</td>
<td>professions</td>
<td>RegistryPro</td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: Sponsored gTLDs

- Among second batch of seven new TLDs in 2000
- Sponsor represents narrow community
- Responsible for TLD policy
  - who can register
  - substructure, e.g. SLDs

<table>
<thead>
<tr>
<th>TLD</th>
<th>SLD</th>
<th>Use</th>
<th>Sponsor</th>
</tr>
</thead>
<tbody>
<tr>
<td>.aero</td>
<td>IATA airport code</td>
<td>air travel</td>
<td>SITA</td>
</tr>
<tr>
<td></td>
<td>IATA airline designator</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>direct</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.coop</td>
<td>direct</td>
<td>co-ops</td>
<td>DotCooperation</td>
</tr>
<tr>
<td>.museum</td>
<td>direct</td>
<td>museums</td>
<td>MuseDoma</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: Sponsored gTLDs

- Third batch of new TLDs in 2005
- Sponsor represents narrow community
- Responsible for TLD policy
  - who can register
  - substructure, e.g. SLDs

<table>
<thead>
<tr>
<th>TLD</th>
<th>SLD</th>
<th>Use</th>
<th>Sponsor</th>
</tr>
</thead>
<tbody>
<tr>
<td>.jobs</td>
<td>validated company</td>
<td>job ads</td>
<td>Soc. for HR Management</td>
</tr>
<tr>
<td></td>
<td>others</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.travel</td>
<td>validated company</td>
<td>museums</td>
<td>Travel Partnership Corp.</td>
</tr>
</tbody>
</table>
Domain Name System

Name Structure: Sponsored gTLDs

- Newly approved TLDs in 2005 and 2006
- Many additional proposals followed
  - .xxx initially denied by ICANN
    - serous constitutional implications
    - RFC 3675: *sex Considered Dangerous*, Feb. 204

<table>
<thead>
<tr>
<th>TLD</th>
<th>SLD</th>
<th>Use</th>
<th>Sponsor</th>
</tr>
</thead>
<tbody>
<tr>
<td>.asia</td>
<td></td>
<td>Asia–Pacific regional</td>
<td>DotAsia</td>
</tr>
<tr>
<td>.cat</td>
<td></td>
<td>Catalan language/culture may <em>not</em> be used for cats</td>
<td>Fundació puntCAT</td>
</tr>
<tr>
<td>.mobi</td>
<td></td>
<td>mobile devices</td>
<td>Mobi JV corp. consortium</td>
</tr>
<tr>
<td>.post</td>
<td>yes</td>
<td>post offices</td>
<td>Universal Postal Union</td>
</tr>
<tr>
<td>.tel</td>
<td>E.164 digits direct</td>
<td>telephony services</td>
<td>Telname Ltd.</td>
</tr>
</tbody>
</table>
Domain Name System

Name Structure: Unrestricted gTLDs

- ICANN accepted recommendation to open TLDs
  - applicants can choose (almost) any TLD
  - will accommodate non-Latin TLD names
  - implementation has begun; over 1000 new g
- Mechanisms will be used for trademarks, e.g.
  - *sunrise*: applications only from trademark holders
  - *land-rush*: applications from anyone meeting requirements
- This is widely recognised as a *terrible* idea
Domain Name System
Name Structure: Unrestricted gTLD Problems

- Expensive: $185,000 application fee
  - biased against third world
  - biased against small businesses
    - can’t afford trademark protection

- Multiple registrations for trademark protection
  - make registrars wealthy; ICN estimates $200M/yr for .xxx
  - e.g. needing apple.sucks, apple.xxx

- Legal nightmare
  - reopen disputes long settled in SLDs
    - e.g. apple.com vs. applemusic.com
  - will make the lawyers wealthy
Domain Name System
Name Structure: Pseudo-TLDs

- Used in DNS-like names
  - indicates routing within or to non-DNS networks
  - BITNET, CSNET, and UUCP
    - if they had persisted might have been real DNS root entries

<table>
<thead>
<tr>
<th>TLD</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>.bitnet</td>
<td>historic BITNET</td>
</tr>
<tr>
<td>.csnet</td>
<td>historic CSNET</td>
</tr>
<tr>
<td>.uucp</td>
<td>historic UUCP</td>
</tr>
<tr>
<td>.local</td>
<td>Apple Mac OSX bonjour/rendezvous zeroconf LAN protocol</td>
</tr>
<tr>
<td>.onion</td>
<td>Onion anonymous routing</td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: Country Code TLD

- ccTLD: country code top level domain
  - based on [ISO 3166]
  - administered per nation
    - some with profit potential, e.g. .tv (Tuvalu)
    - many *domain hacks* possible e.g. jam.es
  - examples
    - .us .ca .uk .de .ch .jp
Domain Name System

Name Structure: ccTLD ISO Exceptions

• ccTLD exceptions to [ISO 3166] codes
  – European Union domain .eu
  – United Kingdom uses .uk (ISO 3166 is gb)
    • both .uk and .gb were simultaneously in use
    • permitted by IANA to choose one
  – Ascension Island uses .ac
    • grandfathered based on previous IANA policy (postal code)
    • should be .sh (dependency of Saint Helena)
  – Australia originally used .oz for ACSnet
    • assigned compliant .au
    • reassigned to .oz.au
Domain Name System
Name Structure: Unused ccTLDs

- Unused ccTLDs

<table>
<thead>
<tr>
<th>ccTLD</th>
<th>Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>.bv</td>
<td>Bouvet Island</td>
<td>in root but unused</td>
</tr>
<tr>
<td>.eh</td>
<td>Western Sahara</td>
<td>disputed by Morocco and SADR</td>
</tr>
<tr>
<td>.kp</td>
<td>North Korea</td>
<td>largely unused</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DPRK intranet isolated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>kcce.kp hosted in Berlin</td>
</tr>
<tr>
<td>.sj</td>
<td>Svalbard and Jan Mayen Islands</td>
<td>Norwegian territory: .no used</td>
</tr>
<tr>
<td>.so</td>
<td>Somalia</td>
<td>suspended and unused</td>
</tr>
<tr>
<td>.um</td>
<td>US Minor Outlying Islands</td>
<td>currently unused</td>
</tr>
</tbody>
</table>
## Domain Name System
### Name Structure: Obsolete ccTLDs

- **Obsolete ccTLDs**

<table>
<thead>
<tr>
<th>ccTLD</th>
<th>Use</th>
<th>Replaced By</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>.bu</td>
<td>Burma → Myanmar</td>
<td>.mm</td>
<td>never used</td>
</tr>
<tr>
<td>.cs</td>
<td>Czechoslovakia</td>
<td>.cz .sk&lt;br&gt; Serbia and Montenegro .rs .me</td>
<td>phased out in 1993&lt;br&gt; never used (.yu)</td>
</tr>
<tr>
<td>.dd</td>
<td>DDR (East Germany)</td>
<td>.de</td>
<td>never used</td>
</tr>
<tr>
<td>.su</td>
<td>Soviet Union</td>
<td>.ru</td>
<td>still in use</td>
</tr>
<tr>
<td>.tp</td>
<td>Portuguese → East Timor</td>
<td>.tl</td>
<td>phasing out</td>
</tr>
<tr>
<td>.yu</td>
<td>Yugoslavia</td>
<td>.rs</td>
<td>phasing out</td>
</tr>
<tr>
<td>.zr</td>
<td>Zaire → DR Congo</td>
<td>.cd</td>
<td>phased out in 2001</td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: US ccTLD

- US country code TLD
  - policy specified in [RFC 1480]
  - US Dept. of Commerce is sponsor and determines policy
  - registry is Neustar (also the NANPA)

- SLD
  - some structured by state, e.g. .state.mn.us
    (Kansas is using kansas.gov) .hopkinton.k12.ma.us
    .ci.cambridge.ma.us
  - some reserved e.g. .kids.us
  - non-conflicting direct assignments since 2002
Domain Name System
Name Structure: US SLDs

- US country code SLDs
  - reserved SLDs
  - SLDs based on two-character state postal code

<table>
<thead>
<tr>
<th>SLD.ccTLD</th>
<th>Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>.fed.us</td>
<td>federal government</td>
<td>.gov far more common</td>
</tr>
<tr>
<td>.nsn.us</td>
<td>native sovereign nations</td>
<td>e.g. hopi.nsn.us</td>
</tr>
<tr>
<td>.isa.us</td>
<td>interstate agencies</td>
<td>not commonly used</td>
</tr>
<tr>
<td>.dni.us</td>
<td>distributed national institutes</td>
<td>not commonly used</td>
</tr>
<tr>
<td>.&lt;xx&gt;.us</td>
<td>state government</td>
<td>RFC 1480 substructure</td>
</tr>
<tr>
<td>.kids.us</td>
<td>restricted content deemed appropriate for kids under 13</td>
<td>compliance reviewed by content managers</td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: US State 3LDs

- US state 3LDs for state governments and entities

<table>
<thead>
<tr>
<th>3LD.SLD.ccTLD</th>
<th>Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>.state.&lt;xx&gt;.us</td>
<td>state government</td>
<td>many states use &lt;xx&gt;.gov</td>
</tr>
<tr>
<td>.district.&lt;xx&gt;.us</td>
<td>regional state entity</td>
<td>not commonly used</td>
</tr>
<tr>
<td>.gen.&lt;xx&gt;.us</td>
<td>general use</td>
<td>not commonly used</td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: US State School 3LDs

