PART VI

INTERNET PROTOCOL: CONNECTIONLESS DATAGRAM DELIVERY

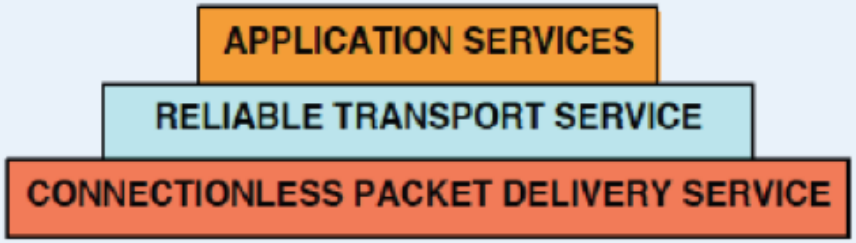
Internet Protocol

- One of two major protocols in TCP/IP suite
- Major goals
  - Hide heterogeneity
  - Provide the illusion of a single large network
  - Virtualize access

The Concept

IP allows a user to think of an internet as a single virtual network that interconnects all hosts, and through which communication is possible; its underlying architecture is both hidden and irrelevant.

Internet Services And Architecture Of Protocol Software



- Design has proved especially robust

IP Characteristics

- Provides connectionless packet delivery service
- Defines three important items
  - Internet addressing scheme
  - Format of packets for the (virtual) Internet
  - Packet forwarding

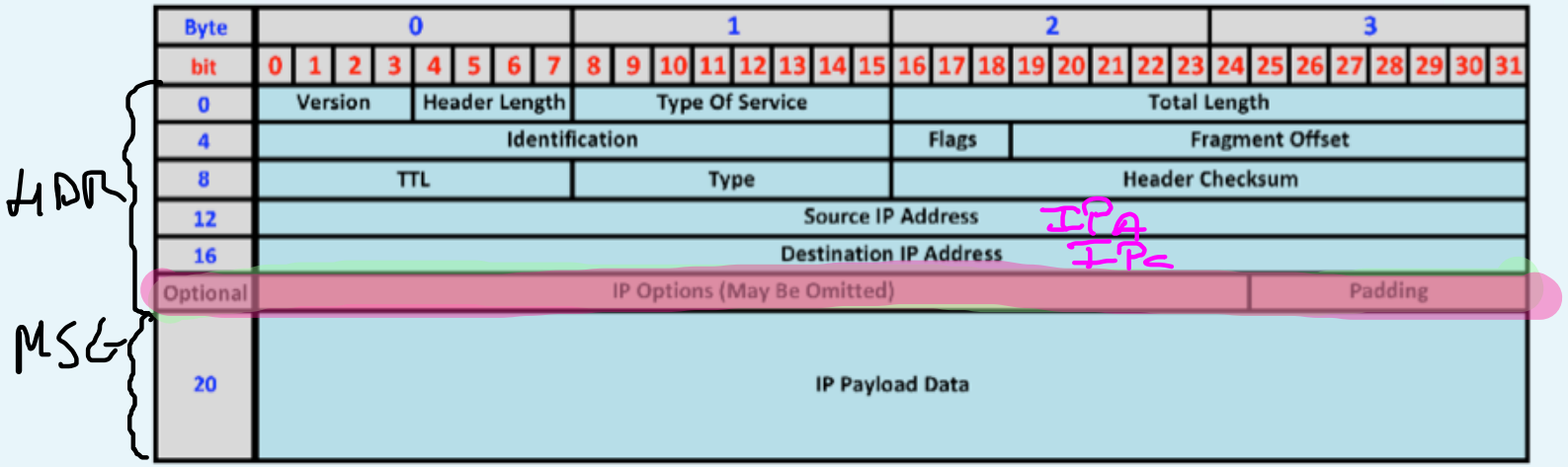
Internet Packet

- Analogous to physical network packet
- Known as *IP datagram*

IP Datagram Layout

- Header contains
  - Source Internet address
  - Destination Internet address
  - Datagram type field
- Payload contains data being carried

Datagram Header Format



Addresses In The Header

- SOURCE is the address of original source
- DESTINATION is the address of ultimate destination

IP Versions

- Version field in header defines version of datagram
- Internet currently uses version 4 of IP, IPv4
- Preceding figure is the IPv4 datagram format
- IPv6 discussed later in the course

