

Lecture 7 Internetworking

References:

- CN Sections 5.5, 5.6.
- RFCs: 1519 (CIDR), 3022 (NAT), 2993 (NAT deficiencies), 2328 (OSPF), 1771–1774 (BGP), 1112 (IGMP), 1075 (DVMRP), 2460–2466 (IPv6).

Network(0234B)

The aims

- Many computers are already in LANs or WANs.
- If a “few” links are added, every computer is connected, so in theory **every host can talk to any other host**.
- There is an obstacle: **network talks different protocols**.
- For various reasons it is **impractical to ask all the world to switch to a single protocol**, at least not at the same time.
- An internetwork **connects networks talking different protocols**, allowing every pair of hosts to communicate.
- To do this, some routers must **knows multiple protocols**, e.g., can talk a broadcast protocol with its LAN and a point-to-point protocol with a WAN peer.

Network(0234B)-7.1

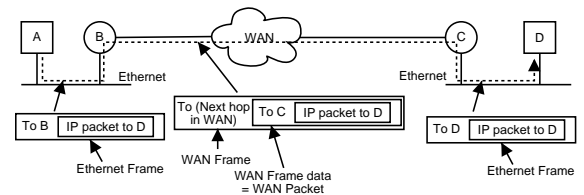
Multi-protocol routers and tunneling

- In an ideal case, the router can receive network packets from one link, **extract the transport layer content**, encode it in another network layer format, and forward it to the next router.
- We call such routers “**multi-protocol routers**”: they can **translate** between multiple protocol.
- In practice, this is **very difficult**. E.g., each network layer protocol has its own addressing scheme, maximum packet size, etc. One might be connection oriented, the other based on datagram.
- In general, it is **impossible** to take a packet of one protocol and translate it to an equivalent packet in another protocol.
- But if the **two ends** talk the **same** protocol (although intermediate routers talk something else), **tunneling** can be used.

Network(0234B)-7.2

Tunneling: illustration

The whole network packet is just treated as **packet data** when passing through a network that doesn't talk the protocol...



Note that **routers in WAN don't need** to look at the internetwork packet.

Routers in WAN don't speak IP, so they can't communicate freely. **Eventually all hosts must speak IP** like B and C. Tunneling buys us **transition time**: non-participating routers won't stop others communicating.

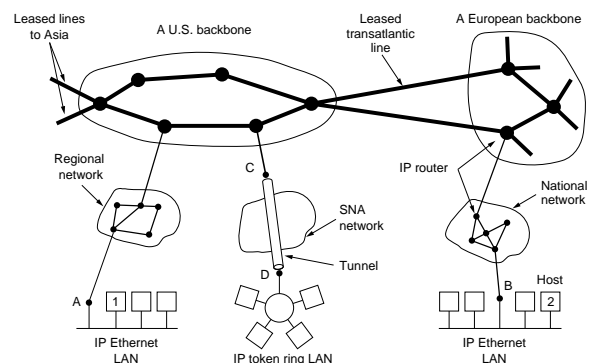
Network(0234B)-7.3

Internet Protocol: the basic idea

- Internet can be treated as an unstructured **collection of subnetworks**, or **Autonomous Systems**.
The Internet is always written with a capital I, to notify that we are talking about that internetwork used by the whole world.
- All participating routers and hosts speaks the network layer protocol called **Internet Protocol (IP)**. Each is given an **IP address**.
- **IP assumes very little from the data link layer**, so that **all existing networks** can be used to **transport (tunnel) IP data**.
Even if they don't speak IP themselves.
- In particular, IP is connectionless, provide **best effort delivery**, does not assume **error checking** and **flow control** in the data link under it.
- Currently, most of the Internet runs on version 4 IP (IPv4), with some starting to switch to version 6 (IPv6).