- US state 3LD structure for schools and libraries

<table>
<thead>
<tr>
<th>.4LD.3LD.SLD.ccTLD</th>
<th>Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>.&lt;school&gt;.k12.&lt;xx&gt;.us</td>
<td>public schools</td>
<td>school or district name</td>
</tr>
<tr>
<td>.&lt;school&gt;.pvt.k12.&lt;xx&gt;.us</td>
<td>private schools</td>
<td></td>
</tr>
<tr>
<td>.&lt;name&gt;.cc.&lt;xx&gt;.us</td>
<td>community colleges</td>
<td></td>
</tr>
<tr>
<td>.&lt;name&gt;.tec.&lt;xx&gt;.us</td>
<td>vocational/technical</td>
<td></td>
</tr>
<tr>
<td>.&lt;library&gt;.lib.&lt;xx&gt;.us</td>
<td>libraries</td>
<td></td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: US City and County 3LDs

- US ccTLD structure for cities and counties
  - city and county governments use .ci and .co
  - other entities, e.g.
    - chambers of commerce, businesses, organisations
- Many states and most cities do *not* follow RFC 1480
  - e.g. kansas.gov lawrenceks.org opkansas.org
  - chicken-&-egg: uneducated sysadmins serving naïve public

<table>
<thead>
<tr>
<th>.4LD.3LD.SLD.ccTLD</th>
<th>Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>.ci&lt;(city)&gt;.(xx).us</td>
<td>city government</td>
<td>hyphenated city name</td>
</tr>
<tr>
<td>.co&lt;(county)&gt;.(xx).us</td>
<td>county government</td>
<td>hyphenated county name</td>
</tr>
<tr>
<td>.&lt;name&gt;.(locality).(xx).us</td>
<td>any other entity</td>
<td>locality is city or county</td>
</tr>
</tbody>
</table>
Domain Name System

Name Structure: UK ccTLD

- UK ccTLD: country code domain structure
  - most administered by Nominet www.nik.uk
  - structured SLD equivalent to gTLD (essentially a gSLD)
- All domain registrations *must* be at the 3rd level
  - strict rules for assignment in all but .co.uk and .or.uk
# Domain Name System

Name Structure: UK SLDs

<table>
<thead>
<tr>
<th>SLD.ccTLD</th>
<th>Use</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>.co.uk</td>
<td>commercial</td>
<td></td>
</tr>
<tr>
<td>.ltd.uk</td>
<td>private limited co.</td>
<td>algorithm based on co. name</td>
</tr>
<tr>
<td>.me.uk</td>
<td>individual person</td>
<td>must be individual person</td>
</tr>
<tr>
<td>.net.uk</td>
<td>ISP with IP block and AS#</td>
<td>based on ISP name</td>
</tr>
<tr>
<td>.org.uk</td>
<td>non-profit or public service</td>
<td></td>
</tr>
<tr>
<td>.plc.uk</td>
<td>public limited co.</td>
<td>algorithm based on co. name</td>
</tr>
<tr>
<td>.nic.uk</td>
<td>network information centre</td>
<td>only for use by UK NIC</td>
</tr>
<tr>
<td>.sch.uk</td>
<td>schools</td>
<td>⟨name⟩.⟨LEA⟩.sch.uk</td>
</tr>
<tr>
<td>.ac.uk</td>
<td>academic institutions</td>
<td>JANET UKERA</td>
</tr>
<tr>
<td>.gov.uk</td>
<td>government</td>
<td>Cabinet Office GSI</td>
</tr>
<tr>
<td>.mod.uk</td>
<td>military and related</td>
<td>Ministry of Defense DINSA</td>
</tr>
<tr>
<td>.nhs.uk</td>
<td>National Health Service</td>
<td>NHSIA</td>
</tr>
<tr>
<td>.police.uk</td>
<td>police services &amp; organisations</td>
<td>PITO</td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: Examples

• Example DNS names
  – deep-thought.eecs.ku.edu
    user 4LD  dept 3LD  inst SLD  gTLD
  – wopr.labs.gte.com
    user 4LD  div 3LD  corp SLD  gTLD
  – www.tik.ee.ethz.ch
    host 5LD  group 4LD  dept 3LD  inst SLD  Swiss ccTLD
  – www.comp.lancs.ac.uk
    host 5LD  dept 4LD  inst 3LD  UK-SLD  UK ccTLD
  – jam.es
    SLD  ccTLC  domain hack for name “James”
Domain Name System

Name Server Structure

- Example: resolve `www.amazon.com`
  - client queries a root server to find .com TLD DNS server
  - client queries .com TLD server to get `amazon.com` server
  - client queries amazon.com DNS server to get IP address
Domain Name System

Name Servers: Root

- **Root name servers** [www.root-servers.org]
  - responsible for resolution to proper TLD server
  - 13 root servers contracted identified as A – M

- **Configuration file** `named.root`
  - used by other DNS servers to locate root servers
  - contain list of A–M servers and their IP addresses
  - `ftp://ftp.internic.net/domain/named.root`
Domain Name System

Root Name Servers

A  VeriSign, Dulles VA
B  ISI, Marina Del Rey CA
C  Cogent, Herndon VA + 3
D  UMd, College Park MD
E  NASA Ames, Mountain View CA
F  ISC, 37 sites
G  US DOD NIC
H  ARL, Aberdeen MD
I  Autonomica/NORDUnet, Stockholm + 28 mirrors
J  VeriSign, Dulles VA + 16 mirrors
K  RIPE, London + 16 mirrors
L  ICANN, Los Angeles Ca
M  WIDE, Tokyo

selected mirrors shown
Domain Name System
Name Servers: TLD

- Root name server
- Top-level domain (TLD) server
  - responsible for all gTLD and ccTLD resolution
  - ICANN contracts each gTLD name server
  - nations responsible for administering or delegating ccTLD
Domain Name System

Name Servers: Authoritative

- Root name server
- Top-level domain (TLD) server
- Authoritative DNS server
  - DNS servers for organisation or corporate entity
  - provide *authoritative* hostname resolution
    - hosts within its own domain
    - e.g. ku.edu Web and mail servers
  - can be maintained by
    - organization itself (e.g. ku.edu)
    - outsourced to service provider (e.g. sterbenz.org)
Domain Name System

Name Servers: Local

- Root name server
- Top-level domain (TLD) server
- Authoritative DNS server
- Local name server (default name server)
  - maintained by each service provider or enterprise
  - resolves host DNS queries
  - acts as a proxy, forwards query into hierarchy
Domain Name System
Updating and Caching Records

• Root name server
  – contacted by local name server that can not resolve name
  – contacts authoritative name server if mapping not known
  – returns mapping to local name server

• When name server learns mapping it is *cached*
  – cache entries timeout (disappear) after some time

• TLD servers typically cached in local servers
  – thus root name servers not often visited

• Update/notify mechanisms [RFC 2136]
Domain Name System
Essential Tool: whois

- whois
  - client–server protocol
  - server listens to port 43
    - uses *whois database* (attempts underway to standardise)
    - originally stored information on people (Internet white pages)
    - now stores domain registry information

- Client usage (CLI)
  - whois ⟨SLD⟩.⟨TLD⟩
  - whois ⟨3LD⟩.⟨SLD⟩.⟨ccTLD⟩ some structured ccTLD: .uk
  - Web interface: http://www.internic.net/whois
    - only for gTLDs (but not .edu .mil .gov)

*try it*

- whois ku.edu
- whois lancs.ac.uk
Domain Name System

Essential Tool: dig

- **dig** (domain information grouper)
  - client program to query DNS server
  - more comprehensive information than nslookup

- **Usage (CLI)**
  - `dig <DNS-name>`
    - returns DNS records and IP address(es) of `<DNS-name>`
  - `dig -x <IP-address>`
    - reverse lookup: returns DNS name and records of `<IP-address>`

*try it*
```
dig www.ku.edu
```
```
dig -x 129.237.33.3
```
```
dig www.cnn.com
```
Domain Name System
Essential Tool: nslookup

- nslookup (name server lookup)
  - client program to query DNS server
  - use when dig not available (e.g. Windows boxes)

- Usage (CLI)
  - nslookup
    - returns DNS server of local host and enters interactive mode
  - nslookup <DNS-name>
    - returns IP address(es) of <DNS-name>
  - nslookup <IP-address>
    - reverse lookup: returns DNS name of <IP-address>

*try it*  
nslookup www.ku.edu  
nslookup 129.237.33.3  
nslookup www.cnn.com
Domain Name System
Web-Based Tools

- Many DNS and IP utilities on the Web
  - http://dnsstuff.com is particularly comprehensive
  - http://kloth.net/services

- Caveats
  - services aren’t being invoked locally

implication?
Domain Name System
Web-Based Tools

- Many DNS and IP utilities on the Web
  - http://dnsstuff.com is particularly comprehensive
  - http://kloth.net/services

- Caveats
  - services aren’t being invoked locally
    - e.g. ping will still test liveness but delay number is useless
    - e.g. traceroute will *not* trace route from local machine
    - local DNS information will not be obtained
  - pages come and go on the Web
Domain Name System
DNS Lookup Iterated Query Example

