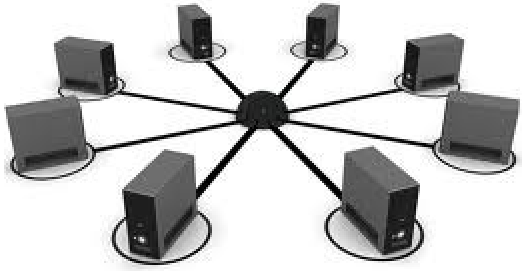
Network Layer



Goals

Understand principles behind network layer services:

- network layer service models
- forwarding versus routing
- how a router works
- routing (path selection)
- dealing with scale
- advanced topics: IPv6, mobility

Network layer

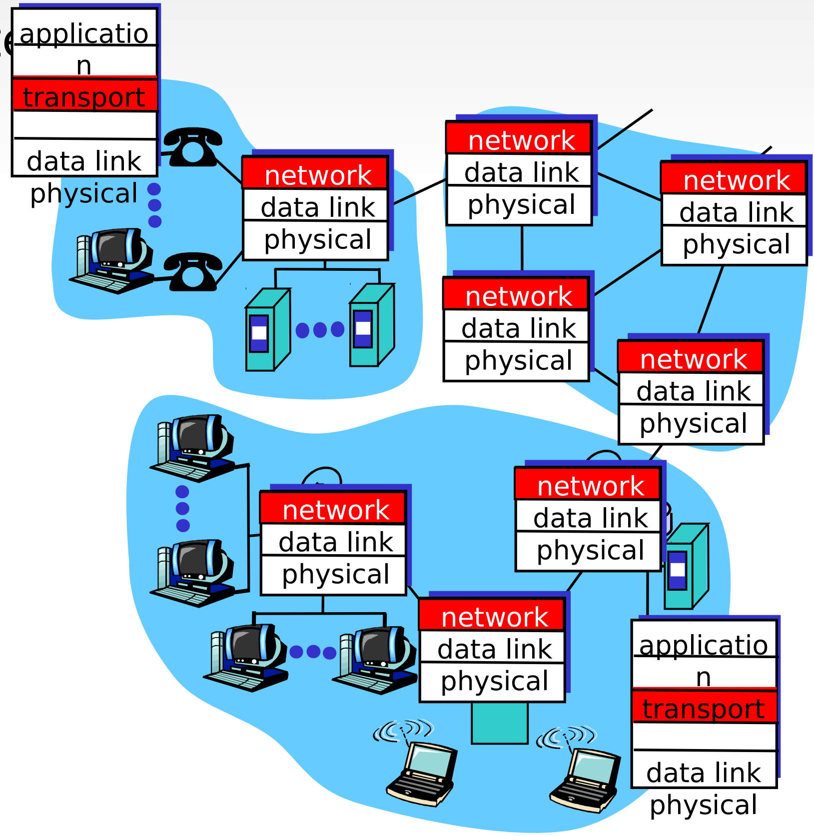
-transport segment from sending to receiving host