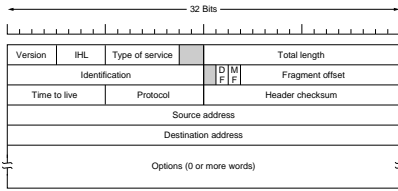
Network(0234B)-7.4

Internet Topology



Network(0234B)-7.5

IPv4 packet



- **Version:** constant 4. This allows different versions to coexist.
- **IHL:** length of the variable size header in 32-bit word.
- **Type of service:** generally ignored by routers. Supposed to differentiate packets of higher requirements.
- **Total length:** size of whole packet including header, in byte.
- **Header checksum:** “1’s complement sum” detecting memory error.

Network(0234B)-7.6

IP packet, continued

- **Time to live (TTL):** Decrement every time an IP router is passed, and packet is discarded if it goes 0. Thus preventing a routing loop from getting infinite amount of data.
- **Protocol:** What type of protocol (TCP, UDP, ICMP, IGMP, etc) the packet is for. **Every other network and transport protocol** of TCP/IP are exchanged between hosts **using IP packets**. The OS use it to find what to do next when the packet is received.
- **Source and Destination:** the network layer address of the ultimate source and destination of the packet. Source for sending error packets.
- **Options:** control the path used for the packet, ask every router to record the route and put a timestamp in the packet, etc. Since 4-bit IHL is at most 15, subtracting 5 for the 20-byte header, we only have at most 15-5=10 words (each 32-bit) here. In general it is insufficient.

Network(0234B)-7.7

Fragmentation

- Different network has different maximum frame (and thus packet) size. When a **large packet** pass a network allowing only small packets, something must be done.
- One strategy: split it. The **MF** (More Fragment) flag, the **Identification** field and **Fragment offset** field support this strategy.

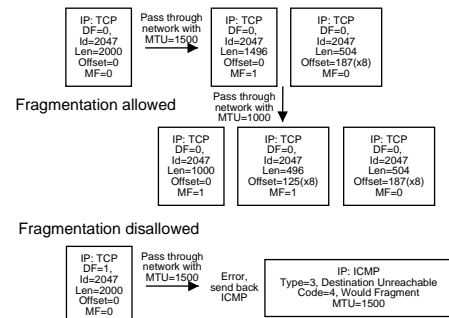
The last fragment has MF=0, all others have MF=1. All have the same identification, and fragment offset identify the position of the fragment.

- Packets are only reassembled **when arriving the final destination**. Some router does reassembling as well, though.
- Another strategy: flag error, asking sender to **use smaller packets**.

If DF=1 (Don't Fragment), an error ICMP packet is sent back to the sender in case the maximum packet size of a network is too small.

Network(0234B)-7.8

Illustration



It is possible for a **fragment to be too big** for the next hop. It will be broken up again.

Network(0234B)-7.9

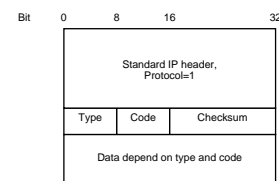
Basic operation

- Transport layer takes data, **breaks it to datagram** for network layer. Of size at most 65536 bytes, but most use 1500 to allow passing through Ethernet without fragmentation.
- Each datagram is processed by the network layer, **route** through it, and arrives the destination, perhaps fragmented.
- The destination **reassemble** the packets and **forward** to a protocol handler, identified by the “protocol” field.
- The transport protocol handler find the **recipient port number**, and map it to the **process** (or socket) that should receive the packet.
- The protocol handler **append** the transport level data to the input stream of the process (or socket).

Network(0234B)-7.10

Example: ICMP

- One possible data within IP packets is **Internet Control Message Protocol** packets, for routers to query routes and notify routing problems.
- ICMP packets are **encapsulated in IP packets** with Protocol=1. Let's see how the encapsulation is done.



From now on, we won't draw the standard IP header.

It has its own checksum, since **IP doesn't have one for the data**. IP packets has a checksum for the header only.

Network(0234B)-7.11

Roles of ICMP

What is done by ICMP? Let's see the simplified table.