Datagram Encapsulation

- Datagram *encapsulated* in network frame
- Network hardware treats datagram as data
- Frame type field identifies contents as datagram
  - Set by sending computer
  - Tested by receiving computer

Datagram Encapsulation For Ethernet

- Ethernet header contains Ethernet hardware addresses
- Ethernet type field set to 0x0800

FROM A ⟶ R

Ethernet Frame Format

Destination Hardware Address	Source Hardware Address	Frame Type	Frame Data
6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes

HW_{R1} HW_A

- Header format fixed (Destination, Source, Type fields)
- Frame data size can vary from packet to packet
  - Maximum 1500 octets
  - Minimum 46 octets

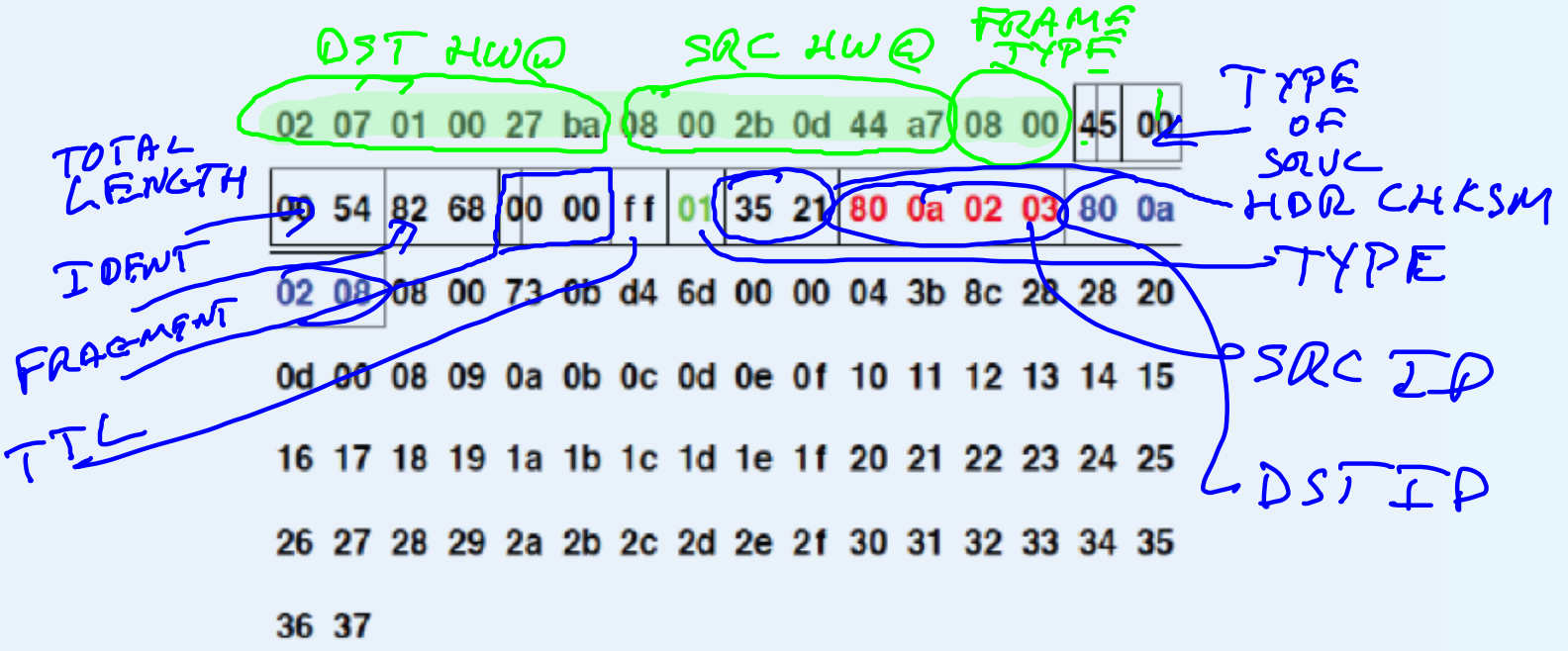
Ethernet Frame Format

Destination Hardware Address	Source Hardware Address	Frame Type	Frame Data
6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes

HW$_C$ HW$_{R_2}$

- Header format fixed (Destination, Source, Type fields)
- Frame data size can vary from packet to packet
  - Maximum 1500 octets
  - Minimum 46 octets