-on sending side encapsulates segments into datagrams

-on rcving side, delivers segments to transport layer

-network layer protocols in *every* host, router

-Router examines header fields in all
 IP datagrams passing through it



Two Key Network-Layer Functions

forwarding: move packets from router's input to appropriate router output

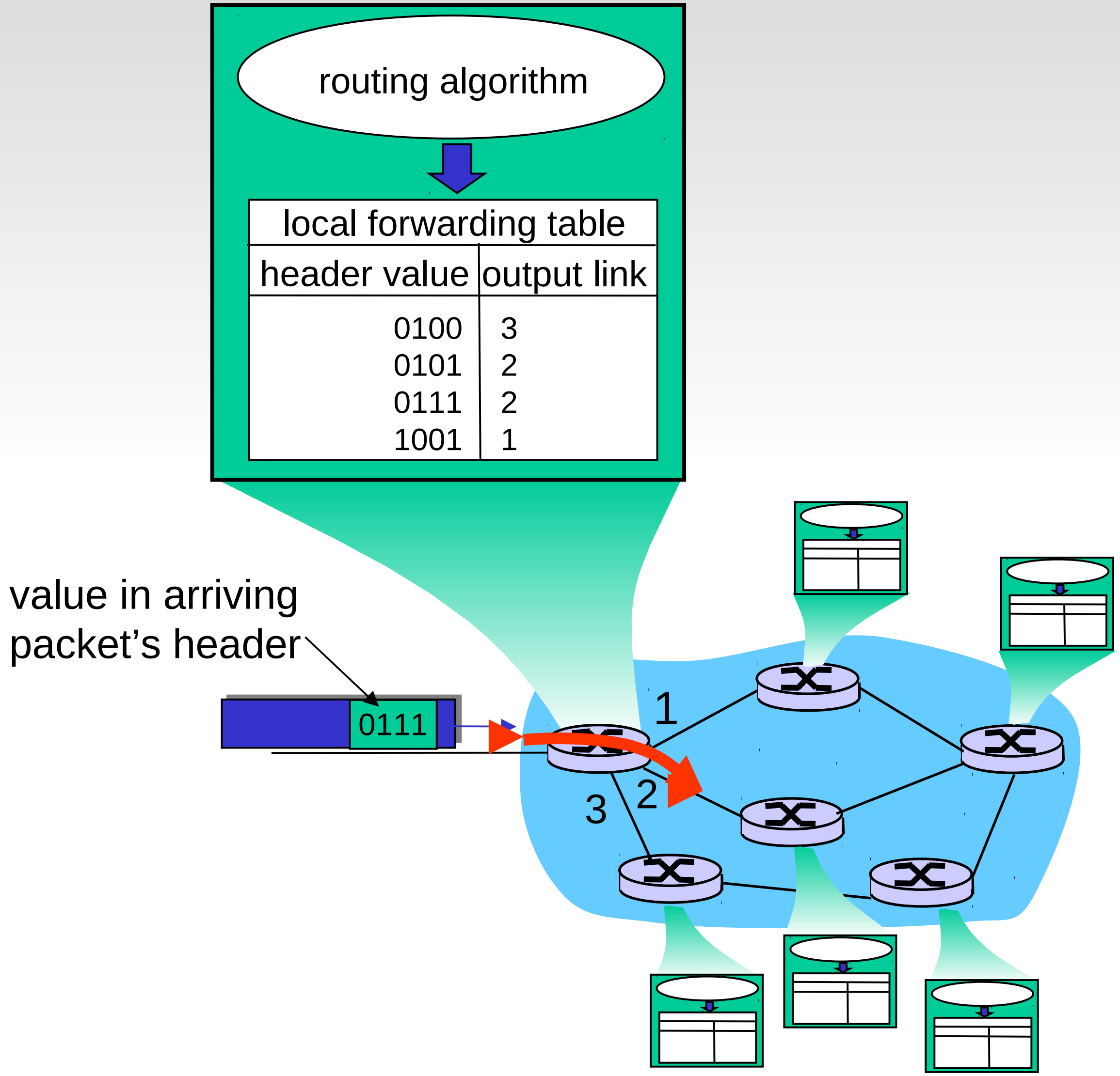
routing: determine route taken by packets from source to dest.

routing algorithms

analogy:

- routing: process of planning trip from source to dest
- forwarding: process of getting through single interchange

Interplay between routing and forwarding



Connection setup

Important function in *some* network architectures:

- ATM, frame relay, X.25

before datagrams flow, two end hosts *and* intervening routers establish virtual connection

- routers get involved

network vs transport layer connection service:

- network: between two hosts (may also involve intervening routers in case of VCs)
- transport: between two processes

Network layer service models:

Network Architecture	Service Model	Guarantees ? Bandwidth	Loss	Order	Timing	Congestion feedback
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