Message type	Description
3, Destination unreachable	Packet could not be delivered
11, Time exceeded	TTL hits 0
12, Parameter problem	Invalid header field
4, Source quench	Choke packet (seldom used)
5, Redirect	Tell router to send elsewhere
8, Echo	Ask a machine if it is alive (ping)
0, Echo reply	Yes, I am alive
13, Timestamp request	Same as 8, with timestamp
14, Timestamp reply	Same as 0, with timestamp

What are the code of each type? Either read our reference, or read the standard Unix C header file `<netinet/ip_icmp.h>`.

Network(0234B)-7.12

Addressing

- Each network interface of each host and router has a **network number** and a **host number** within the network.
- These numbers are **encoded in a unique 32-bit IP address**. Each IP address has some bits for network number and some for host number.
- The 32-bit IP address is usually written in "dotted decimal", i.e., 4 8-bit decimal numbers like "147.8.178.10" (which is 2466820618). This is the IP address of ns.csis.hku.hk, our primary DNS server.
- The network number is **the first some bits**. In our case, the first 22 bits are network number. IP with host number 0 identifies the network. It is written like "147.8.176.0/22". 176 is the first 6 bits of 178.
- It is also common to write 22 bits of 1's as the "**netmask**", like "255.255.252.0", to indicate that 22 bits are used for network number. If you take "bit-wise and" of an IP address and the netmask, you get the IP address of the network. This gives an easy way to find whether a host is local.

Network(0234B)-7.13

Address allocation

- An organization (ICANN, Internet Corporation for Assigned Names and Numbers) is in charge of **allocating IP addresses**.
- ICANN gives ranges of IP addresses to **regional authorities**, which in turns allow individuals to request for network numbers from it. This makes sure that everyone gets a different IP addresses range.
- E.g., if somebody wants 400 IP addresses, it is rounded up to the next power of 2 (so 512), and a network of this size (with a 23 bit network address, netmask 255.255.254.0) is given to him.
- The address with host part being all 1's are reserved for **broadcast address**, any host in the Internet can send to this address for broadcast. So both all 0's and all 1's host parts are reserved. In the last example 510 addresses are usable. These has to be considered when determining the number of addresses to request for.

Network(0234B)-7.14

Special IP address ranges

Some addresses are **never allocated** by ICANN for host numbers:

- **Initial 8 bits of 0s**. 0.0.0.0 identifies a host itself. DHCP use it as source address (since it doesn't know the IP address).
- **Initial bit pattern 1110**. Reserved for multicast addresses.
- **Initial 4 bits of 1s** (Class E addresses). 255.255.255.255 is called the "limited broadcast" address (to subnet), again used by DHCP.
- **Non-routable IP addresses** 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16: used for hosts that wants to be **visible only within a local network**. Regional routers are configured to filter out all these addresses.
- The addresses in 127.0.0.0/8: "local **loopback** address". Hosts bind this to a very fast "interface" for packets towards itself. Thus local communication within host can be done using the same network program, without (too much) loss in efficiency.

Network(0234B)-7.15

Subnetting

- The local network administrator who has some IP addresses can **further allocate it to local networks**, called **subnet**.
- E.g., our campus has a "class B" (/16) network 147.8.0.0. It allocates the range 147.8.176.0/22 and another range 147.8.175.0/24 to the CSIS department.
- Each subnet has its own network address and broadcast address.
- Subnetting is invisible externally. Packets are just **routed to the primary network** (in our case, a router in 147.8.176.0/22), which in turn routes them according to the subnet number.
- Internal to a subnet, **computers don't care that it is a subnet** under a larger network address. (It looks like a normal network.)
- The routers of subnets can run a routing protocol locally to send subnet packets optimally. External traffic **always** goes to the main router.

Network(0234B)-7.16

Shortage of IP addresses

- Just like every successful systems, IP is victim of its own success: **IP addresses are getting completely used up**.
- Before adoption of the scheme we have just seen, there is no address mask. All network addresses are either /8 (class A), /16 (class B) or /24 (class C), identified by bit pattern of the network part. Class A, B and C addresses start with bits 0, 10 and 110 respectively.
- Most people needs class B (only 16384 are available) inefficiently (only around 400 addresses of 65534 are used).
- Class B addresses run out soon after Internet is commercialized. The current scheme fix edthe problem. It is called **CIDR** (Classless InterDomain Routing), named so because it uses no class.
- But with an exponential increase in the number of connected hosts, even this runs short: we simply have more hosts who want to participate than the number of possible IP address.

