Network Layer Part 1

SERVICE MODEL, DATA GRAM AND VIRTUAL CIRCUIT, INTERNET PROTOCOL, DHCP

Network layer

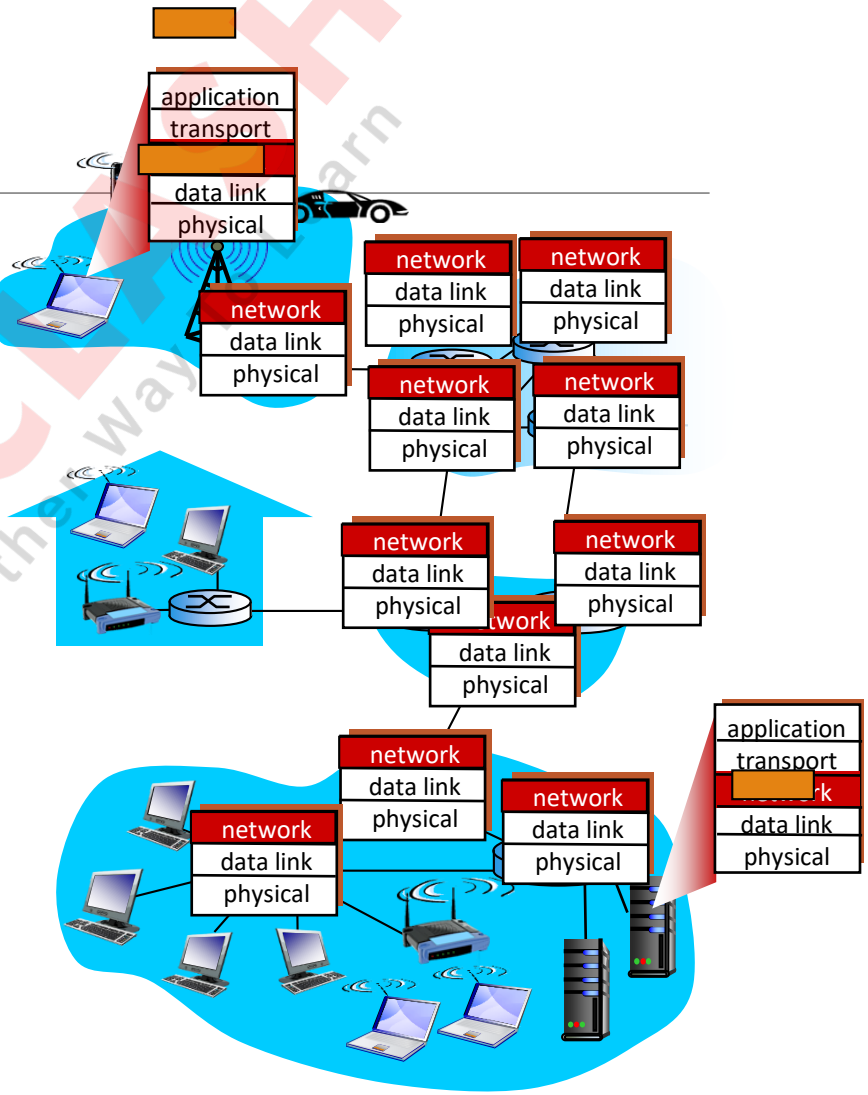
transport segment from sending to receiving host

on sending side encapsulates segments into datagrams

on receiving 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

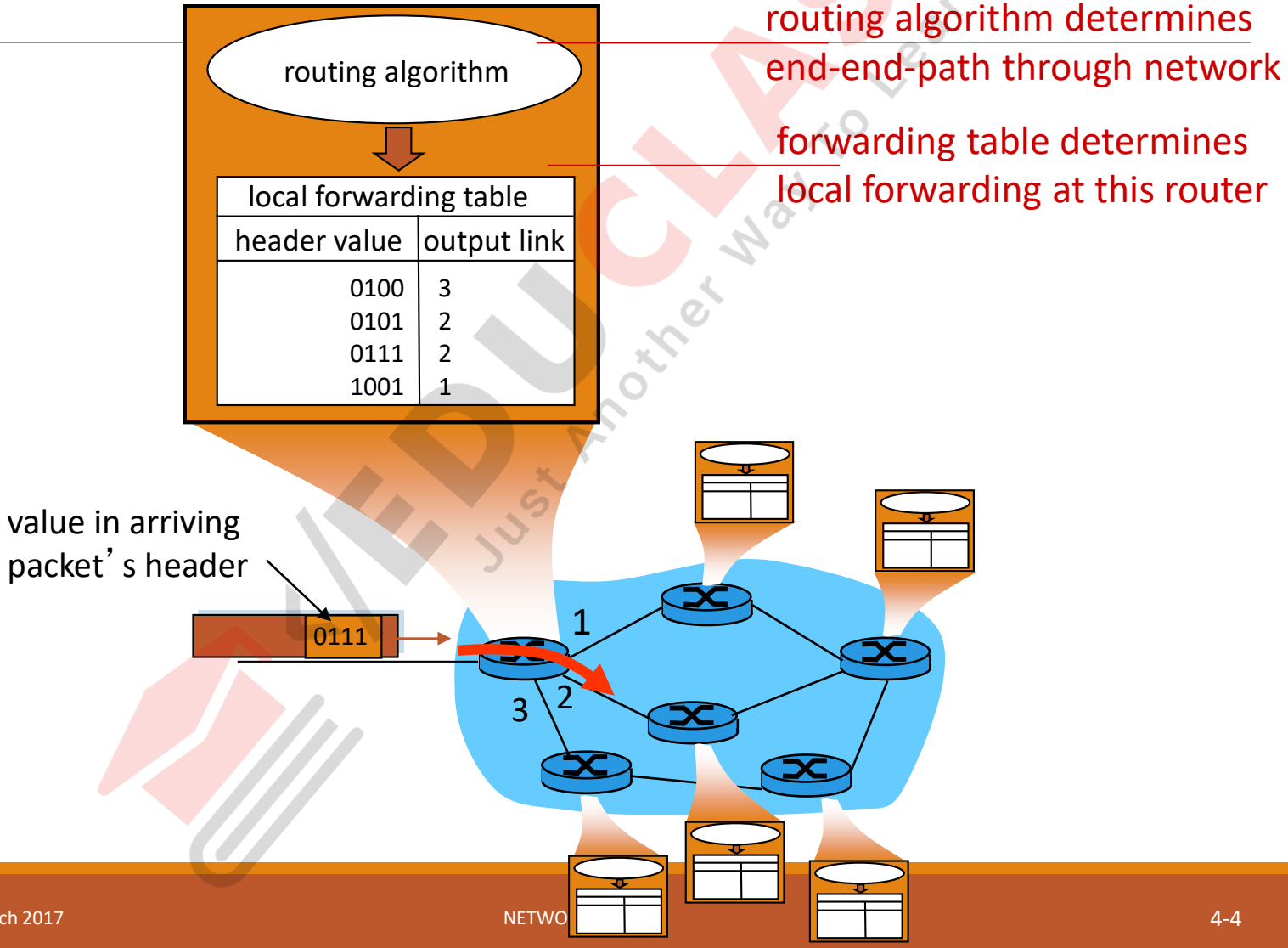
routing: determine route taken by packets from source to dest.

