Group work

- Group list and topics to be submitted by Fri 27th Jan
- Topics available from Wednesday on the webpage http://www.dsg.cs.tcd.ie/~reynoldv/4BA2
- 3 people per group
- To be submitted on the 17th Feb
- Suggested topics to be submitted to me before Wednesdays lecture. (Topic, Group members)
- Webpage

1

Internet Transmission Paradigm

- Source host:
- Forms datagram
- Includes destination address
- Sends to nearest router
- Intermediate routers:
- Forward datagram to next router
- Final router:
- Delivers to destination host

2

Datagram Transmission

- IP internet layer:
- Constructs datagram
- Determines next hop (routing)
- Hands to network interface layer
- Network interface layer
- Binds next hop address to hardware address
- Prepares datagram for transmission
- Network hardware doesn't understand IP; how is the datagram transmitted?
- Network interface layer *encapsulates* IP datagram as data area in hardware frame
- Hardware ignores IP datagram format
- Standard defines data type for IP datagram and others (e.g., ARP)
- Receiving protocol stack interprets data area based on frame type

3

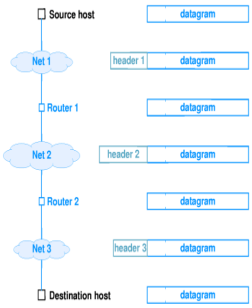
IP Encapsulation

- Entire datagram treated like data
- Frame type identifies contents as IP datagram
- Frame destination address gives next hop
- Frame address:
- Hardware (MAC) address
- Next hop
- Datagram address:
- IP address
- Ultimate destination
- Datagram survives entire trip across an internet
- Frame only survives one hop

4

Encapsulation Across Multiple Hops

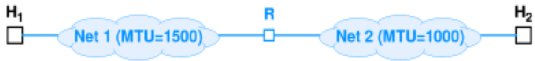
- Each router in the path from the source to the destination:
  - *Unencapsulates* incoming datagram from frame
  - Processes datagram - determines next hop
  - *Encapsulates* datagram in outgoing frame
- The datagram may be encapsulated in different hardware formats at each hop
- The datagram itself is unchanged



5

MTU and Datagram Transmission

- Every hardware technology specification includes the definition of the maximum size of the frame data area
- Called the *Maximum Transmission Unit* (MTU)
- Any datagram encapsulated in a hardware frame must be smaller than the MTU for that hardware
- IP datagrams can be larger than most hardware MTUs
- Heterogeneous networks?
  - An internet *may* have networks with different MTUs
  - Suppose downstream network has *smaller* MTU than local network?



6

Datagram Fragmentation

- One technique: limit datagram size to *smallest* MTU of *any* network
- IP uses *fragmentation:*
  - Datagrams can be split into pieces to fit in a network with small MTU
- Router detects datagram larger than network MTU:
  - Splits it into pieces
  - Each piece *smaller* than outbound network MTU
- Each fragment is an independent datagram:
  - Includes all header fields
  - Bit in header indicates datagram is a fragment
  - Other fields have information for reconstructing original datagram
  - FRAGMENT OFFSET gives original location of fragment
- Router uses local MTU to compute size of each fragment
- Puts parts of data from original datagram in each fragment
- Puts other information into the header

7

Datagram Reassembly

- Reconstruction of original datagram is call *reassembly*
- Ultimate destination performs reassembly

- Fragments may arrive out of order; header bit identifies fragment containing end of data from original datagram
- Last fragment identified via the FLAGS field of the IP header

8

Fragment Identification

- How are fragments associated with original datagram?
- IDENT field in each fragment matches IDENT field in original datagram
- Fragments from different datagrams can arrive out of order and still be sorted out
- Fragment loss
  - IP may drop fragment
  - What happens to original datagram?
    * Destination drops *entire* original datagram
  - How does destination identify lost fragment?
    * Sets timer with each fragment
    * If timer expires before all fragments arrive, fragment assumed
    * Datagram dropped
- *Source* (transport or application layer protocol) assumed to retransmit

9

Fragmenting a Fragment

- Fragment may encounter subsequent network with even smaller MTU
- Router fragments the fragment to fit
- Arbitrary sub-fragmentation possible
- Resulting sub-fragments look just like original fragments (except for size)
- No need to reassemble hierarchically
  - Sub-fragments include position in *original* datagram
  - Offset given with respect to original datagram
  - Destination cannot distinguish sub-fragments

10

IPv4 Review

- Has been extremely successful:
  - Internet handles heterogeneous networks
    * Uniform packet format (IP datagram)
  - Accommodated changes in hardware technologies
  - Used over networks several orders of magnitude faster than the networks it was designed for
- If IP works so well, why do we need to change?
  - Limited address space
  - New Internet applications (audio, video)
  - New mechanisms for addressing and routing

