UDP

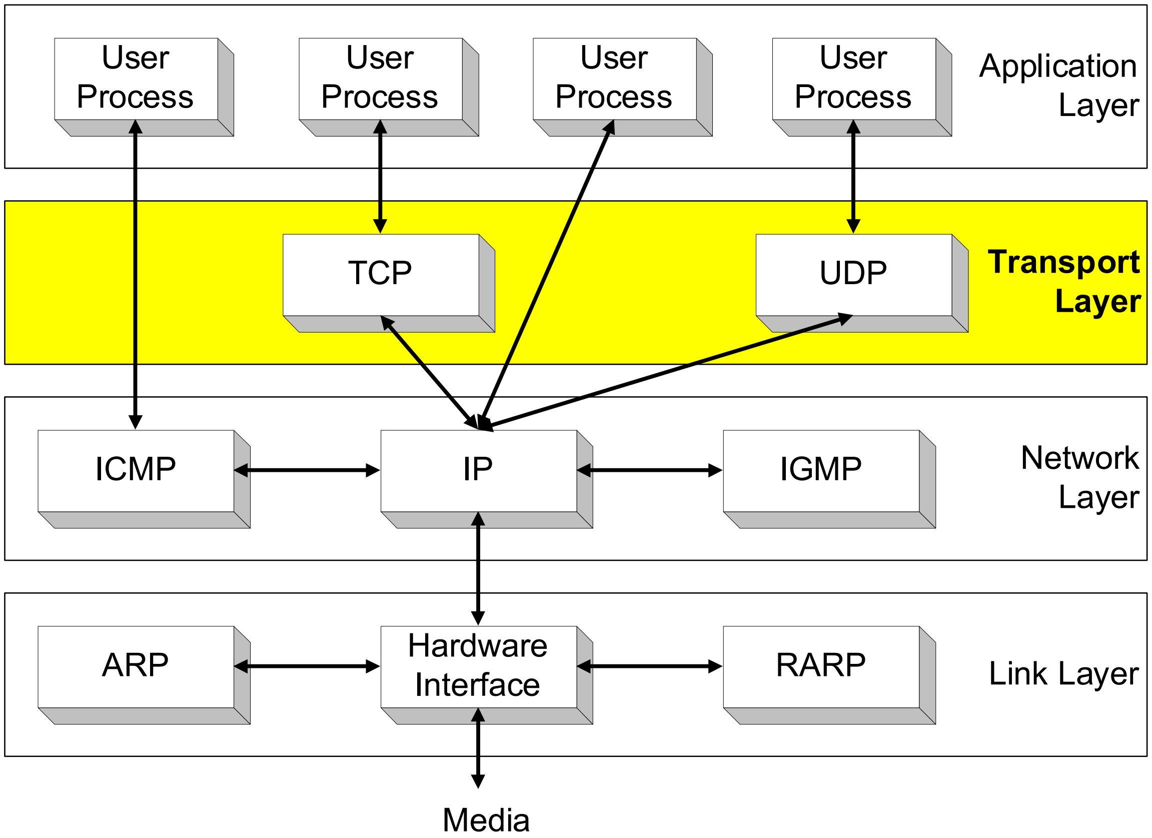
Mobile Computing

Prof. Jongwon Yoon

Intelligent Machines Lab.