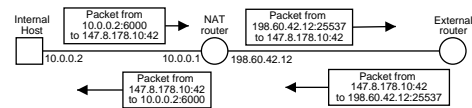
Network(0234B)-7.17

Network Address Translation (NAT)

- One strategy to dampen the exhaustion: **don't allocate** an address for **each** host who want to be connected!
- Instead, every site has **one** computer, the router, who has a "real" IP address. Each other host use a **non-routable IP address**.
- Without a real IP, it cannot communicate with the rest of the world directly. The idea: **the router do it for the host**.
- Recall that a single host (i.e., IP) can make **many transport layer** connections to another computer, so every router can serve many hosts.
- All out-going connections of the local network **are sent to the router**, who **create a transport layer entity** to make the real connection.
- All future outgoing packets for the same connection are **translated**, so that it looks like the router is making the request.
and incoming packets are forwarded back to the requesting host.

Network(0234B)-7.18

NAT (cont'd)



- To the host with IP 10.0.0.1, it simply looks like 10.0.0.2 is its **default route**, i.e., all non-local traffic should be sent to it.
- But the scheme has many limitations:
 - The external host is **completely invisible** from outside.
E.g., traditionally ftp requires the outside to make a connection **back** to the initiating host of a different port number. This fails miserably.
 - If the transport level data (segment) uses or contain the IP address of the initiator, it will not be translated. The protocol will fail.
 - The NAT router can only handle a fix ednumber of connections.
Since there are only 65536 port numbers for both UDP and TCP.

Network(0234B)-7.19

Internetwork routing

- As far as the network layer is concerned, **everyone runs IP**.
Some of them use tunneling to work through other network, but to the network layer of IP, they look like point-to-point links.
- And **every network** known by the routers **has an IP address**.
Some networks are hidden under NAT, but routers don't need to care for them.
- In principle, there should be no difficulty applying distance vector or link state routing for it.
- But there is a problem... **optimal route doesn't please everybody!**
E.g., there's a problem if packets from white house to South Korea route through North Korea.
- There is also problems related to **cost model**... not all routers agrees to **carry transit traffic from every peer**.
- These problem comes from **administration**.

Network(0234B)-7.20

IGP and EGP

- The Internet is thought to compose of **autonomous systems (ASes)**, network under the same administration.
- Within the same AS, the administration can **choose his own routing protocol**. We call it **Interior Gateway Protocol (IGP)**.
- Between ASes, everybody must use the same protocol, called **Exterior Gateway Protocol (EGP)**.
- IGP's strikes to find **optimal** routes.
- EGP strikes to find a **good route** that **satisfies limitations** imposed by the administration.
- Initially, **RIP** (Routing Information Protocol) is used as the Internet IGP. It is no longer recommended due to its problem of slow convergence.
- Replacement: **OSPF** ("Open Shortest Path First"), an implementation of link state routing.

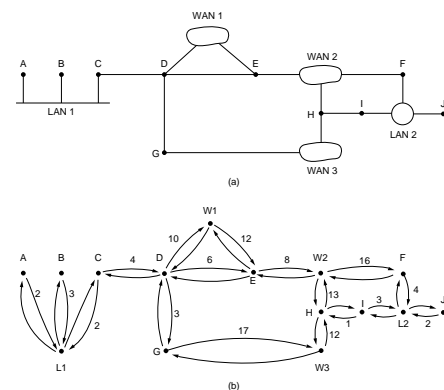
Network(0234B)-7.21

OSPF: representing costs

- The complication: **multi-access** networks (with multiple routers in it), possibly with **broadcast** capability (e.g., Ethernet).
- Solution: Abstract the AS as a **directed graph**, with nodes and links. Each link has a **cost**, exchanged by the link state protocol for calculating **source trees** (shortest path from a node to every other).
- Routers and **broadcast** networks (with a network number and netmask) are represented as a graph **node**. **Non-broadcast** networks are emulated as either many point-to-point links, or a broadcast network.
- Point-to-point link are represented as **pair between router nodes**. Routers in broadcast multi-access networks have a link connecting the router node and the network node.
- A network can be configured as **stub nodes**, i.e., no transit traffic.
Stubs networks only has incoming edge, so source trees never transits them.