Network layer connection and connection-less service

Datagram network provides network-layer connectionless service

VC network provides network-layer connection service

analogous to the transport-layer services, but:

service: host-to-host

no choice: network provides one or the other

implementation: in network core

Virtual circuits

"source-to-dest path behaves much like telephone circuit"

performance-wise

network actions along source-to-dest path

- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- *every* router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)

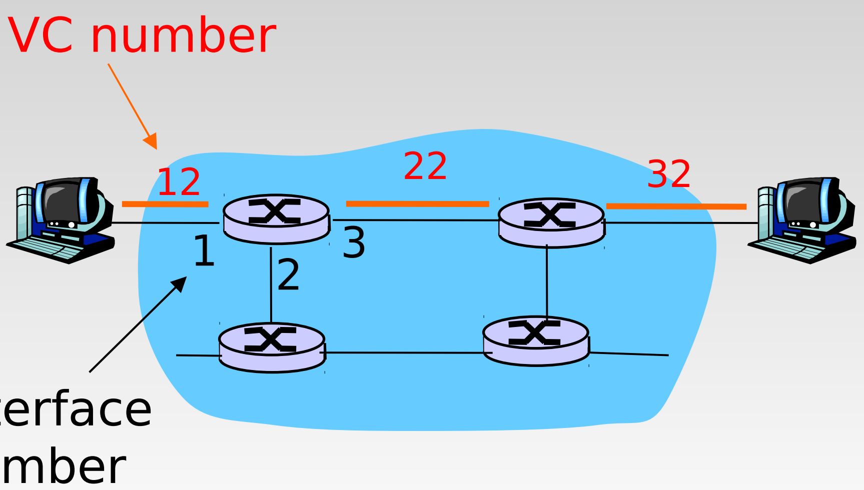
VC implementation

a VC consists of:

- path from source to destination
- VC numbers, one number for each link along path
- entries in forwarding tables in routers along path

- packet belonging to VC carries VC number (rather than dest address)
- VC number can be changed on each link.
  - New VC number comes from forwarding table

Forwarding table



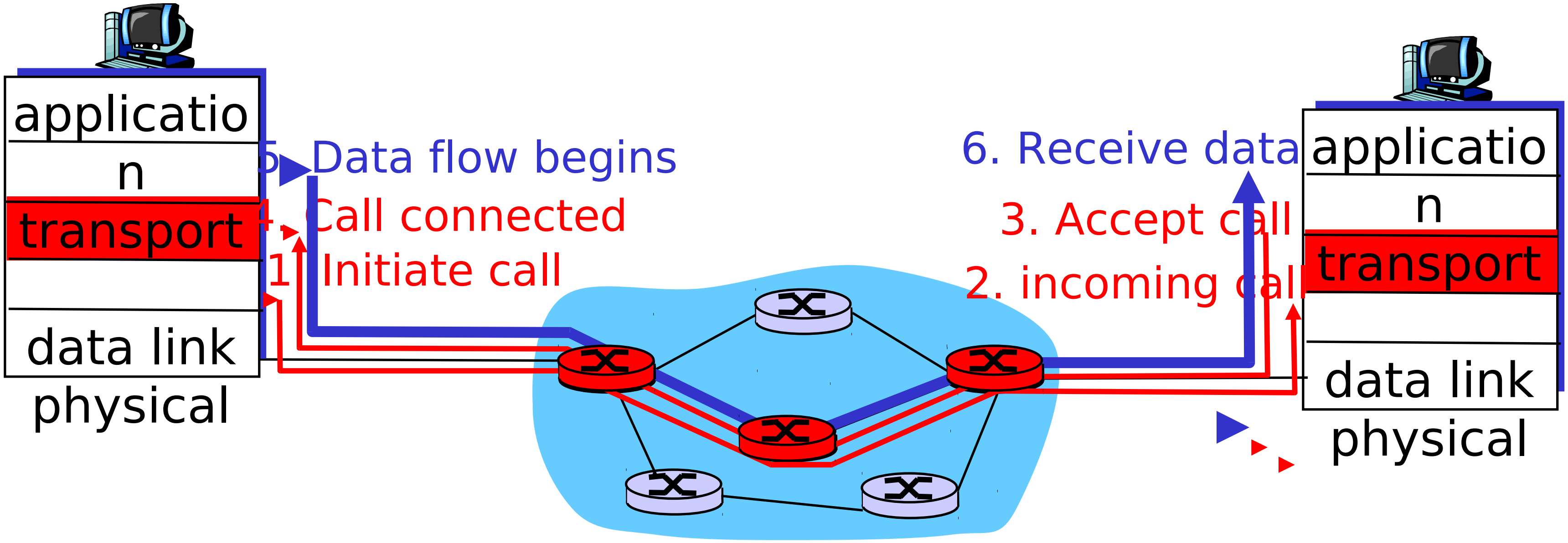
Forwarding table in northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