11

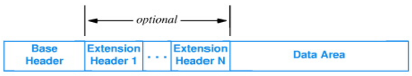
IPv6: IP – The Next Generation

- IPv6 retains many of the IPv4 features
  - Is connectionless
  - Contains destination and source address
  - Each datagram routed independently
  - Max number of hops before discarded
- Basic concepts are the same, but details change
  - Address size: 128 bits instead of 32 bits (4 times larger)
  - Header format: Completely different
  - Extension headers: Base header plus zero or more extension headers followed by the payload
  - Support for audio and video: Establishment of high-quality paths and association of datagrams with these paths
  - Extensible protocol: Does not specify all possible protocol features; a sender can add additional information to a datagram

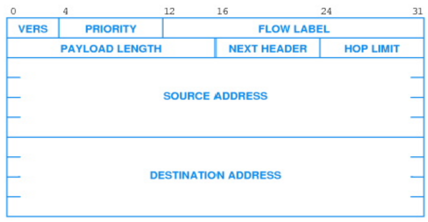
12

IPv6 Datagram Format

- Begins with base header
- Followed by zero or more extension headers
- Followed by the payload
- An extension header may be larger than the base header



IPv6 Base Header (1/3)



IPv6 Base Header (2/3)

- VERS: Identifies the protocol version as 6
- SRC, DST: Most space of the header
- TRAFFIC CLASS: Specifies general characteristics that the datagram needs from the underlying network hardware

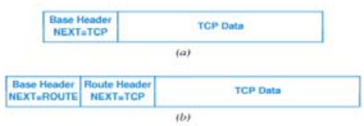
- Established path has delay less than 100 ms

- PAYLOAD LENGTH: The size of the payload
- HOP LIMIT: IPv4 TTL
- FLOW LABEL: Associates the datagram with a particular underlying network path

IPv6 Base Header (3/3)

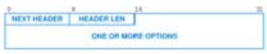
- NEXT HEADER: Specifies the type of information that follows the current header

- Type of extension header
- Type of payload

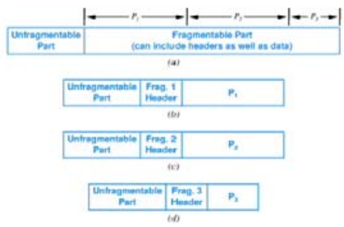


IPv6 Multiple Headers

- Based on the value in the NEXT HEADER field
  - Receiver passes the datagram to a software module to handle the data, or
  - IP software parses the header and interprets its contents
- How does IPv6 software know where a particular header ends?
  - Base header: Fixed size of 40 bytes
  - Extension headers:
    - * Some types also have a fixed size
    - * Others do not. include HEADER LEN field



IPv6 Fragmentation (1/2)



IPv6 Fragmentation (2/2)

- Fragment size == MTU
- Sending host responsible for fragmentation, not intermediate routers
- Minimum MTU: Path MTU
- Path MTU discovery:
  - A hosts sends a sequence of various-size datagrams to the destination
  - Once a datagram is small enough to reach the destination, the hosts chooses a datagram size equal to the path MTU
  - Minimum packet size is 576 bytes

IPv6 Addressing

- Prefix: Identifies network
- Suffix: Identifies interface
- Division on arbitrary bound
  - Multilevel hierarchy
  - Assignments are not fixed
    - * E.g.: ISP, organization, site, and so on
- Each address one of three types:
  - Unicast: Single computer
  - Multicast: Set of computers
  - Anycast: Set of computers that share a prefix, delivered to exactly one of them
- No broadcast address
  - IPv4: Last address of the subnet, e.g.: 192.168.1.255

IPv6 Colon Hexadecimal Notation

- Dotted decimal:
  - 105.220.136.100.255.255.255.255.0.0.18.128.140.10.255.255
- Groups of 16 bits written in hexadecimal:
  - 69dc:8864:ffff:ffff:0:1280:8c0a:ffff
- Zero compression replaces sequences of zeroes with two colons
  - ff0c:0:0:0:0:0:0:b1
  - ff0c::b1
- IPv4 addresses can be represented as IPv6 addresses that start with 96 zero bits
  - ::192.31.20.46

21

Why Do We Still Not Have IPv6 Dammit?

- Everything has to change (end-to-end)
- Applications and APIs have to change
- Domain Name System (DNS) has to change
- Routing protocols have to change
- Evolution:
  - The 6Bone ``6 over 4'' tunnels, RFC 2529 (1996)
  - IPv6 RFC 2460 (1998)
  - Native IPv6 backbones (1999)
  - ``Bump in the stack'', RFC 2767
    * IPv4 traffic is translated to IPv6 traffic and vice versa

22