- DNS iterated lookup example
  - jpgs.ittc.ku.edu needs IP address of gaia.cs.umass.edu
  - ns1.ittc.ku.edu is local DNS server 129.237.125.220 obtained via DHCP
Domain Name System
DNS Lookup Iterated Query Example

- DNS iterated lookup example
  1Q. A? gaia.cs.umass.edu to local DNS server
  ns1.ittc.ku.edu 129.237.125.220

![Diagram showing DNS lookup process with queries to local DNS server and responses for gaia.cs.umass.edu and jpgs.ittc.ku.edu]
Domain Name System
Iterated Query Example

- DNS lookup example
  1Q. local DNS query
  2Q. A? gaia.cs.umass.edu to
      root nameserver in named.root
      h.rootservers.net 128.63.2.53

\[\text{local} \rightarrow \text{h.rootservers.net} 128.63.2.53\]

\[\text{jpgs.ittc.ku.edu} \quad \text{gaia.cs.umass.edu}\]
Domain Name System
Iterated Query Example

- DNS lookup example
  1Q. local DNS query
  2Q. A? gaia.cs.umass.edu to root nameserver in named.root
  2R. list of .edu NSs including NS a3.nstld.com
      A 192.5.6.32

1Q. local DNS query
2Q. A? gaia.cs.umass.edu to root nameserver in named.root
2R. list of .edu NSs including NS a3.nstld.com
   A 192.5.6.32

HTTP Error

DNS lookup example
1Q. local DNS query
2Q. A? gaia.cs.umass.edu to root nameserver in named.root
2R. list of .edu NSs including NS a3.nstld.com
   A 192.5.6.32

HTTP Error
Domain Name System
Iterated Query Example

• DNS lookup example
  1Q. local DNS query
  2QR. root DNS query/response
  3Q. A? gaia.cs.umass.edu to TLD a3.nstld.com 192.5.6.32
Domain Name System
Iterated Query Example

- DNS lookup example
  1Q. local DNS query
  2QR. root DNS query/response
  3Q. A? gaia.cs.umass.edu to TLD a3.nstld.com 192.5.6.32
  3R. list of .edu NSs including
      NS ns1.umass.edu
      A 128.119.100.21
      NS unix1.cs.umass.edu
      A 128.119.40.22

  note: contains L3 & L4 entries
  use most specific & authoritative entry

jpgs.ittc.ku.edu  gaia.cs.umass.edu
Domain Name System

Iterated Query Example

- DNS lookup example
  1Q. local DNS query
  2QR. root DNS query/response
  3QR. TLD DNS query/response
  4Q. A? gaia.cs.umass.edu to auth unix1.cs.umass.edu
      128.119.40.22

1Q. local DNS query
2QR. root DNS query/response
3QR. TLD DNS query/response
4Q. A? gaia.cs.umass.edu to auth unix1.cs.umass.edu
    128.119.40.22
Domain Name System
Iterated Query Example

- DNS lookup example
  1Q. local DNS query
  2QR. root DNS query/response
  3QR. TLD DNS query/response
  4Q. A? gaia.cs.umass.edu to auth unix1.cs.umass.edu
       128.119.40.22
  4R. A 128.119.245.12
Domain Name System
Iterated Query Example

- DNS lookup example
  1Q. local DNS query
  2QR. root DNS query/response
  3QR. TLD DNS query/response
  4QR. auth DNS query/response
  1R. A 128.119.245.12

• 1Q. local DNS query
• 2QR. root DNS query/response
• 3QR. TLD DNS query/response
• 4QR. auth DNS query/response
• 1R. A 128.119.245.12
Domain Name System
Iterated Query Example

- DNS lookup example
  1Q. local DNS query
  2QR. root DNS query/response
  3QR. TLD DNS query/response
  4QR. auth DNS query/response
  1R. A 128.119.245.12
  5Q. ping gaia.cs.umass.edu

Example:
1Q. local DNS query: jpps.ittc.ku.edu
2QR. root DNS query/response: root
3QR. TLD DNS query/response: .edu
4QR. auth DNS query/response: auth
1R. A 128.119.245.12
5Q. ping gaia.cs.umass.edu
Domain Name System
Iterated Query Example

- DNS lookup example
  1Q. local DNS query
  2QR. root DNS query/response
  3QR. TLD DNS query/response
  4QR. auth DNS query/response
  1R. A 128.119.245.12
  5Q. ping gaia.cs.umass.edu
  5R. ...reply from 128.119.245.12...
Domain Name System
Iterated Query Example

*Iterated query performance?*
Domain Name System
Iterated Query Performance

- Iterated query performance problems
  - multiple round trips per host query
  - at least 4, perhaps more

*Can we do better?*
Domain Name System
Recursive Queries

- Iterated query performance problems
- Recursive queries
  - name server resolves
    - reduces load on host
    - doesn’t reduce round trips
  - support optional
  - many servers don’t allow
    - load problems
    - security problems *Lecture SR*
Domain Name System
Recursive Queries

- Iterated query performance problems
- Recursive queries
  - name server resolves
    - reduces load on host
    - doesn’t reduce round trips
  - support optional
  - many servers don’t allow
    - load problems
    - security problems

Performance problem?
Domain Name System
Recursive Queries

- Iterated query performance problems
- Recursive queries
  - name server resolves
    - reduces load on host
    - doesn’t reduce round trips
  - support optional
  - many servers don’t allow
    - load problems
    - security problems
- Performance problem
  - round trip per resolution *solution?*
Domain Name System

Caching

- Iterated query performance problems
- Caching
  - each name server caches queries
  - significantly reduces
    - load on root and TLD servers
    - round trip latency typically 1 or 2
  - hosts cache mapping
    - no resolution for recent mappings
Domain Name System

Reverse Lookup

- DNS provides mapping function
  - hostname to IP address

*Problem: how to get hostname given IP address?*
Domain Name System

Reverse Lookup

- DNS provides mapping function
  - hostname to IP address
- Reverse lookup:
  - map IP address to hostname
Domain Name System
Reverse Lookup Example

- Reverse lookup example
  - `jjpgs.ittc.ku.edu`
    needs hostname of
    `128.119.245.12`
Domain Name System
Reverse Lookup Example

- DNS iterated lookup example
  1Q. PNTR? 12.245.119.128.in-addr.arpa to local DNS server
tns1.ittc.ku.edu 129.237.125.220

```
local
129.237.125.220
```

```
jpgs.ittc.ku.edu 128.119.245.12
```

```
ns1.ittc.ku.edu 129.237.125.220
```
Domain Name System
Reverse Lookup Example

• DNS lookup example
  1Q. local reverse DNS query
  2Q. PNTR? 12.245.119.128.in-addr.arpa to root nameserver in named.root
Domain Name System

Reverse Lookup Example

- DNS lookup example
  1Q. local reverse DNS query
  2Q. PNTR? 12.245.119.128.in-addr.arpa to root nameserver in named.root
  2R. NS figwort.arin.net A 192.42.93.32
      zone 128.in-addr.arpa

RIR: figwort.arin.net 192.42.93.32
Root nameserver: named.root
Local nameserver: jpgs.ittc.ku.edu

128.119.245.12
Domain Name System
Reverse Lookup Example

- DNS lookup example
  1Q. local reverse DNS query
  2QR. root reverse DNS query
  3Q. PNTR? 12.245.119.128.in-addr.arpa to
       RIR figwort.arin.net 192.5.6.32

Local Name Server

RIR: figwort.arin.net 192.42.93.32

-root

 Reverse Lookup Example

128.119.245.12
jags.itcc.ku.edu
Domain Name System
Reverse Lookup Example

- DNS lookup example
  1Q. local reverse DNS query
  2QR. root reverse DNS query
  3Q. PNTR? 12.245.119.128.in-addr.arpa to RIR figwort.arin.net 192.5.6.32
  3R. NS ns1.umass.edu A 128.119.100.21 zone 119.128.in-addr.arpa

1Q. local reverse DNS query
2QR. root reverse DNS query
3Q. PNTR? 12.245.119.128.in-addr.arpa to RIR figwort.arin.net 192.5.6.32
3R. NS ns1.umass.edu A 128.119.100.21 zone 119.128.in-addr.arpa
Domain Name System
Reverse Lookup Example

- DNS lookup example
  1Q. local reverse DNS query
  2QR. root reverse DNS query
  3QR. RIR reverse DNS query
  4Q. PNTR? 12.245.119.128.in-addr.arpa to
       ns1.umass.edu 128.119.100.21

  jpgs.ittc.ku.edu 128.119.245.12
  ns1.umass.edu 128.119.100.21
Domain Name System
Reverse Lookup Example

- DNS lookup example
  1Q. local reverse DNS query
  2QR. root reverse DNS query
  3QR. RIR reverse DNS query
  4Q. PNTR? 12.245.119.128.in-addr.arpa to ns1.umass.edu 128.119.100.21
  4R. PTR gaia.cs.umass.edu A 128.119.245.112
Domain Name System
Reverse Lookup Example

- DNS lookup example
  1Q. local reverse DNS query
  2QR. root reverse DNS query
  3QR. RIR reverse DNS query
  4QR auth reverse DNS query
  1R. PTR gaia.cs.umass.edu
       A 128.119.245.112

root

RIR

auth

local

jpgs.ittc.ku.edu

gaia.cs.umass.edu

128.119.245.12
Domain Name System
Name Structure: Alternative TLDs

• Alternative DNS root servers
  – generally used to bypass ICANN in providing new TLD