Network(0234B)-7.22

Example



Network(0234B)-7.23

OSPF: Routing information exchange

- In link state routing, some messages are exchanged to **measure link cost**, and the resulting information is **flooded**.
- The former is done using **Hello** packets, the latter using **Link State Update** (and Ack) packets, encapsulated in IP (protocol 89).
- There is a problem: flooding means **every pair of neighbours** will exchange packets, a serious problem on broadcast network.
- The Hello packets is used to **elect one router** within a broadcast network as **designated**. Every router in the network consider itself adjacent **only** to it, so as to limit the amount of information to flood.
- Routers connected to other AS (running EGP) will **broadcast external information** among the AS. This allows every routers to calculate **closest path to each external AS** (although not to each external router).
I.e., hierarchical routing. It is done in another level...

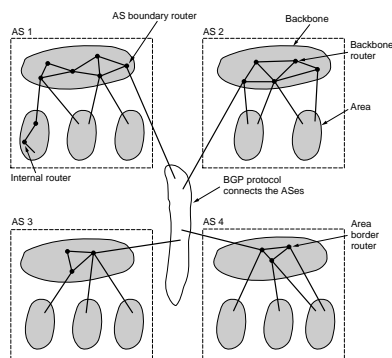
Network(0234B)-7.24

OSPF: Areas

- Many ASes are very large, making it **inappropriate** to let have all routers storing the whole graph and calculate complete source tree.
- OSPF support these ASes by allowing a **further level** of hierarchical routing. Routers can be **grouped into areas**.
- Only **one** area can have links to external network. The area is called the **backbone** area, consisting of **backbone routers**. Some, connected to other ASes, are **AS boundary routers**.
- Some backbone routers, called **area border routers**, are connected to other areas. These routers are responsible for forwarding inter-area and inter-AS traffic.
- Other routers within an area (**Internal routers**) only knows routing within an area, and how to reach an AS boundary router.

Network(0234B)-7.25

Example

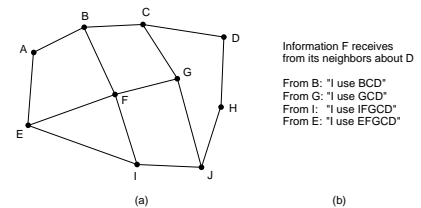


Network(0234B)-7.26

EGP: Border Gateway Protocol (BGP)

The Internet EGP is BGP, which is a **variant of distance vector routing**.

The significant difference: instead of just a cost to every destination, it store also a **path**. This allows **political decisions** to be made **based on the path**, and fix the slow convergence problem.



F might decide to use B for traffic towards D, and if later it is down it won't choose I or E since F appears in their path.

Network(0234B)-7.27

Internet Multicasting

- Many services require some server(s) sending the same packets to many, but not all, hosts simultaneously.
E.g., weather or stock report, radio broadcast, chat room, etc.
- Using unicast is possible, but it **consumes unnecessarily large bandwidth**: the same message is sent to the same interface multiple times.
- Using broadcasting **force all hosts to process and discard** the packets, and nobody want them to be forwarded out of the local network.
- Many broadcast media actually **provide multicasting facility**, so they can be used. But there are three problems:
 - We need an **OS interface** for it. Applications can only send datagrams in the network layers, not frames in the data link layers.
 - We need some mechanism for **extending it to other networks**.

Network(0234B)-7.28

Multicast Interface: Class D addresses

The interface chosen by IP: a set of IP addresses.