- *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

3rd 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 service model

Q: What *service model* for "channel" transporting datagrams from sender to receiver?

example services for individual datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

example services for a flow of datagrams:

- in-order datagram delivery
- guaranteed minimal bandwidth
- Guaranteed maximum jitter

Network layer service models:

Network Architecture	Service Model	Guarantees ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
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

Connection, connection-less service

- *datagram* network provides network-layer *connectionless* service
- *virtual-circuit* network provides network-layer *connection* service
- analogous to TCP/UDP connecton-oriented / connectionless 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)

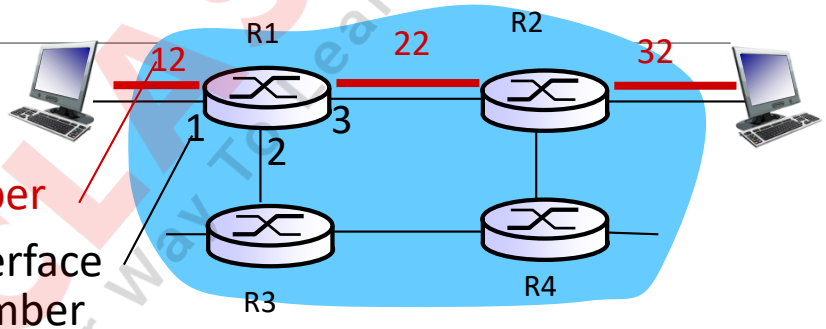
VC implementation

a VC consists of:

1. *path* from source to destination
2. *VC numbers*, one number for each link along path
3. *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

VC forwarding table



forwarding table in northwest router:

VC number

interface number

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

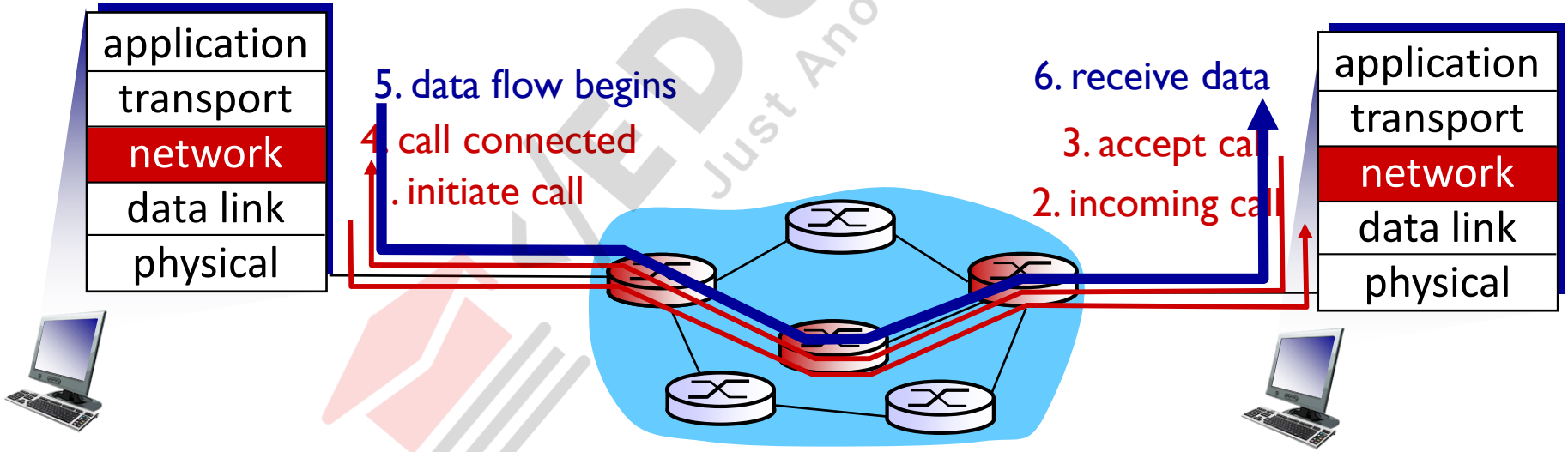
VC routers maintain connection state information!