Datagram Encapsulated In Ethernet Frame



- 20-octet IP header follows Ethernet header
- IP source: 128.10.2.3 (800a0203)
- IP destination: 128.10.2.8 (800a0208)
- IP type: 01 (ICMP)

Destination Hardware Address	Source Hardware Address	Frame Type	Frame Data
6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes

Byte	0								1								2								3							
bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Version				Header Length				Type Of Service								Total Length															
4	Identification																Flags				Fragment Offset											
8	TTL								Type								Header Checksum															
12	Source IP Address																															
16	Destination IP Address																															
Optional	IP Options (May Be Omitted)																								Padding							
20	IP Payload Data																															

8B	BD	EB	D5	A3	00	9E	CF	04	7F	AC	2B	08	00	05	11
00	72	FC	53	14	8D	47	36	9A	84	D2	92	8E	39	49	16
75	02	D6	1B	BE	C8	33	02	C7	DF	1A	12	AF	D3	91	AF
BE	91	3D	25	0D	6E	4F	5E	61	0A	E5	42	F0	6C	B1	0E
4C	E7	57	89	4E	9D	C7	2D	7E	74	A8	AF	FE	7B	1A	FF
1E	1B	45	4A	3D	5B	5E	7A	95	8E	31	C4	83	3E	A8	47
E0	A1	95	35	99	33	07	2D	D6	7F	3F	E5	E8	5C	20	9D
80	25	CC	B1	EF	7F	69	44	B6	AC	A4	EE	03	88	5C	B0

Destination Hardware Address	8B	BD	EB	D5	A3	00	10001011	10111101	11101011	11010101	10100011	00000000
Source Hardware Address	9E	CF	04	7F	AC	2B	10011110	11001111	00000100	01111111	10101100	00101011
Frame Type	08	00					00001000	00000000				
Vers & Len	45						01000101					
Type Of Service	11						00010001					
Total Length	00	72					00000000	01110010				
Ident	FC	53					11111100	01010011				
Flags & Fragment Offset	14	8D					00010100	10001101				
Flags												
Fragment Offset	14	8D					00010100	10001101				
TTL	47						01000111					
Type	36						00110110					
Header Checksum	9A	84					10011010	10000100				
Source IP Address	D2	92	8E	39			11010010	10010010	10001110	00111001		
Class	C											
Network	D2	92	8E				11010010	10010010	10001110			
Host	39						00111001					
Source IP Address (Decimal)	210	146	142	57								
Destination IP Address	49	16	75	02			01001001	00010110	01110101	00000010		
Class	A											
Network	49						01001001					
Host	16	75	02				00010110	01110101	00000010			
Destination IP Address (Decimal)	73	22	117	2								
Payload Data	D6	1B	BE	C8	33		11010110	00011011	10111110	11001000	00110011	

Destination Hardware Address	Source Hardware Address	Frame Type	Frame Data
6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes

Byte	0								1								2								3							
bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Version				Header Length				Type Of Service								Total Length															
4	Identification																Flags				Fragment Offset											
8	TTL								Type								Header Checksum															
12	Source IP Address																															
16	Destination IP Address																															
Optional	IP Options (May Be Omitted)																								Padding							
20	IP Payload Data																															

4B	A8	A0	4E	76	1B	D1	1A	9D	41	79	BD	08	00	45	61
00	72	CA	0C	33	6A	BC	EA	F7	21	21	09	7F	7A	C1	2B
02	5D	7E	7D	37	33	97	91	CA	7A	0C	5A	AE	A0	A7	8A
CF	56	07	DC	79	35	EA	BD	DE	11	6E	12	77	81	D8	33
88	73	1E	75	02	70	20	7B	2C	96	61	DE	E2	27	75	29
19	52	9A	87	C4	CA	1A	96	1C	72	0C	BF	A8	2F	84	A3
BF	C7	CC	A4	6E	37	99	13	44	48	C5	D8	39	22	94	72
95	13	D5	DD	91	F7	A7	EF	E0	AB	30	7F	8E	54	7B	B4