Routers maintain connection state information!

Virtual circuits: signaling protocols

used to setup, maintain teardown VC
used in ATM, frame-relay, X.25
not used in today’s Internet



Datagram networks

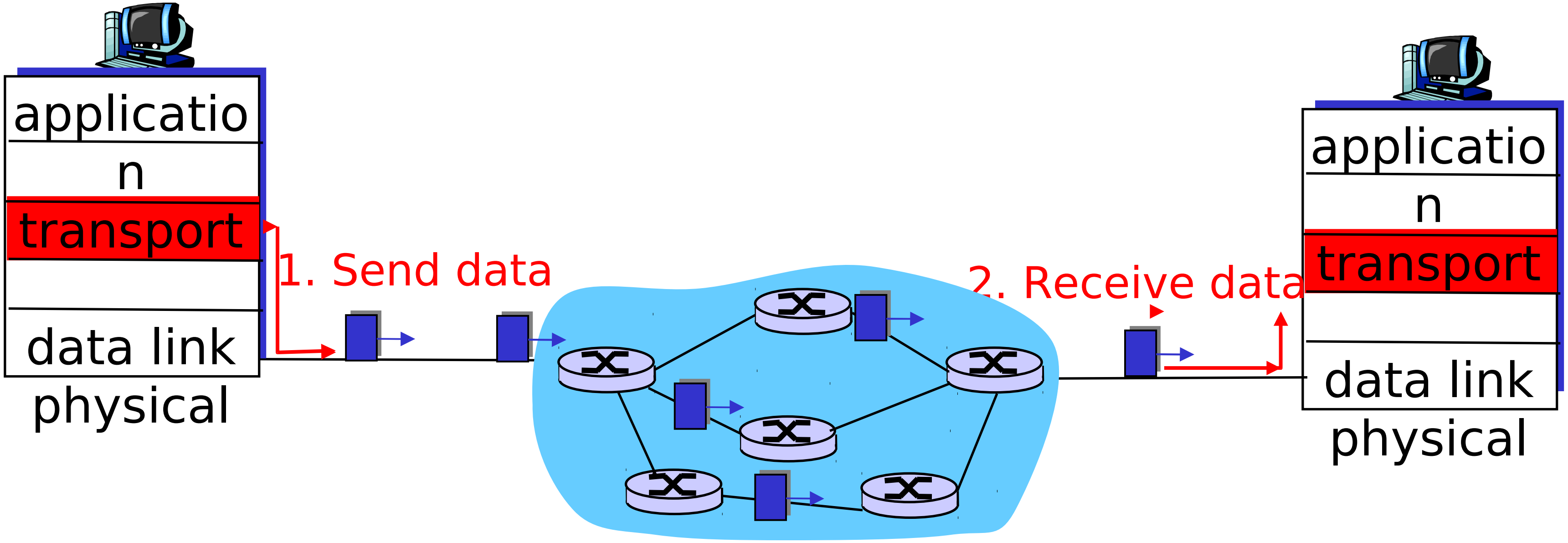
-no call setup at network layer

-routers: no state about end-to-end connections

- no network-level concept of "connection"

-packets forwarded using destination host address

- packets between same source-dest pair may take different paths



Forwarding table

4 billion
possible entries

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Longest prefix matching

Prefix Match	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

Examples

DA: 11001000 00010111 00010110 10100001   Which interface?

