S-38.188 - Computer Networks - Spring 2005

Problem

- **Aim:** Build networks connecting millions of users around the globe
  - also spanning networks based on *any* technology

- **Problems:** heterogeneity and scalability
  - bridges can be used to connect different LANs (extended LANs)
  - heterogeneity: need to support different LANs, point-to-point technologies, switched networks, different addressing formats
  - scalability: addressing (management and configuration) and routing must be able to handle millions of hosts
  - in this lecture, we examine the (original) IP protocol, IP addressing, packet forwarding
Outline

• Internet architecture
• IP service model
• IP forwarding
• Address translation (ARP)
• Automatic host configuration (DHCP) and error reporting (ICMP)
• Virtual Private Networks (VPNs)

IP Internet

• Terminology
  – link = network based on LAN or extended LAN technology
  – internet = “network of networks”
  – Internet = internet using IP
  – IP = Internet Protocol, current version IPv4 (IP Version 4)
  – node = something implementing IP
  – router = node connecting networks, forwarding packets on other’s behalf
  – host = node that is not a router
  – interface = an node’s attachment to a link
  – IP address = IP identifier for an interface
    • If a node has multiple interfaces, it will have multiple IP addresses
  – Packet = IP header + payload sent on a link
  – Datagram = IP header + application layer payload (“service data unit”)
    (unfragmented or re-assembled)
IP Internet

- Sample internetwork

\[ \begin{aligned}
\text{Network 2 (Ethernet)} & \hspace{1cm} \text{Network 3 (FDDI) (point-to-point)} \\
\text{Network 1 (Ethernet)} & \\
\end{aligned} \]

IP design principles

- Cerf and Kahn’s internet design principles (1974)
  - minimalism, autonomy
    - no internal changes required to interconnect networks
    - network is self-configuring as much as possible
    - network can survive node and link failures
  - best effort service model
    - packets are not offered any guarantees
    - simplifies packet processing
  - stateless routers
    - network does not store information of any “connections” or user state
    - routers forward autonomous packets
  - decentralized control
    - enables high survivability (in presence of, e.g., link or node failures)
Internet architecture

- Internet architecture has only 4 layers
  - L7 (Application layer): FTP, HTTP, ...
  - L4 (Transport layer): TCP (reliable byte transfer) and UDP (unreliable datagram delivery) provide logical channels to applications
  - L3 (IP layer): IP protocol interconnects multiple networks into a single logical network
  - L1/2 ("Link" layer): wide variety of LAN and point-to-point protocols

- Internet architecture features
  - Does not imply strict layering
  - IP defines a common way for exchanging packets among widely differing networks
  - "Hour glass"-shape

- Aim: heterogeneity and scalability
Internet Protocol “Suite”

Application protocols:
- SMTP
- LDAP
- HTTP
- SIP/SDP/SAP
- POP3/IMAP4
- FTP
- NFS
- RTSP
- TELNET
- X11
- RTP

Transport layer:
- TCP
- UDP

Internetworking layer:
- IP

Mapping:
- IPCP
- ARP
- PPP
- 802
- ATM

Physical networks:
- ISDN
- Ethernet
- Fiber
- POTS
- FDDI
- Copper
- GSM
- WLAN
- 155 / 622 Mbit/s

IP protocol stack
IETF (Internet Engineering Task Force)

- Majority of Internet development (standardization) done by IETF
  - offers a mutual forum for the development of the Internet to vendors, users, researchers, service providers and network managers
  - develops architectures and protocols for solving technical issues
  - gives recommendations on the use of protocols
  - performs dissemination of the recommendations of IRTF (Internet Research Task Force) which is responsible for long term development of Internet
  - IETF requires always working implementations before any protocol specification is accepted as a standard (*we believe in running code*)
- Working methods
  - has meetings 3 times a year
  - work conducted within working groups (> 100 working groups)
    - joining a group done via e-mail to the mailing list
    - working groups belong to 8 different areas
      - Internet, Routing, Operations & Mgmt, Transport, Application, Security, General
      - New: Real-time Applications and Infrastructure (RAI)
- Work reported in Internet drafts and RFCs (Request for Comments)
  - Internet drafts have no official status (expire after 6 months), serve as basis for RFCs
  - Not all RFCs are standards (Informational, Best Current Practice, ...)
  - http://www.ietf.org