*Is this a good thing?*
Domain Name System
Name Structure: Alternative TLDs

- Alternative DNS root servers
  - generally used to bypass ICANN in providing new TLD
- Strongly opposed by the IAB [RFC 2826]
- Parts of Internet not accessible to all users
  - DNS names may conflict with one another
  - poor accessibility from commercial ISPs
    - most users don’t know how to configure DNS servers
  - poor record of financial stability of providers
    - many have gone belly-up and abandoned domain holders
- Examples
  .shop .love
Domain Name System
Name Structure: Chinese Pseudo-TLDs

• Chinese pseudo-TLDs
  – details on implementation are sketchy
    [www.lightbluetouchpaper.org/2006/03/01-new-chinese-tlds]
    [www.circleid.com/posts/chinas_new_domain_names_lost_in_translation]

• Server behaviour: effectively a split DNS
  – DNS servers within China add 3 new entries to resolve
    • but apparently not a alternate root server

• Client behaviour permits operation outside China
  – i-DNS plugin for MSIE rewrites URL resolvable in .cn
    • .中国  →  .cn
    • .公司  →  公司.cn
    • .网络  →  .网络.cn
Domain Name System

Name Structure: Chinese TLDs

- Chinese TLDs now approved by ICANN
  - administered by CNNIC cnnic.net.cn
  - DNS names may be ASCII and/or Chinese characters
  - content severely restricted by PRC government, e.g. not
    - “harm the glory and interests of the state”
    - “disseminate rumors, disturb social order, social stability”

<table>
<thead>
<tr>
<th>TLD</th>
<th>Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>.cn</td>
<td>China</td>
<td>ISO 3166 ICANN approved</td>
</tr>
<tr>
<td>.中国</td>
<td>ZhōngGuó (China) .cn parallel</td>
<td>now approved by ICANN</td>
</tr>
<tr>
<td>.公司</td>
<td>gōngsī (company) .com equivalent also resolves as .公司.cn</td>
<td>not approved by ICANN</td>
</tr>
<tr>
<td>.网络</td>
<td>wǎngluò (network) .net equivalent also resolves as .网络.cn</td>
<td>not approved by ICANN</td>
</tr>
</tbody>
</table>
Domain Name System
Name Structure: Alternate Int. TLDs

- Alternate Internationalised TLDs
- Example: Russian alternate TLD IDNs
  - administered by RegTime regtime.net using i-DNS.net
  - TLDs are Cyrillic transliterations of gTLDs and ccTLD
  - DNS names may contain Latin and/or Cyrillic characters

<table>
<thead>
<tr>
<th>TLD</th>
<th>Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>.ru</td>
<td>Russia</td>
<td>ISO 3166 ICANN approved</td>
</tr>
<tr>
<td>.рф</td>
<td>Russian Federation .ru equivalent</td>
<td>approved, ISO-Latin friendly</td>
</tr>
<tr>
<td>.ру</td>
<td>.ru transliteration .ru equivalent</td>
<td>could be confused with .py</td>
</tr>
<tr>
<td>.ком</td>
<td>.com transliteration .com.ru equiv.</td>
<td>not approved by ICANN</td>
</tr>
<tr>
<td>.нет</td>
<td>.net transliteration .net.ru equiv.</td>
<td>not approved by ICANN</td>
</tr>
<tr>
<td>.орг</td>
<td>.org transliteration .org.ru equiv.</td>
<td>not approved by ICANN</td>
</tr>
</tbody>
</table>
Domain Name System

Name Structure: Internationalised TLDs

- Current TLDs must be in ASCII [a..z,0..9,-]
- Some countries already register non-Latin TLDs
  - e.g. China and Russia
- ICANN instituting international domain names (IDNs)
  - wiki and testing at idn.icann.org, e.g:
    - http://مثال.اختبار/
    - http://例子.測試/
    - http://παράδειγμα.δοκιμή/
    - http://उदाहरण.परीक्षा/
    - http://실례.테스트/
    - http://пример.испытание/
Domain Name System
Name Structure: Internationalised TLDs

- Fast-track process instituted late 2009
  - alternative ccTLDs to nationalal authorities
- IDN ccTLDs restricted to non-Latin similar characters
  - to reduce chance phishing attacks
  - e.g. \( .рф \) instead of \( .py \) for \( .ru \) equivalent
    - \( .py \neq .py \)
  - may be difficult for some countries
    - Ukraine / Україна \( .ua \Rightarrow ? \)
## Domain Name System

### Name Structure: Internationalised TLDs

- **IDN ccTLDs with 1st-step fast-track approval**

<table>
<thead>
<tr>
<th>Country</th>
<th>ccTLD</th>
<th>Punycode</th>
<th>Script</th>
</tr>
</thead>
<tbody>
<tr>
<td>United Arab Emirates</td>
<td>.ae</td>
<td>xn--mgbaam7a8h</td>
<td>Arabic</td>
</tr>
<tr>
<td>دولة الإمارات العربية المتحدة</td>
<td>.امارات</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Egypt</td>
<td>.eg</td>
<td>xn--wgbh1c</td>
<td>Arabic</td>
</tr>
<tr>
<td>مصر</td>
<td>.مصر</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Russian Federation</td>
<td>.ru</td>
<td>xn--p1ai</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>Росийская Федерация</td>
<td>.рф</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saudi Arabia</td>
<td>.sa</td>
<td>xn--mgberp4a5d4ar</td>
<td>Arabic</td>
</tr>
<tr>
<td>المملكة العربية السعودية</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Domain Name System

## International TLDs: Cyrillic

<table>
<thead>
<tr>
<th>Country</th>
<th>ccTLD</th>
<th>Punycode</th>
<th>Script</th>
</tr>
</thead>
<tbody>
<tr>
<td>Russian Federation</td>
<td>.ru</td>
<td>xn--p1ai</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>Беларусь</td>
<td>.by</td>
<td>xn--90ais</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>Украина</td>
<td>.ua</td>
<td>xn--j1amh</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>България</td>
<td>.bg</td>
<td>xn--90ae</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>Република Сербия</td>
<td>.rs</td>
<td>xn--90a3ac</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>Kazakhstan</td>
<td>.kz</td>
<td>xn--80ao21a</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>Монгол Улс</td>
<td>.mn</td>
<td>xn--d1alf</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>Монгол Улс</td>
<td>.mn</td>
<td>xn--l1acc</td>
<td>Cyrillic</td>
</tr>
</tbody>
</table>
## Domain Name System

### International TLDs: Other Eurasian

<table>
<thead>
<tr>
<th>Country</th>
<th>ccTLD</th>
<th>Punycode</th>
<th>Script</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armenia</td>
<td>.am</td>
<td>xn--y9a3aq</td>
<td>Armenian</td>
</tr>
<tr>
<td>Հայաստան</td>
<td>.հայ</td>
<td></td>
<td>Armenian</td>
</tr>
<tr>
<td>Georgia</td>
<td>.ge</td>
<td>xn--node</td>
<td>Georgian</td>
</tr>
<tr>
<td>საქართველო</td>
<td>.გე</td>
<td></td>
<td>Georgian</td>
</tr>
<tr>
<td>Greece</td>
<td>.gr</td>
<td></td>
<td>Greek</td>
</tr>
<tr>
<td>Ελλάς</td>
<td>.ελ</td>
<td></td>
<td>Greek</td>
</tr>
</tbody>
</table>
## Domain Name System

### International TLDs: Sel. Chinese & East Asian

<table>
<thead>
<tr>
<th>Country</th>
<th>ccTLD</th>
<th>Translit.</th>
<th>Aurh/Script</th>
<th>Punycode</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>China 中国</td>
<td>.cn</td>
<td>Zhōngguó</td>
<td>CNNIC Hanzi (simpl.)</td>
<td>xn--fiqs8s xn--fiqz9s</td>
<td>in use</td>
</tr>
<tr>
<td></td>
<td>.中国</td>
<td>Zhōngguó</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.中國</td>
<td>Zhōngguó</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hong Kong 香港</td>
<td>.hk</td>
<td>Hoeng¹gong²</td>
<td>KHIRC Hanzi</td>
<td>xn--j6w193g</td>
<td>in use</td>
</tr>
<tr>
<td>Taiwan 中華民國</td>
<td>.tw</td>
<td>Táiwān</td>
<td>TWNIC Hanzi (trad.)</td>
<td>xn--kpry57d xn--kprw13d</td>
<td>ICANN approved</td>
</tr>
<tr>
<td></td>
<td>.台灣</td>
<td>Táiwān</td>
<td>Hanzi (simpl.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.台灣</td>
<td>Táiwān</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Japan 日本</td>
<td>.jp</td>
<td>Nihon</td>
<td>JIDNC/JPRS Kanji</td>
<td></td>
<td>JIDNC in Japan</td>
</tr>
<tr>
<td>Korea 대한민국</td>
<td>.kr</td>
<td>Hanguk</td>
<td>KISA Hangul</td>
<td>xn--3e0b707e</td>
<td>ICANN delegated</td>
</tr>
<tr>
<td></td>
<td>.한국</td>
<td>Hanguk</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Domain Name System

