CS2105

An *Awesome* Introduction to Computer Networks

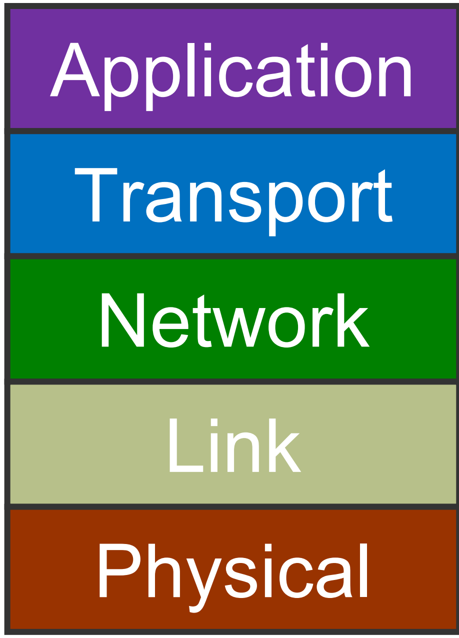
Conclusion

NUS National University of Singapore

Department of Computer Science
School of Computing

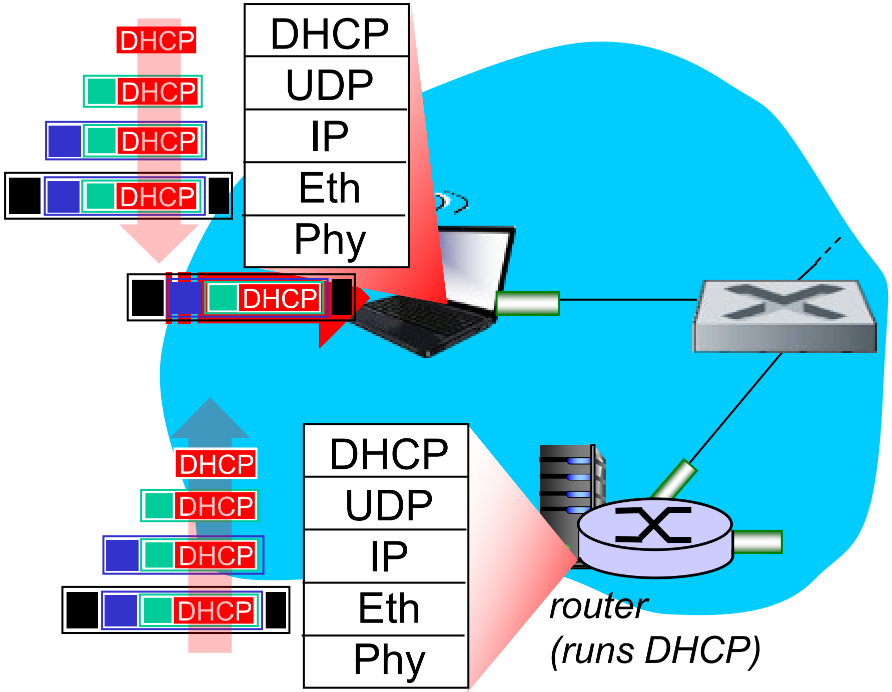
Last Lecture!

Synthesis: a day in the life of a web request



- Journey down protocol stack complete!
  - application, transport, network, link
- Putting-it-all-together: *synthesis*!
  - *goal:* identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - *scenario:* student attaches laptop to campus network, requests/receives www.google.com

A day in the life… connecting to the Internet

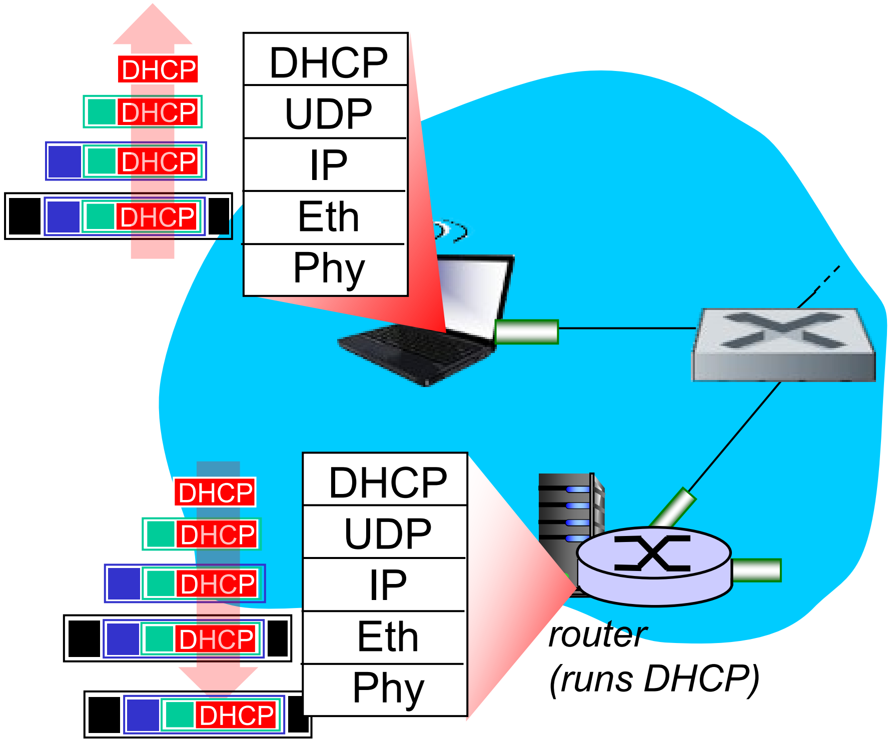


- connecting laptop needs to get its own
  - IP address,
  - addr of first-hop router,
  - addr of DNS server

DHCP

- DHCP request encapsulated in UDP,
  - encapsulated in IP,
    - encapsulated in 802.3 Ethernet
- Ethernet frame broadcast (dest: FF:FF:FF:FF:FF:FF) on LAN, received at router running DHCP server
  - switch learning
- Ethernet demuxed to IP
  - demuxed to UDP
    - demuxed to DHCP

A day in the life… connecting to the Internet



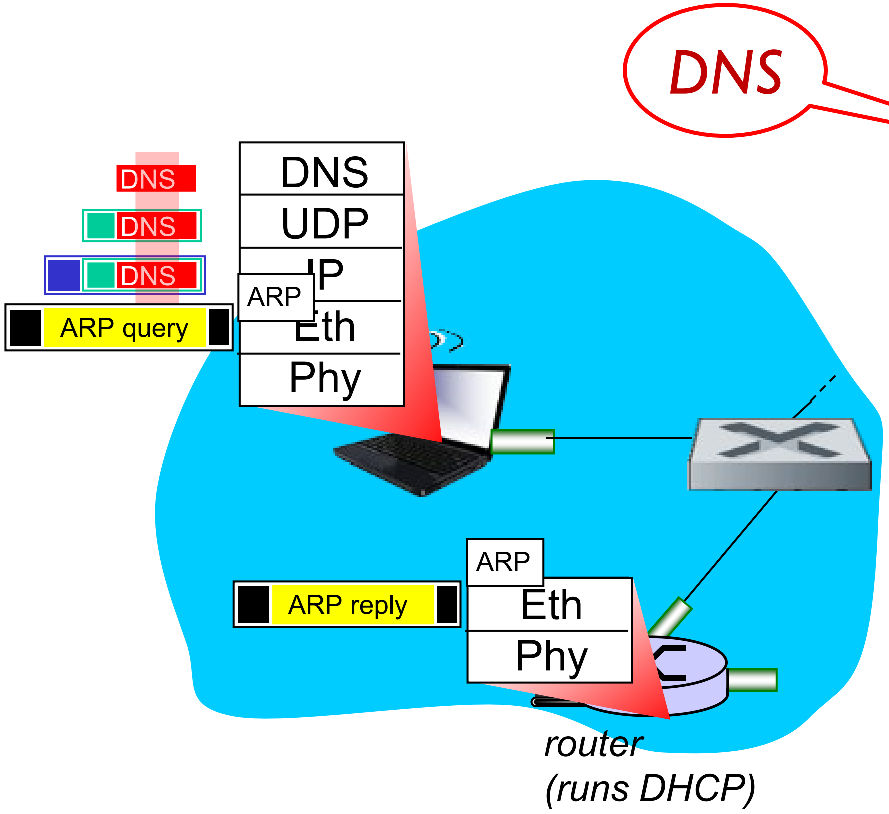
- DHCP server formulates *DHCP ACK* containing
  - client's IP address,
  - IP address of first-hop router for client,
  - name & IP address of DNS server

- encapsulation at DHCP server,
  - frame forwarded (switch learning) through LAN,

- Demultiplexing at client
  - client receives DHCP ACK reply

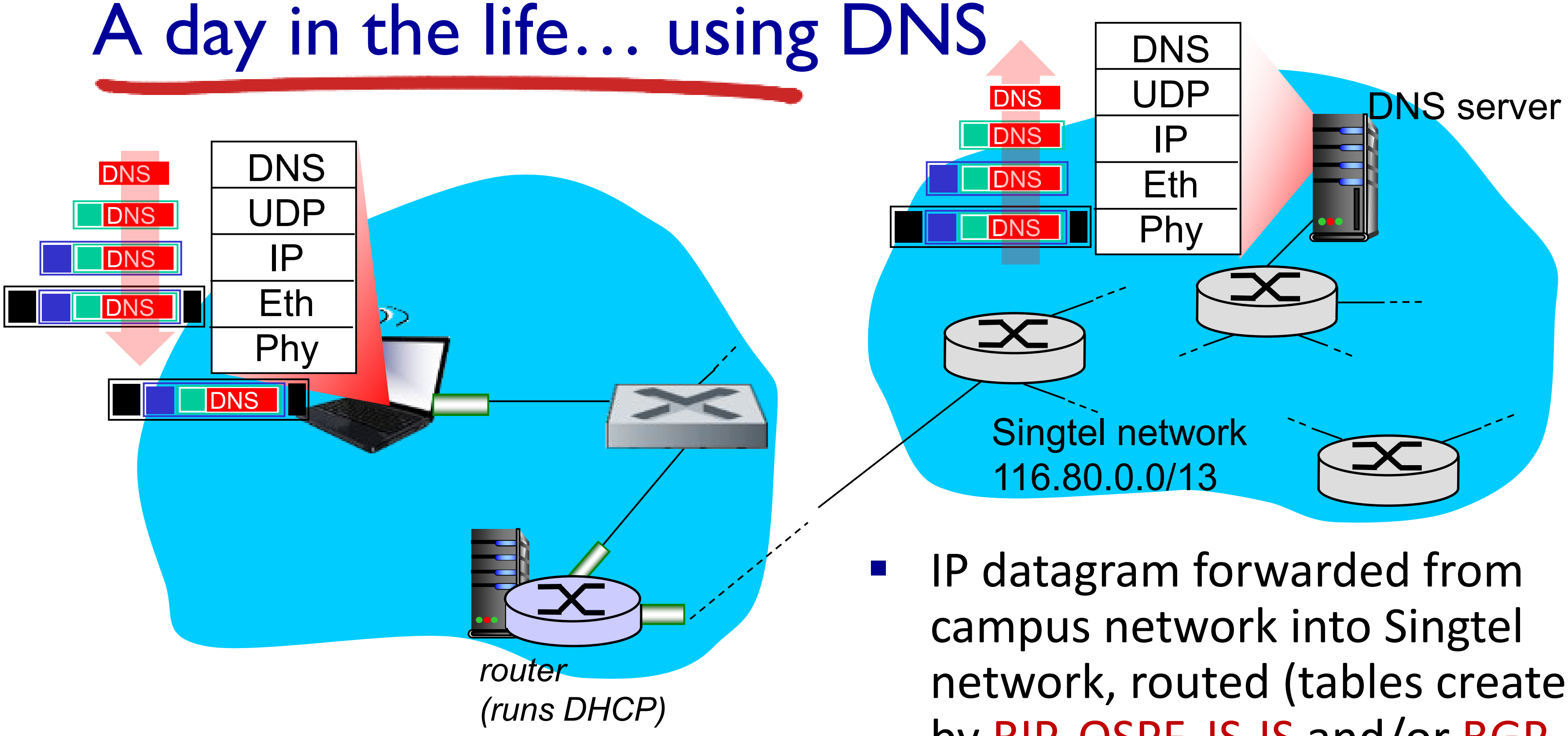
Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

A day in the life… ARP (before DNS, before HTTP)



- before sending *HTTP* request, need IP address of www.google.com
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth.
  - To send frame to router, need MAC address of router interface
- ARP query broadcast, received by router, which replies with ARP reply giving MAC address of router interface
- client now knows MAC address of first hop router, so can now send frame containing DNS query

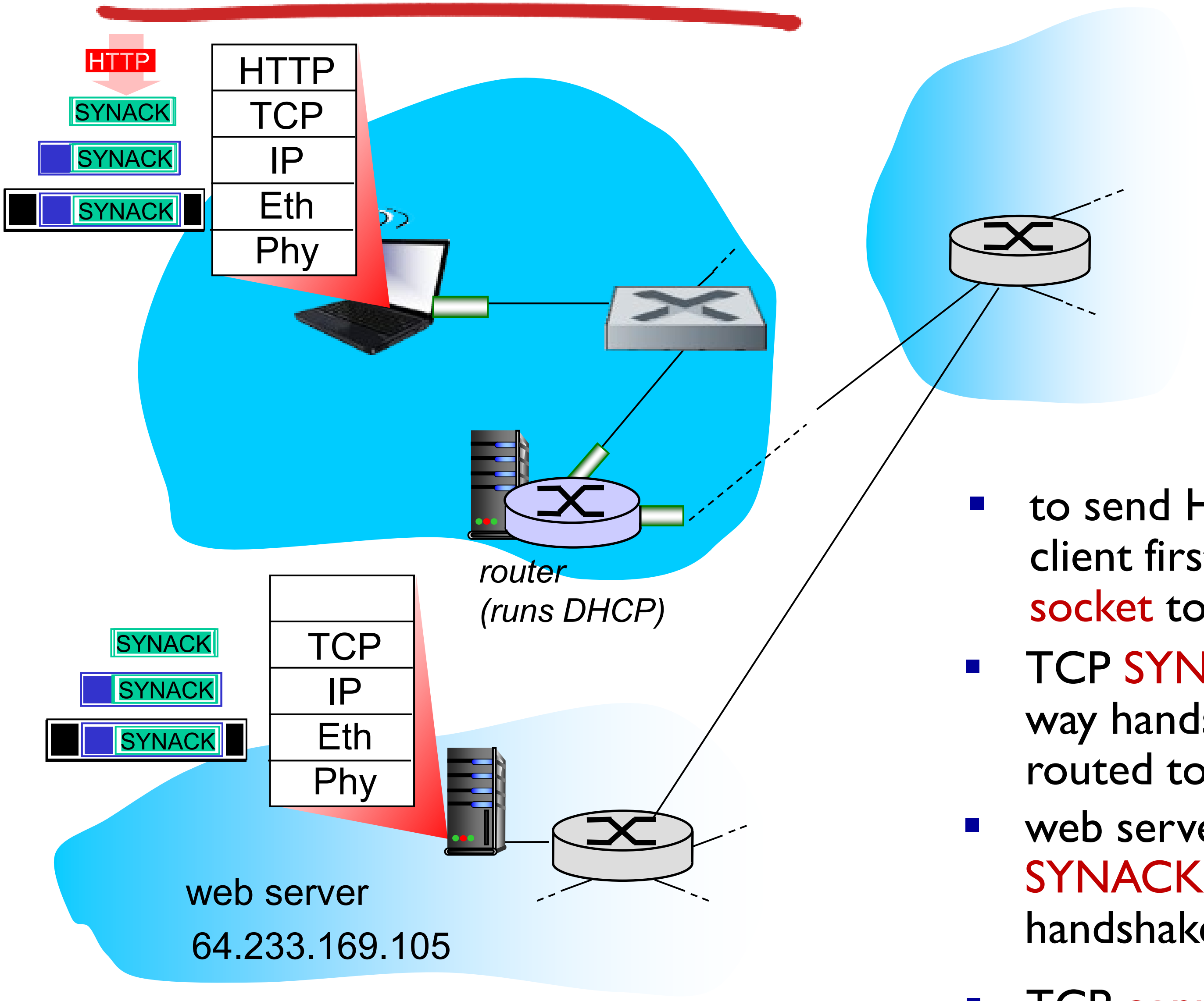
A day in the life… using DNS



- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router

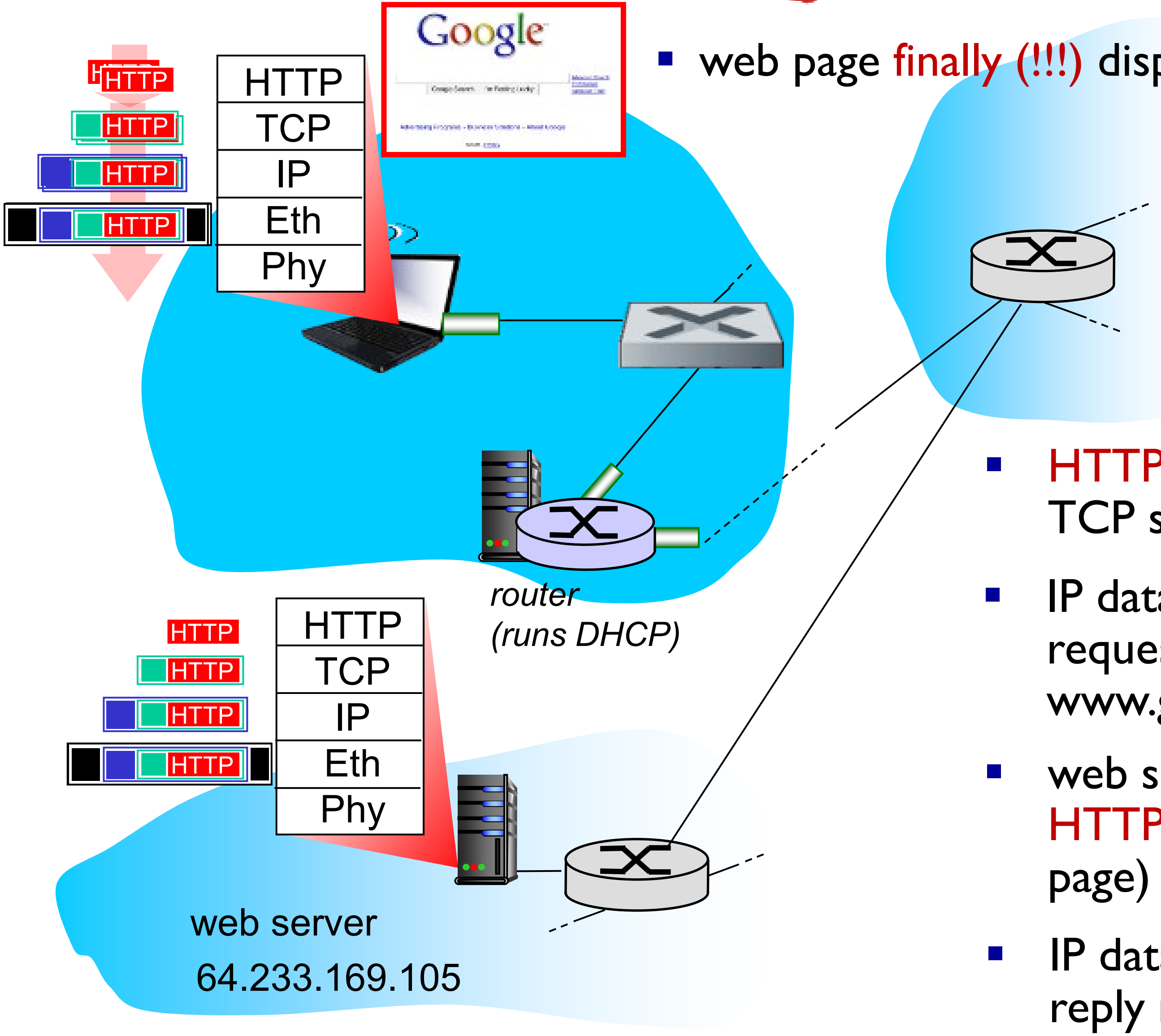
- IP datagram forwarded from campus network into Singtel network, routed (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols) to DNS server
- demuxed to DNS server
- DNS server replies to client with IP address of www.google.com

A day in the life…TCP connection carrying HTTP



- to send HTTP request, client first opens TCP socket to web server
- TCP SYN segment (step 1 in 3-way handshake) inter-domain routed to web server
- web server responds with TCP SYNACK (step 2 in 3-way handshake)
- TCP connection established!

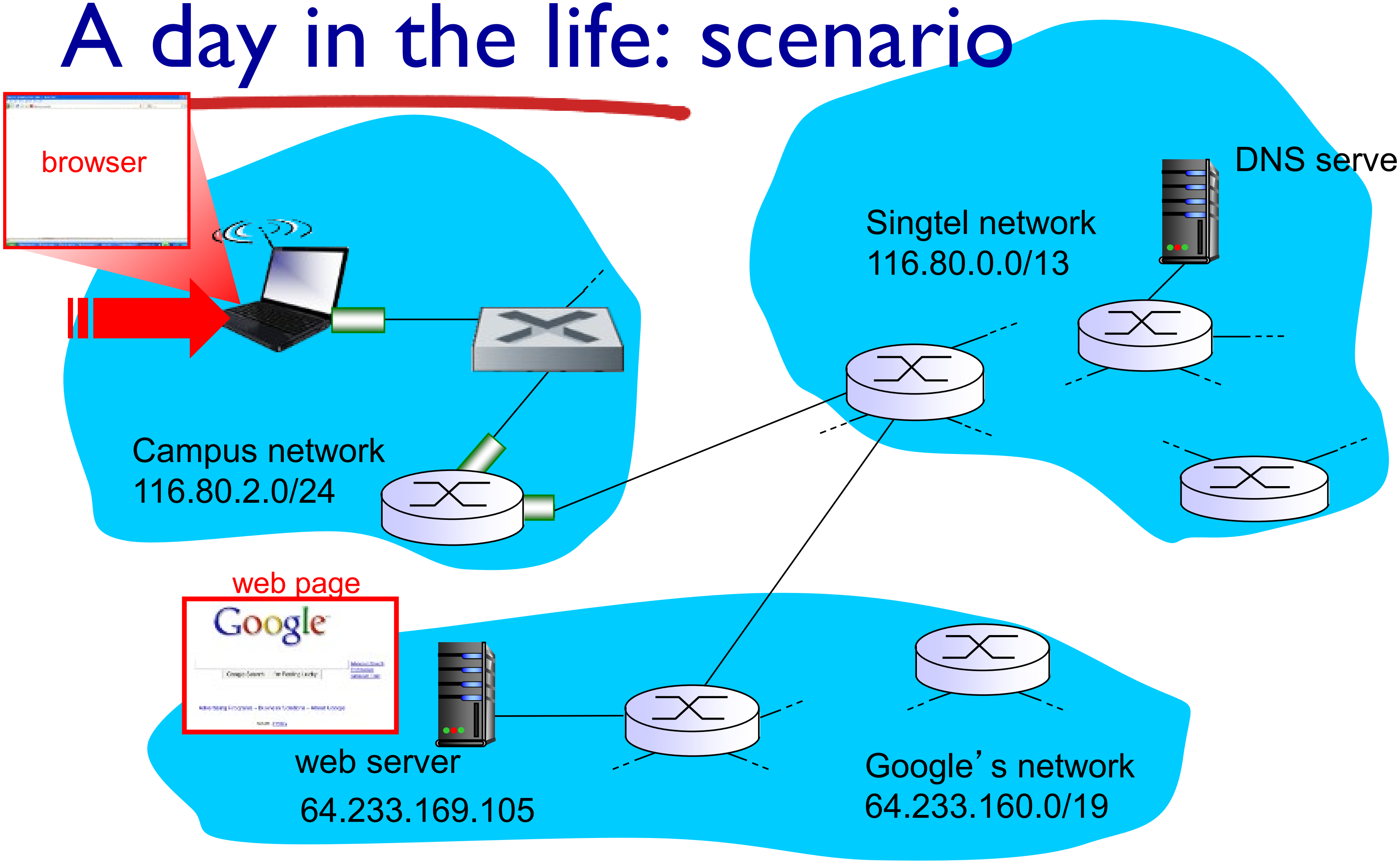
A day in the life… HTTP request/reply



- web page finally (!!!) displayed

- HTTP request sent into TCP socket
- IP datagram containing HTTP request routed to www.google.com
- web server responds with HTTP reply (containing web page)
- IP datagram containing HTTP reply routed back to client

A day in the life: scenario



browser

DNS server

Singtel network
116.80.0.0/13

Campus network
116.80.2.0/24

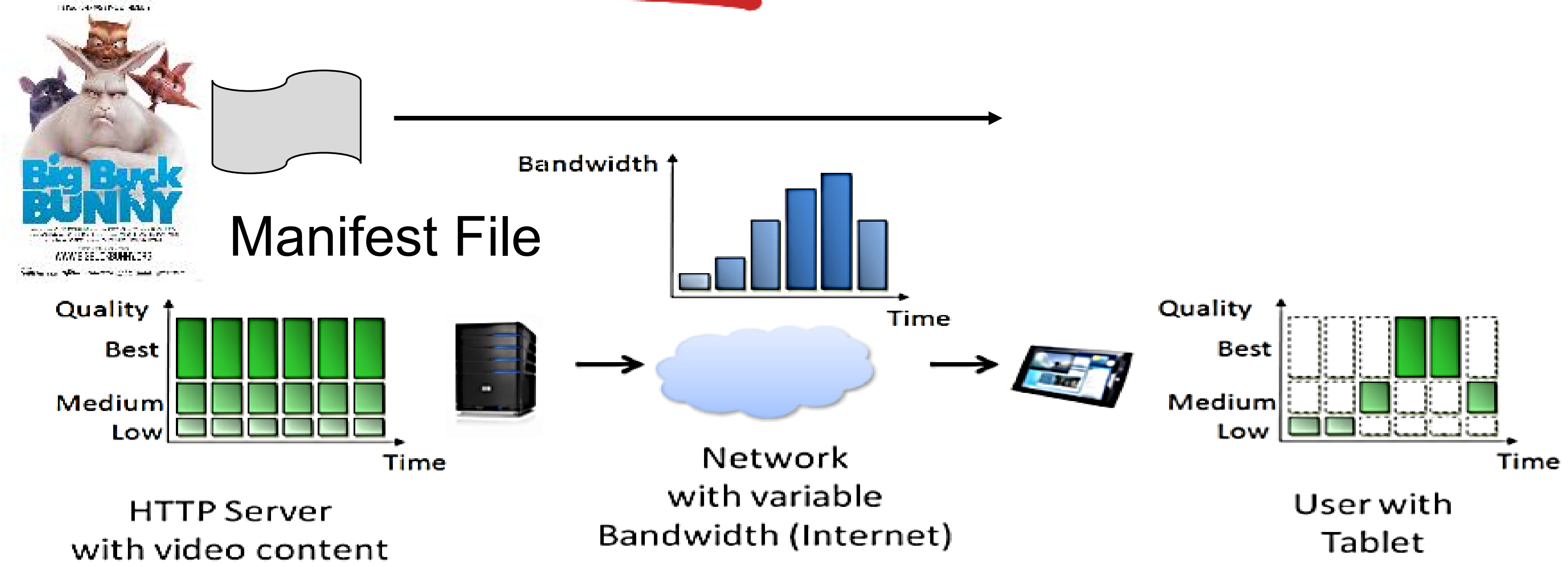
web page

web server

64.233.169.105

Google’s network
64.233.160.0/19

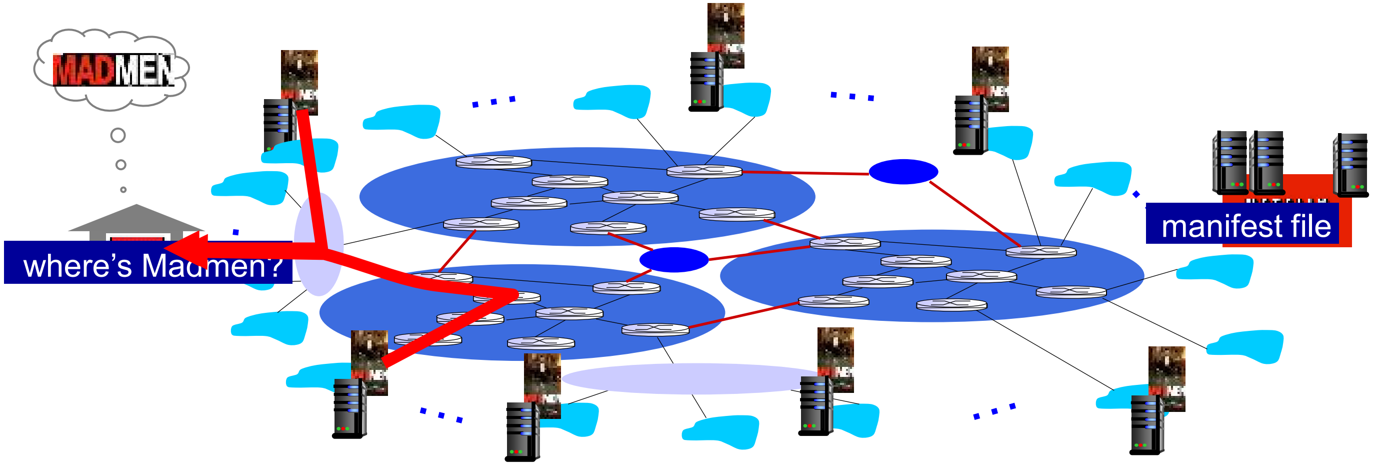
Streaming of Video: Dash



- Data is encoded into different qualities and cut into short segments (streamlets, chunks).
- Client first downloads *Manifest File*, which describes the available videos and qualities.
- Client/player executes an adaptive bitrate algorithm (ABR) to determine which segment do download next.

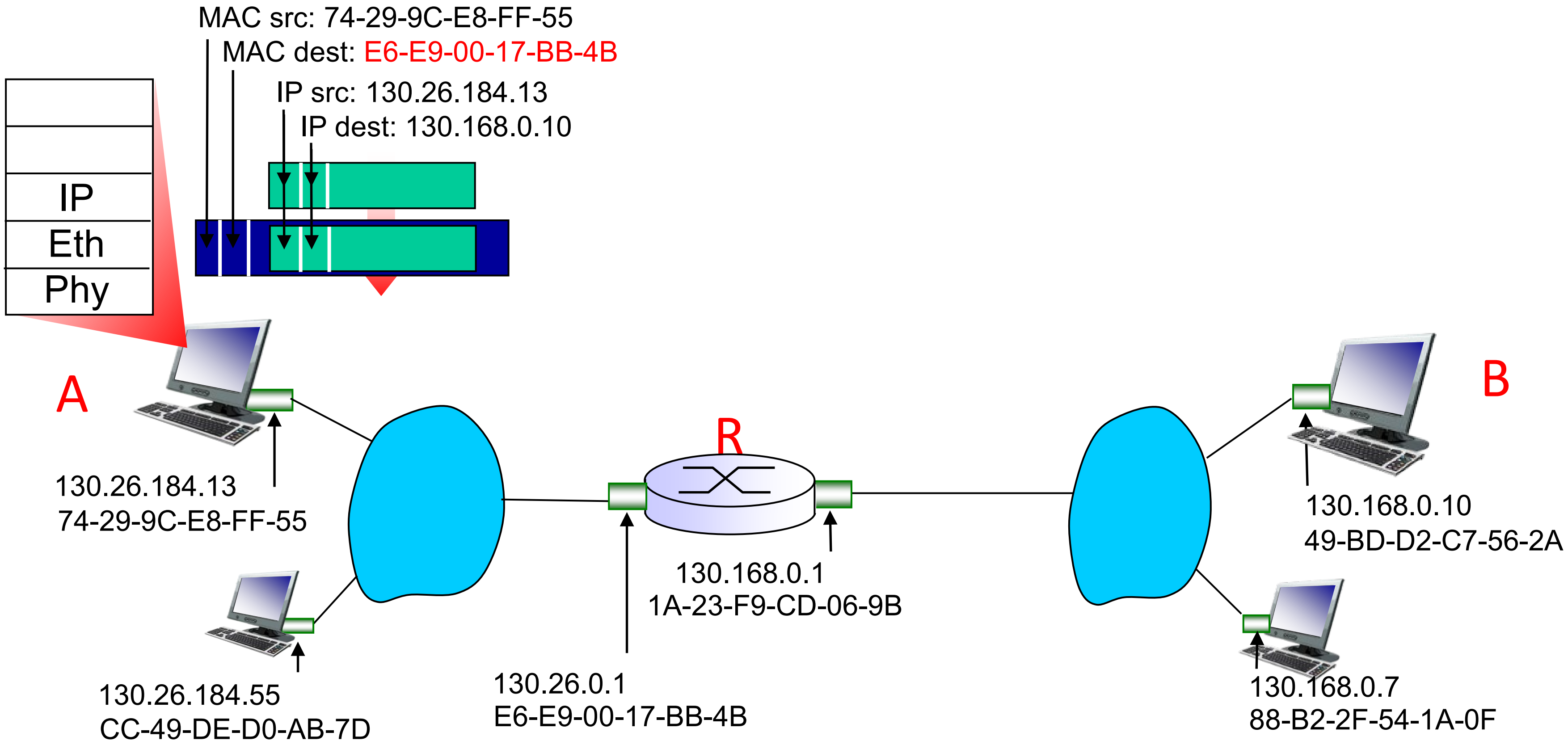
Content distribution networks (CDNs)

- CDN: stores copies of content (e.g. MADMEN) at CDN nodes
- Client requests content
  - service provider returns manifest
- using manifest, client retrieves content at highest supportable rate
- may choose different rate or copy if network path congested



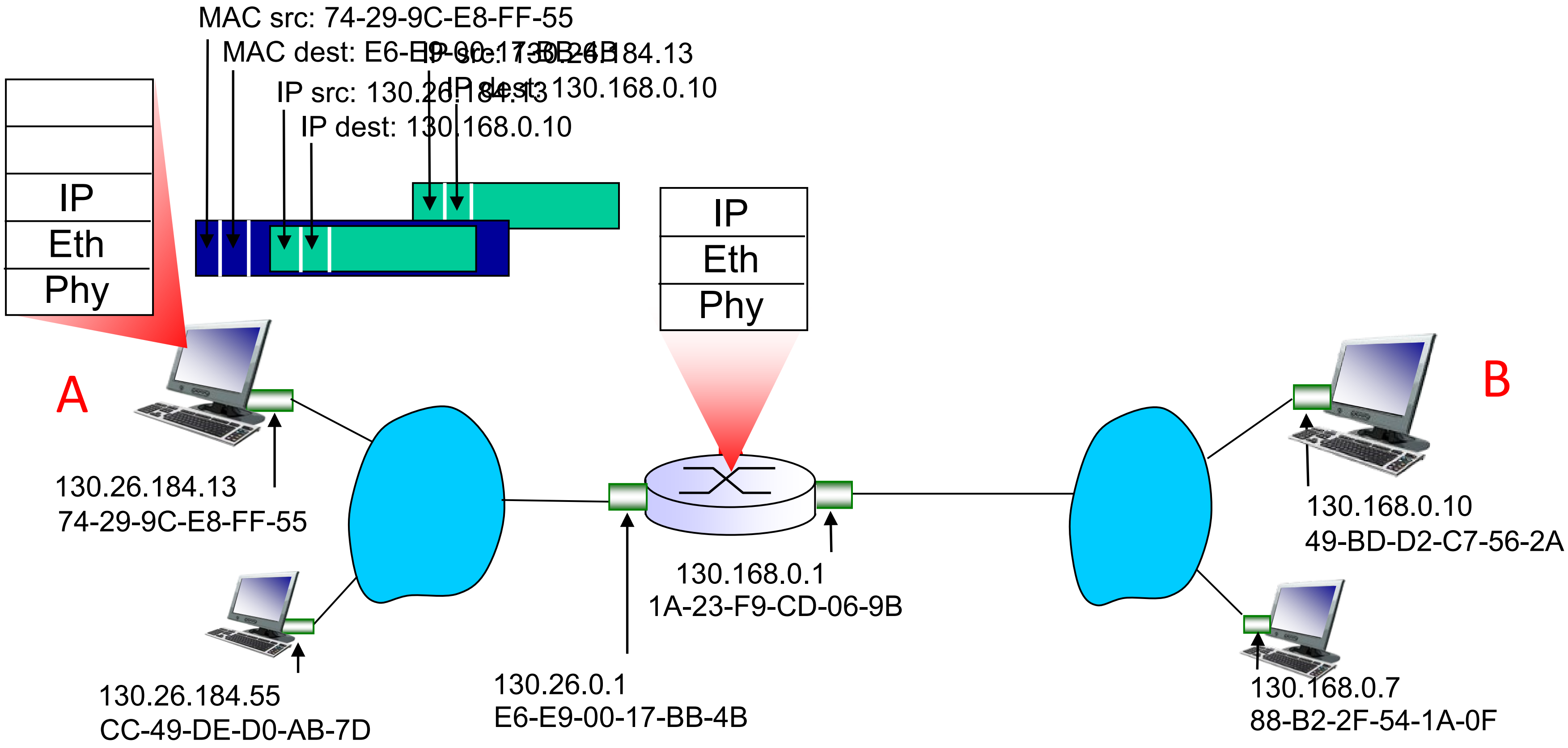
Sending Frame to Another Subnet

- *A* creates IP datagram with IP source *A*, destination *B*
- *A* creates link-layer frame with *R*'s MAC address as destination address, frame contains A-to-B IP datagram



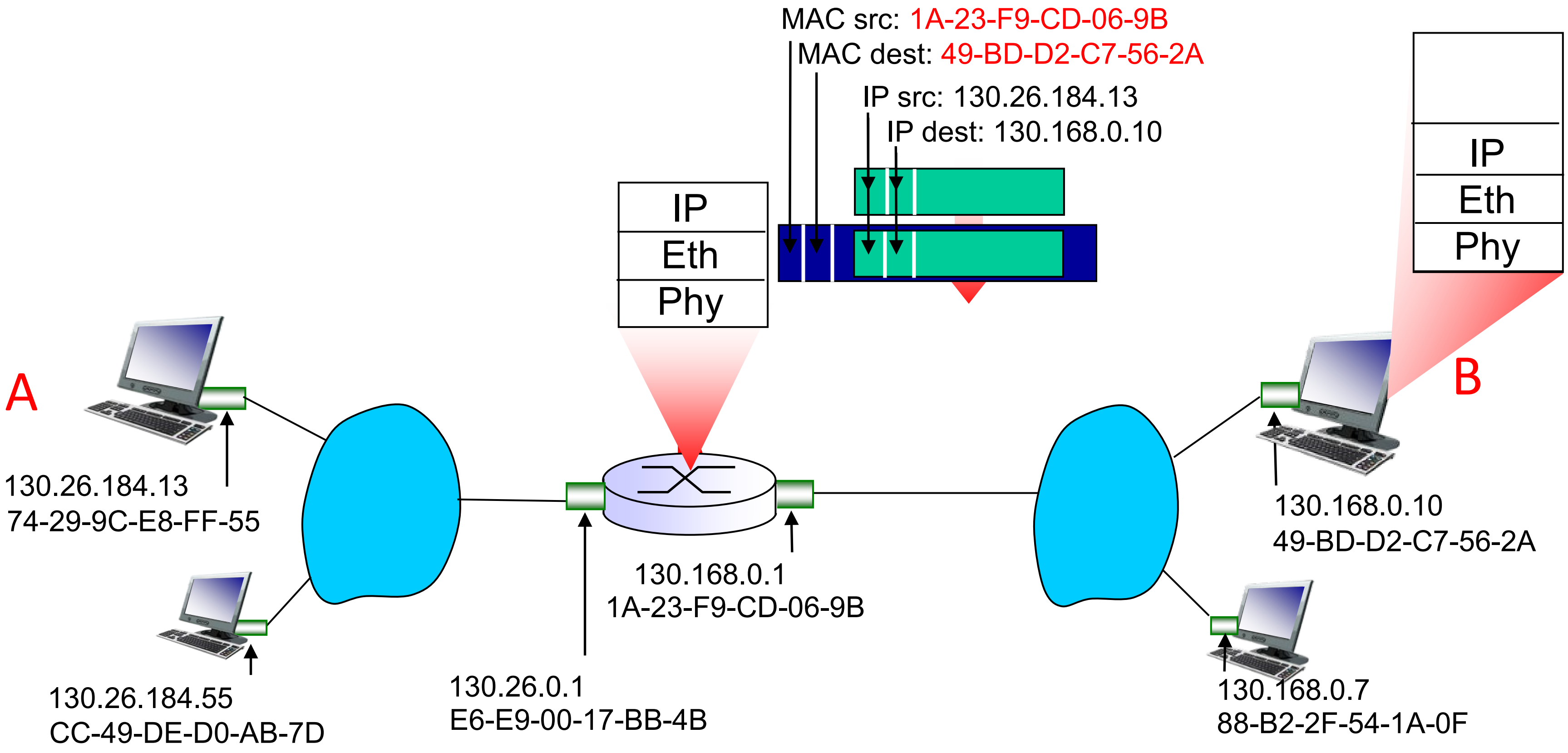
Sending Frame to Another Subnet

- frame sent from *A* to *R*
- frame received at *R*, datagram removed, passed up to IP



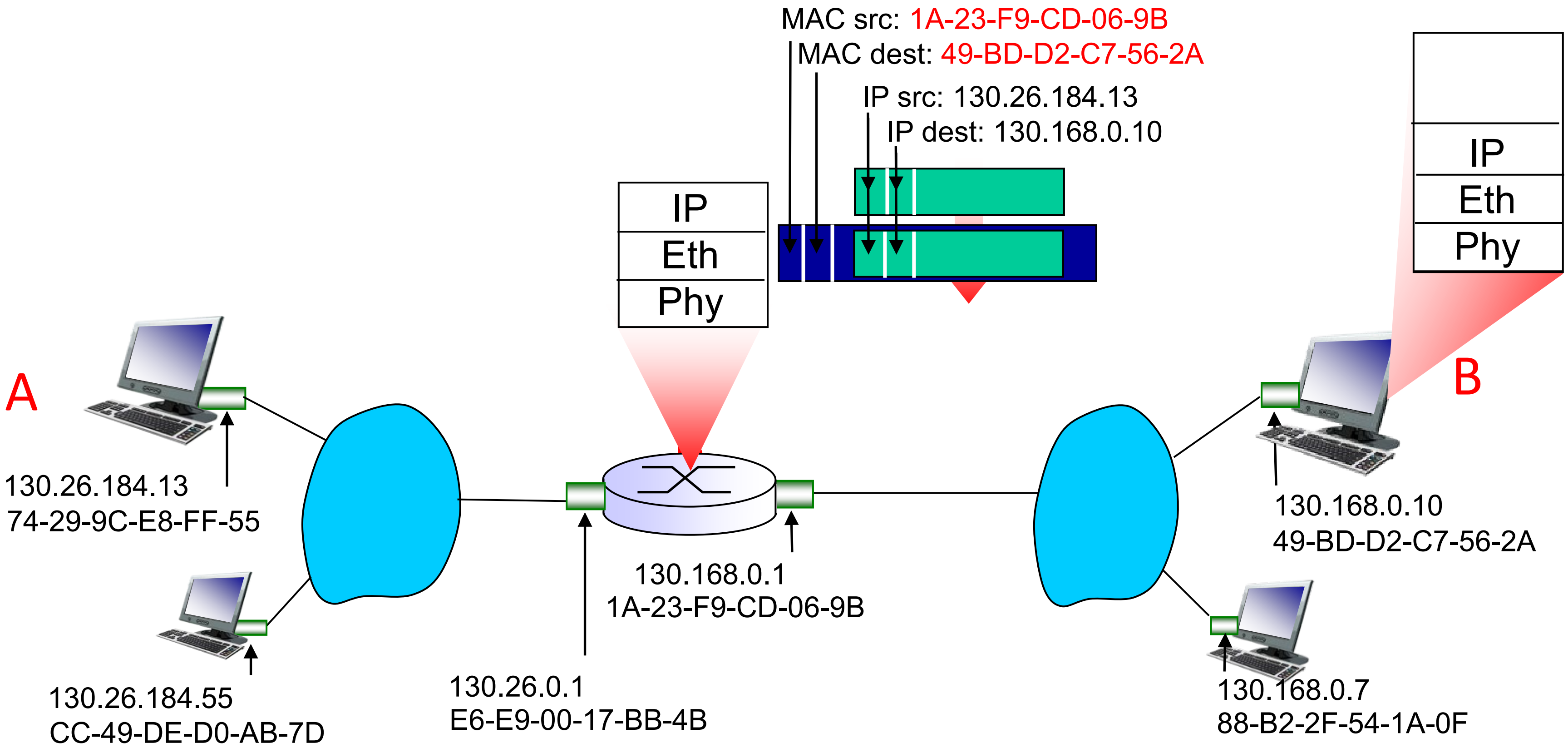
Sending Frame to Another Subnet

- *R* forwards datagram with IP source *A*, destination *B*
- *R* creates link-layer frame with *B* 's MAC address as destination address, frame contains A-to-B IP datagram



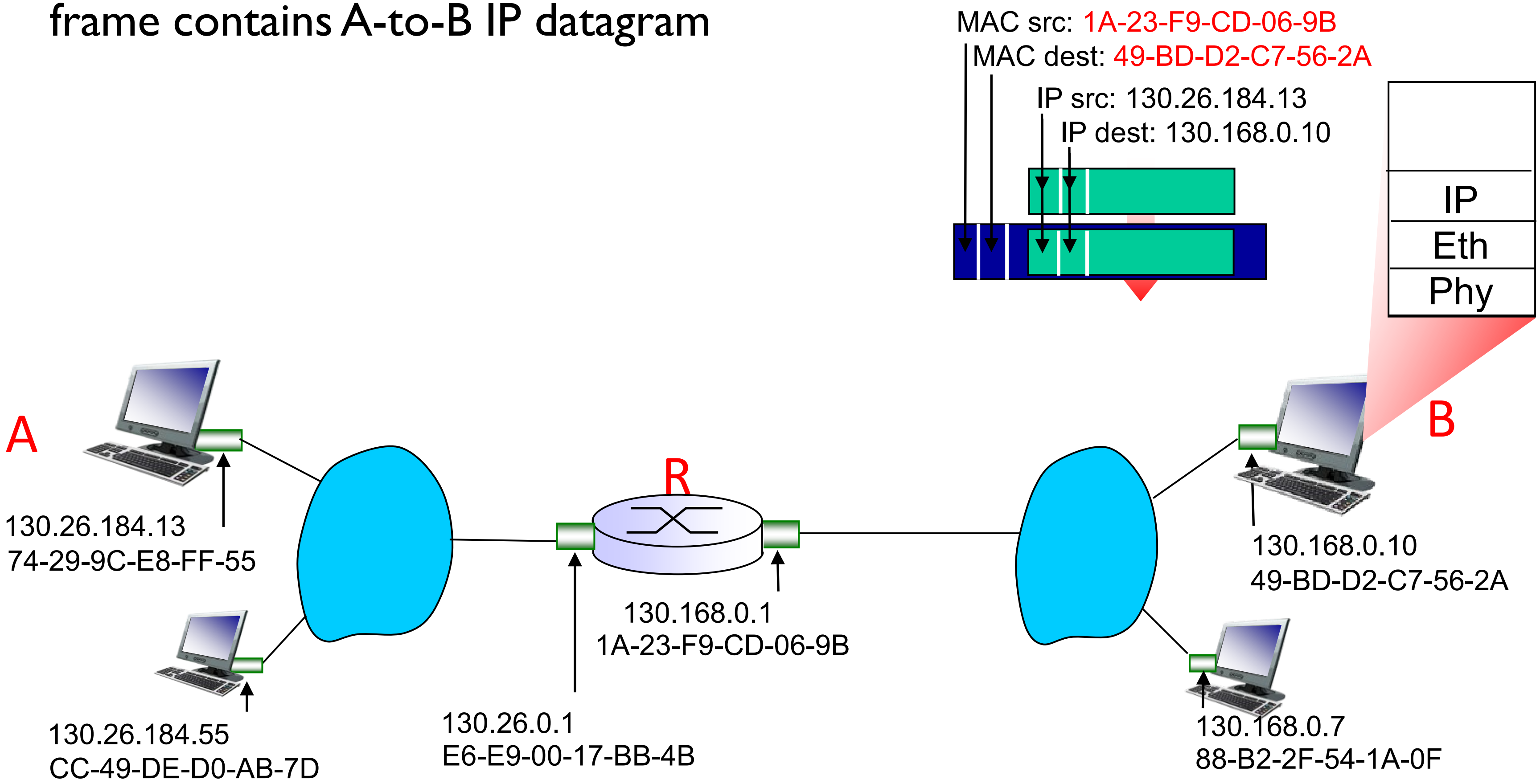
Sending Frame to Another Subnet

- *R* forwards datagram with IP source *A*, destination *B*
- *R* creates link-layer frame with *B* 's MAC address as destination address, frame contains A-to-B IP datagram



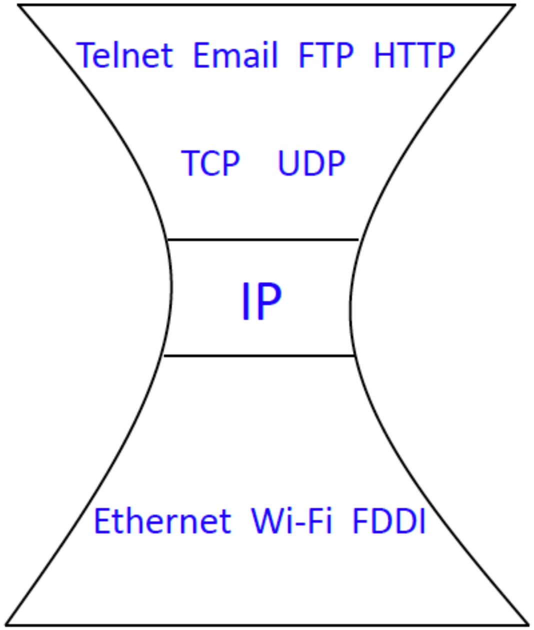
Sending Frame to Another Subnet

- *R* forwards datagram with IP source *A*, destination *B*
- *R* creates link-layer frame with *B* 's MAC address as destination address, frame contains A-to-B IP datagram



Lessons from CS2105

- Network systems are complex!
  - There are many issues to consider, to support different applications running on a large number of hosts through different access technologies and physical media.

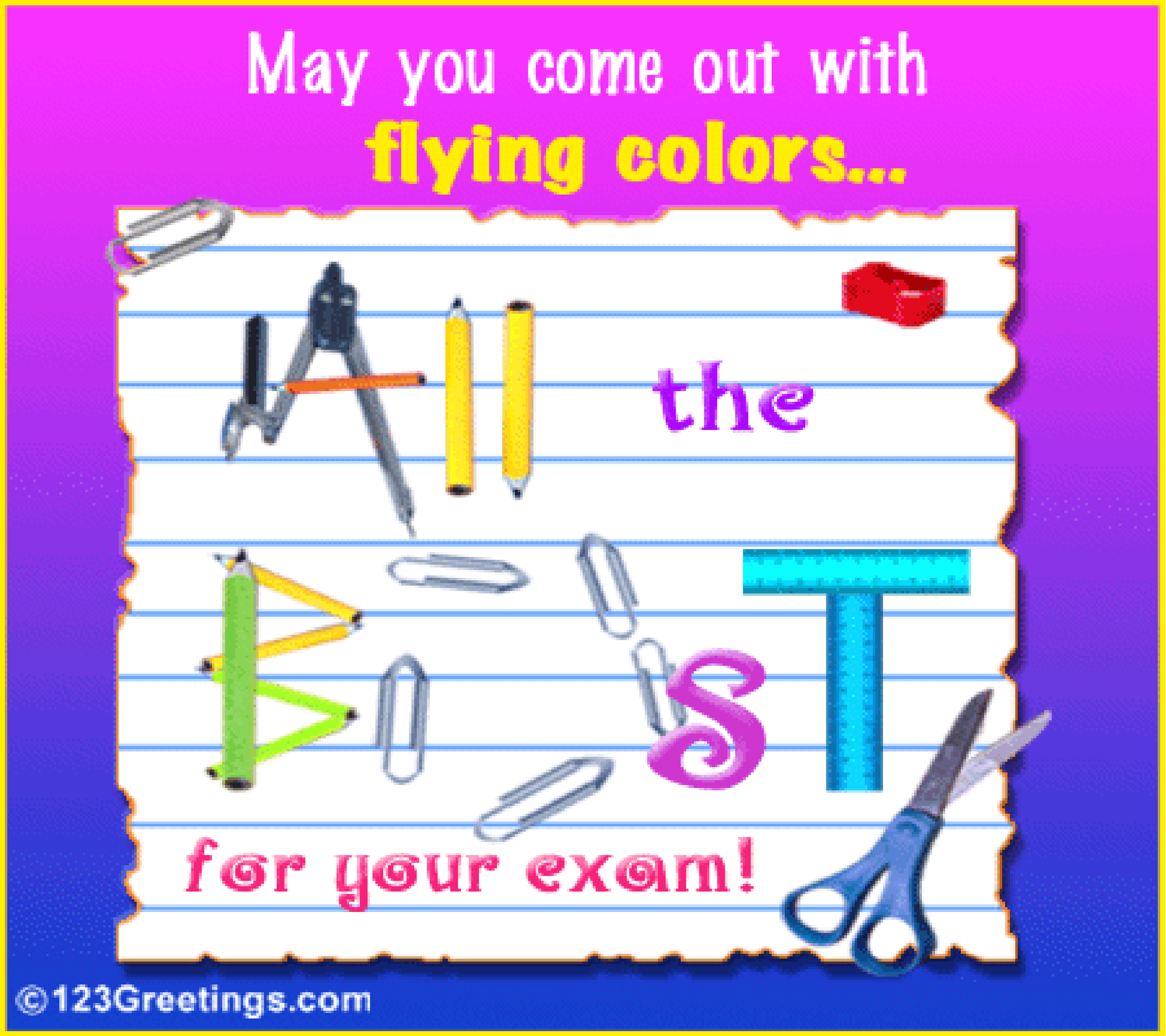


- To deal with complexity:
  - Separation of concerns
  - 5 protocol layers
- To deal with scalability:
  - Hierarchical systems

What’s Next?

- CS3103 Computer Networks Practice
    - Continuation of CS2105 in selected areas.
    - Use the same textbook as ours.
    - Cover network management, TCP congestion control and routing protocols in more details.
- CS4222 Wireless Networking
- CS4226 Internet Architecture
    - CS5229 Advanced Computer Networks
- CS3235 Computer Security
    - CS5321 Network security

Thank you!


May you come out with
flying colors...
All the BEST
for your exam!
©123Greetings.com