Virtual circuits: signaling protocols

used to setup, maintain teardown VC

used in ATM, frame-relay, X.25

not used in today's Internet



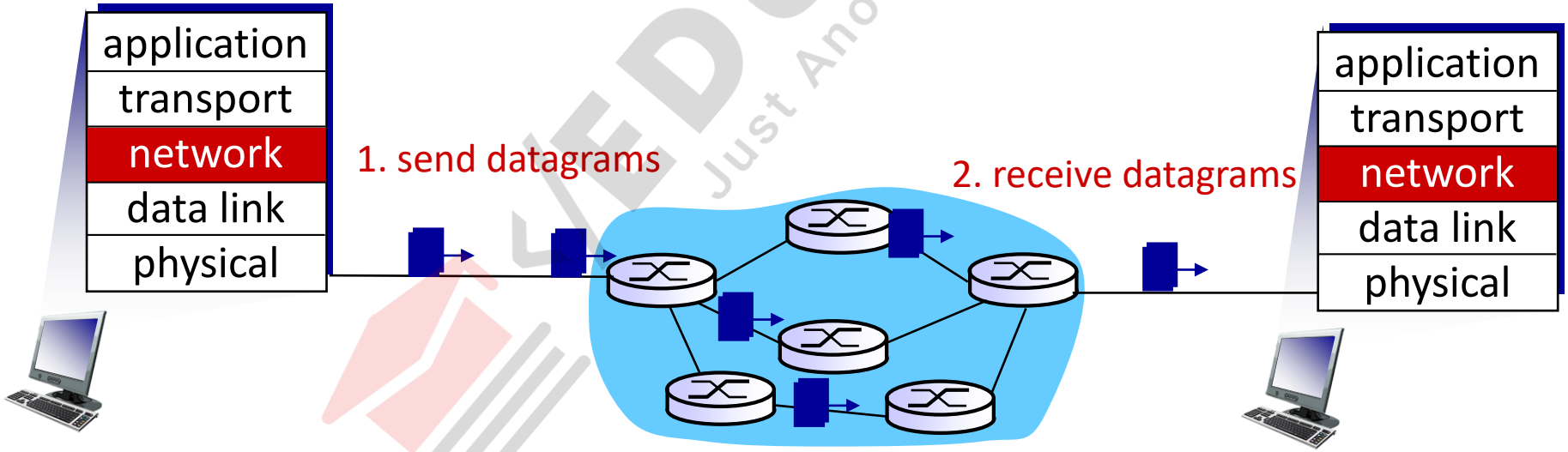
Datagram networks

no call setup at network layer

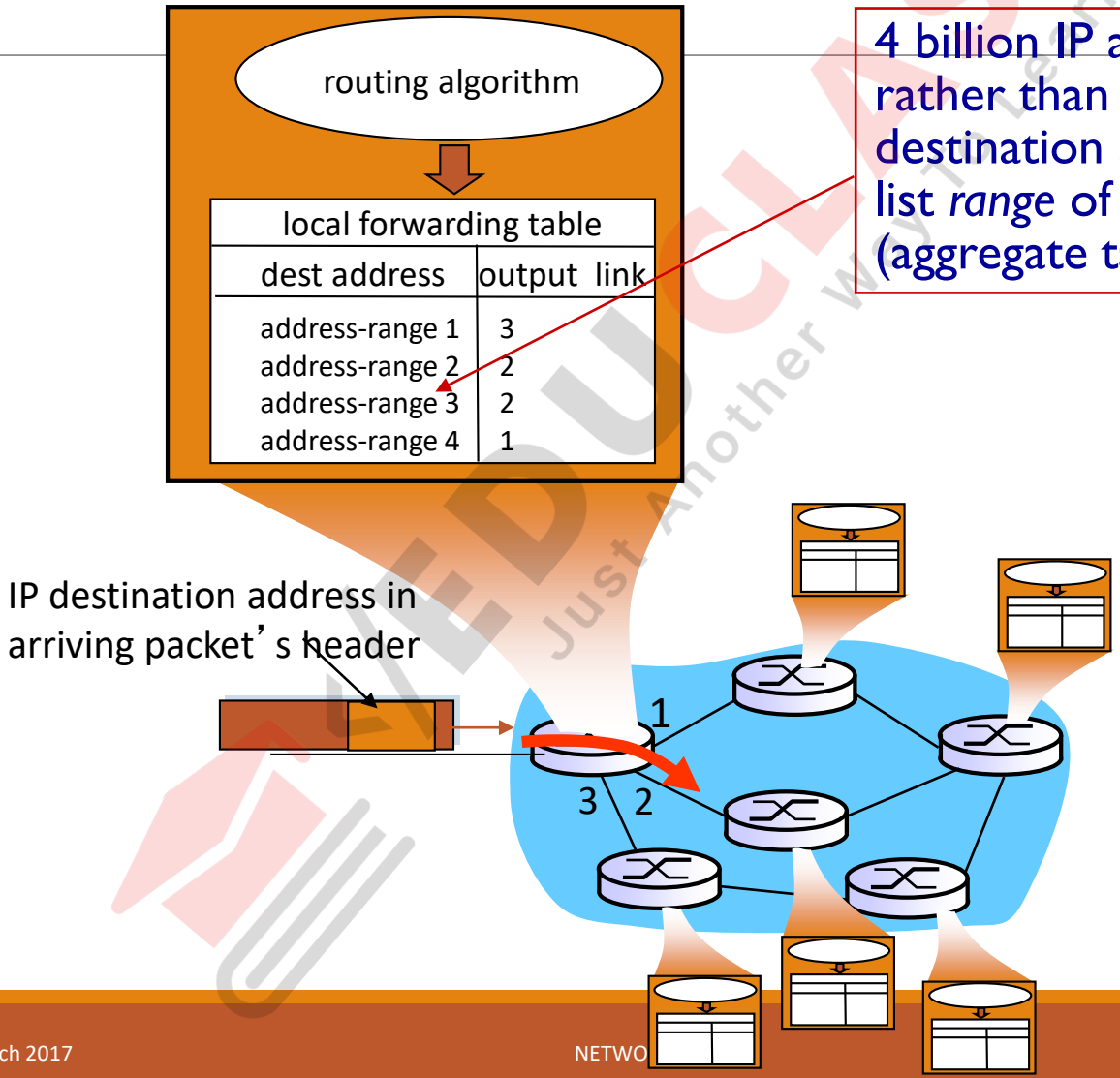
routers: no state about end-to-end connections

- no network-level concept of "connection"

packets forwarded using destination host address



Datagram forwarding table



Datagram forwarding table

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

longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	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?

Datagram or VC network: why?

Internet (datagram)

data exchange among computers
- "elastic" service, no strict timing req.

many link types
- different characteristics
- uniform service difficult

"smart" end systems (computers)
- can adapt, perform control, error recovery
- ***simple inside network, complexity at "edge"***

ATM (VC)

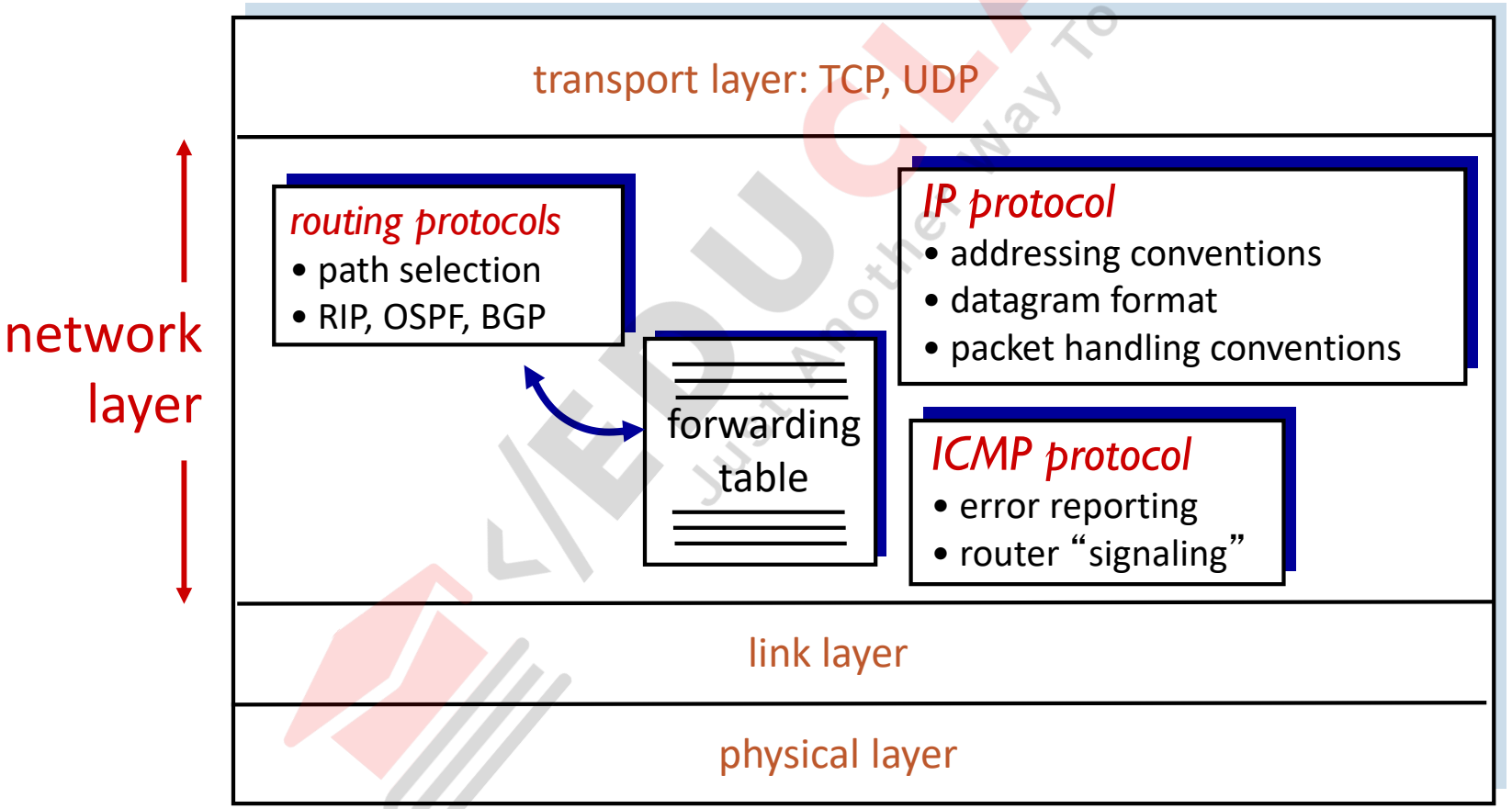
evolved from telephony

human conversation:
- strict timing, reliability requirements
- need for guaranteed service

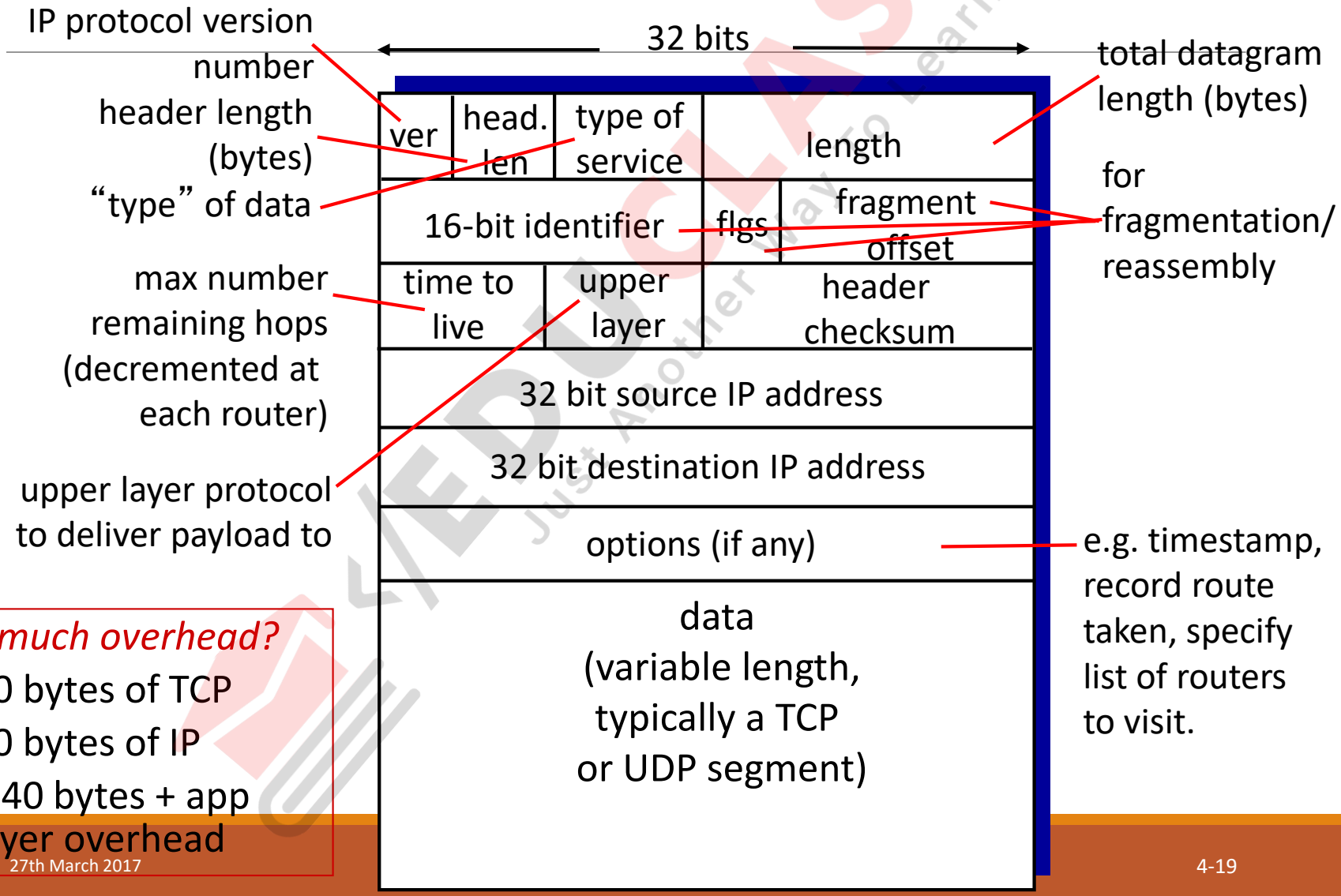
"dumb" end systems
- telephones
- ***complexity inside network***

The Internet network layer

host, router network layer functions:



IP datagram format



how much overhead?

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

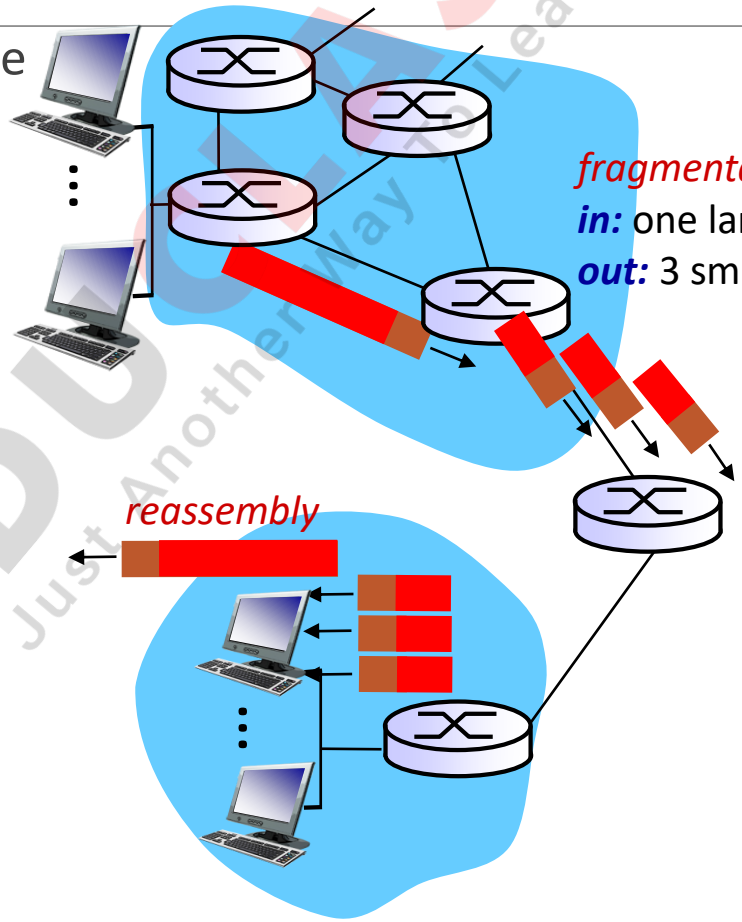
IP fragmentation, reassembly

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

- different link types, different MTUs

large IP datagram divided ("fragmented") within net

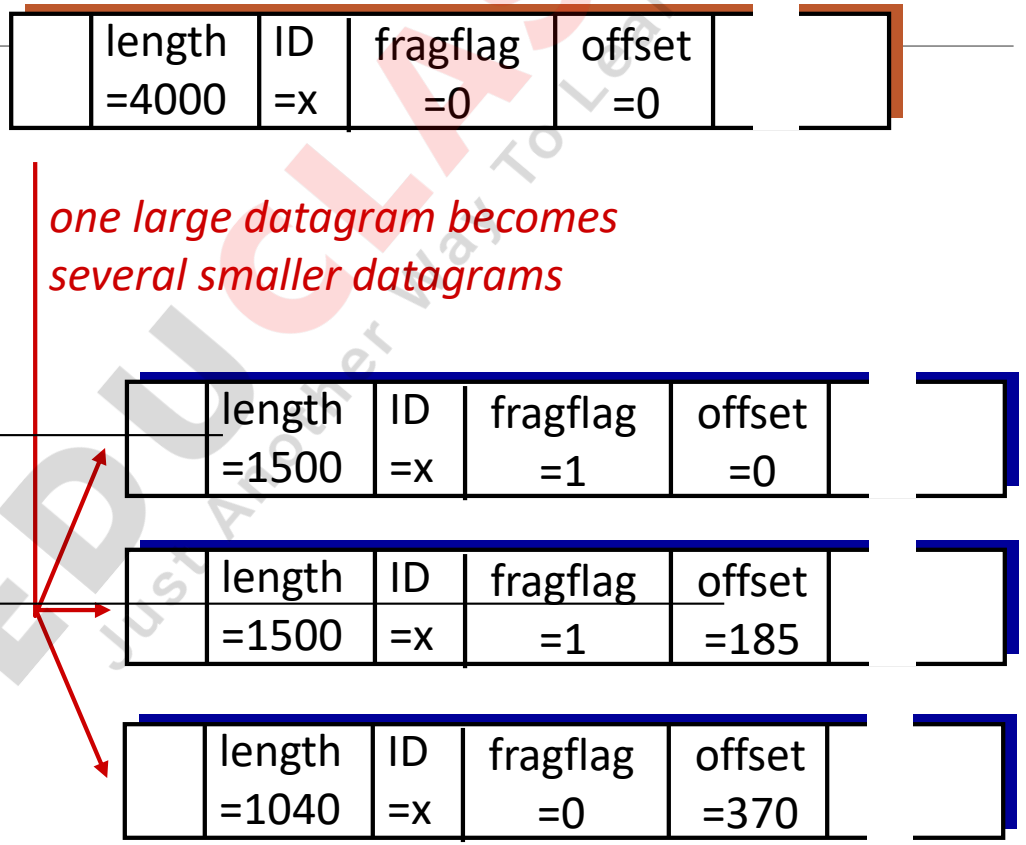
- one datagram becomes several datagrams
- "reassembled" only at final destination
- IP header bits used to identify, order related fragments



IP fragmentation, reassembly

example:

- 4000 byte datagram
- MTU = 1500 bytes



IP addresses: how to get one?

Q: How does a *host* get IP address?

hard-coded by system admin in a file

- Windows: control-panel->network->configuration->tcp/ip->properties
- UNIX: /etc/rc.config

DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server

- "plug-and-play"

DHCP: Dynamic Host Configuration Protocol

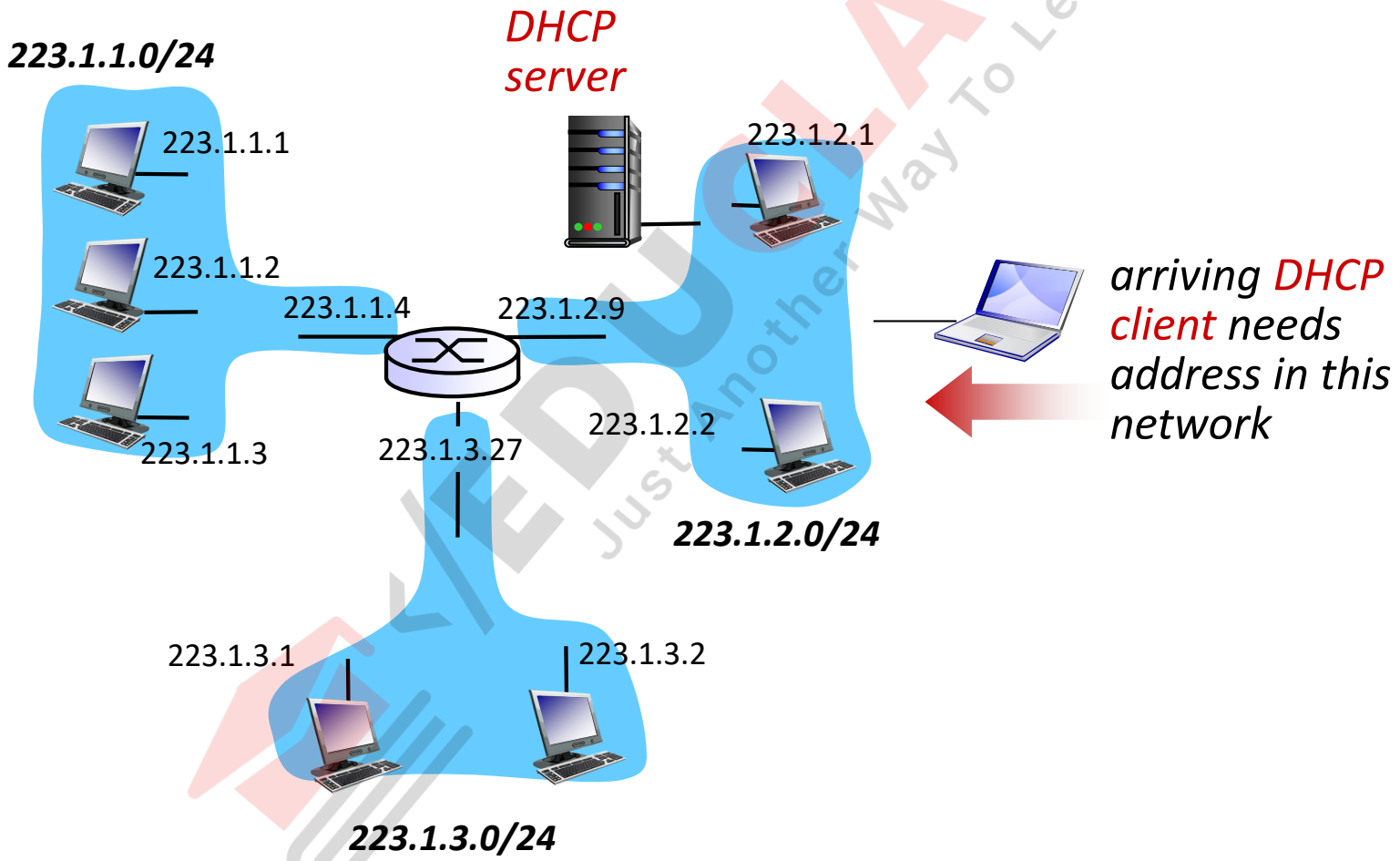
goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network (more shortly)

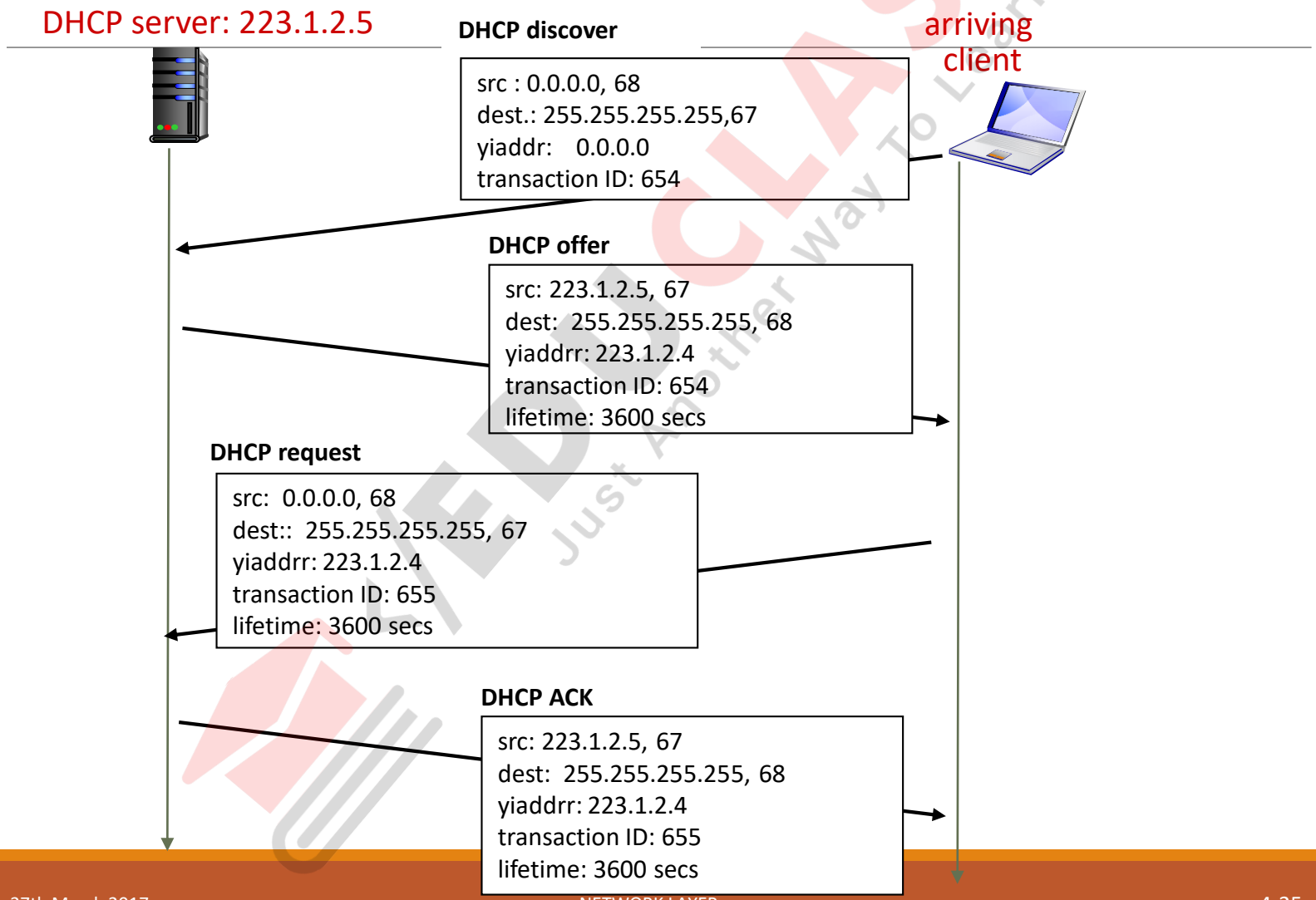
DHCP overview:

- host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario



DHCP client-server scenario

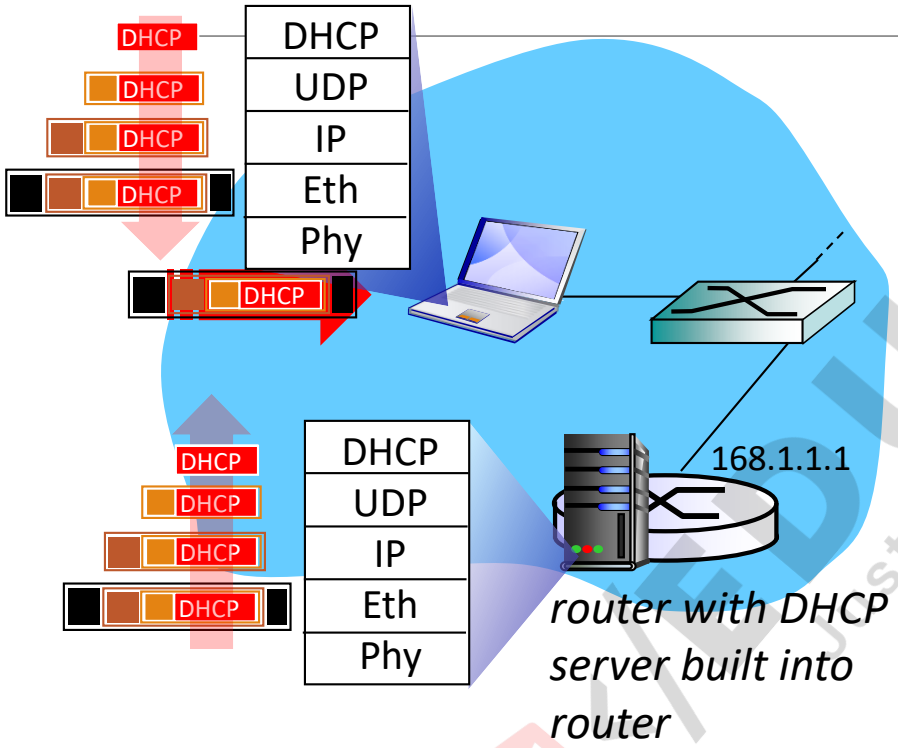


DHCP: more than IP addresses

DHCP can return more than just allocated IP address on subnet:

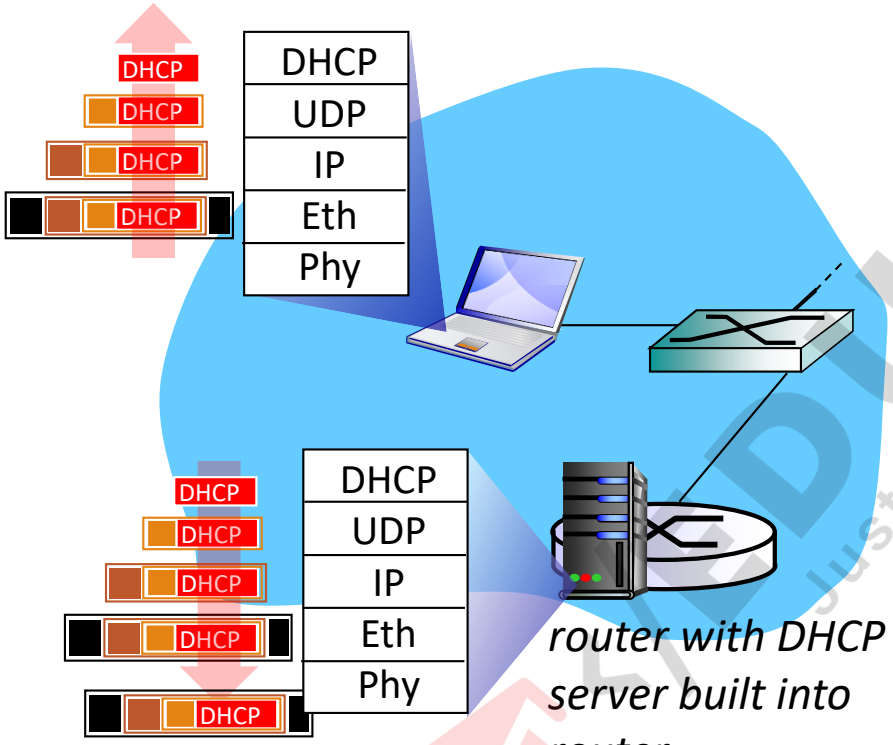
- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)

DHCP: example



- connecting laptop needs its IP address, addr of first-hop router, addr of DNS server: use DHCP
  - DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet
  - Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
  - Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

DHCP: example



DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server

- encapsulation of DHCP server, frame forwarded to client, demuxing up to DHCP at client
- client now knows its IP address, name and IP address of DSN server, IP address of its first-hop router

IP addresses: how to get one?

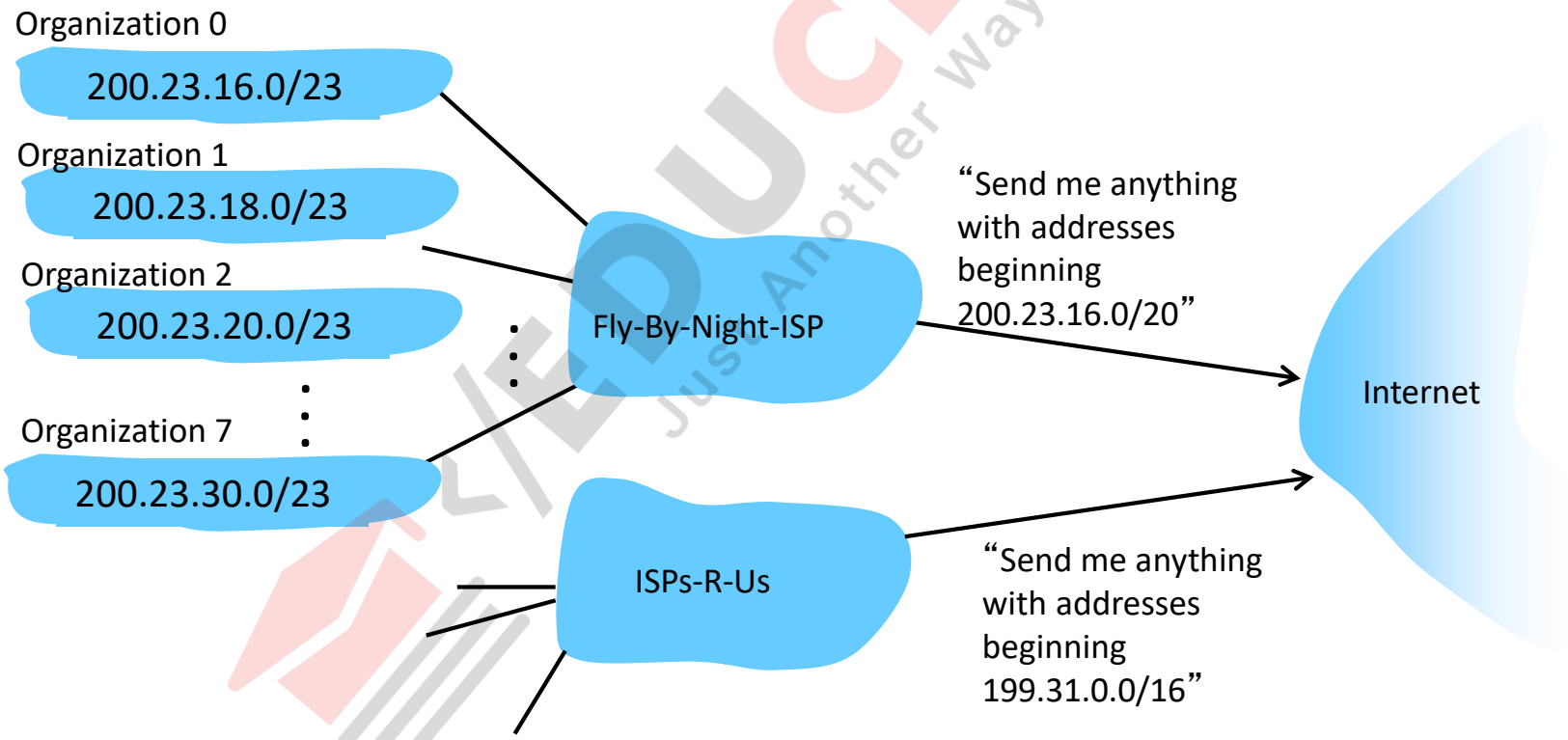
Q: how does *network* get subnet part of IP addr?

A: gets allocated portion of its provider ISP's address space

ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

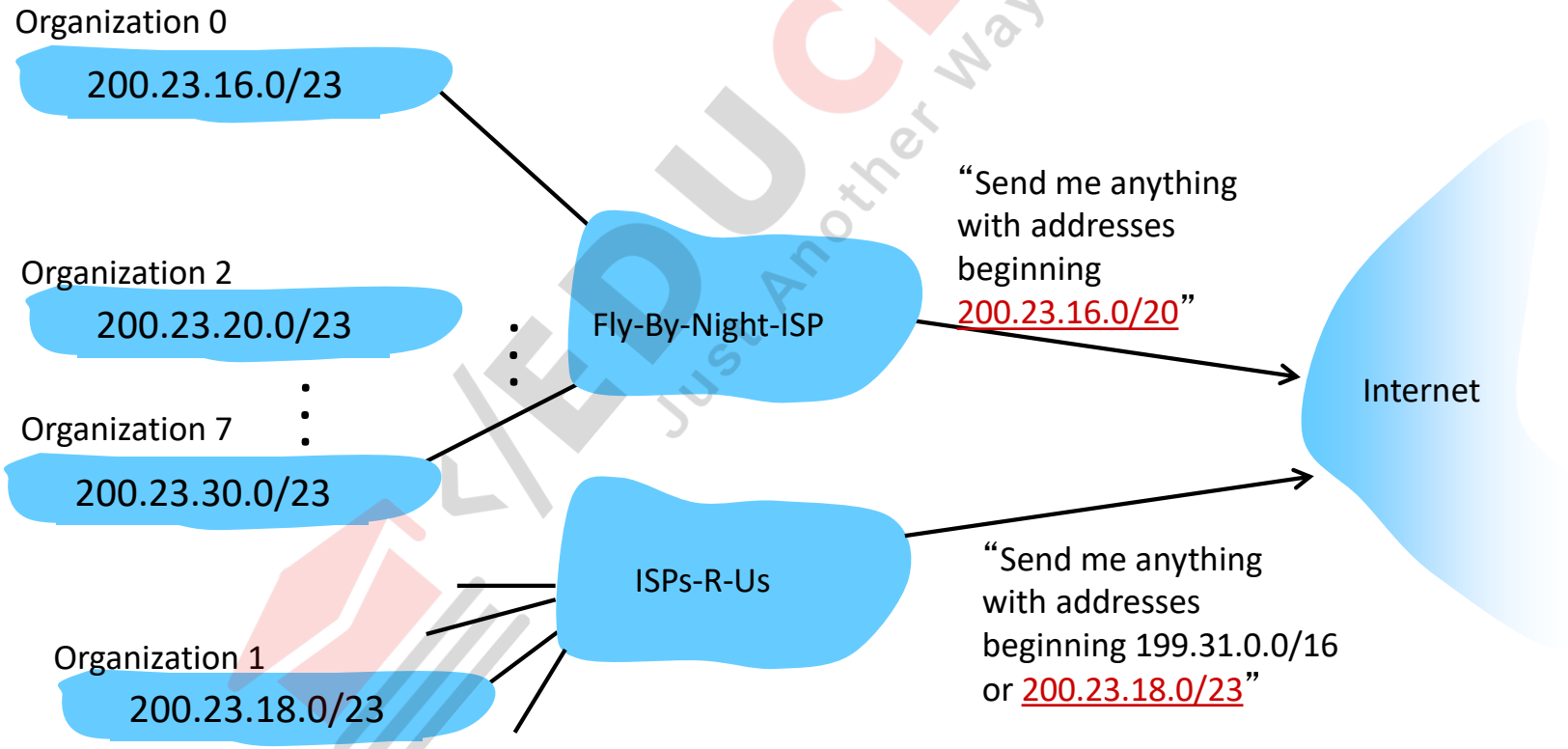
Hierarchical addressing: route aggregation

hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1



IP addressing: the last word...

Q: how does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers http://www.icann.org/

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes