

Network Layer Part 2

Mark Allman mallman@case.edu

EECS 325/425 Fall 2018

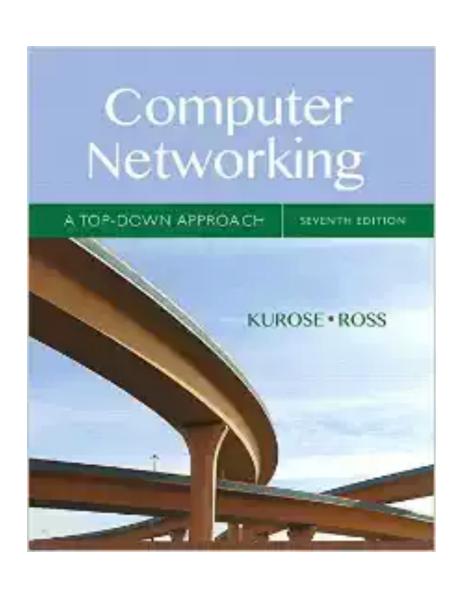
#Rally

These slides are more-or-less directly from the slide set developed by Jim Kurose and Keith Ross for their book "Computer Networking: A Top Down Approach, 5th edition".

The slides have been lightly adapted for Mark Allman's EECS 325/425 Computer Networks class at Case Western Reserve University.

All material copyright 1996-2010 J.F Kurose and K.W. Ross, All Rights Reserved

Reading Along ...



- Network layer is chapters 4
 - 4,2: what's inside a router?

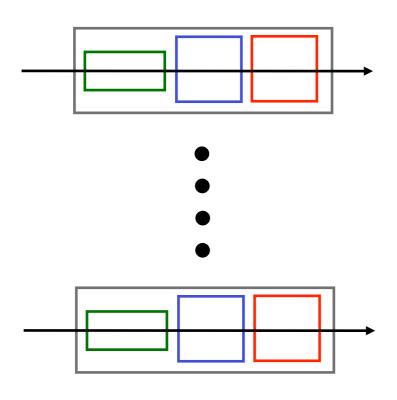
two key router functions:

two key router functions:

- *run routing algorithms/protocol (OSPF, BGP)
- forwarding datagrams from incoming to outgoing link

two key router functions:

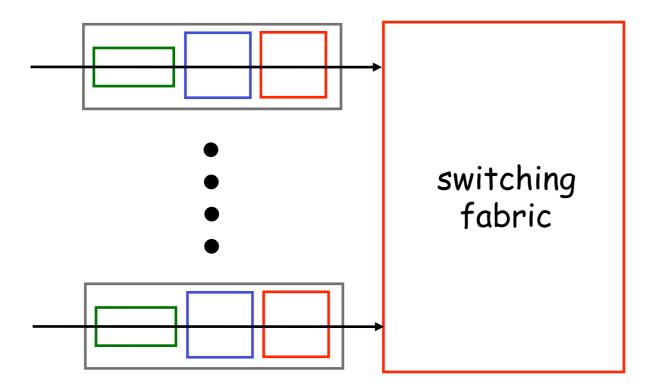
- *run routing algorithms/protocol (OSPF, BGP)
- *forwarding datagrams from incoming to outgoing link



router input ports

two key router functions:

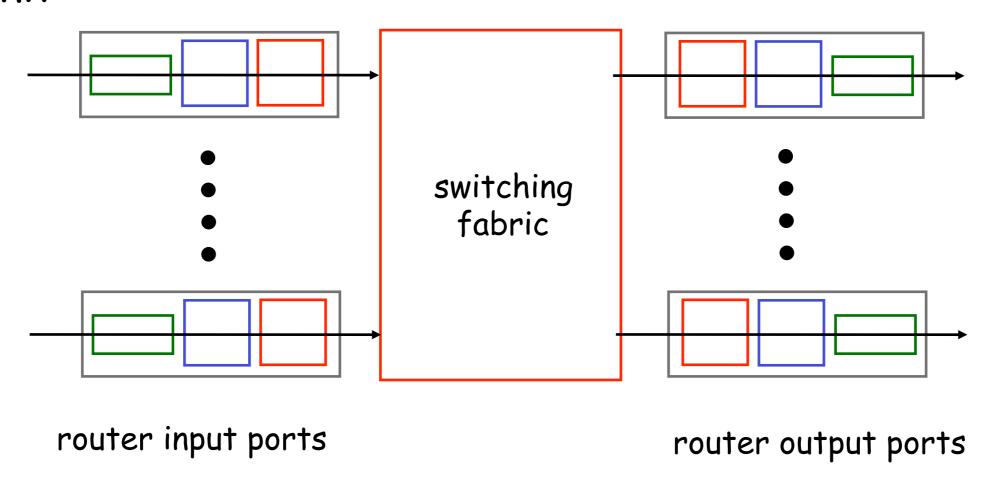
- *run routing algorithms/protocol (OSPF, BGP)
- *forwarding datagrams from incoming to outgoing link