## International TLDs: Sel. Indian & South Asian

<table>
<thead>
<tr>
<th>Country</th>
<th>ccTLD</th>
<th>Translit.</th>
<th>Aurh/Script</th>
<th>Punycode</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>India</td>
<td>.in</td>
<td>Bhārat</td>
<td>NIXI / INRegistry Devanagari</td>
<td></td>
<td>in use</td>
</tr>
<tr>
<td></td>
<td>.भारत</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Singapore</td>
<td>.sg</td>
<td>Sinkapo</td>
<td>SGNIC Hanzi Tamil</td>
<td>xn--yfro4167o</td>
<td></td>
</tr>
<tr>
<td></td>
<td>대한민국</td>
<td>Cinkappūr</td>
<td></td>
<td>xn--c1chc0ea0b2g2a9gcd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.சிங்கப்பூர்</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Domain Name System

UDP vs. TCP

- DNS uses UDP or TCP for transport

*why?*
Domain Name System
UDP vs. TCP

- DNS uses UDP or TCP for transport
- UDP
  - generally used for small queries
    - name resolution and reverse name lookup
  - avoids delay of TCP 3-way handshake
  - automatically retries with TCP if answer too long
- TCP
  - generally used for large queries
    - e.g. zone transfers
## Domain Name System

### DNS Message Header Format

- **Header** [12b]
- **Query/response fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>12 bits</td>
</tr>
<tr>
<td>flags</td>
<td></td>
</tr>
<tr>
<td>QDCOUNT</td>
<td>question (variable number, typ=1)</td>
</tr>
<tr>
<td>ANCOUNT</td>
<td>answer RRs (variable number)</td>
</tr>
<tr>
<td>NSCOUNT</td>
<td>authority RRs (variable number)</td>
</tr>
<tr>
<td>ARCOUNT</td>
<td>additional RRs (variable number)</td>
</tr>
</tbody>
</table>

- **32 bits**
Domain Name System
DNS Message Header Format

- **ID**
  - 16-bit identifier matches query to response
- **Flags**
- **Count fields**

<table>
<thead>
<tr>
<th>ID</th>
<th>flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>QDCOUNT</td>
<td>ANCOUNT</td>
</tr>
<tr>
<td>NSCOUNT</td>
<td>ARCOUNT</td>
</tr>
</tbody>
</table>

- **flags**
  - (variable number, typ=1)
- **answer RRs**
  - (variable number)
- **authority RRs**
  - (variable number)
- **additional RRs**
  - (variable number)
## Domain Name System

### DNS Message Header Format

- **ID**
- **Flags**
  - QR: query/reply
  - opcode [4b]:
    - 0 = normal
    - 1 = inverse
    - 2 = stats req
  - AA: reply is authoritative
  - TC: truncated to 512B
  - RD: recursion desired
  - RA: recursion available
  - rcode: return code [4b]:
    - 1 = normal
    - 2 = server fail
    - 3 = name err
    - 4 = not impl
    - 5 = refused

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>12B</td>
<td>ID field in message header</td>
</tr>
<tr>
<td>QDCOUNT</td>
<td></td>
<td>Question count (variable number, typ=1)</td>
</tr>
<tr>
<td>NSCOUNT</td>
<td></td>
<td>Authority RRs count (variable number)</td>
</tr>
<tr>
<td>ARCOUNT</td>
<td></td>
<td>Additional RRs count (variable number)</td>
</tr>
<tr>
<td>QR</td>
<td>1</td>
<td>Query/Reply flag</td>
</tr>
<tr>
<td>Opcode</td>
<td>1</td>
<td>Opcode field</td>
</tr>
<tr>
<td>RCode</td>
<td>1</td>
<td>Return code field</td>
</tr>
<tr>
<td>ANCOUNT</td>
<td></td>
<td>Answer RRs count (variable number)</td>
</tr>
<tr>
<td>TRRR</td>
<td></td>
<td>Authority RRs (variable number)</td>
</tr>
<tr>
<td>ARRR</td>
<td></td>
<td>Additional RRs (variable number)</td>
</tr>
</tbody>
</table>
## Domain Name System
### DNS Message Header Format

- **ID**
- **Flags**
- **Count fields**
  - QDCOUNT: # questions
  - ANCOUNT: # answer RRs
  - NSCOUNT: # authority RRs
  - ARCOUNT: # additional RRs

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>32 bits</td>
</tr>
<tr>
<td>QDCOUNT</td>
<td>question (variable number, typ=1)</td>
</tr>
<tr>
<td>ANCOUNT</td>
<td>answer RRs (variable number)</td>
</tr>
<tr>
<td>NSCOUNT</td>
<td>authority RRs (variable number)</td>
</tr>
<tr>
<td>ARCOUNT</td>
<td>additional RRs (variable number)</td>
</tr>
</tbody>
</table>
Domain Name System
DNS Query Format

- **QDCOUNT typically = 1**
- Questions DNS queries
  - **QNAME**: query DNS name
    - sequence of \( \langle \text{count-byte} \rangle \langle \text{string} \rangle \) pairs terminated by 0 byte
    - \( \langle \text{count-byte} \rangle \) is number of characters in \( \langle \text{string} \rangle \)
    - variable length
    - no padding
  - **QTYPE**
  - **QCLASS**:  
    1 = Internet address

<table>
<thead>
<tr>
<th>ID</th>
<th>flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>QDCOUNT = 1</td>
<td>ANCOUNT = 0</td>
</tr>
<tr>
<td>NSCOUNT = 0</td>
<td>ARCOUNT = 0</td>
</tr>
</tbody>
</table>

QNAME

| QTYPE | QCLASS = 1 |

17 October 2016
Domain Name System
DNS Query Format

- **QDCOUNT typically = 1**
- **Questions: DNS queries**
  - QNAME
  - **QTYPE [16b]**
    - 1=A IP address
    - 2=NS name server
    - 5=CNAME canonical name
    - 12=PTR pointer record
    - 13=HINFO host info
    - 15=MX mail exch. record
    - 252=AXFR req for zone xfer
    - 255=ANY req all records
  - **QCLASS**

<table>
<thead>
<tr>
<th>ID</th>
<th>flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>QDCOUNT = 1</td>
<td>ANCOUNT = 0</td>
</tr>
<tr>
<td>NSCOUNT = 0</td>
<td>ARCOUNT = 0</td>
</tr>
<tr>
<td>QNAME</td>
<td></td>
</tr>
<tr>
<td>QTYPE</td>
<td>QCLASS = 1</td>
</tr>
</tbody>
</table>
Domain Name System

DNS Records

- DNS records: \( \langle \text{name, value, type, TTL} \rangle \)
- A record for DNS resolution
  - name: DNS hostname
  - type: IP address
- NS record for resolution to authoritative name server
  - name: domain
  - type: IP address of authoritative name server for domain
- CNAME record for DNS alias resolution
  - name: alias DNS name, e.g. www.ku.edu
  - type: canonical (real) name, e.g. raven.cc.ku.edu
Domain Name System

DNS Records

- DNS records: \{(name, value, type, TTL)\}
- MX record for mail server alias resolution
  - name: DNS hostname
  - type: IP address
- PTR record for pointer queries (reverse lookup)
  - name: reverse IP DNS name under .in-addr.arpa
  - type: DNS name
- HINFO record
  - name: DNS hostname
  - type: CPU type and operating system
Domain Name System
DNS Response Format

- 3 types of responses
  - multiple per response
- Answers
  - **ANCOUNT**: # answer RRs
  - answer RR(s)
- Authority NS answers
  - **NSCOUNT**: # authority RRs
  - authority RRs
- Additional RRs
  - **ARCOUNT**: # additional RR
  - additional RR(s)
Domain Name System
DNS Response Format

- **RRs: resource records**
  - **NAME:** DNS name
    - same format as QNAME
  - **TYPE** (same as QTYPE)
  - **CLASS** 1=Internet
  - **TTL:** time to live
    - #sec. RR to be cached
    - 0 = do not cache
  - **RDLENGTH:**
    - RDATA length in bytes
  - **RDATA:** resource data
    - response to query
    - depends on **TYPE**

```plaintext
<table>
<thead>
<tr>
<th>ID</th>
<th>flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>QDCOUNT = 0</td>
<td>ANCOUNT</td>
</tr>
<tr>
<td>NSCOUNT</td>
<td>ARCOUNT</td>
</tr>
<tr>
<td></td>
<td>NAME</td>
</tr>
<tr>
<td>TYPE</td>
<td>CLASS = 1</td>
</tr>
<tr>
<td>TTL</td>
<td></td>
</tr>
<tr>
<td>RDLENGTH</td>
<td>RDATA</td>
</tr>
</tbody>
</table>
```
Domain Name System
Record Insertion Example

- **Example**: just created startup “Example Networks, Inc.”
  - register name example.com at a .com **registrar**
  - provide registrar with auth name server information
    - primary: ns1.example.com  192.0.2.1
    - secondary: ns2.example.com  192.0.2.2
  - registrar inserts RR resource recs. into the .com TLD server:
    example.com, ns1.example.com, NS
    example.com, ns2.example.com, NS
    ns1.example.com, 192.0.2.1, A
    ns2.example.com, 192.0.2.2, A
  - to get email and Web servers to resolve add:
    www.example.com, 192.0.2.10, A
    mail.example.com, 192.0.2.20, MX
Network Layer