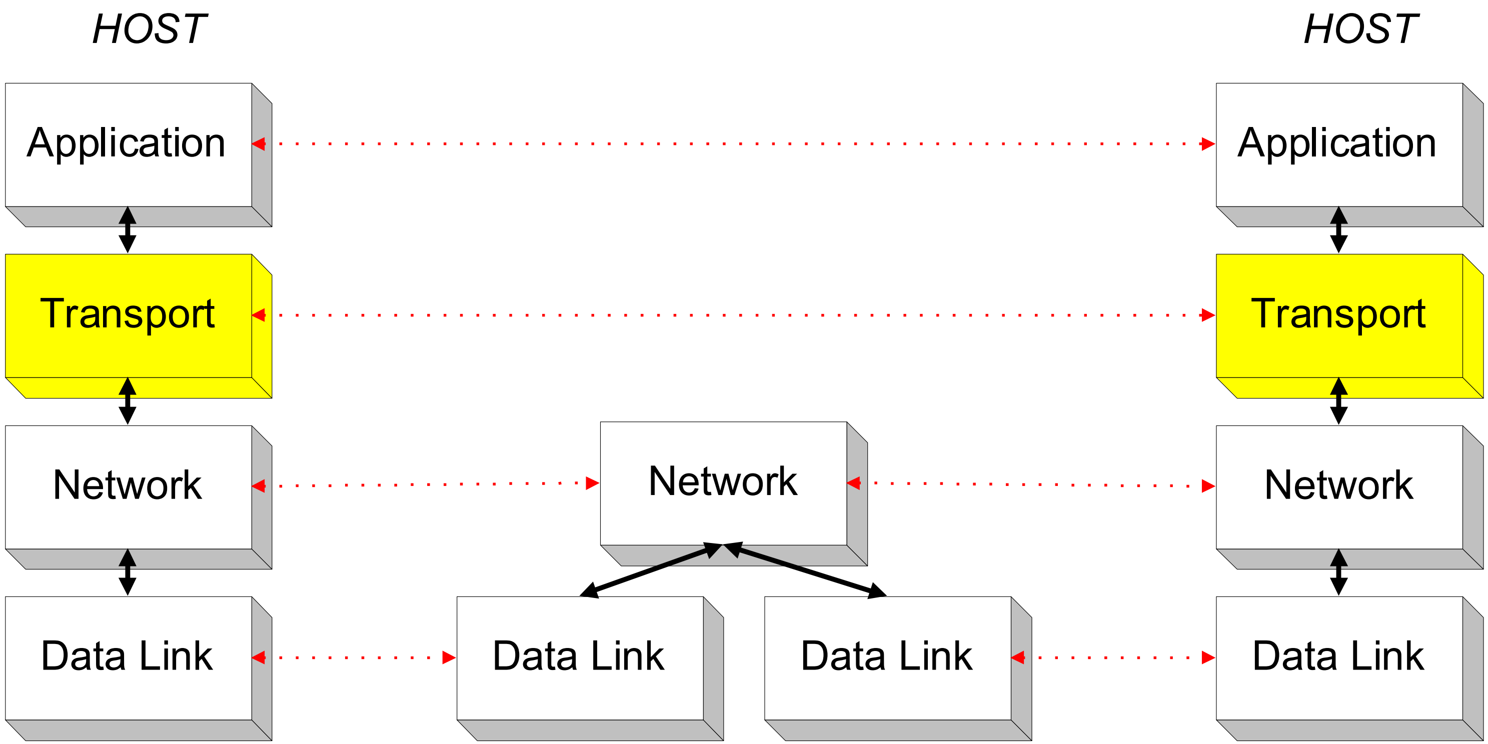
Overview

- We move one layer up and look at the transport layer.



Overview

- Transport layer protocols are end-to-end protocols
  (smart terminals and dumb minimal networks)
- They are only implemented at the hosts



Transport Protocols

The Internet supports 2 transport protocols

UDP - User Datagram Protocol

- datagram oriented
- unreliable, connectionless
- simple
- unicast and multicast
- useful only for few applications, e.g., multimedia applications
- used a lot for services

  - network management (SNMP), routing (RIP), naming (DNS), etc.
- 3 uses of UDP: non-unicast, real-time, short transactions