router input ports

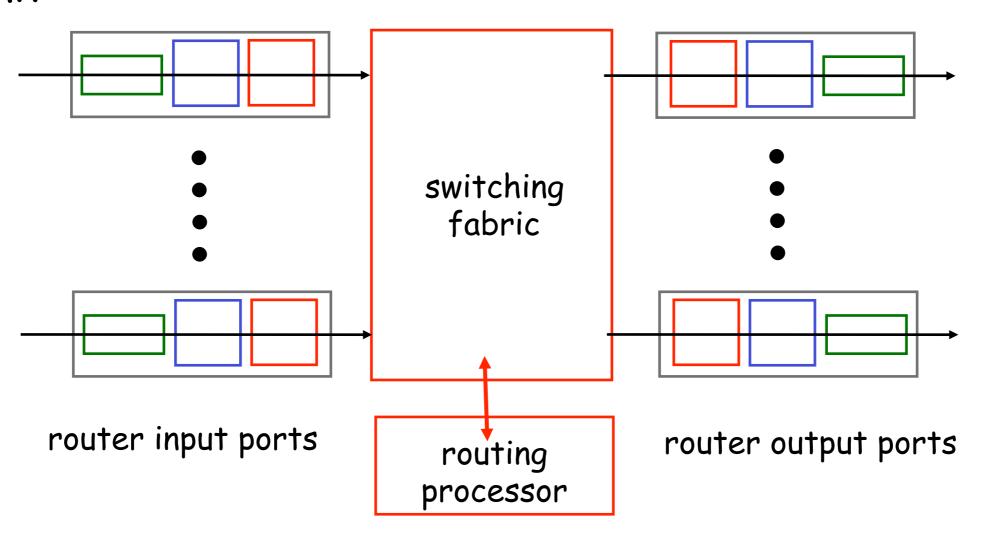
two key router functions:

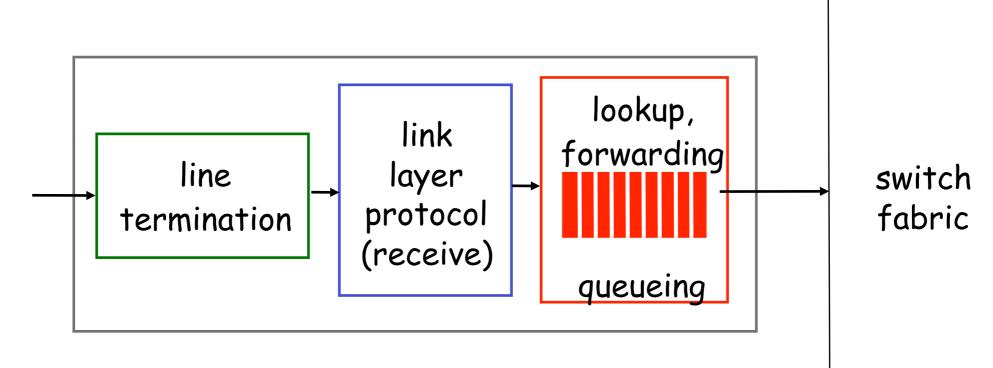
- *run routing algorithms/protocol (OSPF, BGP)
- *forwarding datagrams from incoming to outgoing link

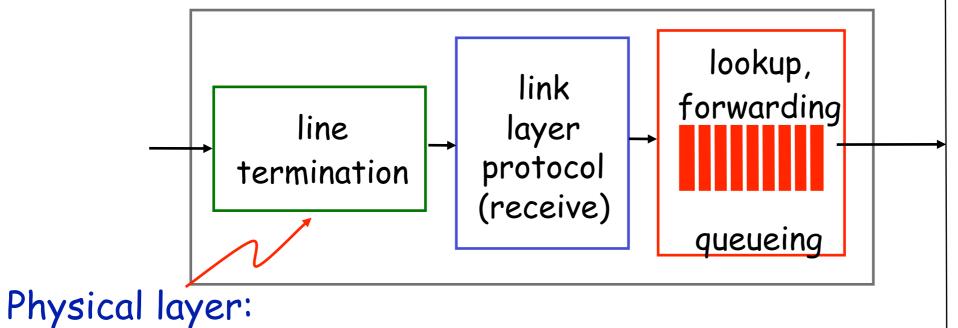


two key router functions:

- *run routing algorithms/protocol (OSPF, BGP)
- forwarding datagrams from incoming to outgoing link



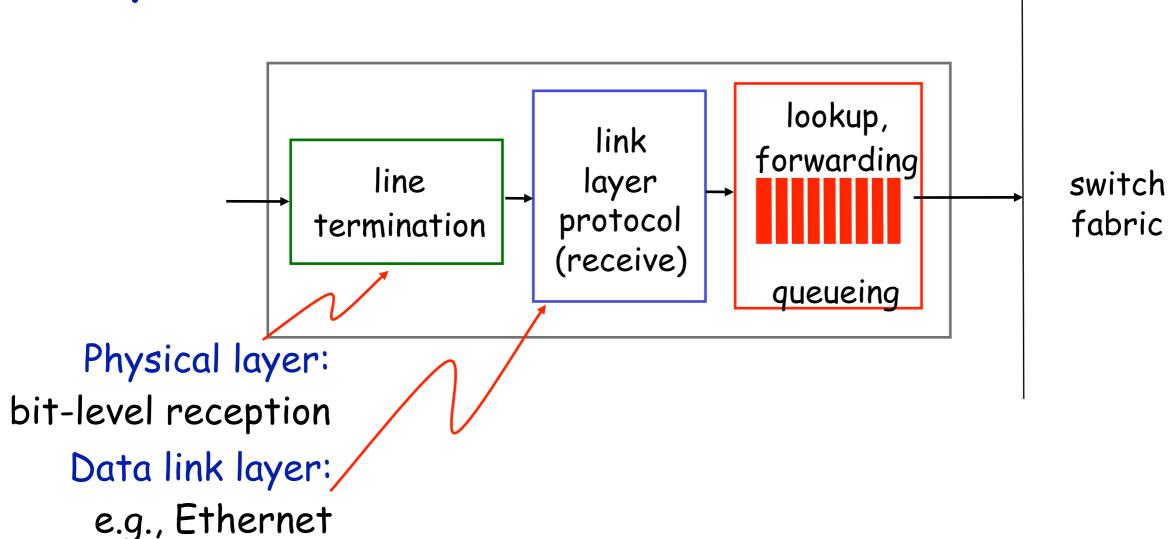




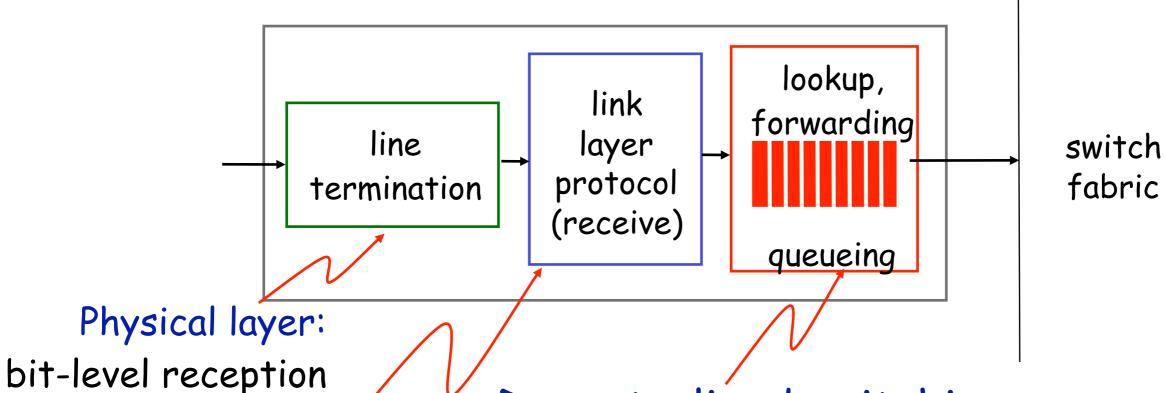
bit-level reception

switch

fabric



Input Port Functions lookup, link forwarding line layer switch protocol fabric termination (receive) queueing Physical layer: bit-level reception Decentralized switching: Data link layer: e.g., Ethernet

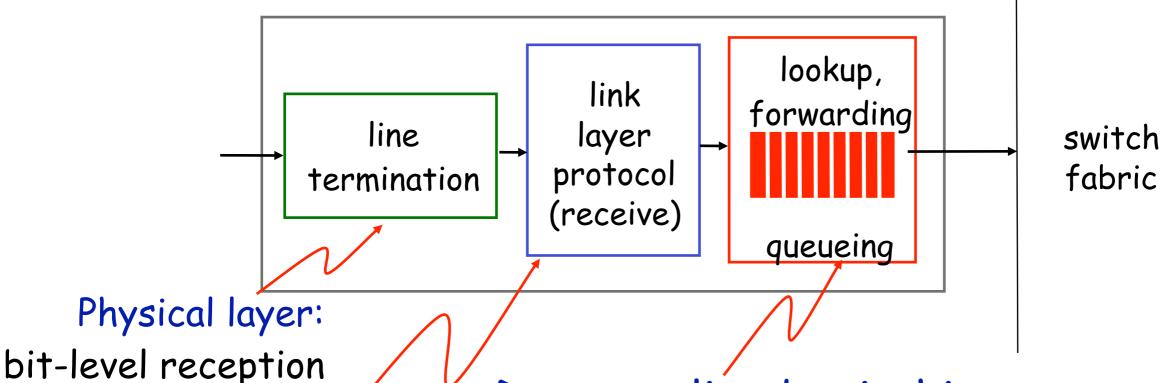


Data link layer:

e.g., Ethernet

Decentralized switching:

*given datagram dest., lookup output port using forwarding table in input port memory



Data link layer:

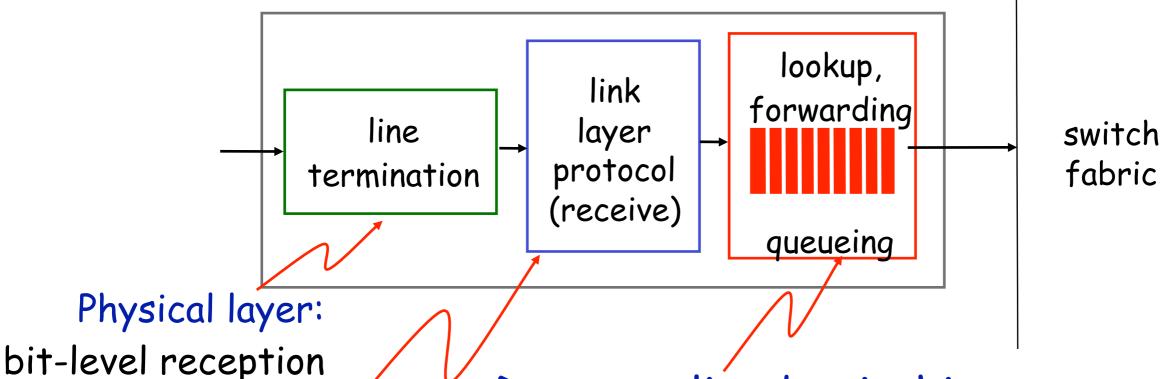
e.g., Ethernet

Decentralized switching:

- *given datagram dest., lookup output port using forwarding table in input port memory
- *goal: complete input port processing at 'line speed'

Data link layer:

e.g., Ethernet



Decentralized switching:

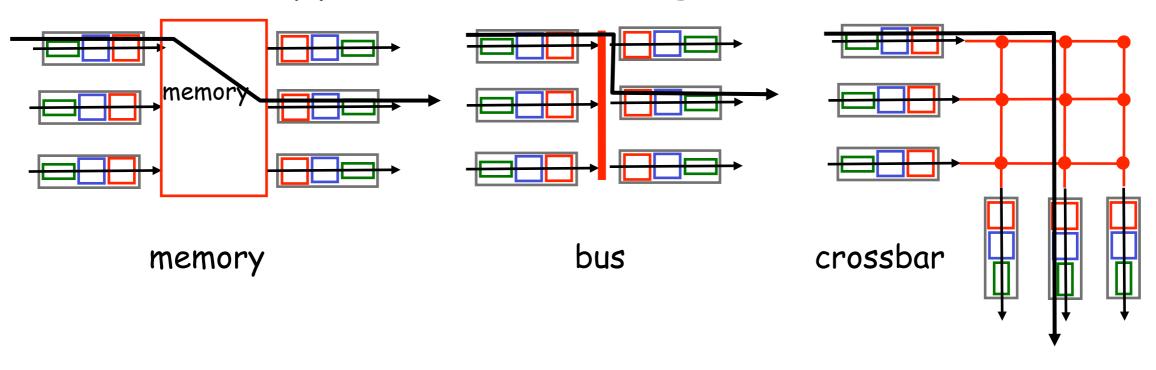
- *given datagram dest., lookup output port using forwarding table in input port memory
- *goal: complete input port processing at 'line speed'
- *queuing: if datagrams arrive faster than forwarding rate into switch fabric

Network Layer

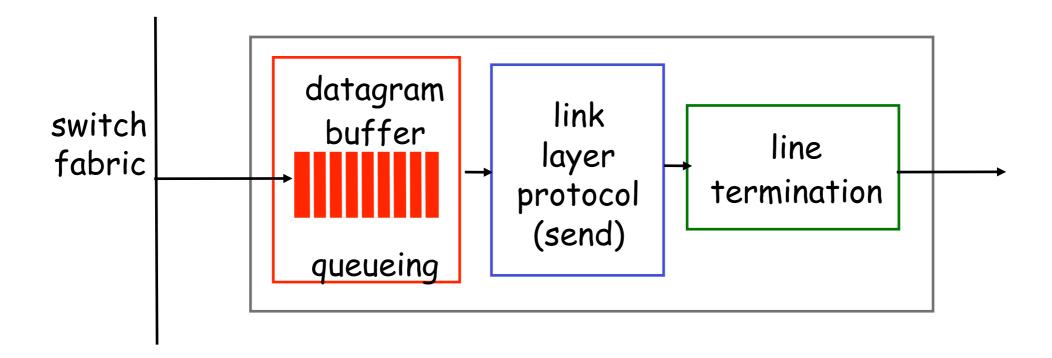
*transfer packet from input buffer to appropriate output buffer