NL.4.6 Examples: IP, ICMP, and IPv6

NL.1 Network layer functions and services
NL.2 Network signalling paradigms
NL.3 Switches and packet structure
NL.4 Examples
  NL.4.1 PSTN
  NL.4.2 X.25 CONS
  NL.4.3 ISDN and Frame Relay
  NL.4.4 B-ISDN and ATM
  NL.4.5 MPLS
  NL.4.6 Internet: DNS, IP, ICMP and IPv6

NL.5 Fast datagram routers
Internet Protocol
Overview

• IP (Internet Protocol)
  – waist of the global information infrastructure (GII)
  – addressing and forwarding
  – version 4 (IPv4) [RFC 0791 / STD 0005]

• ICMP (Internet control message protocol)
  – signalling for IP
  – version 4 (ICMPv4) [RFC 0792 / STD 0005]
Internet Protocol
IP Packet Format

- **IP version number**
  - rest of packet
    - *version dependent*

- **Maintained by IANA**
  
  00 = reserved
  01 = TCP1 [RFC 675]
  02 = TCP2 [IEN 5]
  03 = TCP3 [IEN 21] (before TCP/IP split)
  04 = IPv4
  05 = ST [IEN 119] (experimental)
  06 = IPv6
  07 = TP/IX [RFC 1475] (proposed IPng)
  08 = PIP [RFC 1621] (proposed IPng)
  09 = TUBA [RFC 1347] (proposed IPng)
  15 = reserved
Internet Protocol
IPv4 Packet Format: Header Control Fields

- **IP version number** = 04
- **IHL**: header length
  - [32-b words]
- **TOS**: type of service
  - not generally used
- **TTL**: time to live
- **Protocol** to demux
  - TCP, UDP, etc.
- **Header checksum**
  - 1’s comp of 1’s comp Σ

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHL</td>
<td>header length</td>
</tr>
<tr>
<td>TOS</td>
<td>type of service</td>
</tr>
<tr>
<td>TTL</td>
<td>time to live</td>
</tr>
<tr>
<td>Protocol</td>
<td>protocol to demux</td>
</tr>
<tr>
<td>Source</td>
<td>source address</td>
</tr>
<tr>
<td>Destination</td>
<td>destination address</td>
</tr>
<tr>
<td>Options</td>
<td>options</td>
</tr>
<tr>
<td>Payload</td>
<td>payload</td>
</tr>
</tbody>
</table>

- **Connection**: 20B
Internet Protocol
IPv4 Packet Format: Payload Related Fields

- **Total length** of datagram
  - header and data [B]
- **Payload**

<table>
<thead>
<tr>
<th>04</th>
<th>IHL</th>
<th>TOS</th>
<th><strong>total length</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>fragment id</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>frag offset</td>
</tr>
<tr>
<td>TTL</td>
<td>protocol</td>
<td>header checksum</td>
<td></td>
</tr>
<tr>
<td></td>
<td>source address</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>destination address</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>options</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(= hl – 20B)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

payload

(= length – hl – 20B)
Internet Protocol
IPv4 Packet Format: Addresses

- 32 bit IP addresses
  - host or router interface
- **Destination address**
  - used by forwarding
- **Source address**
  
  *why?*

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td>IHL</td>
<td>4</td>
</tr>
<tr>
<td>08</td>
<td>TOS</td>
<td>8</td>
</tr>
<tr>
<td>08</td>
<td>Fragment length</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>Protocol</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Header checksum</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>Options</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>Payload</td>
<td>20</td>
</tr>
</tbody>
</table>

- 32 bit IP addresses
  - host or router interface
- **Destination address**
  - used by forwarding
- **Source address**
  
  *why?*
Internet Protocol
IPv4 Packet Format: Addresses

- 32 bit IP addresses
  - host or router interface
- Destination address
  - used by forwarding
- Source address
  may be needed for:
  - network to record senders
  - destination to reply

<table>
<thead>
<tr>
<th>04</th>
<th>IHL</th>
<th>TOS</th>
<th>total length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fragment id</td>
<td>flag</td>
<td>frag offset</td>
</tr>
<tr>
<td></td>
<td>TTL</td>
<td>protocol</td>
<td>header checksum</td>
</tr>
</tbody>
</table>

source address

destination address

options
(= hl – 20B)

payload
(= length – hl – 20B)
Internet Protocol

IP Addresses

• All interfaces that use IP have an address
  - host–network interfaces
    • many hosts have more than one
  - router ports

• 32-bit addresses
  - e.g. www.eecs.ku.edu to 129.237.87.18
  - example: www.eecs.ku.edu (resolved via DNS to)
    10000001 111101101 01010111 00010010

• Dotted decimal notation:
  - \( b_7 b_6 \cdot b_5 b_4 \cdot b_3 b_2 \cdot b_1 b_0 \) converted to decimal in \( 4 \times 8 \) b chunks
  - example: 129.237.87.18
Internet Protocol

IP Special Addresses

- **Localhost**
  - 0.0.0.0 during boot
  - 127/8 (127.x.x.x); generally 127.0.0.1

- **Link local communication** [RFC 3927]
  - 169.254/16 (169.254.X.X)

- **Broadcast**
  - 255.255.255.255 broadcast for this subnet
  - subnet address followed by all 1s: broadcast on subnet
  - generally restricted in use *why?*

- **Example for documentation**
  - 192.0.2/24 (192.0.2.X)
Internet Protocol
IPv4 Address Assignment

- IP addresses not randomly assigned to hosts
  - *every* table would have to contain *every* Internet host
  - *billions* of entries
Internet Protocol
IPv4 Address Assignment

- IP addresses not randomly assigned to hosts
  why?
Internet Protocol
IPv4 Address Hierarchy

- IP addresses assigned *hierarchically*
  - address aggregation dramatically improves scalability
  - forwarding table only needs to contain *network address*
  - routing advertisements only contain network address prefix

```
<table>
<thead>
<tr>
<th>IP Address</th>
<th>Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>200.23.16.4</td>
<td>X</td>
</tr>
<tr>
<td>200.23.16.45</td>
<td></td>
</tr>
<tr>
<td>200.23.16.12</td>
<td>2</td>
</tr>
<tr>
<td>199.31.0.4</td>
<td></td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>IP Address</th>
<th>Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>200.23.16</td>
<td>A</td>
</tr>
<tr>
<td>199.31.0</td>
<td>B</td>
</tr>
</tbody>
</table>
```

ISP_A
ISP_B
IP Addressing
Class-Based Addressing Hierarchy

- Divide IP address into 3 level hierarchy
  - class, network address, host address
  - byte aligned
  - simple IP address lookup (3 major cases)
  - class D for multicast addresses

<table>
<thead>
<tr>
<th>Class</th>
<th>Network</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1110</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>1111</td>
<td></td>
</tr>
</tbody>
</table>

Lecture NR
IP Addressing
Class-Based Addressing Problems

- Principle behind division
  - A: very large network providers
  - B: large organisations
  - C: LANs

- Reality: rigid structure
  - doesn’t match all organisations perfectly
  - doesn’t match many organisations well
    - especially class B: “three bears problem”

- Inefficient partitioning of address space
  - large fraction of unusable addresses
  - imminent exhaustion of IP address space led to...
IP Addressing

Subnets

16K networks × 64 subnets × 1024 hosts

<table>
<thead>
<tr>
<th>net</th>
<th>subnet</th>
<th>host</th>
</tr>
</thead>
</table>

• Subnets [RFC 0950 / STD 0005]
  – originally way to divide address class within organisation
  – example: 6b subnet to class B
  – subnet mask

• Hosts in subnet share upper IP address bits
  – natural to cluster similar IP addresses
  – efficient IP routing to subnet
  – switched layer 2 LAN with no layer 3 routing

Lecture LL
IP Addressing
Classless Addressing (CIDR)

- CIDR: classless interdomain routing [RFC 1519]
  - eliminate assignment of IP address blocks by class
  - $b_7.b_6.b_5.b_4.b_3.b_2.b_1.b_0/x$
    - x-bit prefix = arbitrary number of network bits
  - example: $11001000 \ 00010111 \ 0001000000000000$
    - $200.23.16.0/23$

- Service providers get variable IP block
  - based on need from RIR (or NIR)

- Significant improvement in IP address use
  - at the cost of significant increase in complexity of IP lookup
IP Address Assignment

Administrative Delegation

• IP address blocks originally assigned directly by IANA
  – as class A, B, or C
  – later managed by InterNIC operated by NSI

• Address assignment now delegated [RFC 2050]
  – IANA manages and allocates
    • http://www.iana.org/assignments/ipv4-address-space
  – regional internet registries (RIR) allocate within their range
    • ARIN, RIPE, APNIC, LACNIC, AfriNIC
    • Number Resource Organization www.nro.net
  – national internet registries operate within APNIC
  – local internet registries: typically ISPs
IP Address Assignment
Organisational Subnetworks

- Service provider assigns sub-blocks to subscribers
  - CIDR enables arbitrary subnetting at multiple levels
- Example
  - ISP assigned by 200.23.16.0/20 RIR (regional Internet registry)
  - ISP assigned subnets base on each organisation need

<table>
<thead>
<tr>
<th>ISP</th>
<th>Subnet Block</th>
<th>Prefix Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org0</td>
<td>11001000 00010111 00010000 00000000</td>
<td>200.23.16.0/20</td>
</tr>
<tr>
<td>Org1</td>
<td>11001000 00010111 00010010 00000000</td>
<td>200.23.18.0/23</td>
</tr>
<tr>
<td>Org2</td>
<td>11001000 00010111 00010100 00000000</td>
<td>200.23.20.0/23</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Org7</td>
<td>11001000 00010111 00011110 00000000</td>
<td>200.23.30.0/23</td>
</tr>
</tbody>
</table>