Outline

- Internet architecture
- IP service model
- IP forwarding
- Address translation (ARP)
- Automatic host configuration (DHCP) and error reporting (ICMP)
- Virtual Private Networks (VPNs)
Service model

- Idea in the Internet service model:
  - Make it undemanding enough that IP can be run over anything
    • Minimal requirement: carry packets of some size
  - Avoid doing "smart" (= complex!) things in the IP network
    • Applications may not need it anyway
    • Just Fragmentation required to deliver packets end-to-end
  - One of the major reasons for the success of IP technology

- Service model consists of 2 parts:
  - Model for data delivery
  - Addressing scheme

Data delivery model

- Data delivery in Internet
  - IP network connectionless (datagram-based)
  - IP network offers best-effort delivery (unreliable service)
    • packets may be lost (they often are)
    • packets may be delivered out of order (but usually are not)
    • duplicate copies of a packet may be delivered (but usually are not)
    • packets can be delayed for a long time
    • "intelligence" implemented at the end hosts
  - datagram format (next slide)
IP datagram format details

- Format aligned at 32 bit words
  - simplifies packet processing in sw
- Fields
  - Version: currently version 4 (6 is coming)
  - HLen: header length, 32 bit words (min 5)
  - TOS: type of service, used to give priorities to packets (QoS lecture)
  - Length: datagram+header length, in bytes
  - 2nd word for fragmentation/reassembly
  - TTL: time to live, not times packet allowed to be forwarded (not hops), default 64, detects packets caught in routing loop
  - Protocol: identifies upper layer protocols, TCP (6), UDP (17)
  - Checksum: erroneous packets discarded
  - Addresses: global Internet addresses
  - Options: rarely used

<table>
<thead>
<tr>
<th>Version</th>
<th>HLen</th>
<th>TOS</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fragmentation and reassembly

- Each network has an MTU (Maximum Transfer Unit)
  - Ethernet 1500 bytes, FDDI 4500 bytes, PPP 512 bytes
- Strategy
  - fragment when necessary (MTU < datagram length)
  - try to avoid fragmentation at source host
    - host sets datagram size equal to MTU of home network
    - for ATM MTU based on CS-PDU size (not cell size)
    - hosts may determine Path MTU
  - fragments are self-contained datagrams
    - each fragment contains a common identifier in Identi field
    - Flags (M-bit) and Offset used to guide fragmentation process
      - Offset measured in 8B units
    - fragmented packet can be again re-fragmented
  - reassembly performed only at destination host
  - reassembly does not try to recover from lost fragments
Fragmentation/reassembly example

- Original message 1400B + 20B header

IP addressing

- Properties
  - globally unique, 32 bits
  - hierarchical: network + host
  - address identifies interface
    - end host has 1 interface
    - router has many interfaces
  - IP address ≠ domain name

- Original classful addressing
  - class A, B and C networks
  - defines different sized networks
  - idea: small nof WANs, modest nof campus networks, large nof LANs

- Dot Notation
  - 32 bit addresses represented as group of 8 bit integers
  - e.g., 10.3.2.4, 128.96.33.81
Outline

• Internet architecture
• IP service model
  • IP forwarding
  • Address translation (ARP)
  • Automatic host configuration (DHCP) and error reporting (ICMP)
  • Virtual Private Networks (VPNs)

Routing Concept

IP layer routing all the way

Network internal routing to next hop
IP forwarding (1)

• Some terminology:
  