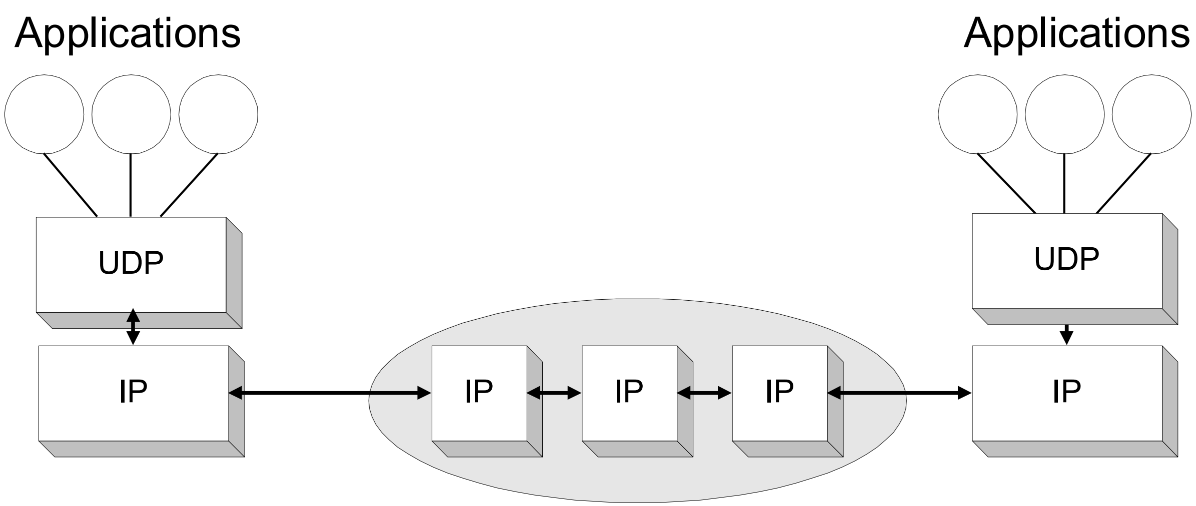
TCP - Transmission Control Protocol

- stream oriented
- reliable, connection-oriented
- complex
- only unicast
- used for most Internet applications:

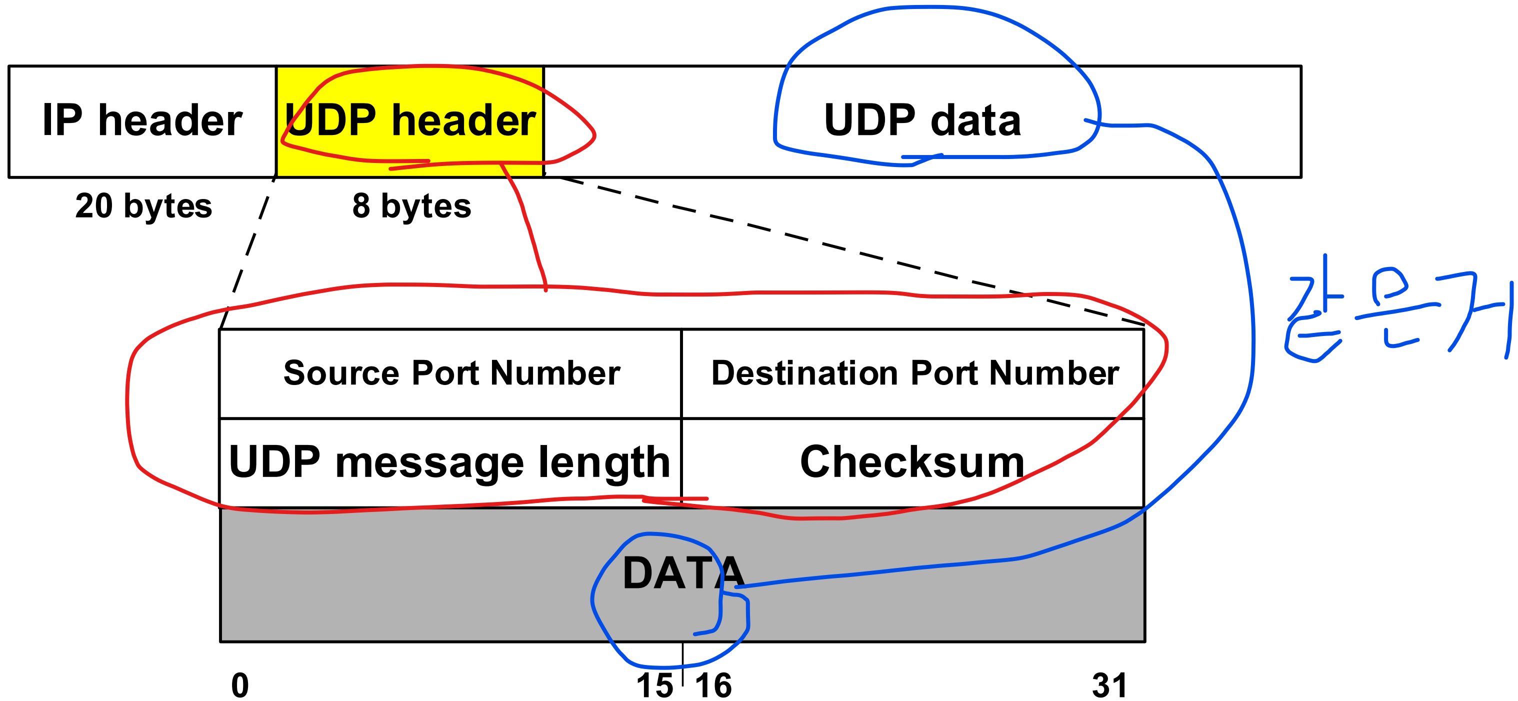
  - web (http), email (smtp), file transfer (ftp), terminal (telnet), etc.

