Connectionless Service

- End-to-end delivery service is *connectionless*
- Extension of LAN abstraction

 - Universal addressing

 - Data delivered in packets (frames), each with a header

- Combines collection of physical networks into single, virtual network
- Transport protocols use this connectionless service to provide connectionless data delivery (UDP) and connection-oriented data delivery (TCP)

Virtual Packets (1/2)

- *Packets* serve same purpose in internet as frames on LAN
- Created and understood only by software
- Contain sender and destination addresses
- Each has a header
- *Routers* (formerly *gateways*) forward between physical networks
- Packets have a uniform, hardware-independent format

  - Includes header and data

  - Can't use format from any particular hardware
- Encapsulated in hardware frames for delivery across each physical network

Virtual Packets (2/2)

- Because it can connect heterogeneous networks, a router cannot transmit a copy of a frame that arrives on one network across another.
- To accommodate heterogeneity, an internet must define a hardware-independent packet format.

IP Datagram Format

- Formally, the unit of IP data delivery is called a *datagram*
- Includes header area and data area
- Datagrams can have different sizes
- Header area *usually* fixed (20 bytes) but can have options that increase the size
- Data area (payload) can contain between 1 and 65,535 bytes

  - No minimum size
- Usually, data area much larger than header

Best-effort Delivery

- IP provides a service equivalent to LAN
- Does *not* guarantee to prevent:

 - Duplicate datagrams
 - Delayed or out-of-order delivery
 - Corruption of data
 - Datagram loss

- *Reliable delivery* provided by *transport layer*
- *Network layer* (IP) can *detect* and *report* errors without actually *fixing* them

 - Network layer focuses on datagram delivery
 - Application layer not interested in differentiating among delivery problems at intermediate routers

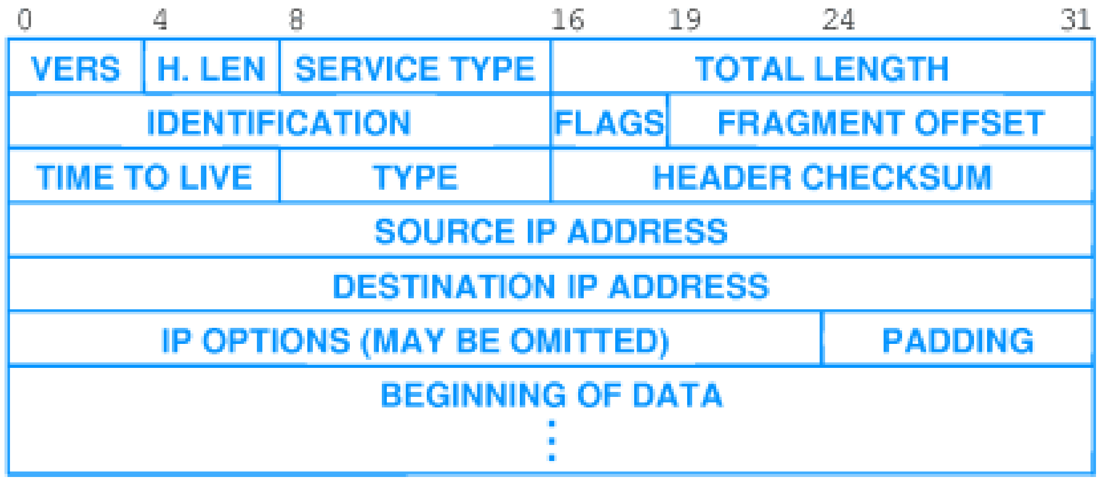
Network Byte Order

- One problem that often arises is that different machines represent integers in different ways:
  - *Big Endian* machines such as IBM and Sun computers store the most significant byte of a 32-bit integer in the lowest memory address of the word (e.g. to the left)

    * The integer 0x01020304 is laid out in memory as bytes 0x01, 0x02, 0x03, and 0x04
  - *Little Endian* machines such as Intel computers store the most significant byte at the highest address

    * The integer 0x01020304 is laid out in memory as bytes 0x04, 0x03, 0x02, 0x01
- Other machines (such as DEC-10s) use 36-bit words to hold integers
- As with all network protocols, the standards specify the meanings of all bits in each field, right down to the bit and byte order
- The Internet defines a network Big Endian standard byte order that is used when referring to the fields of Internet datagrams

IPv4 Datagram Header Format



- VERS -- Version of IP (currently 4)
- H. LEN -- Header length (in units of 32 bits)
- SERVICE TYPE -- Sender's preference for low latency, high reliability (rarely used)
- TOTAL LENGTH -- Total bytes in datagram
- IDENT, FLAGS, FRAGMENT OFFSET -- Used with fragmentation
- TTL -- *Time-To-Live*; decremented at each router; datagram discarded when TTL == 0
- TYPE -- Type of protocol carried in datagram; e.g.: TCP, UDP
- HEADER CHECKSUM -- 1s complement of 1s complement sum
- SOURCE, DEST IP ADDRESS -- IP addresses of *original* source and *ultimate* destination

IPv4 Header Fields (1/6)

- Version number (4-bits):

  - The current protocol version is 4
  - Including a version number allows a future version of IP be used along side the current version, facilitating migration to new protocols

- Header length (4-bits):

  - Length of the datagram header (excluding data) in 32-bit words (or units)
  - The minimum length is 5 words (== 20 bytes), but can be up to 15 words if IPv4 options are used
  - In practice, the length field is used to locate the start of the data portion of the datagram