  – forwarding:
    • process of taking a packet from input interface, and …
    • based on the contents of the **forwarding table**, determining the correct output interface for the packet
  
  – routing:
    • process of constructing forwarding tables that enable efficient routing of traffic in the network (lecture 4)

IP forwarding (2)

• Preliminaries
  
  – Every datagram contains destination’s address
  – Every node has a forwarding table
    • normal hosts with one interface have only **default router** configured
    • routers maintain forwarding tables with multiple entries (constructed via routing process)
    • forwarding table maps network number into next hop router number or local interface number

• Strategy
  
  – Any node receiving a packet (router/host) checks destination **network address** of datagram and …
    • if directly connected to destination network, then forward to host
      • need to map IP address to physical LAN address \( \Rightarrow \) **ARP**
    • if not directly connected to destination network, then forward to next hop router
IP forwarding example

- H1 → H3: forwarding on the same network
- H1 → H8: via R1 and R2

### Forwarding table of H1

<table>
<thead>
<tr>
<th>NetworkNum</th>
<th>NextHop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R3</td>
</tr>
<tr>
<td>2</td>
<td>R1</td>
</tr>
<tr>
<td>3</td>
<td>Interface 1</td>
</tr>
<tr>
<td>4</td>
<td>Interface 0</td>
</tr>
</tbody>
</table>

### Forwarding table of R2

<table>
<thead>
<tr>
<th>NetworkNum</th>
<th>NextHop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interface 0</td>
</tr>
<tr>
<td>2</td>
<td>R1</td>
</tr>
<tr>
<td>3</td>
<td>Interface 1</td>
</tr>
<tr>
<td>4</td>
<td>Interface 0</td>
</tr>
</tbody>
</table>

Routers vs. bridges

- **Bridge (+/-)**
  + bridge operation simple, requires less processing
  + transparent (no configuration needed when new nodes added to LAN)
    - restricted topology (forwarding determined by a spanning tree)
    - LANs use a flat addressing space (no hierarchical network structure)
- **Router (+/-)**
  + arbitrary topologies, enables use of efficient routing algorithms for distributing traffic (helps traffic management)
  + hierarchical addressing enables scalability:
    - scalability requires minimization of address info stored in routers
    - routing based on network numbers ⇒ forwarding tables contain info on all networks, **not** all nodes
  - requires IP address configuration
  - packet processing more demanding
- **Summary**: bridges do well in small (~ 100 hosts) networks while routers are used in large networks (1000s of hosts)
Outline

• Internet architecture
• IP service model
• IP forwarding
• Address translation (ARP)
• Automatic host configuration (DHCP) and error reporting (ICMP)
• Virtual Private Networks (VPNs)

Address translation

• Earlier, we skipped the part what to do when router/host notes that it is connected directly to the network where an arriving packet is destined.

• Need to map IP addresses into physical LAN addresses
  – destination host
  – next hop router

• Techniques
  – encode physical LAN address in host part of IP address
    • not scalable
  – table-based (maintain IP address, PHY address pairs)
    • ⇒ ARP
ARP details

- ARP (Address Resolution Protocol)
  - utilizes LAN’s broadcast capabilities
  - each node maintains table of IP to physical LAN address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical LAN address

- ARP request contains also source addresses (physical and IP)
  - all “interested” parties can learn the source address

- Node (host/router) actions:
  - table entries timeout in about 10 minutes
  - if node already has an entry for source, refresh timer
  - if node is the target, reply and update table with source info
  - if node not target and does not have entry for the source, ignore source info

- ARP info can be incorporated in the contents of forwarding table

ARP Packet Format

- Request Format
  - HardwareType: type of physical network (e.g., Ethernet)
  - ProtocolType: type of higher layer protocol (e.g., IP)
  - HLen & PLen: length of physical and upper layer addresses
  - Operation: request or response
  - Physical/IP addresses of Source and Target
Classical IP over ATM

- Problem: ARP uses broadcast, but
  - ATM is connection oriented
    (no broadcasting)
- Solution:
  - LANE not useful if nodes spread
    over large area
  - Classical IP over ATM and
    ATMARP server
- Classical IP over ATM
  - group nodes of ATM network into several LIS (Logical IP Subnet)
  - nodes in same LIS have same IP network number
  - nodes in same LIS communicate with each other directly using ATM (AAL5)
  - nodes in different LIS communicate via IP router
  - can connect large nof hosts and routers to a big ATM network without
    assigning addresses from same IP network
  - scalability: ATMARP handles smaller nof hosts

ATMARP

- ATMARP server
  - resolves ATM addresses to IP addresses (like ARP translates ETH to IP)
  - does not rely on broadcast
- Functionality
  - each node in a LIS sets up VC to ATMARP and registers (sends own (ATM, IP) address pair)
  - ARP server builds table of (ATM, IP) address pairs for all registered nodes
  - nodes make queries to ARP server
  - nodes can keep cache of (ATM, IP) address mappings
    - like in traditional ARP
  - VC to a destination can be kept alive as long as needed
- Note! In Classical IP over ATM two nodes in same ATM network cannot
  communicate directly if they are in different subnets.
Outline

- Internet architecture
- IP service model
- IP forwarding
- Address translation (ARP)
- Automatic host configuration (DHCP) and error reporting (ICMP)
- Virtual Private Networks (VPNs)

Network management and scalability

- Mechanisms in IP that enable heterogeneity and scalability
  - heterogeneity:
    - best effort service model that makes minimal assumptions on underlying network capabilities
    - common packet format, fragmentation used for networks with different MTUs
    - global address space (ARP maps physical addresses to IP)
  - scaling:
    - hierarchical aggregation of routing information (network/host number)
    - above focuses on minimizing network state info in devices
- Important also to consider management complexity as network grows
  - example: configuration of IP addresses via DHCP
Need for automatic configuration

- IP addresses need to be reconfigurable
  - Ethernet addresses hardwired onto the network adapter
  - IP address consists of network and host part
  - hosts can move between networks ⇒ host gets new address in each network
- Need for automated host configuration
  - hosts need other configuration info, e.g., the default router
  - manual configuration impossible (too much work and errors)
  - ⇒ Dynamic Host Configuration Protocol (DHCP)
- DHCP server
  - at least one DHCP server for each administrative domain
  - centralized repository for configuration info
  - two operation modes:
    - administrator chooses host addresses and configures them to DHCP
    - DHCP manages the addresses by allocating addresses dynamically from a pool of available addresses (more sophisticated)

DHCP operation

- Server discovery: host sends DHCPDISCOVER msg to IP broadcast address (255.255.255.255)
- Msg broadcasted only on same network
- If server on same network, host receives its IP address
- If not, msg picked up by DHCP relay agent
- Relay agent knows address of DHCP server, forwards the msg to DHCP server and host receives its IP address
- Use of DHCP relay agent makes it possible to have fewer DHCP servers (relay agent configuration simpler than DHCP server configuration)
DHCP packet format, etc.

- Packet format
  - carried on top of UDP
  - based on older protocol BOOTP (unused fields)
  - client puts its hardware address in chaddr
  - DHCP server puts client’s IP address in yiaddr
  - other info placed in options (default router, subnet mask, DNS server)

- Handling dynamic addresses
  - problem: hosts may not return addresses (host crashes, is turned off, ...)
  - DHCP addresses only “leased” for a period of time
  - if lease is not refreshed, address placed back in pool

- DHCP improves manageability of network

Internet Control Message Protocol

- ICMP
  - carried in IP packets, but is functionally part of IP

- Functionality
  - Error conditions
    - notification about reasons for non-delivery of datagrams
    - time exceeded, re-assembly timeout, fragmentation needed
    - destination unreachable (network, host, port, protocol)
  - Routing support (for hosts)
    - router redirect, router selection / advertisement, ...
  - Diagnostics support
    - echo request/reply, traceroute, timestamp, ...
  - Extensions for new IP features
    - IPv6, mobility
Diagnostic Tools (1)