- *transfer packet from input buffer to appropriate output buffer
- *switching rate: rate at which packets can be transfer from inputs to outputs
 - often measured as multiple of input/output line rate
 - N inputs: switching rate N times line rate desirable

- *transfer packet from input buffer to appropriate output buffer
- *switching rate: rate at which packets can be transfer from inputs to outputs
 - often measured as multiple of input/output line rate
 - N inputs: switching rate N times line rate desirable
- *three types of switching fabrics

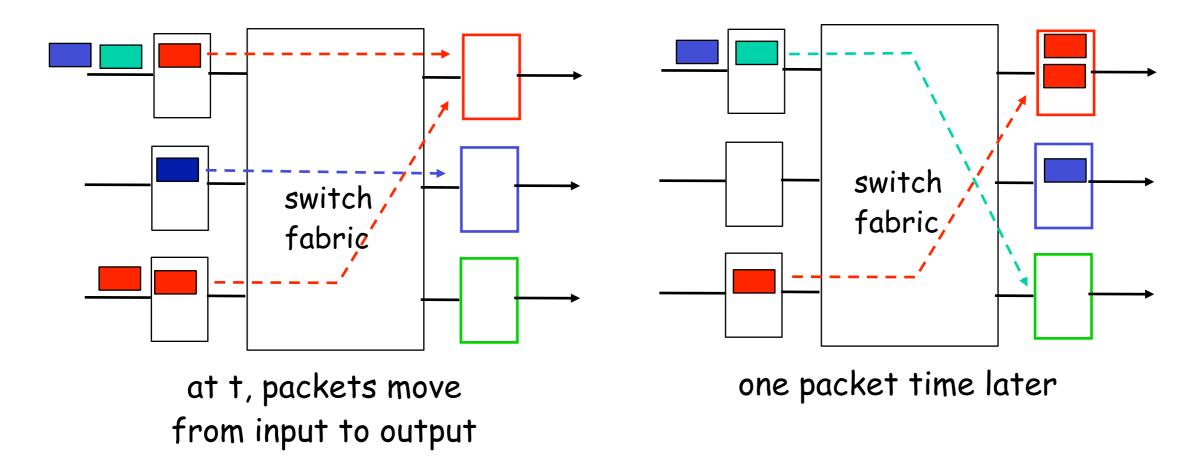


Output Ports



- *buffering required when datagrams arrive from fabric faster than the transmission rate
- *scheduling discipline chooses among queued datagrams for transmission

Output port queueing



- *buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!

How much buffering?

How much buffering?

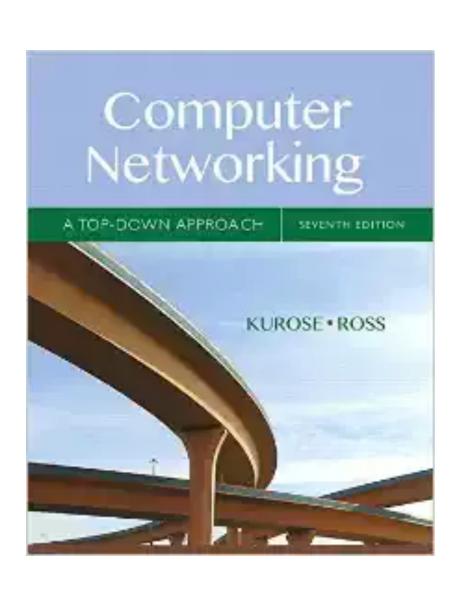
- *RFC 3439 rule of thumb: average buffering equal to "typical" RTT (say 100 msec) times link capacity C
 - e.g., C = 10 Gpbs link: 1 Gbit buffer
 - comes from simple model of TCP behavior (which we'll get to)

How much buffering?

- *RFC 3439 rule of thumb: average buffering equal to "typical" RTT (say 100 msec) times link capacity C
 - e.g., C = 10 Gpbs link: 1 Gbit buffer
 - comes from simple model of TCP behavior (which we'll get to)

*recent recommendation: with N flows, buffering equal to RTT.C

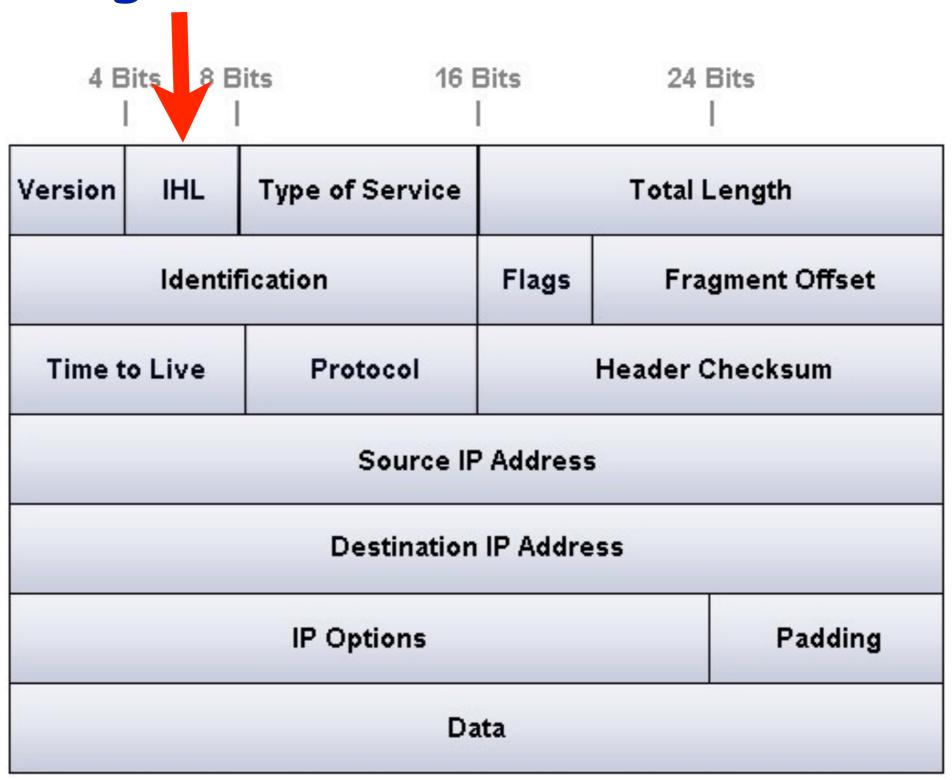
Reading Along ...

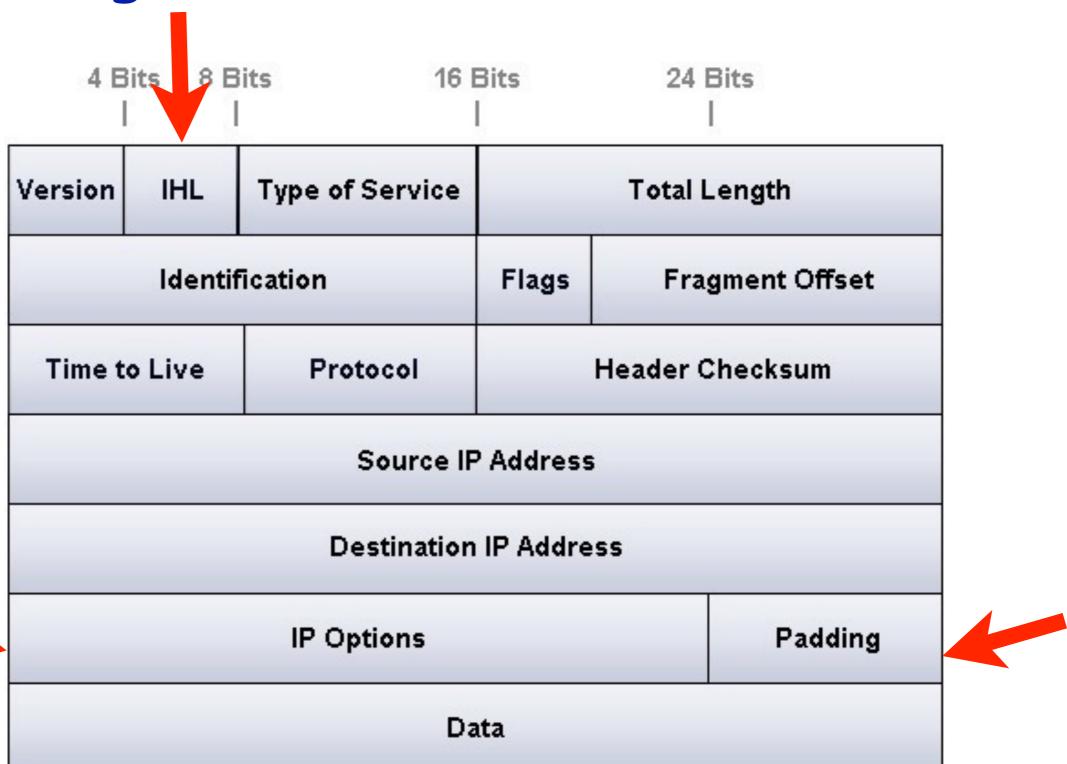


- Network layer is chapters 4
 - 4.3: Internet Protocol
 - datagram format

4 Bi	ts 8B	its 16 E	3its	24 Bits						
Version	IHL	Type of Service		Total Length						
•	ldentif	ication	Flags	Fragment Offset						
Time to	Live	Protocol	Header Checksum							
		Source IP	Address							
		Destination	IP Addre	ss						
IP Options Padding										
		Da	ta							

4 Bi	its 8 E	Bits 16 B	Bits	24 Bits				
Version	IHL	Type of Service		Total Length				
	Identii	fication	Flags	Fragment Offset				
Time to	Live	Protocol	Header Checksum					
	Source IP Address							
		Destination	IP Addre	ss				
IP Options Padding								
		Da	ita					

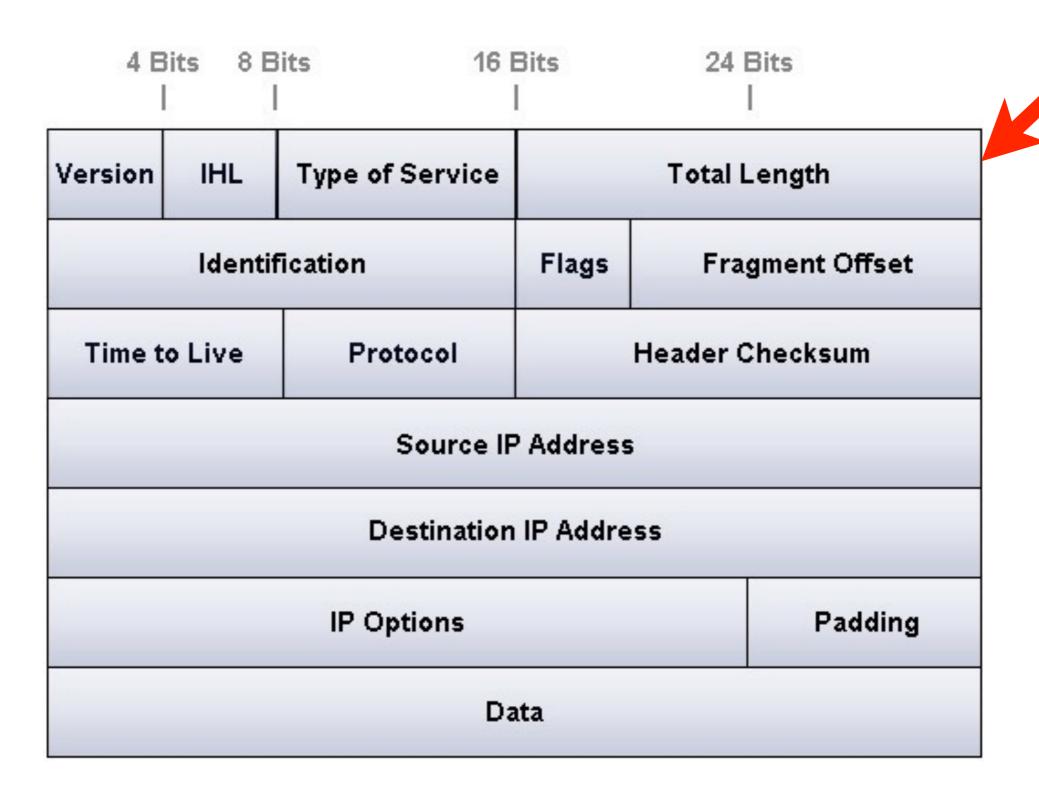


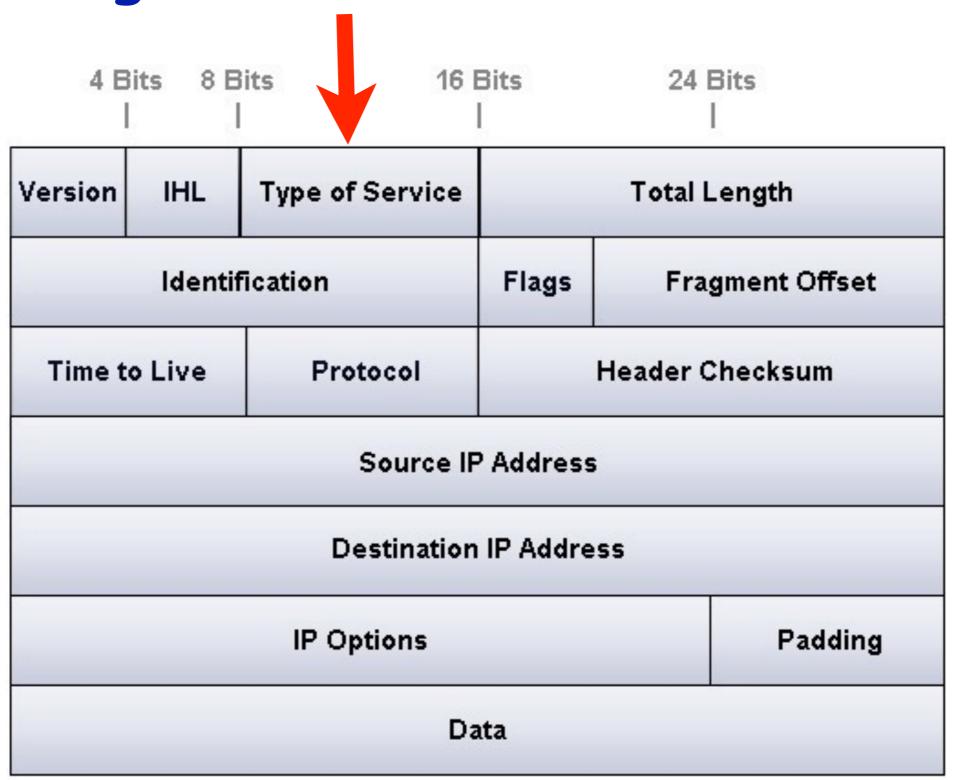


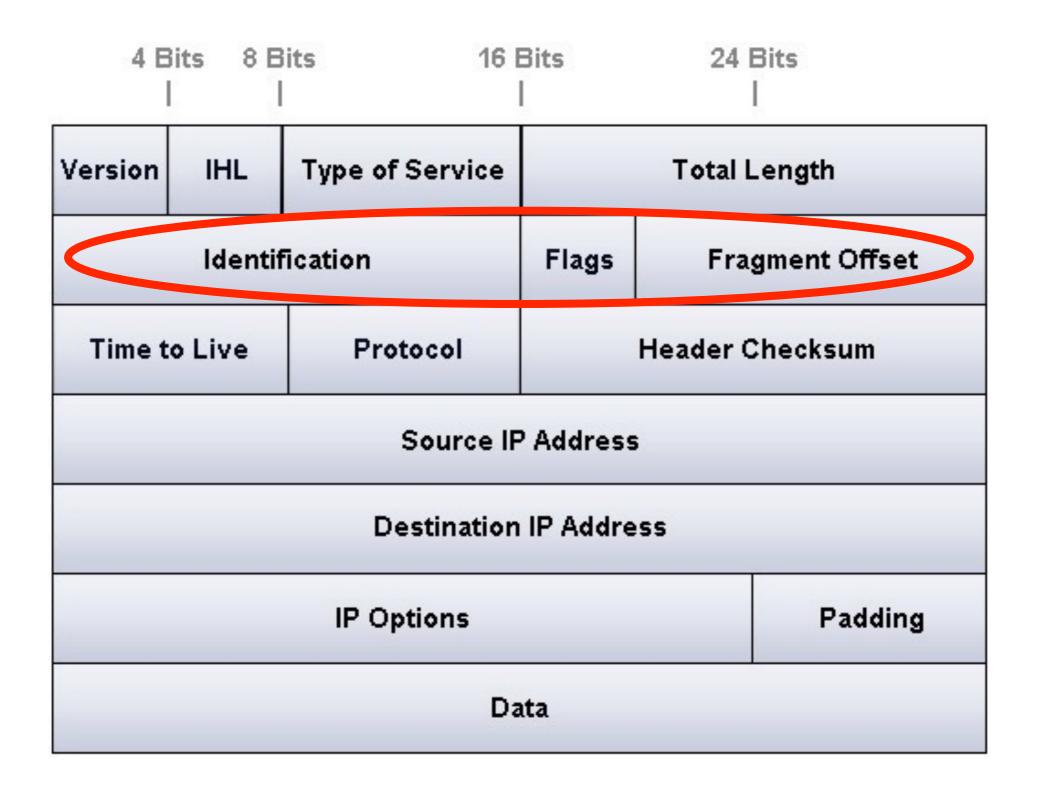
TCP Packet Length Issue ...

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	3
	Source Po	rt												Des	stina	tion	Port						
					S	Gequ	enc	e N	umbe	er													
				Д	ckn	owle	edge	mei	nt Nu	mbe	r												
HLEN Reserved R C S S Y I Window																							
	Checksum	1												Urg	gent	Poir	nter						
		Opt	ions	(if a	ny)			ė.											Pad	ding			
							D:	ata								•							
							10.	·.															

4 Bi	ts 8B	its 16 E	3its	24 Bits						
Version	IHL	Type of Service		Total Length						
•	ldentif	ication	Flags	Fragment Offset						
Time to	Live	Protocol	Header Checksum							
		Source IP	Address							
		Destination	IP Addre	ss						
IP Options Padding										
		Da	ta							







	4 B	its 8 B	its 16 I	Bits	24 Bits						
	Version	IHL	Type of Service		Total Length						
		ldentif	ication	Flags	Fragment Offset						
>	Time to	Live	Protocol		Header Checksum						
		Source IP Address									
	Destination IP Address										
	IP Options Padding										
		Data									