UDP (User Datagram Protocol)

- UDP supports unreliable transmissions of datagrams (low overhead)
- UDP merely extend the host-to-host delivery service of IP datagram to an application-to-application service
- The only thing that UDP adds is multiplexing and demultiplexing

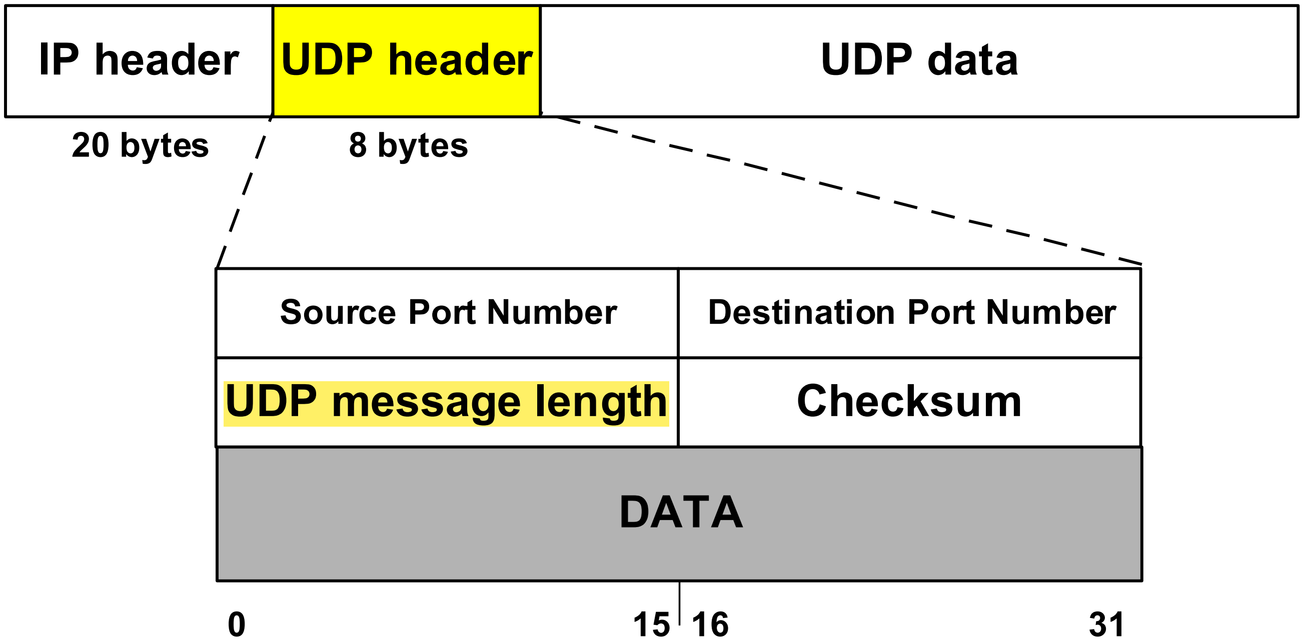