- **Ping**
  - check reachability / availability of destination node
  - sends an ICMP echo request
  - destination responds with an echo reply

```
rini>ping -sv presto.cs.tu-berlin.de
64 bytes from presto.cs.tu-berlin.de (130.149.25.1): icmp_seq=0. time=23. ms
64 bytes from presto.cs.tu-berlin.de (130.149.25.1): icmp_seq=1. time=27. ms
64 bytes from presto.cs.tu-berlin.de (130.149.25.1): icmp_seq=2. time=42. ms
64 bytes from presto.cs.tu-berlin.de (130.149.25.1): icmp_seq=3. time=16. ms
64 bytes from presto.cs.tu-berlin.de (130.149.25.1): icmp_seq=4. time=19. ms
^C
```

---presto.cs.tu-berlin.de PING Statistics-----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms) min/avg/max = 16/25/42

Diagnostic Tools (2)

- **Traceroute (Windows: tracert.exe)**
  - datagrams sent to target
    - addressed to "unlikely" destination port
  - with increasing TTL
    - starting from TTL=1
    - TTL decremented by 1 at each router
    - when TTL reaches zero: router sends ICMP time exceeded message back to the source
    - if target reached, ICMP destination port unreachable comes back
  - provides step by step the route to the destination
    - assumption: the route does not change while tracing
Diagnostic Tools (3)

- Traceroute example

```
ruin->traceroute presto.cs.tu-berlin.de
traceroute to presto.cs.tu-berlin.de (130.149.20.22): 1-30 hops, 38 byte packets
1  irizfn.informatik.uni-bremen.de  (134.102.224.250)   4.9 ms  23.7 ms  22.5 ms
2  Uni-Bremen1.Win-IP.DFN.DE       (188.1.3.57)        1.8 ms   1.7 ms   2.2 ms
3  ZR-Hamburg1.Win-IP.DFN.DE       (188.1.3.53)        5.0 ms   5.3 ms   5.3 ms
4  ZR-Berlin1.Win-IP.DFN.DE        (188.1.144.17)       11.7 ms  37.3 ms  10.9 ms
5  TU-Berlin1.Win-IP.DFN.DE        (188.1.162.38)       14.2 ms  13.7 ms  12.8 ms
6  KR-TU-Berlin1.Win-IP.DFN.DE     (188.1.1.110)       13.0 ms  14.0 ms  15.0 ms
7  130.149.6.3                     (130.149.6.3)      15.5 ms 14.4 ms  23.3 ms
8  sombrero.cs.tu-berlin.de (130.149.17.8)     34.6 ms  24.1 ms  25.8 ms
9  presto.cs.tu-berlin.de  (130.149.25.1)     21.9 ms  32.4 ms  15.9 ms
```
Virtual private networks (VPN)

- Problem:
  - group of isolated networks
  - geographically distant from each other
  - need to connect different networks together into a "private" network
  - e.g., company with many branch offices

- Solution:
  - VPN
  - connect individual networks together through a public network

- Technologies
  - leased virtual circuits from an ATM network operator or Frame Relay operator
  - possible with IP, but requires IP tunneling

VPN and IP tunneling

- Problem with IP
  - not possible to connect to Internet via router without the whole Internet also knowing about your network

- Tunneling
  - virtual point-to-point link btw. two nodes separated by arbitrary nof networks
  - created in R1 by providing it with address of R2
  - R1 encapsulates original packet in a new packet addressed to R2
  - packet forwarded normally inside IP network
  - R2 receives packet and strips off packet header and notices payload contains an encapsulated packet addressed to some host inside network 2

- IP tunneling used in
  - VPNs, Mobile IP
  - building logical networks of multicast or QoS enabled routers