Destination Hardware Address	4B	A8	A0	4E	76	1B	01001011	10101000	10100000	01001110	01110110	00011011
Source Hardware Address	D1	1A	9D	41	79	BD	11010001	00011010	10011101	01000001	01111001	10111101
Frame Type	08	00					00001000	00000000				
Vers & Len	45						01000101					
Type Of Service	61						01100001					
Total Length	00	72					00000000	01110010				
Ident	CA	0C					11001010	00001100				
Flags & Fragment Offset	33	6A					00110011	01101010				
Flags	More Fragments											
Fragment Offset	13	6A					00010011	01101010				
TTL	BC						10111100					
Type	EA						11101010					
Header Checksum	F7	21					11110111	00100001				
Source IP Address	21	09	7F	7A			00100001	00001001	01111111	01111010		
Class	A											
Network	21						00100001					
Host	09	7F	7A				00001001	01111111	01111010			
Source IP Address (Decimal)	33	9	127	122								
Destination IP Address	C1	2B	02	5D			11000001	00101011	00000010	01011101		
Class	C											
Network	C1	2B	02				11000001	00101011	00000010			
Host	5D						01011101					
Destination IP Address (Decimal)	193	43	2	93								
Payload Data	7E	7D	37	33	97		01111110	01111101	00110111	00110011	10010111	

Destination Hardware Address	Source Hardware Address	Frame Type	Frame Data
6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes

Byte	0								1								2								3							
bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Version				Header Length				Type Of Service								Total Length															
4	Identification																Flags			Fragment Offset											
8	TTL								Type								Header Checksum															
12	Source IP Address																															
16	Destination IP Address																															
Optional	IP Options (May Be Omitted)																								Padding							
20	IP Payload Data																															

FRAG OFFSET
0A9A

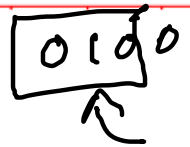
FLAGS
010

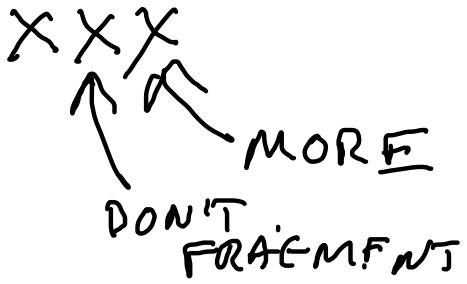
87	36	8F	42	77	95	83	1A	AB	39	16	D9	08	00	45	EE
00	72	B3	45	4A	9A	E0	E0	BC	20	39	16	85	EA	DE	78
12	02	B3	86	BE	AA	B7	06	0C	15	71	87	B1	85	28	59
F9	68	E9	13	C5	B7	76	2C	A9	B4	C9	78	1C	42	39	AE
8C	54	EB	E7	DA	BB	05	CF	F4	BA	FD	5B	1C	42	4A	8D
61	FD	13	4F	2B	02	36	99	30	67	43	28	C1	98	C7	03
F1	80	ED	5F	1F	31	05	04	E6	41	70	E5	26	47	4A	19
A6	1C	CD	DA	14	5C	CA	AD	D2	72	CB	71	42	93	08	01

4A 9A

FLAGS 0 10 | 0 1010 1001 1010 FRAG OFFSET

Destination Hardware Address	87	36	8F	42	77	95	10000111	00110110	10001111	01000010	01110111	10010101
Source Hardware Address	83	1A	AB	39	16	D9	10000011	00011010	10101011	00111001	00010110	11011001
Frame Type	08	00					00001000	00000000				
Vers & Len	45						01000101					
Type Of Service	EE						11101110					
Total Length	00	72					00000000	01110010				
Ident	B3	45					10110011	01000101				
Flags & Fragment Offset	4A	9A					01001010	10011010				
Flags	Don't Fragment											
Fragment Offset	0A	9A					00001010	10011010				
TTL	E0						11100000					
Type	E0						11100000					
Header Checksum	BC	20					10111100	00100000				
Source IP Address	39	16	85	EA			00111001	00010110	10000101	11101010		
Class	A											
Network	39						00111001					
Host	16	85	EA				00010110	10000101	11101010			
Source IP Address (Decimal)	57	22	133	234								
Destination IP Address	DE	78	12	02			11011110	01111000	00010010	00000010		
Class	C											
Network	DE	78	12				11011110	01111000	00010010			
Host	02						00000010					
Destination IP Address (Decimal)	222	120	18	2								
Payload Data	B3	86	BE	AA	B7		10110011	10000110	10111110	10101010	10110111	