- Addresses starting with bits 1110 are called **Class D** IP addresses, and are used solely for multicasting.
Some addresses are assigned to fix edservices, e.g., all multicast routers listen on 224.0.0.2. There is no way to tell whether someone else is using an address.
- The address can be used for **connectionless** IP protocols like ICMP, IGMP and UDP. Connection-oriented protocols ignore them.
- When a host sends a packet containing the address, it is **translated** to a **data link layer multicast address** for building a frame.
If that is not available, broadcast or unicast address can still be used. Multicasting on hardware reduce the load of uninterested hosts.
- **Multicast-enabled hosts** will configure the network interface to capture traffic on data link multicast address that it is interested in, thus listening to all senders sending to the right address.

Network(0234B)-7.29

Programming interface

- Sending to multicast address needs no special attention: everybody can send by using the correct multicast address.
- Sockets allows a program to **receive** from a multicast address.
- The program creates a **UDP socket**, **bind** it to all host with the port number it wants to listen.
Multicast UDP also has a port number, and packets are delivered only if it matches a listening port number.
- Then the `IP_ADD_MEMBERSHIP` IP option is used to **start listening** on a multicasting address.
- Once that is not needed, the `IP_DROP_MEMBERSHIP` option is used to stop listening.
- See the man page `ip(7)` for details.

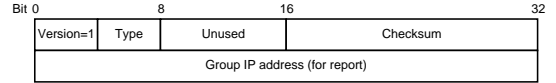
Network(0234B)-7.30

Group management and IGMP

If that service is to extend over LANs, there must be **multicasting routers** which forward messages across networks.

Router need to know whether a host is interested in an address. **Internet Group Management Protocol (IGMP)** provides this information.

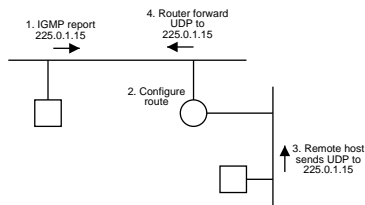
- It is again **encapsulated in IP**, with protocol=2, with the following format:



- Type is either 1 (*query*) or 2 (*report*).
- When a host **join** a group, it sends a report to all routers.
- Routers periodically sends **query** to all multicast hosts, asking them to “renew” their membership. Without it, the membership is dropped.

Network(0234B)-7.31

Example



- Routing of multicast packets is experimental, e.g., **DVMRP** (Distance Vector Multicast Routing Protocol) uses distance vector with tunneling.
- **Pruning** (last chapter) is done to avoid sending to multicast groups that no one is interested in.
- Many problems remain unsolved, though. E.g., how to find a free multicast group, which groups are local and which are global, etc.

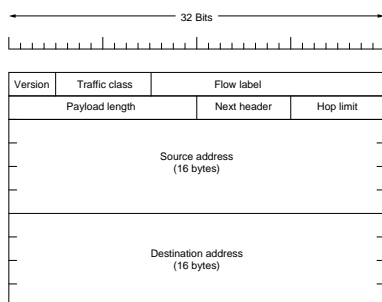
Network(0234B)-7.32

IPv6: introduction

- CIDR and NAT delays the death of IPv4, but ultimately **IP addresses will exhaust**, and the pain is already being felt (by having to use NAT).
- IPv6 expands the **address space** from 32 bits to **128 bits**.
Nearly every molecule in the earth can have one!
- Rarely used fields (e.g., fragments) are **moved to options** to compensate for the lengthened address.
- Main header is **fixed size**, eliminating the length field. Options are **specified in the protocol field**, renamed to “Next header”.
This solves the problem of small option size.
- Options have similar format, allowing routers to ignore them or discard the packet if the router doesn’t support it.
- **Encryption support** is in a **mandatory option** (i.e., one that every router must support).

Network(0234B)-7.33

IPv6: main header



- Hop limit is the original TTL, renamed to reflect the real usage.
- Flow label allows **virtual circuit** to be implemented.
Virtual circuit provides easier congestion control and service guarantees.

Network(0234B)-7.34