[Kurose–Ross p.336]
IP Address Assignment
Static Host IP Addresses

- Static IP address assignment
  - configuration file (e.g. Unix /etc/rc.config)

*Advantages and disadvantages?*
IP Address Assignment
Static Host IP Addresses

• Static IP address assignment
  – configuration file (e.g. Unix /etc/rc.config)

• Advantages
  – relative stable IP addresses better for incoming requests
  – allows systematic addresses for use by network admins

• Disadvantages
  – network administrators must manually assign addresses
  – users must manually configure their computers
IP Address Assignment

DHCP

- DHCP: dynamic host configuration protocol
  - [RFC 2131] – DHCPv6 [RFC 3315]
  - dynamic IP address assignment
  - automatic configuration of DNS servers and default routers
  - allows user to “plug into” network and it just works
    - most of the time

DHCP
IP Address Assignment

DHCP Address Assignment

- DHCP address assignment procedure
  - client broadcasts `discover` message
    - UDP datagram to 255.255.255.255 port 67
  - DHCP server broadcasts `offer` message
  - client broadcasts `request` message
  - DHCP server ACKs

`todo`: details and figure
IP Address Assignment
Strict Hierarchy with CIDR

- Forwarding table entries unique to networks

issue?

[Tier1X]

[199.31.0.0/16] B
[200.23.30.0/20] A

[Kurose–Ross p.347]
IP Address Assignment
Strict Hierarchy

- Forwarding table entries unique to networks
  - all organisations *must* change IP address with ISP change
IP Address Assignment

Loose Hierarchy

- Forwarding table entries not unique to networks
  - longest prefix is used for forwarding (most specific)
IP Addressing
Network Address Translation

- **NAT**: network address translation [RFC 2663, 3022]
  - translates public Internet address ↔ *private addresses*

**Benefits?**

```
org1 200.23.18.0/23
org2 172.16/12
org7 172.16/12

ISP_A 200.23.30.0/20

ISP_A 200.23.30.0/20

Tier1_X

<table>
<thead>
<tr>
<th>199.31.0.0/16</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>200.23.30.0/20</td>
<td>A</td>
</tr>
</tbody>
</table>
```
IP Addressing
Network Address Translation

• **NAT:** network address translation [RFC 2663, 3022]
  – translates public Internet address ↔ private addresses

• **Benefits**
  – provide multiple private addresses for one public address
    • e.g. home networks without buying multiple IP addresses
  – change private addresses without impacting ISP assignment
  – change ISP without impacting private addresses
  – private addresses not explicitly addressable
    • helps resist attacks
IP NAT

Private Address Space

- IP addresses reserved for private Internets [RFC 1918]
  - reserved by IANA

*why?*
IP NAT
Private Address Space

- IP addresses reserved for private Internets [RFC 1918]
  - reserved by IANA
  - private addresses must not conflict with public addresses
    - would prevent routing toward public Internet if conflict

<table>
<thead>
<tr>
<th>CIDR Prefix</th>
<th>Class</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/8</td>
<td>A</td>
<td>10.000.0.0 – 10.255.255.255</td>
</tr>
<tr>
<td>172.16/12</td>
<td>B</td>
<td>172.016.0.0 – 172.031.255.255</td>
</tr>
<tr>
<td>192.168/16</td>
<td>C</td>
<td>192.168.0.0 – 192.168.255.255</td>
</tr>
</tbody>
</table>
IP NAT Implementation

- NAT Implementation

*how to translate multiple private addresses to one public Internet address?*
IP NAT Implementation

• NAT Implementation
  – translate *multiple* private addr. to *one* public Internet addr.

• Hack using TCP ports
  – 16-bit port field allows almost 64K flows

• Outgoing translation
  – choose unused high port number new-port \( q \)
  – \( \langle \text{private-addr}_i, \text{port}_p \rangle \rightarrow \langle \text{public-addr}, \text{new-port}_q \rangle \)
  – store in NAT translation table

• Incoming translation
  – \( \langle \text{public-addr}, \text{new-port}_q \rangle \rightarrow \langle \text{private-addr}_i, \text{port}_p \rangle \)

*Disadvantages?*
IP NAT Example

- Private 10.0.0/24 network NATed behind 138.76.27.9

[Kurose–Ross p.354]
**IP NAT Example**

- Private **10.0.0/24** network NATed behind **138.76.27.9**
  
  1: **10.0.0.1** generates TCP seg. to server **128.119.40.186:80**

### Private to Public Mapping

<table>
<thead>
<tr>
<th>Private</th>
<th>Public</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1</td>
<td>138.76.27.9</td>
</tr>
<tr>
<td>10.0.0.2</td>
<td>138.76.27.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Public</th>
<th>Private</th>
</tr>
</thead>
<tbody>
<tr>
<td>5001</td>
<td>10.0.0.1:21</td>
</tr>
<tr>
<td>5007</td>
<td>10.0.0.2:3345</td>
</tr>
<tr>
<td>5042</td>
<td>10.0.0.1:21</td>
</tr>
</tbody>
</table>

ISP: **128.119.40.186**
IP NAT Example

- Private 10.0.0/24 network NATed behind 138.76.27.9
  1: 10.0.0.1 generates TCP seg. to server 128.119.40.186:80
  2: NAT chooses unused port, translates, adds to table
IP NAT Example

- Private 10.0.0/24 network NATed behind 138.76.27.9
  1: 10.0.0.1 generates TCP seg. to server 128.119.40.186:80
  2: NAT chooses unused port, translates, adds to table
  3: normal interaction with server
**IP NAT Example**

- **Private** 10.0.0.0/24 network NATed behind 138.76.27.9
  1: 10.0.0.1 generates TCP segment to server 128.119.40.186:80
  2: NAT chooses unused port, translates, adds to table
  4: normal interaction with server

<table>
<thead>
<tr>
<th>Public</th>
<th>Private</th>
</tr>
</thead>
<tbody>
<tr>
<td>138.76.27.9</td>
<td>10.0.0.1:3345</td>
</tr>
<tr>
<td>128.119.40.186</td>
<td>10.0.0.2:3345</td>
</tr>
<tr>
<td></td>
<td>10.0.0.1:21</td>
</tr>
<tr>
<td></td>
<td>128.119.40.186</td>
</tr>
</tbody>
</table>
IP NAT Example

- Private 10.0.0/24 network NATed behind 138.76.27.9
  1: 10.0.0.1 generates TCP seg. to server 128.119.40.186:80
  2: NAT chooses unused port, translates, adds to table
  5: normal interaction with server

<table>
<thead>
<tr>
<th>Public</th>
<th>Private</th>
</tr>
</thead>
<tbody>
<tr>
<td>5001</td>
<td>10.0.0.1:3345</td>
</tr>
<tr>
<td>5007</td>
<td>10.0.0.2:3345</td>
</tr>
<tr>
<td>5042</td>
<td>10.0.0.1:21</td>
</tr>
</tbody>
</table>
IP NAT Example

- Private 10.0.0/24 network NATed behind 138.76.27.9
  1: 10.0.0.1 generates TCP seg. to server 128.119.40.186:80
  2: NAT chooses unused port, translates, adds to table
  3: normal interaction with server
  6: NAT does reverse translation

<table>
<thead>
<tr>
<th>S: 128.119.40.186:80</th>
<th>D: 10.0.0.1:3345</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISP</td>
<td>128.119.40.186</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Public</th>
<th>Private</th>
</tr>
</thead>
<tbody>
<tr>
<td>5001</td>
<td>10.0.0.1:3345</td>
</tr>
<tr>
<td>5007</td>
<td>10.0.0.2:3345</td>
</tr>
<tr>
<td>5042</td>
<td>10.0.0.1:21</td>
</tr>
</tbody>
</table>

1: 10.0.0.1 generates TCP seg. to server 128.119.40.186:80
2: NAT chooses unused port, translates, adds to table
3: normal interaction with server
6: NAT does reverse translation
IP NAT

Disadvantages

- NAT disadvantages
  - eliminates end-to-end address transparency
    - NATed devices not addressable from outside
  - difficult to be a server or peer
    - ugly out-of-band hacks exist
  - violates protocol layer semantics
    - restricts to transport protocols using TCP/UDP socket semantic

- NATs were controversial
  - but have been widely deployed
  - and aren’t going to disappear
Internet Protocol
IPv4 Packet Format: Option Fields

- **Options**
  - optional packet processing
  - not typically in fast path
  - many service providers *ignore* options

- **Examples [IANA]**
  - security
  - source route
  - record route taken
  - router alert
  - timestamp

<table>
<thead>
<tr>
<th></th>
<th>IHL</th>
<th>TOS</th>
<th>total length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fragment id</td>
<td>flag</td>
<td>frag offset</td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td>protocol</td>
<td>header checksum</td>
<td></td>
</tr>
<tr>
<td>source address</td>
<td>destination address</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>options</strong></td>
<td>(= hl – 20B)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>payload</td>
<td>(= length – hl – 20B)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Internet Protocol
IPv4 Fragmentation

- Packet too big for (sub)network must be fragmented
  - generally dictated by link layer MTU (maximum transfer unit)
    - e.g. Ethernet MTU = 1500B

*Performance issues?*
Internet Protocol
IPv4 Packet Format: Fragmentation Fields

- **Fragment id**
  - common ID for fragments of a particular datagram

- **Flags** [3b]
  - 0: reserved
  - DF: may/don’t fragment
  - MF: last/more fragment

- **Fragmentation offset**
  - byte offset within datagram

<table>
<thead>
<tr>
<th>04 IHL TOS</th>
<th>total length</th>
</tr>
</thead>
<tbody>
<tr>
<td>fragment id</td>
<td>DF MF frag offset</td>
</tr>
<tr>
<td>TTL protocol</td>
<td>header checksum</td>
</tr>
<tr>
<td>source address</td>
<td></td>
</tr>
<tr>
<td>destination address</td>
<td></td>
</tr>
<tr>
<td>options (= hl – 20B)</td>
<td></td>
</tr>
<tr>
<td>payload (= length – hl – 20B)</td>
<td></td>
</tr>
</tbody>
</table>
Internet Protocol
IPv4 Fragmentation: Performance

- Packet too big for (sub)network must be fragmented
  - generally dictated by link layer MTU (maximum transfer unit)
    - e.g. Ethernet MTU = 1500B

- Significant performance penalty
  - delay in fragmentation and reassembly
  - buffer space to hold partially fragmented/reassembled

*Alternative?*
Internet Protocol

IPv4 Fragmentation: Path MTU Discovery

  - determine MTU of entire path
  - transport protocol uses to limit packet size
- Done automatically
  - sender sets DF flag (don’t fragment)
  - sender uses local link MTU or min(local-link-MTU,576)
- IP router at each hop
  - forwards if no fragmentation needed
  - discards and returns ICMP (3,4) message
- Sender retries with smaller MTU
Internet Control
ICMP

• Control messages for the Internet
  – carried in IP datagrams
  – not a well organised control plane
    • unlike the PSTN

• ICMP message
  – type, code plus first 8 bytes of IP datagram causing error
  – http://www.iana.org/assignments/icmp-parameters
  – http://www.iana.org/assignments/icmpv6-parameters

• Not a well-organised control plane (unlike the PSTN)
  – rather a set of messages
  – used by other protocols and tools
# Internet Control

ICMP Selected Message Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>echo reply (used by ping)</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>destination unreachable</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>source quench (intended for congestion control)</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>redirect</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>echo (used by ping)</td>
</tr>
<tr>
<td>9</td>
<td>X</td>
<td>router advertisement [RFC 1256]</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>router solicitation</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>TTL exceeded</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>timestamp</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>timestamp reply</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>information request</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>information reply</td>
</tr>
</tbody>
</table>
# Internet Control

ICMP Selected Unreachable Codes

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>network unreachable</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>host unreachable</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>protocol unreachable</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>port unreachable</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>fragmentation needed and don’t fragment was set</td>
</tr>
</tbody>
</table>
Internet Control
Essential Tool: ping

• ping used to test liveness of remote host
  – sends ICMP (8,0) echo message
  – destination replies with ICMP (0,0) echo reply message

• Client usage (CLI)
  – ping ⟨dnsname⟩
    • DNS first resolves hostname to IP address
  – ping ⟨address⟩

  try it  ping www.eecs.ku.edu
         ping www.comp.lancs.ac.uk
Internet Control
Essential Tool: traceroute

- traceroute used to map path HBH through network
  - sends UDP datagrams with increasing TTL and unlikely port
  - each hop drops and returns ICMP (11,0) TTL exceeded
    - many network service provider do not reply
      traceroute times out and shows “*” for these hops
  - done 3 times per hop; source computes and averages RTT
  - destination replies with ICMP (3,3) port unreachable

- Client usage (CLI)
  - traceroute \langle dnsname\rangle
    - DNS first resolves hostname to IP address
  - traceroute \langle address\rangle
    - try it traceroute www.eecs.ku.edu
    - traceroute www.comp.lancs.ac.uk
Internet Protocol
IPv6 Motivation

- IPv4 address space exhaustion
  - CIDR reduced problem with more efficient allocation
  - NATs reuse addresses in edge networks
    - perhaps eliminating the address-motivation for IPv6 completely
- IPv4 didn’t provide explicit support for QoS
  - other than TOS field
- IPv4 was not designed for efficient processing
  - options
  - variable length header
- IPv6 chosen among a number of proposals
Internet Protocol

IPv6 Overview

- IPv6 (Internet Protocol version 6)
  [RFC 2460] draft standard
- ICMPv6 (Internet control message protocol vers. 6)
  - signalling for IPv6 [RFC 4443]
- Support for QoS with flow label
- More efficient structure for high-speed processing
  - fixed 40B header
  - no options
  - but next header can contain options rather than L4 header
  - fragmentation not allowed
Internet Protocol: IPv6
Packet Format: Header Control Fields

- IP version number = 06
- Traffic class [8b]
- Flow label [20b]
  - flow identifier for soft state
    *Lecture TQ*
- Next header
  - protocol # to demux
  - extension header
- Hop limit
  - used as IPv4 TTL
- note: *no header checksum*
Internet Protocol: IPv6
Packet Format: Payload Related Fields

- IP version number = 06
- Payload length [B]
- Payload

<table>
<thead>
<tr>
<th></th>
<th>class</th>
<th>flow label</th>
</tr>
</thead>
<tbody>
<tr>
<td>payload length</td>
<td>next hdr</td>
<td>hop limit</td>
</tr>
</tbody>
</table>

(source address)

(= payload length)
Internet Protocol: IPv6
Packet Format: Addresses

- 128 bit IP addresses
  \[2^{128} = 3 \times 10^{38} = \]
  \[7 \times 10^{23}\text{ addr/m}^2\text{ on earth}\]
  \[\approx 6.022 \times 10^{23}\text{ (Avagadro)}\]
- Structure in allocation
  - reduces usable number
  - IANA administers

<table>
<thead>
<tr>
<th>06</th>
<th>class</th>
<th>flow label</th>
</tr>
</thead>
<tbody>
<tr>
<td>payload length</td>
<td>next hdrl</td>
<td>hop limit</td>
</tr>
</tbody>
</table>

- source address
- destination address
- payload
  (= payload length)
Internet Protocol
IPv4 → IPv6 Transition

If IPv6 replaces IPv4, how to transition?
Internet Protocol
IPv4 → IPv6 Transition

• If IPv6 replaces IPv4, how to transition?
  – “flag day” (won’t happen)
  – tunneling:
    IPv6 carried in IPv4 datagrams among IPv4 routers
  – dual stack
    • most IP routers now support both IPv4 and IPv6
    • permits slow transition with dual addressing
Network Layer

NL.5  Fast Datagram Routers

NL.1  Network layer functions and services
NL.2  Network signalling paradigms
NL.3  Switches and packet structure
NL.4  Examples
NL.5  Fast datagram routers
Fast Datagram Routers

Motivation

• Connection-oriented fast packet switching
  – emerged in ATM standards, but ATM failed
• IP became waist of global network infrastructure
  – increased processing capability enabled fast IP lookups
  – apply fast packet switching to IP datagram forwarding
Fast Datagram Switches

Architecture

- Fast packet switch core
- Input processing
  - IP lookup
  - packet classification
- Output processing
  - packet scheduling
  - fair queueing

- Input processing
- Output processing
- Management
- Routing and Signalling

- Prefixes
- Input processor
- Switch fabric
- Link
- Header update
- Classify

Network layer classification and packet scheduling.
Fast Datagram Switches

Throughput

- Packet processing rate critical [packets/s]
  - packet processing must sustain at least average rate
  - critical path must sustain peak line rate for min size packets

More in EECS 881
Fast Datagram Switches
Software IP Lookup

- Longest prefix match
- Critical parameters
  - worst case lookup time
    - brute force: $O(\log_2 n)$
    - $n$ hundred thousands
  - memory required
  - forwarding table update time

<table>
<thead>
<tr>
<th>prefix</th>
<th>$P_{out}$</th>
<th>$f_{state}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>001*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>101*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10100*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>111*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

101 011 01 payload
$P_{out}$ 101 011 01 payload
- hop count checksum fix
Fast Datagram Switches
Software IP Lookup Example: Trie

- Many algorithms
- Example: trie
  - sparse binary tree
  - valid prefixes are root
  - lookup time $O(a)$
    - $a =$ number of address bits
Fast Datagram Switches

Hardware IP Lookup

- Ternary CAM
  - 1, 0, X (don’t care)
  - expensive and complex
    - relative to RAM
- Simultaneous match
  - lookup time constant
    - $O(1)$
Fast Datagram Switches
Packet Classification

• Packet classification determines how packet treated
  – QOS or diffserv
  – policy based routing
  – security and DOS protection (e.g. firewalls)
  – layer 4 and 7 switching
  – active network processing

• Before queueing to meet most stringent delay class

More in EECS 881
Fast Datagram Switches
Packet Classification

- Multidimensional classification
  - policies may be hierarchal or overlap
  - precedence rules needed
- More complex than longest prefix match
- Hardware and software implementation tradeoffs
Further Reading

Network Layer

Further Reading

Communication Networks

Acknowledgements

Some material in these foils comes from the textbook supplementary materials:
