Computer Networks (CS F303)

BITS Pilani
Pilani Campus

Virendra Singh Shekhawat
Department of Computer Science and Information Systems

innovate achieve lead

BITS Pilani

Pilani Campus

Second Semester 2020-2021

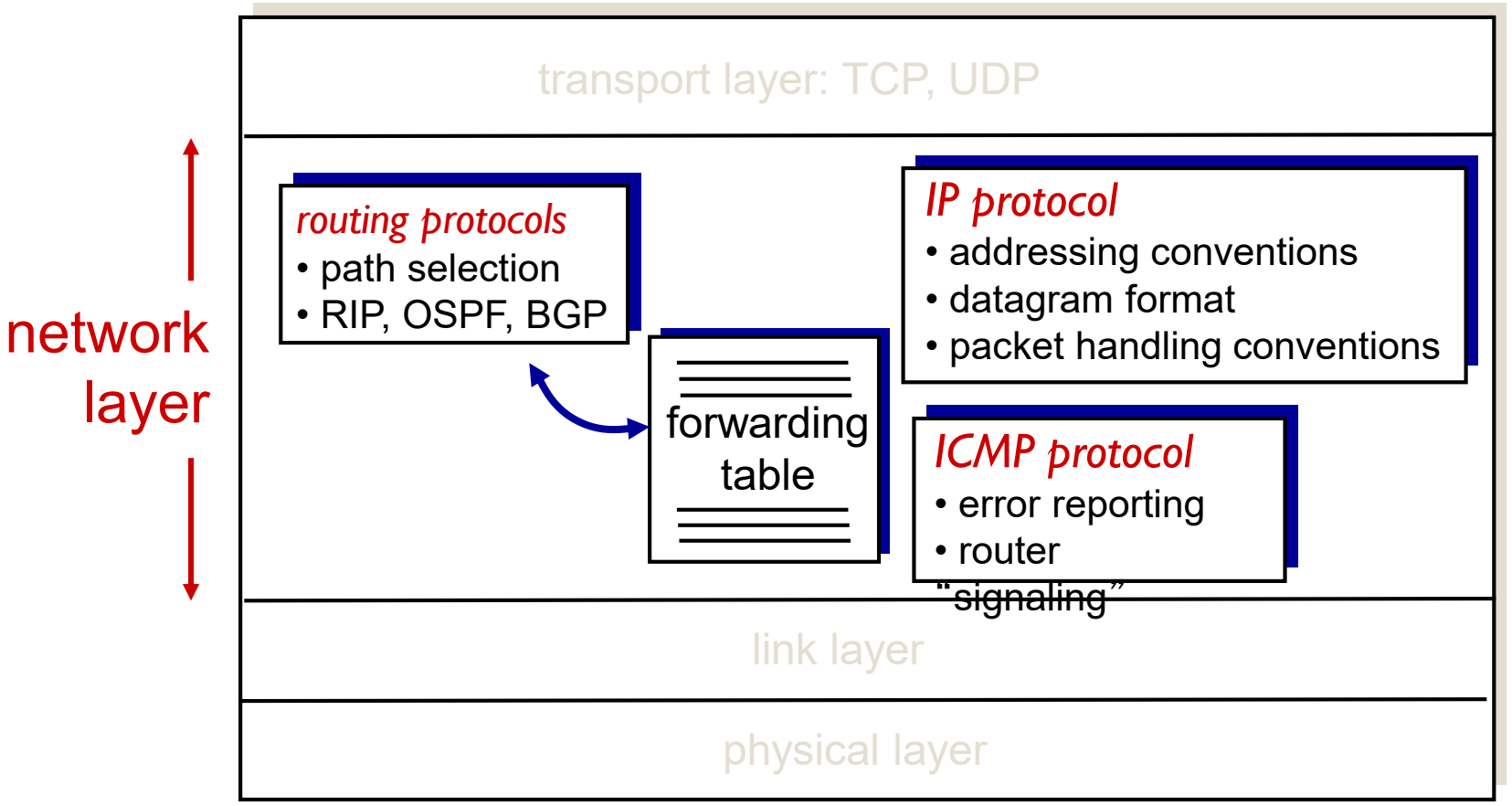
Module-4 <Network Layer>

Agenda

- DHCP Protocol
- IP Datagram Fragmentation
- NAT Firewall

The Internet network layer

host, router network layer functions:



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
  - Automate the process of connecting a host into a network. "plug-and-play"
  - Works as client (arriving host) and Server (DHCP server)
  - Useful where hosts join and leave network frequently

DHCP: Dynamic Host Configuration Protocol

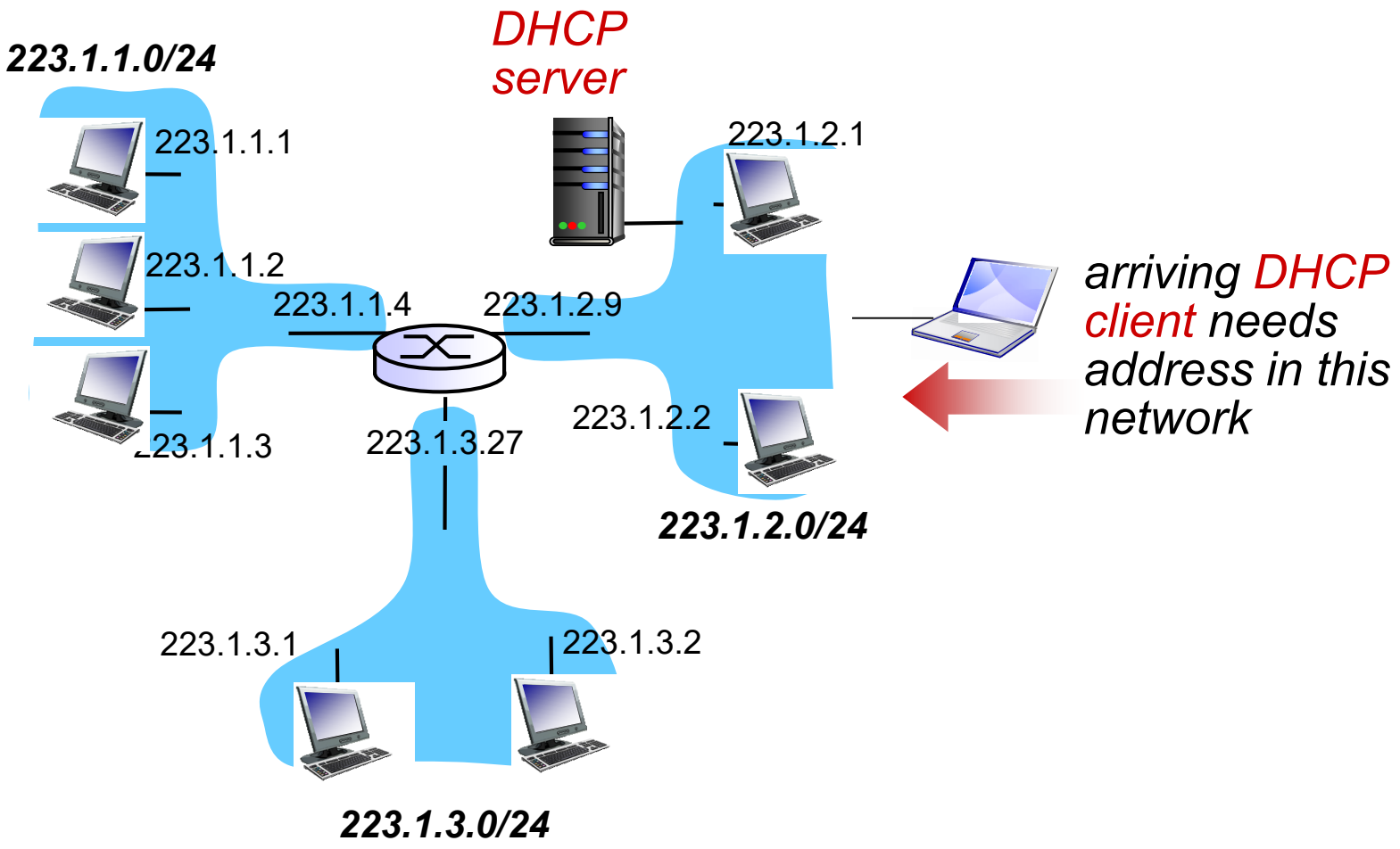
Goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- 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)