DA: 11001000 00010111 00011000 10101010   Which interface?

Refer the research paper uploaded in photon for more details

Datagram or VC network: why?

Internet (datagram)

data exchange among computers

"elastic" service, no strict timing req.

"smart" end systems (computers)

can adapt, perform control, error recovery

simple inside network, complexity at "edge"

many link types

different characteristics

uniform service difficult

ATM (VC)

evolved from telephony

human conversation:

strict timing, reliability requirements

need for guaranteed service

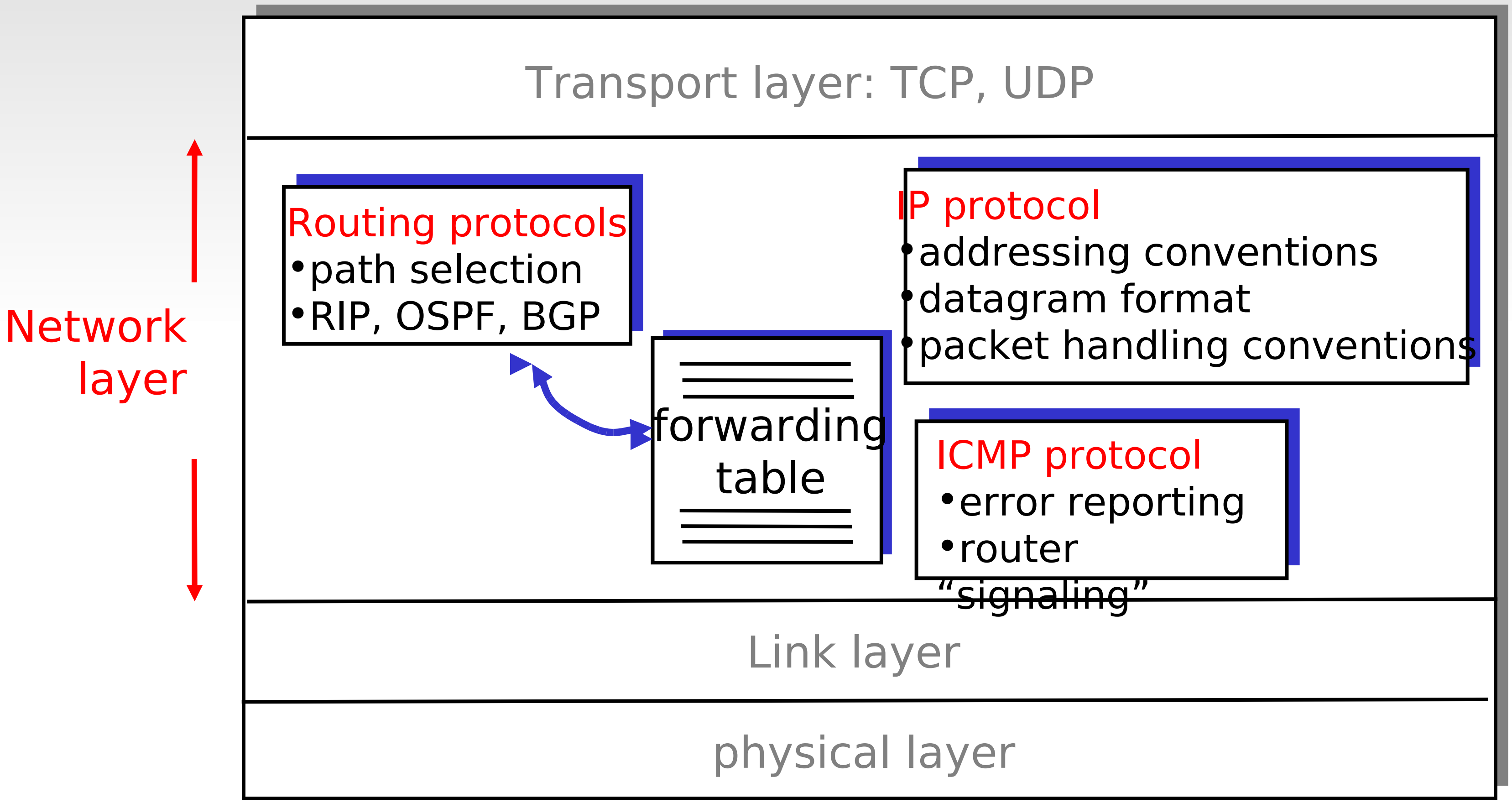
"dumb" end systems

telephones

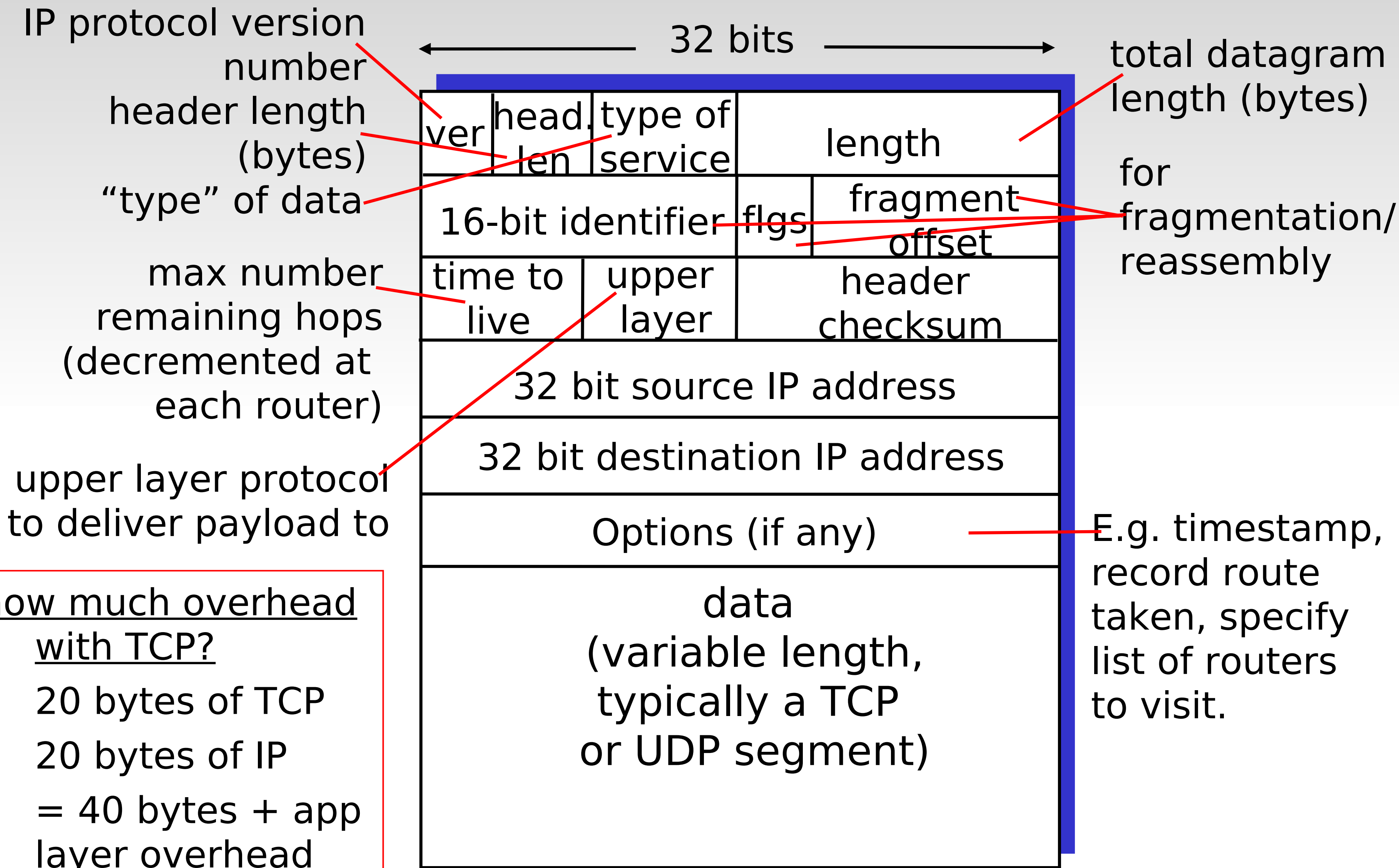
complexity inside network

The Internet Network layer

Host, router network layer functions:



IP datagram format



<u>how much overhead with TCP?</u>

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

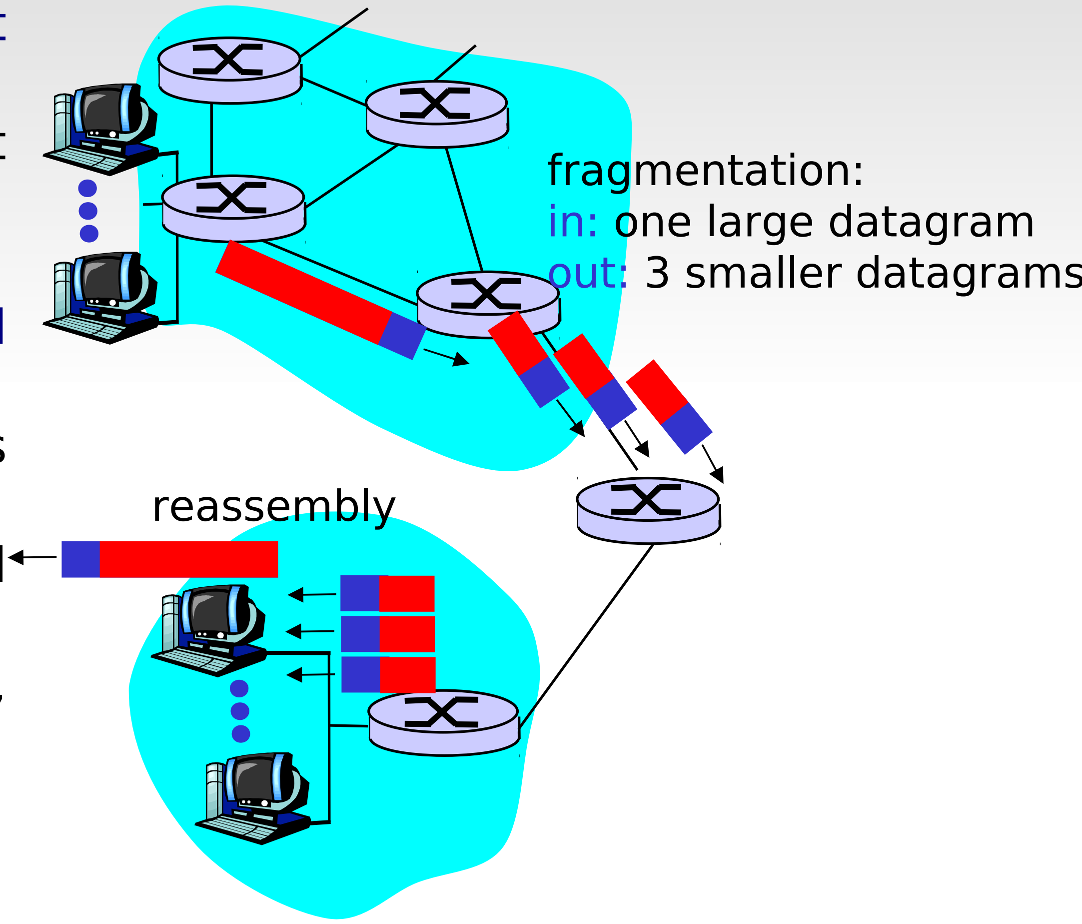
IP Fragmentation & Reassembly

Network links have MTU (max.transfer size) - largest possible link-level frame.

-different link types, different MTUs

Large IP datagram divided ("fragmented") within net

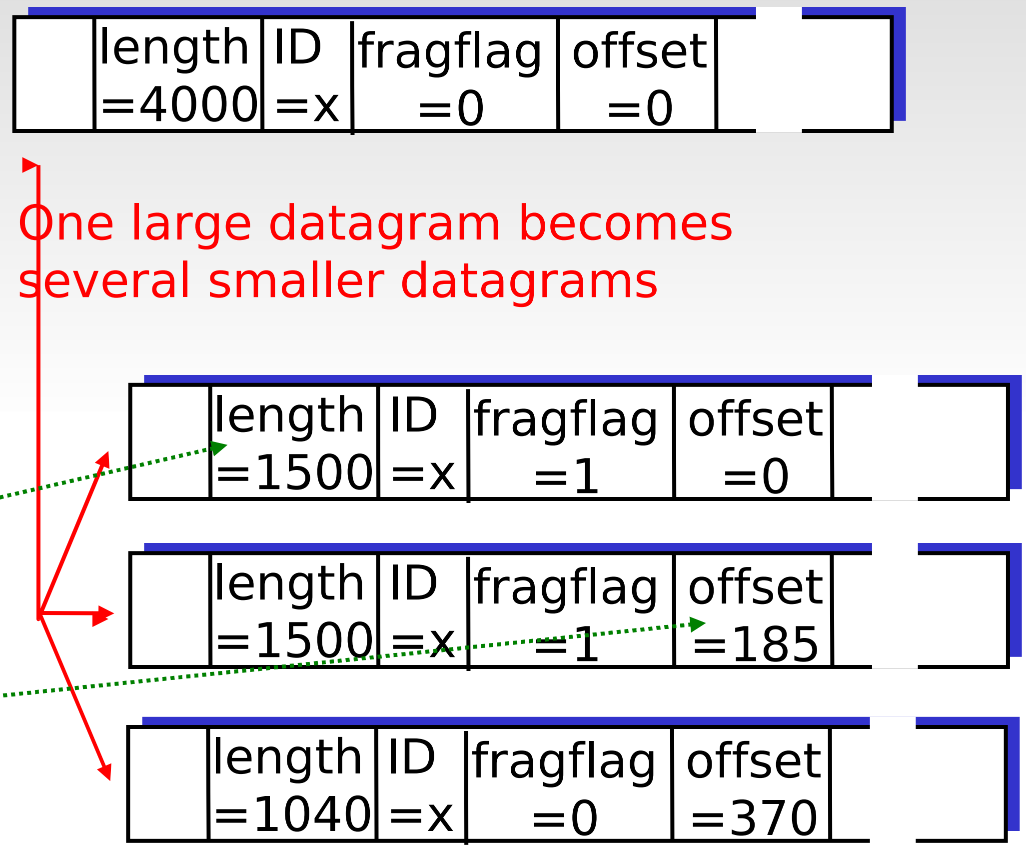
- one datagram becomes several datagrams
- "reassembled" only at final destination
- IP header bits used to identify, order related fragments



IP Fragmentation and Reassembly

<u>Example</u>

- 4000 byte datagram
- MTU = 1500 bytes



1480 bytes in data field

offset = 1480/8