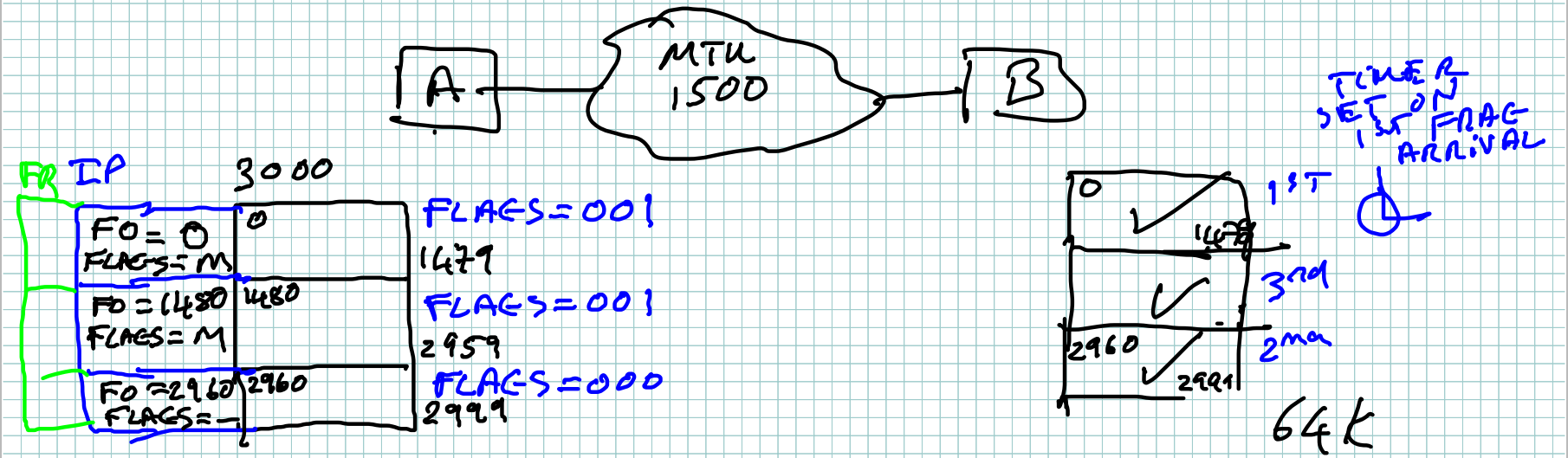
Standards For Encapsulation

- TCP/IP protocols define encapsulation for each possible type of network hardware
  - Ethernet
  - Frame Relay
  - Others

A Potential Problem

64K = 64 x 1024 = 65535

- A datagram can contain up to 65535 total octets (including header)
- Network hardware limits maximum size of frame (e.g., Ethernet limited to 1500 octets)
  - Known as the network *Maximum Transmission Unit* (*MTU*)
- Question: how is encapsulation handled if datagram exceeds network MTU?


A
MTU
1500
B
FR IP
3000
FO = 0
FLAGS = M
0
1479
FLAGS = 001
FO = 1480
FLAGS = M
1480
2959
FLAGS = 001
FO = 2960
FLAGS = -
2960
2999
FLAGS = 000
TIMER SET ON 1ST FRAG ARRIVAL
0
1479
1ST
2960
2999
2ND
3RD
64K

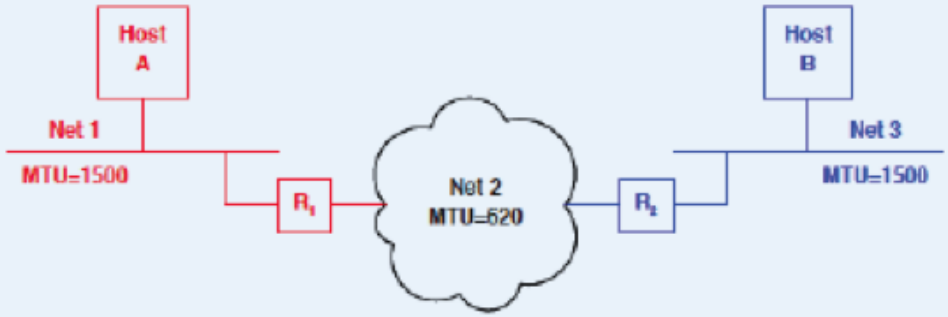
Possible Ways To Accommodate Networks With Differing MTUs

- Force datagram to be less than smallest possible MTU
  - Inefficient
  - Cannot know minimum MTU
- Hide the network MTU and accommodate arbitrary datagram size

Accommodating Large Datagrams

- Cannot send large datagram in single frame
- Solution
  - Divide datagram into pieces
  - Send each piece in a frame
  - Called *datagram fragmentation*

Illustration Of When Fragmentation Needed

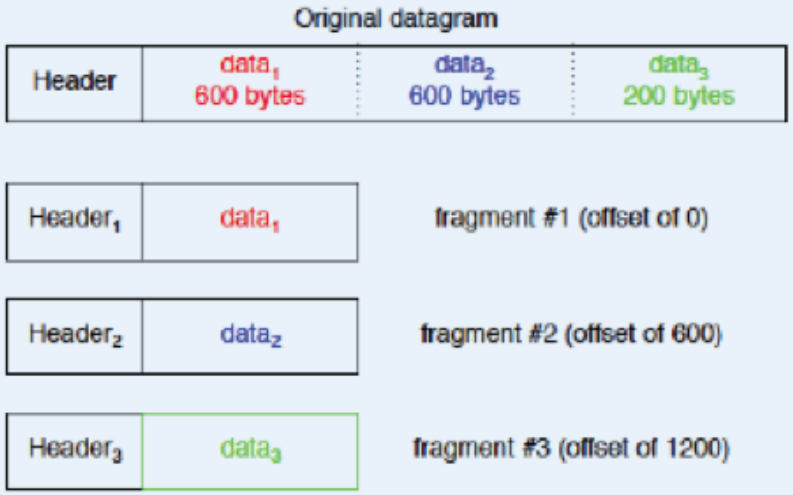


- Hosts A and B send datagrams of up to 1500 octets
- Router R_1 fragments large datagrams from Host A before sending over Net 2
- Router R_2 fragments large datagrams from Host B before sending over Net 2

Datagram Fragmentation

- Performed by routers
- Divides datagram into several, smaller datagrams called fragments
- Fragment uses same header format as datagram
- Each fragment forwarded independently

Illustration Of Fragmentation



- Offset specifies where data belongs in original datagram
- Offset actually stored as multiples of 8 octets
- MORE FRAGMENTS bit turned off in header of fragment #3

Fragmenting A Fragment

- Fragment can be further fragmented
- Occurs when fragment reaches an even-smaller MTU
- Discussion: which fields of the datagram header are used, and what is the algorithm?

Reassembly

- Ultimate destination puts fragments back together
  - Key concept!
  - Needed in a connectionless Internet
- Known as *reassembly*
- No need to reassemble subfragments first
- Timer used to ensure all fragments arrive
  - Timer started when first fragment arrives
  - If timer expires, entire datagram discarded

Time To Live

- TTL field of datagram header decremented at each hop (i.e., each router)
- If TTL reaches zero, datagram discarded
- Prevents datagrams from looping indefinitely (in case forwarding error introduces loop)
- IETF recommends initial value of 255 (max)

Checksum Field In Datagram Header

- 16-bit 1's complement checksum
- Over IP header only!
- Recomputed at each hop

IP Options

- Seldom used
- Primarily for debugging
- Only *some* options copied into fragments
- Are variable length
- Note: padding needed because header length measured in 32-bit multiples
- Option starts with option code octet

IP Semantics

- IP uses best-effort delivery
  - Makes an attempt to deliver
  - Does not guarantee delivery
- In the Internet, routers become overrun or change routes, meaning that:
  - Datagrams can be lost
  - Datagrams can be duplicated
  - Datagrams can arrive out of order or scrambled
- Motivation: allow IP to operate over the widest possible variety of physical networks

Summary

- Internet Protocol provides basic connectionless delivery service for the Internet
- IP defines *IP datagram* to be the format of packets on the Internet
- Datagram header
  - Has fixed fields
  - Specifies source, destination, and type
  - Allows options
- Datagram encapsulated in network frame for transmission

Summary
(continued)

- Fragmentation
  - Needed when datagram larger than MTU
  - Usually performed by routers
  - Divides datagram into fragments
- Reassembly
  - Performed by ultimate destination
  - If some fragment(s) do not arrive, datagram discarded
- To accommodate all possible network hardware, IP does not require reliability (best-effort semantics)