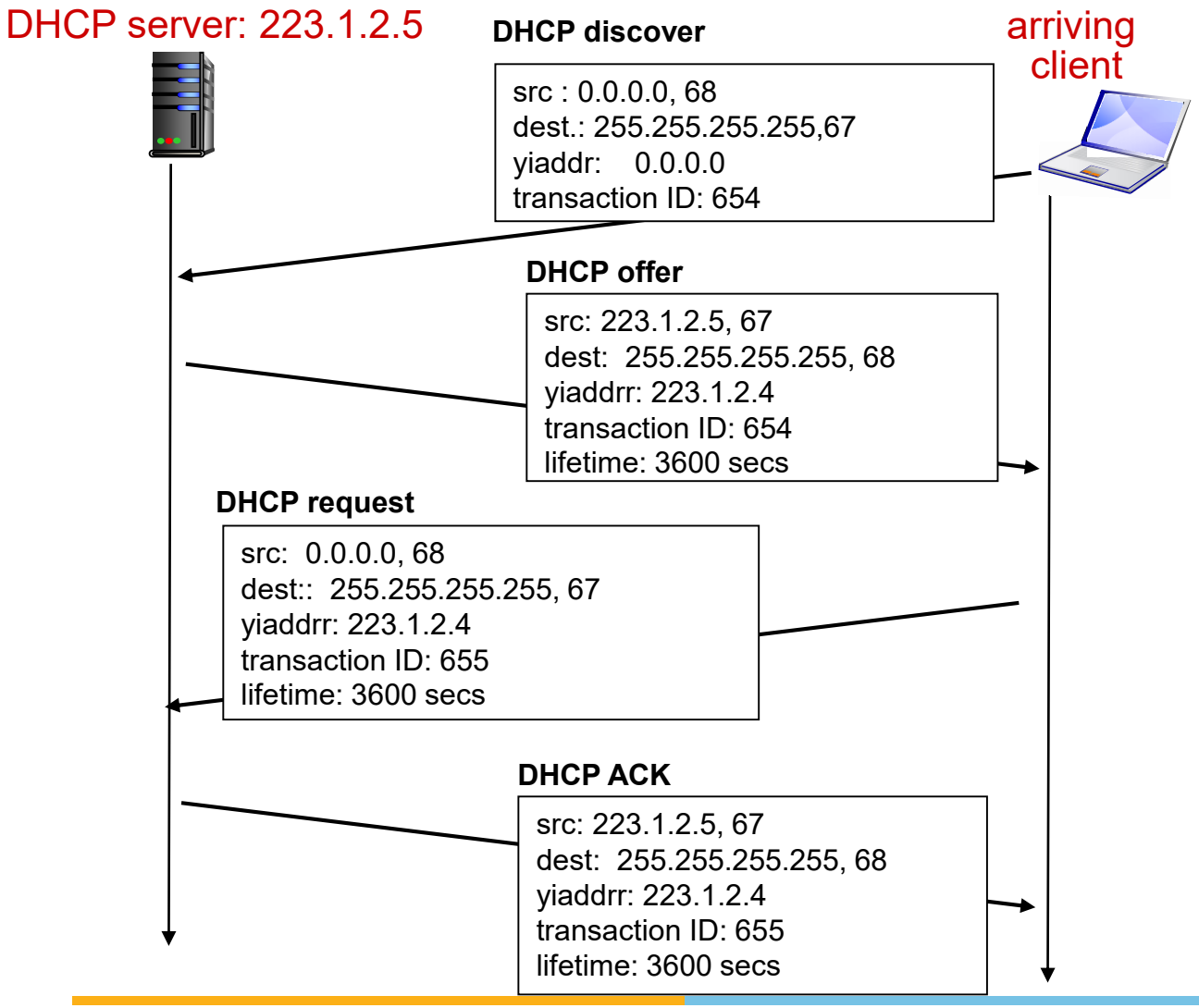
DHCP overview:

- Host broadcasts "DHCP discover" msg
- DHCP server responds with "DHCP offer" msg
- Host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

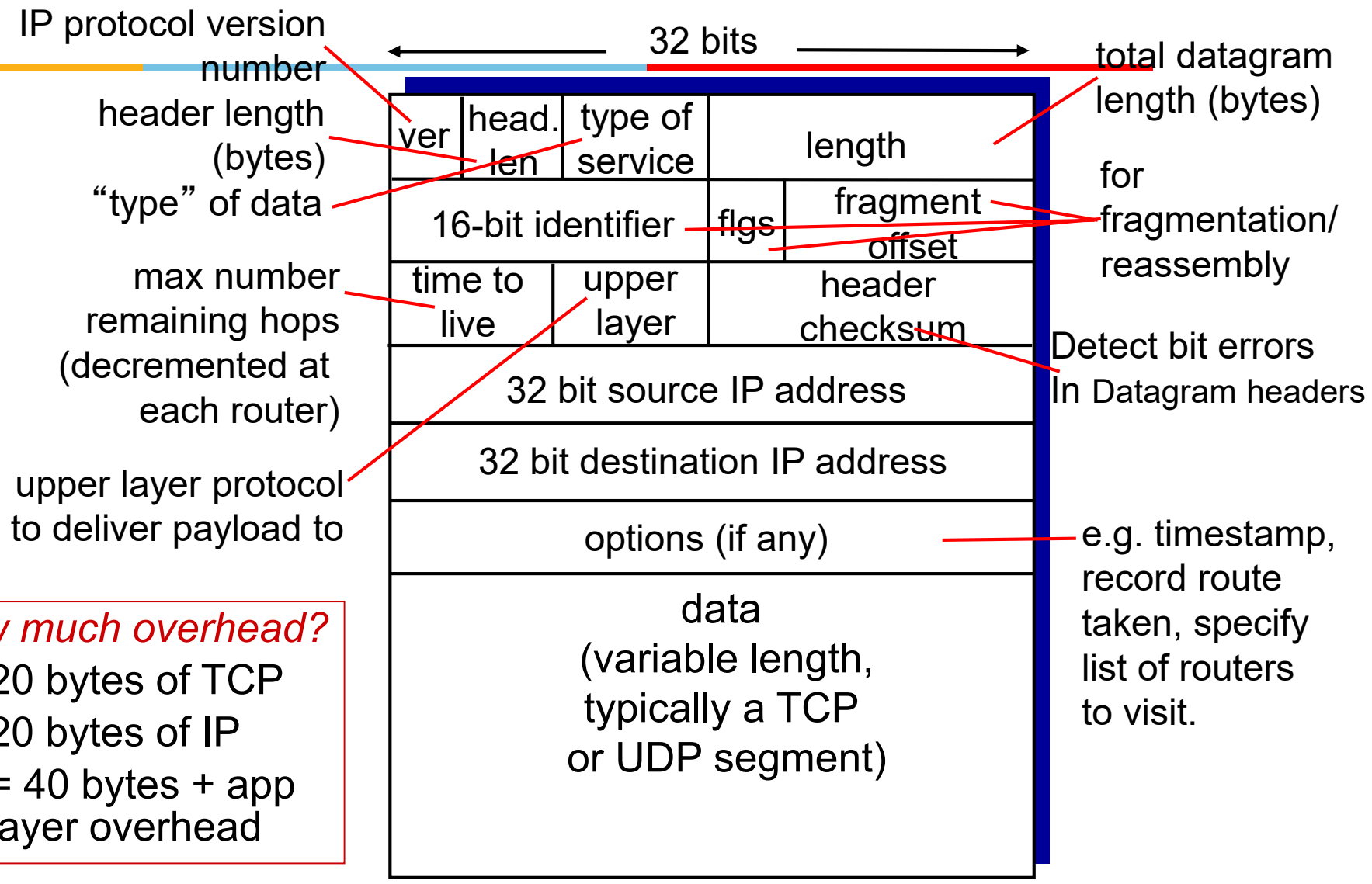
DHCP Client-Server Scenario



DHCP


DHCP server: 223.1.2.5
DHCP discover
arriving client
src : 0.0.0.0, 68
dest.: 255.255.255.255,67
yiaddr: 0.0.0.0
transaction ID: 654
DHCP offer
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddrr: 223.1.2.4
transaction ID: 654
lifetime: 3600 secs
DHCP request
src: 0.0.0.0, 68
dest:: 255.255.255.255, 67
yiaddrr: 223.1.2.4
transaction ID: 655
lifetime: 3600 secs
DHCP ACK
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddrr: 223.1.2.4
transaction ID: 655
lifetime: 3600 secs

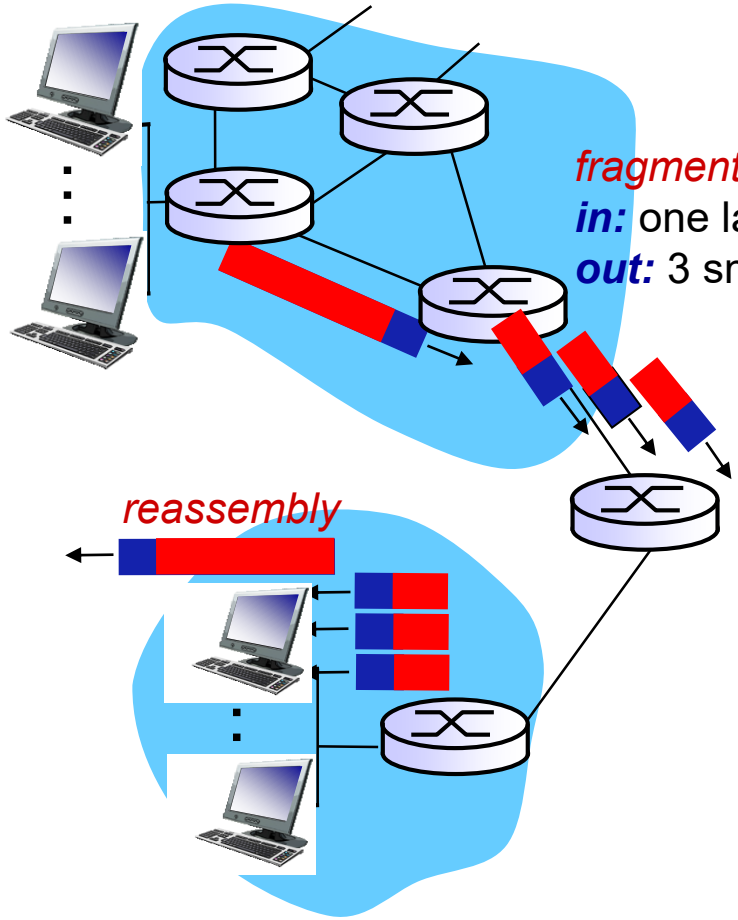
IP Datagram Format



how much overhead?

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

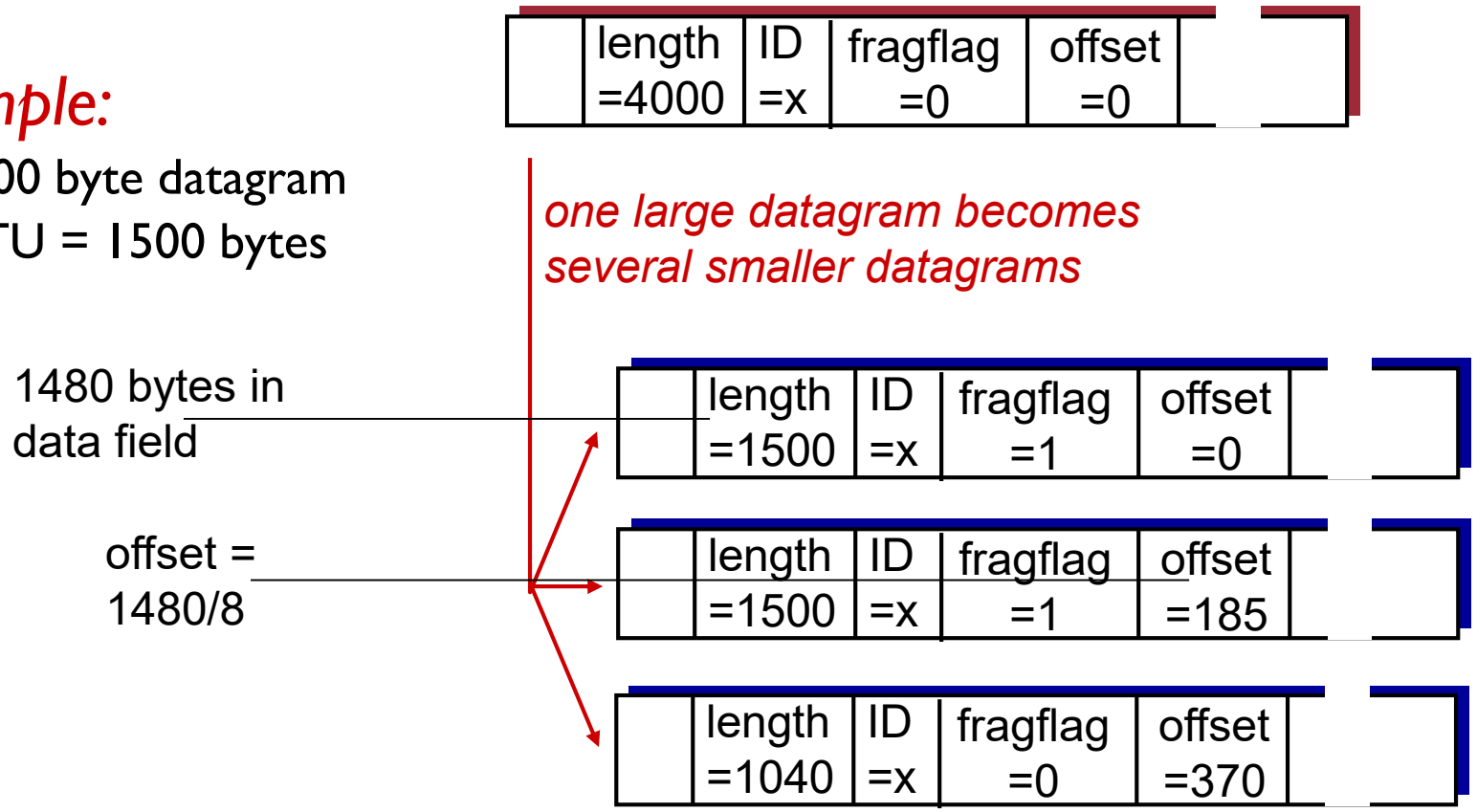
IP Fragmentation & Reassembly [.1]



IP Fragmentation & Reassembly [..2]

example:

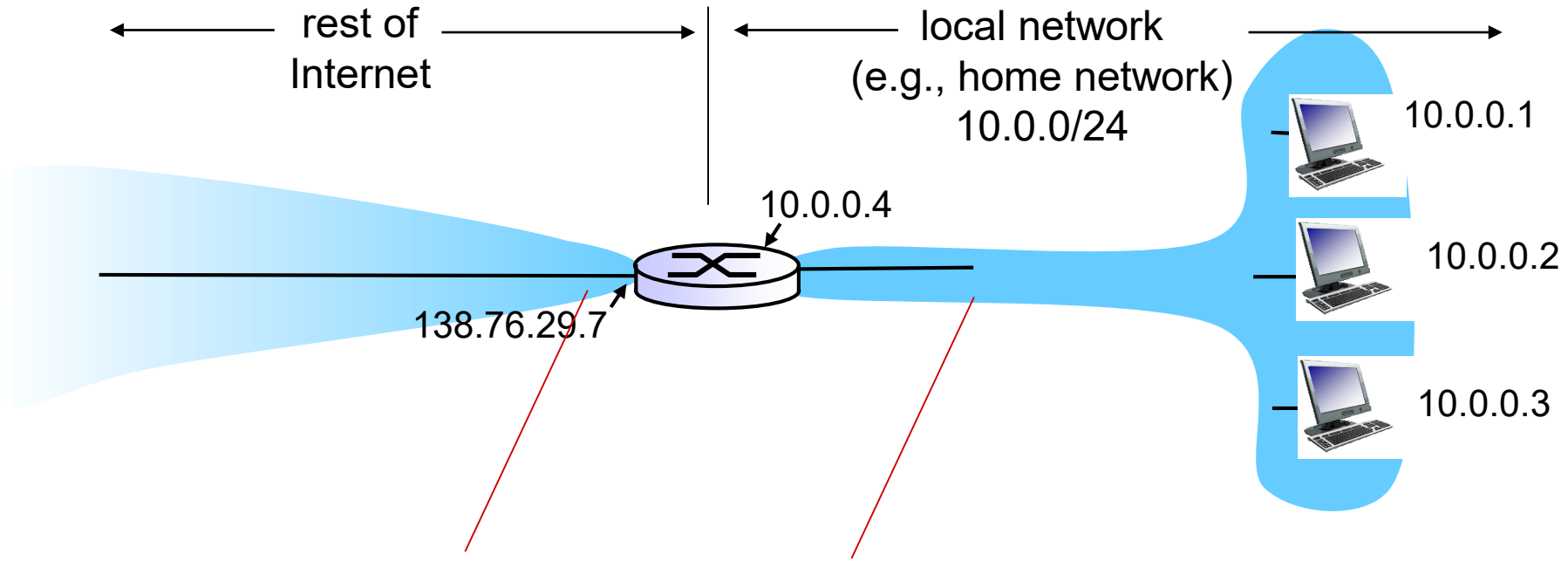
- 4000 byte datagram
- MTU = 1500 bytes



Network Address Translation (NAT)

- *Motivation:* local network uses just one IP address as far as outside world is concerned
  - Can change addresses of devices in local network without notifying outside world
  - Can change ISP without changing addresses of devices in local network
  - Devices inside local net not explicitly addressable, visible by outside world (a security plus)

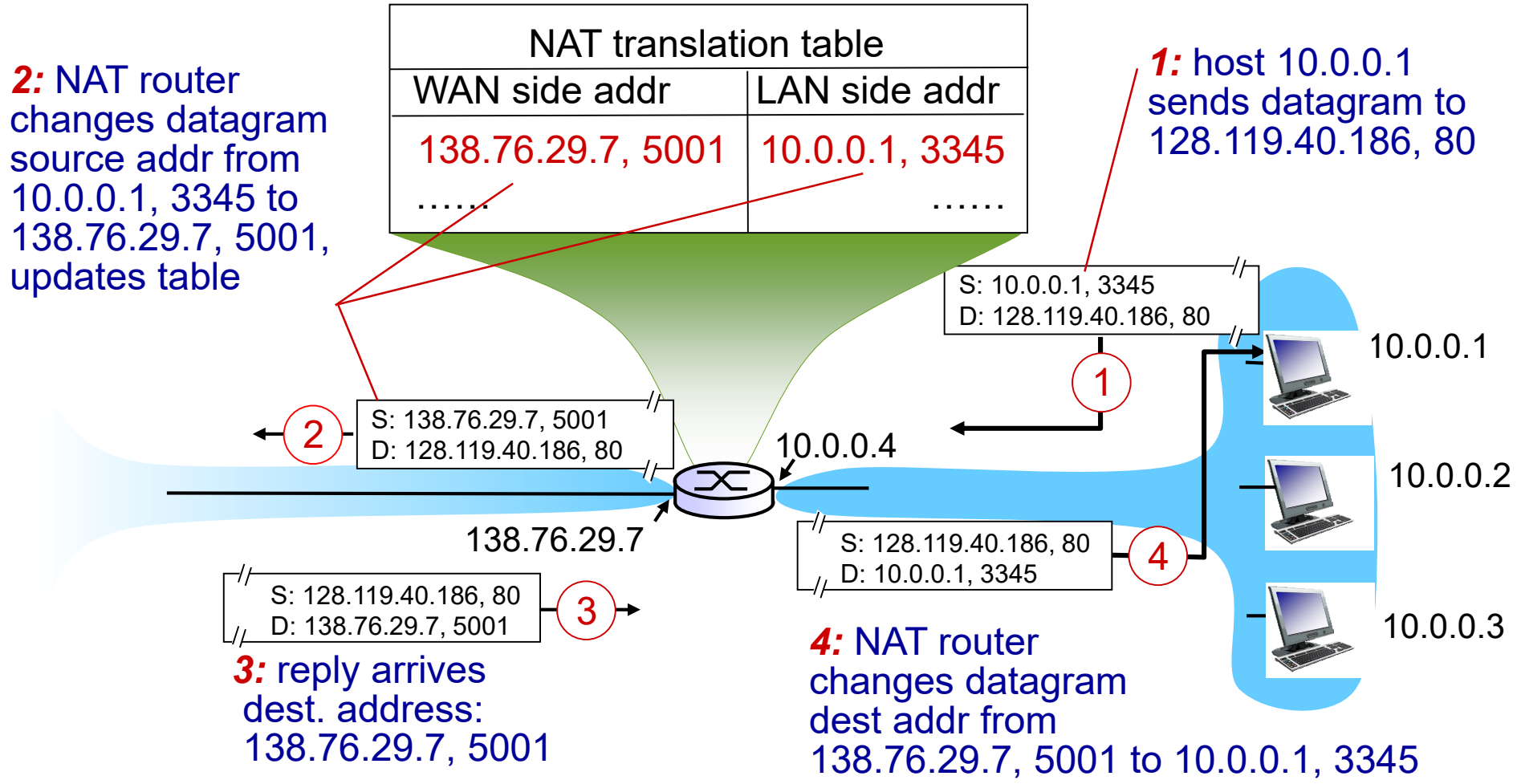
NAT: Motivation???



all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7,different source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

How it Works???


NAT translation table
WAN side addr
LAN side addr
138.76.29.7, 5001
10.0.0.1, 3345
……
……
2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table
1: host 10.0.0.1 sends datagram to 128.119.40.186, 80
S: 10.0.0.1, 3345
D: 128.119.40.186, 80
1
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
138.76.29.7
2
S: 138.76.29.7, 5001
D: 128.119.40.186, 80
3
S: 128.119.40.186, 80
D: 138.76.29.7, 5001
4
S: 128.119.40.186, 80
D: 10.0.0.1, 3345
3: reply arrives dest. address: 138.76.29.7, 5001
4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

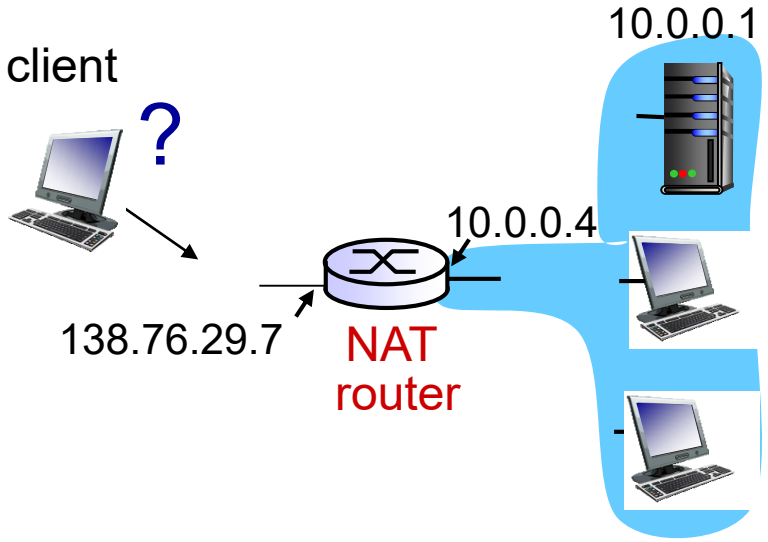
Facts about NAT

- 16-bit port-number field:
  - How many devices can be connected?

- NAT is controversial:
  - Routers should only process up to layer 3
  - Violates end-to-end argument
  - Address shortage should instead be solved by IPv6

NAT Traversal Problem

- Client wants to connect to server with address 10.0.0.1
  - Server address 10.0.0.1 local to LAN (client can't use it as destination address)
  - Only one externally visible NATed address: 138.76.29.7



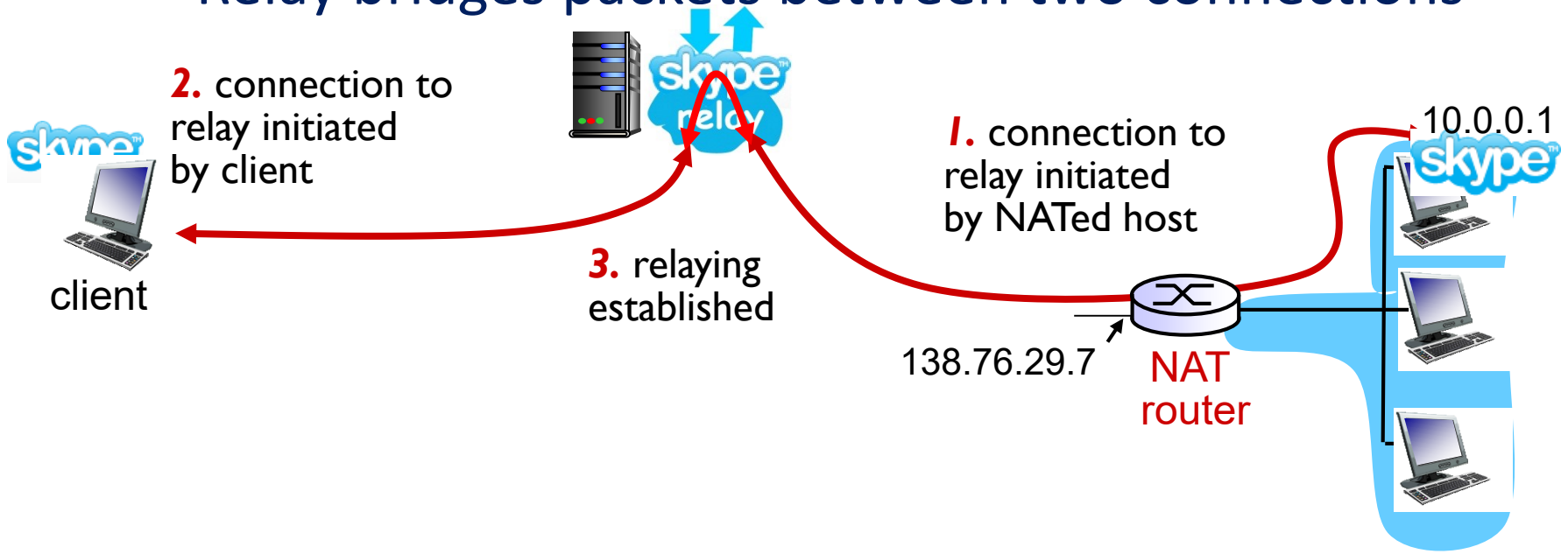
Solutions [.1]

- Statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (138.76.29.7, port 25000) always forwarded to 10.0.0.1 port 25000

- Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATed host to:
  - Learn public IP address (138.76.29.7)
  - e.g., BitTorrent application in the host asks NAT to create a hole that maps (10.0.0.1,3345) to (138.76.29.7,5001)
  - Add/remove port mappings (with lease times)

Solutions [..2]

- Relaying (used in Skype)
  - NATed client establishes connection to relay
  - External client connects to relay
  - Relay bridges packets between two connections



Thank You!