UDP Format



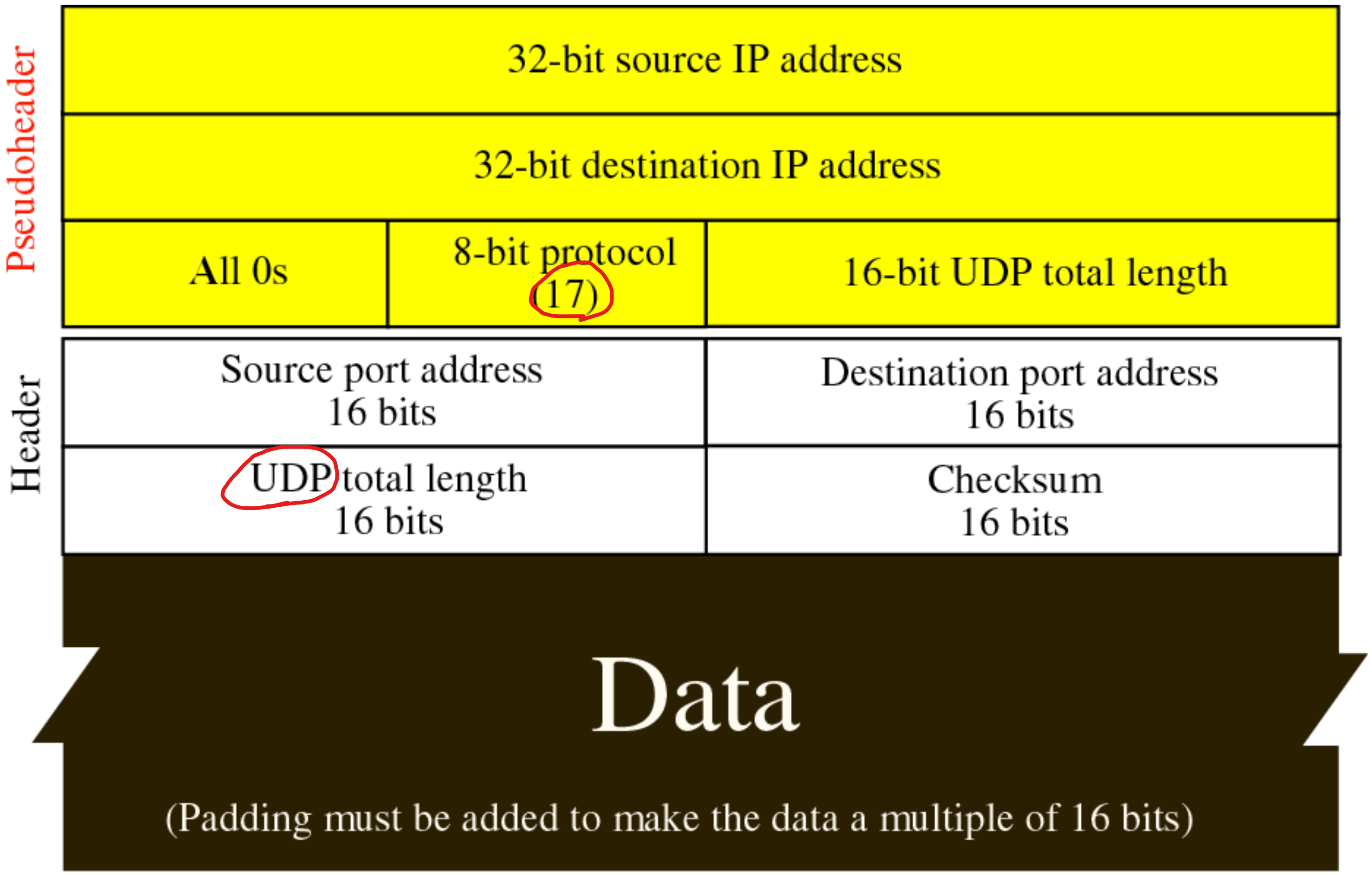
- **Port numbers** identify sending and receiving applications (processes)
  - Required to find the process
    (ICMP error carries the starting part of the dead IP datagram)
  - If no association is found, ICMP Port Unreachable is returned to the source by UDP

UDP Format



- **Message Length** is at least 8 bytes (i.e., includes UDP header, data field can be empty) and at most 65,535 bytes
- **Checksum** covers the UDP header and the UDP data
  - Optional in IPv4, mandatory in IPv6
  - 0 means checksum is not used

Pseudoheader added to the UDP datagram



Checksum calculation of a simple UDP user datagram

1. Add pseudo-header
2. Fill checksum with 0's
3. Divide into 16-bit words (adding padding if required)
4. Add words using 1's complement arithmetic
5. Complement the result and put in checksum field
6. Drop pseudo-header and padding
7. Deliver UDP segment to IP

Checksum calculation of a simple UDP user datagram

153.18.8.105			
171.2.14.10			
All 0s	17	15	
1087		13	
15		All 0s	
T	E	S	T
I	N	G	All 0s

10011001 00010010	→	153.18
00001000 01101001	→	8.105
10101011 00000010	→	171.2
00001110 00001010	→	14.10
00000000 00010001	→	0 and 17
00000000 00001111	→	15
00000100 00111111	→	1087
00000000 00001101	→	13
00000000 00001111	→	15
00000000 00000000	→	0 (checksum)
01010100 01000101	→	T and E
01010011 01010100	→	S and T
01001001 01001110	→	I and N
01000111 00000000	→	G and 0 (padding)
10010110 11101011	→	Sum
01101001 00010100	→	Checksum

Checksum calculation at Receiver

1. Add pseudo-header to the UDP segment
2. Add padding, if needed
3. Divide into 16-bit words and add words using 1's complement arithmetic
4. Complement result
5. If result is all 0's
   - Drop pseudo-header and padding (if any)
   - Accept segment

   Else
   - Drop segment

[e2e] purpose of pseudo header in TCP checksum

David P. Reed dpreed at reed.com
Tue Feb 15 04:39:39 PST 2005

As I was there (in 1976, when we split TCP into IP, TCP, and other protocols, such as UDP) for the decision to separate the checksums and to create a pseudo-header, here is the rationale, which is highly relevant.

TCP (and UDP) are end-to-end protocols. In particular, the TCP checksum is "end-to-end". It is a "private matter" between end points implementing the TCP layer, guaranteeing end-to-end reliability, not hop-by-hop reliability.

IP is a wrapper for TCP, which instructs the transport layer (the gateways and routers) where the packet is to be transported, how big it is, and how it may be fragmented in the process of delivery..

The Source Address, Destination address, length, etc. are part of the meaning of the TCP frame - in that the end point machines use that information in the TCP application.

Thus the function of SA, DA, etc. are "shared" because they are meaningful to both layers (IP and TCP). Rather than include the same information twice in the packet format, the concept of a "virtual header" was invented to encapsulate the idea that IP is not allowed to change the SA and DA because they are meaningful.

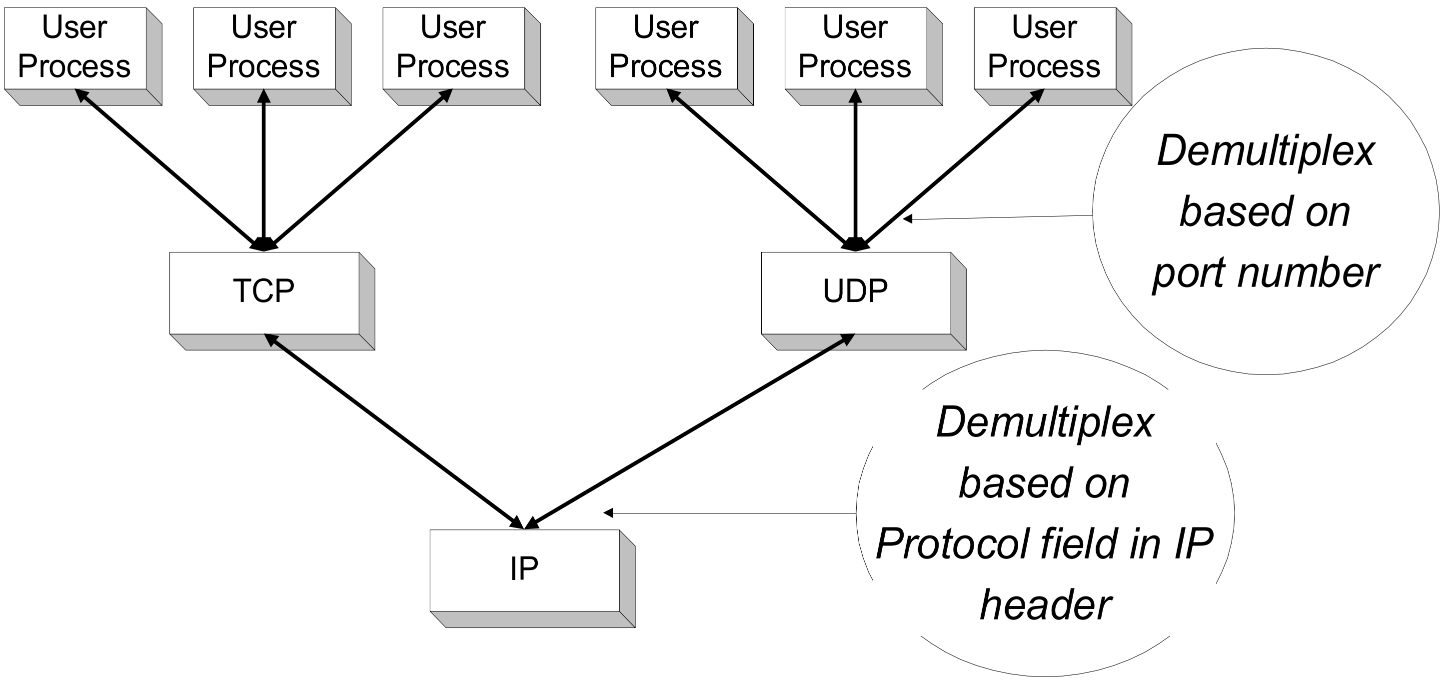
Further, in the case of end-to-end encryption (in 1976 we had a complete design by Steven T. Kent, my office mate, which was blocked by NSA from being deployed) it is essential that all end-to-end meaning be protected. The plan was to leave the SA and DA in the clear, but encrypt the rest of the TCP payload, including the checksum. This would protect against a man-n-the-middle attack that delivered valid packets with an incorrect source address, etc. (yes, to be truly reliable, we would have had to use a DSA instead of the current checksum).

This was a careful design decision, wrecked irrevocably by the terrorists who invented NAT (which doesn't allow end--to-end encryption, because NAT is inherently a "man-in-the-middle" attack!).

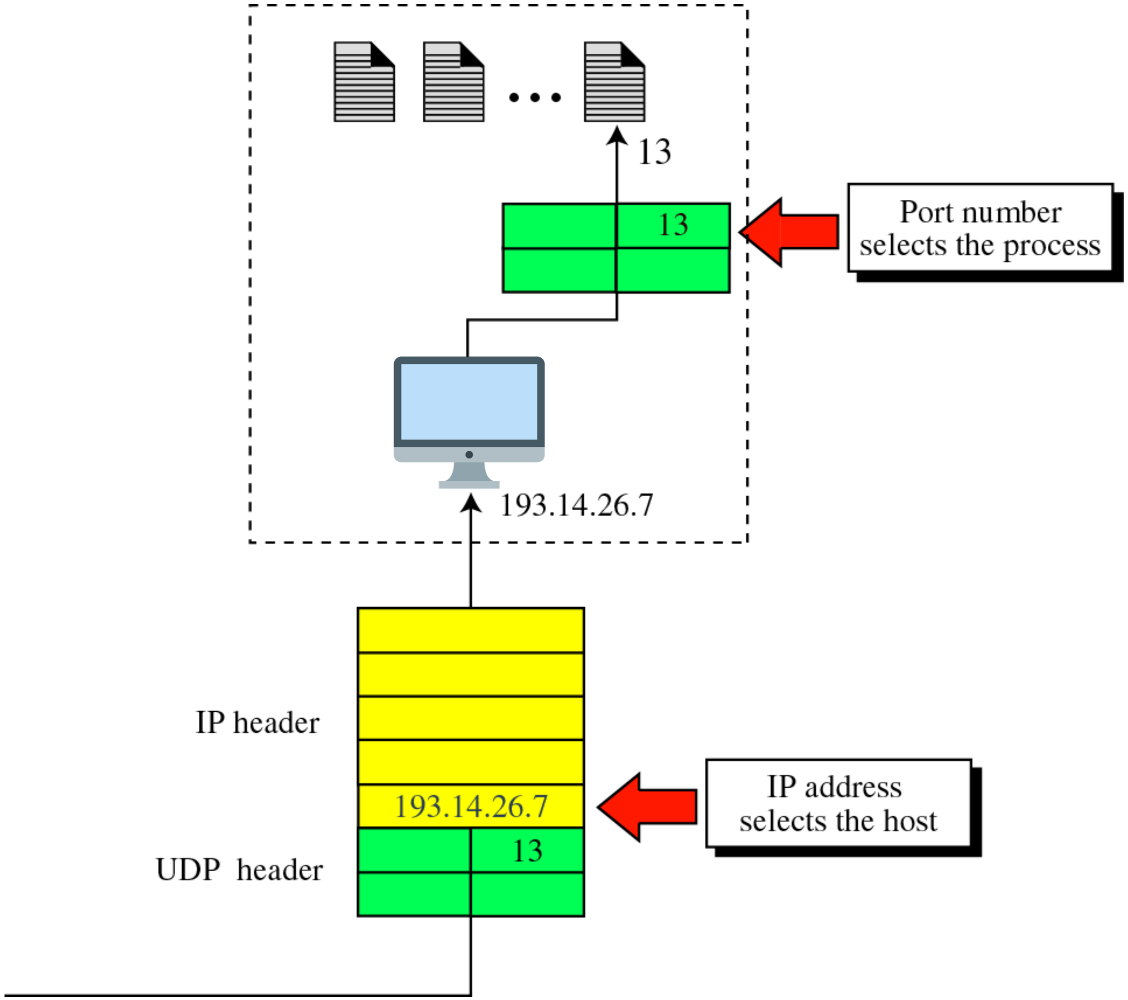
The rise of the middleboxen have now so thoroughly corrupted the Internet protocol design that it's not surprising that the original designs are difficult to decode. If we actually had end-to-end encrypted TCP (now impossible because of the NATs) we would have a much more secure and safe Internet, while preserving its open character. Instead we have a maze of twisty, disconnected passages, vulnerable to a zillion hackers.

Port Numbers

- UDP (and TCP) uses port numbers to identify applications
- A globally unique address at the transport layer (for both UDP and TCP) is a tuple **<IP address, port number>**
- There are 65,535 UDP ports per host.



Port Numbers



Encapsulation and Decapsulation