IPv4 Header Fields (2/6)

- Type-of-service (8-bits):

- A hint to the routing algorithms as to what type of service we desire
- In practice, routers ignore the TOS field in IPv4

  - *Precedence (3-bits)*: A priority indication, where 0 is the lowest and means normal service, while 7 is highest and is intended for network control messages (e.g., routing, congestion control)
  - *Delay (1-bit)*: An application can request low delay service (e.g., for interactive use)
  - *Throughput (1-bit)*: Application requests high throughput
  - *Reliability (1-bit)*: Application requests high reliability

- Does setting the low-delay bit guarantee getting such service? No
- The type-of-service field is meant as a request or hint to the routing algorithms, but does not guarantee that your request can be honored

IPv4 Header Fields (3/6)

- Total length (16-bits):

 - Total length of the IP datagram (in bytes), including data and header

 - The size of the data portion of the datagram is the total length minus the size of the header

- Identification (16-bits):

 - The identification field uniquely identifies fragments of the same original datagram

 - Whenever a host sends a datagram, it sets the identification field of the outgoing datagram and increments its local identification counter.

- Flags (3-bits):

 - *Don't fragment* indication (set by host, honored by routers)

 - *More fragments* field indicates that another fragment follows this one

 - An unfragmented datagram has an offset of 0, and a *More fragments* bit of 0

 - The last fragment of a fragmented datagram contains *More fragments* == 0 and the *Fragment offset* non-zero

- Fragment offset (13-bits):

 - The offset field shows order of the fragments

 - When a gateway fragments a datagram, it sets the offset field of each fragment to reflect at what data offset with respect to the original datagram the current fragment belongs

IPv4 Header Fields (4/6)

- Time-To-Live (8-bits):

- A counter that is decremented by each gateway
- Should this hop count reach 0, discard the datagram
- Originally, the time-to-live field was intended to reflect real time
- In practice, it is now a hop count
- The time-to-live field squashes looping packets
- It also guarantees that packets don't stay in the network for longe, a property needed by higher layer protocols that reuse sequence numbers

- Protocol (8-bits):

- What type of data the IP datagram carries (e.g., TCP, UDP, etc.)
- Needed by the receiving IP to know the higher level service that will next handle the data

An aside (Opus1 traceroute server)

- traceroute to 134.226.1.64 (134.226.1.64), 30 hops max, 40 byte packets
- 1 Firewall (192.245.12.78) 3.906 ms TTL =1
- 2 Opus-GW (207.182.35.49) 3.906 ms TTL =2
- 3 Opus-Login-T3 (204.17.35.105) 3.96 ms TTL=3
- 4 phv-edge-01.inet.qwest.net (65.121.93.133) 10.742 ms TTL = …..
- 5 tmp-core-01.inet.qwest.net (205.171.129.85) 19.530 ms
- 6 bur-core-01.inet.qwest.net (67.14.10.6) 20.507 ms
- 7 lap-brdr-01.inet.qwest.net (205.171.213.106) 20.506 ms
- 8 205.171.1.82 (205.171.1.82) 23.436 ms
- 9 so0-0-0-2488M.ar1.DUB1.gblx.net (67.17.66.6) 170.887 ms
- 10 HEAnet-2.so-3-0-0.ar1.dub1.gblx.net (208.48.23.54) 170.887 ms
- 11 mantova-gige5-2.bh.access.hea.net (193.1.195.136) 171.864 ms
- 12 spike-gig1-5.tcd.access.hea.net (193.1.196.150) 171.864 ms
- 13 134.226.254.45 (134.226.254.45) 172.840 ms

IPv4 Header Fields (5/6)

- Header checksum (16-bits):

- A checksum of the IP header (excluding data)
- The IP checksum is computed as follows:
  * Treat the data as a stream of 16-bit words (appending a 0 byte if needed)
  * Compute the 1's complement sum of the 16-bit words
  * Take the 1's complement of the computed sum
- This checksum is much weaker than CRC
- It has the property that the order in which the 16-bit words are summed is irrelevant
- We can place the checksum in a fixed location in the header, set it to zero, compute the checksum, and store its value in the checksum field
- On receipt of a datagram, the computed checksum calculated over the received packet should be zero
- Checksumming only the header reduces the processing time at each gateway, but forces transport layer protocols to perform error detection (if desired)
- The header must be recalculated at every router since TTL is decremented

IPv4 Header Fields (6/6)

- Source address (32-bits):
  - Original sender's address
  - This is an IP address, not a MAC address
- Destination address (32-bits):
  - Datagram's ultimate destination
  - When a gateway forwards a frame to another gateway, it forwards an Ethernet frame
  - The IP embedded datagram contains the source of the original sender (not the forwarding gateway) and the destination address of the ultimate destination

IPv4 Header Options (1/2)

- IP datagrams allow the inclusion of optional, varying length fields that need not appear in every datagram
- We may sometimes want to send special information, but we don't want to dedicate a field in the packet header for this purpose
- Options start with a 1-byte *option code*, followed by zero or more bytes of *option data*
- The option code byte contains three parts:

- <u>Copy flag (1 bit)</u>:
  * If 1, replicate option in each fragment of a fragmented datagram
  * If 0, option need only appear in first fragment
- <u>Option class (2 bits)</u>:
  * 0: Network control
  * 1: Reserved
  * 2: Debugging and measurement
  * 3: Reserved

IPv4 Header Options (2/2)

- Option number (5 bits):

Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp