CSC 458/2209 – Computer Networking Systems

Handout # 2: Course Logistics and Introduction

Professor Yashar Ganjali

Department of Computer Science

University of Toronto

ganjali7@cs.toronto.edu

http://www.cs.toronto.edu/~yganjali

Today

- Outline
  - What this course is about
- Logistics
  - Course structure, assignments, evaluation
  - What is expected from you
  - What you can expect from this course
- Review
  - Simple example – mail vs. FTP
- Foundations and basic concepts

What is This Course About?

- Undergrad course; can be taken by grads
- Computer networks
  - Basics: layers, naming, and addressing, network (socket) programming, routing, congestion control, …
  - Advanced networking: peer-to-peer, routers and switch architectures, software-defined networking, datacenter networking, networks for ML, …
- Theory vs. Practice
  - CSC 457: focus on foundation, principles, and theory
  - CSC 458: focus on networking systems and programming

Logistics – Prerequisites, Readings

- Prerequisites
  - Algorithms
  - Basic probability theory
  - Strong background in C programming and Unix environment

- Note: CSC 457 is not a prerequisite.

- Readings
  - Will be posted on course schedule web page
  - Read before class

Logistics – Textbooks

- Textbook
  - Computer Networks: A Systems Approach (6th Edition), Peterson, Davie, 2021.

- You can get a print copy of this book. Alternatively, you can find an online version here:
  - https://book.systemsapproach.org/

- Recommended book
  - UNIX Network Programming, Volume I: The Sockets Networking API, W. Richard Stevens, Bill Fenner, and Andrew M. Rudoff, 3rd edition, 2003

Logistics – Sections

- This course is offered in two sections
  - L0101: Tue. 1-3 PM, GB 119
  - L5101: Tue. 6-8 PM, BA1180
- Please only attend your assigned section.
  - Particularly important for quizzes and midterm exam.
- Content/assignments are the same for the two sections.
  - Midterm exam is different.

Logistics – Hours, Web, Announcements

- Office hours
  - **Time:** Tue. 3-4 PM, Tue. 5-6 PM, or by appointment
  - **Location:** BA5238

- Course web page

  http://www.cs.toronto.edu/~yganjali/teaching/csc458-fall-2025/

- Please check the class web page, and the bulletin board regularly for announcements.

Logistics – Teaching Assistants

- Hossein Bijanrostami
- Ehsan Etesami
- David Lin
- Parsa Pazhooheshy (Head TA)
- Farid Zandi Shafagh

- Please check class web site for email addresses, and assignments for which each TA is responsible.

Logistics – TA hours, Tutorials

- Tutorials and discussion session
  - L0101: Thu. 1-2 PM, MP 137
  - L5101: Tue. 8-9 PM, BA 1180

- First tutorial:
  - Next week

Logistics – Mailing List, Bulletin Board

- Bulletin board
  - We will use Piazza for announcements and Q&A
    - https://piazza.com/utoronto.ca/fall2025/csc458csc2209/home
    - Sign up link on class web site
  - Post any questions related to the course.
  - Check previous posts before asking a question.
  - We guarantee to respond within 48 hours.

- Class mailing list
  - Based on e-mail address you have defined on ACORN.
  - The TAs and I will use this list for announcements only.
  - Do not send e-mails to this list!

Logistics – Grading

- Grading for undergraduate AND graduate students
  - Assignments: 40%
    - Two problem sets: 0%
    - Two programming: 20%
    - Two in-class quizzes (based on assignments): 20%
  - Midterm exam: 20% - In class
    - October 21
  - Final exam: 40% - TBA

- Please note that grading is the same for graduate and undergraduate students.

Logistics - Deadlines

- Assignment deadlines
  - One free late submission of 24 hours
    - Use on assignment of your choice
  - 10% deduction for each day late
    - Up to 20%
    - Assignment not accepted after two days

Logistics – Programming Assignments

- Implementing a simple network stack
- To be completed individually.
- You can submit your assignment during a seven-day period before the deadline
  - Your last submission before the deadline will be marked
- Marking:
  - Public tests: 50% of the mark
  - Private tests: 40%
  - Style and documentation: 10%

Logistics – Academic Integrity

- All submissions must present original, independent work.
- We take academic offenses very seriously.
- Please read
  - Handout # 1 (course information sheet)
  - "Guideline for avoiding plagiarism"
  - http://www.cs.toronto.edu/~fpitt/documents/plagiarism.html
  - "Advice about academic offenses"
  - http://www.cs.toronto.edu/~clarke/acoffences/
- Use of AI tools: OK to use for general questions, not specific ones related to assignments.
  - Please see Handout #1 for more information.

Logistics - Accessibility

- Accessibility Needs
  - The University of Toronto is committed to accessibility. If you require accommodations or have any accessibility concerns, please visit accessibility services as soon as possible.
  - https://studentlife.utoronto.ca/department/accessibility-services/

Acknowledgements

- Special thanks to:
  - Prof. Nick McKeown from Stanford University
  - Prof. Jennifer Rexford from Princeton University
  - Prof. David Wetherall from University of Washington
  - Prof. Nick Feamster from University of Chicago
  - Dr. Soheil Abbasloo from Microsoft Research

Quick Survey

- Have you taken another networking course before?
- Are you familiar with
  - Socket programming?
  - Ethernet, framing, encoding, error detection/correction?
  - UDP, TCP and congestion control?
  - DNS, SNMP, BGP?
  - Voice and video over IP?
  - Network security?
  - Software-defined networking?
  - Control plane vs. data path?
  - Datacenter networks?
  - Networks for machine learning?

Questions?

What else do you want to know about this course?

Announcement

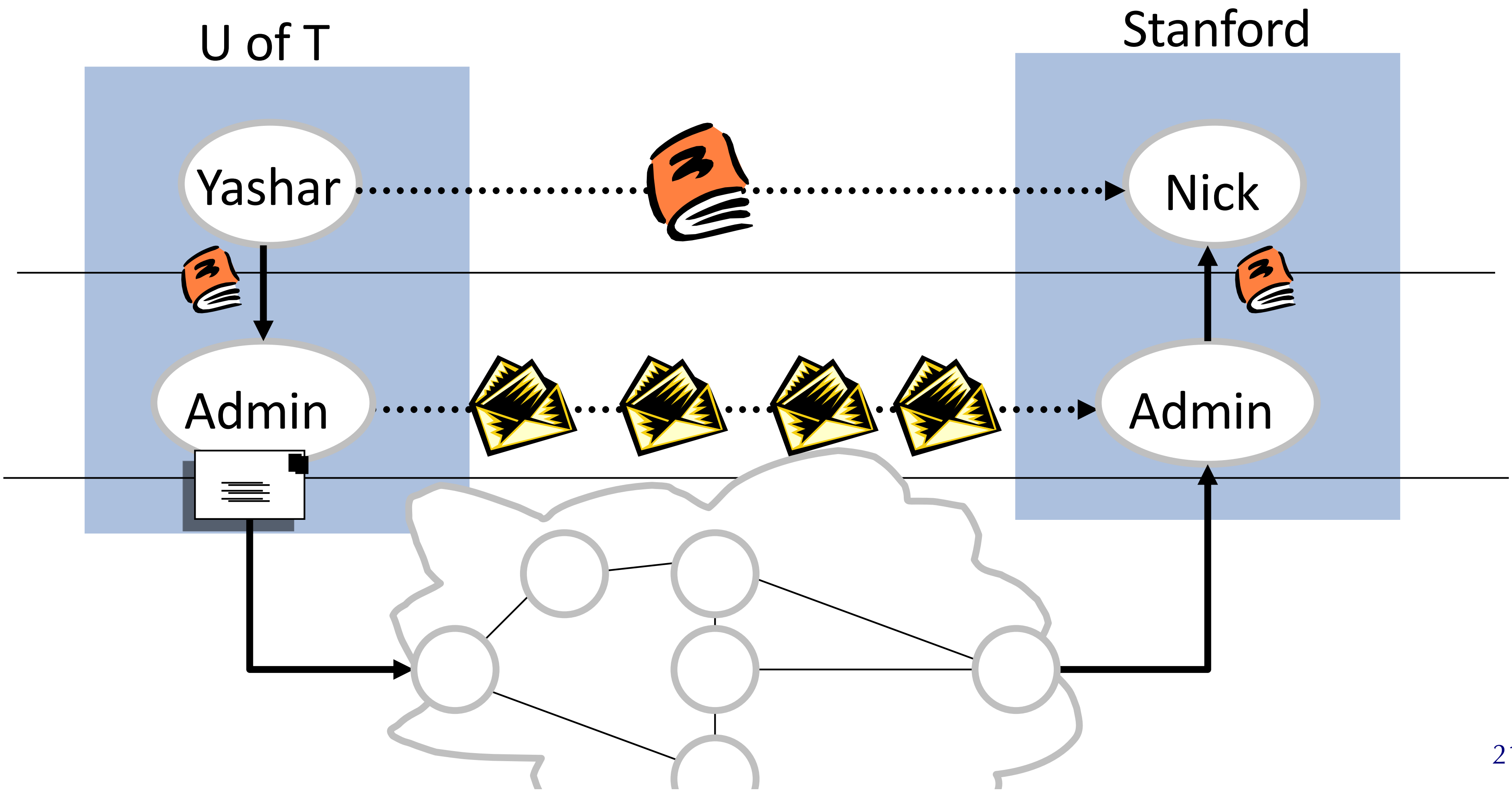
- First tutorial
  - Next week

- Covers *socket programming*

Let's Begin

- An introduction to the mail system
- An introduction to the Internet

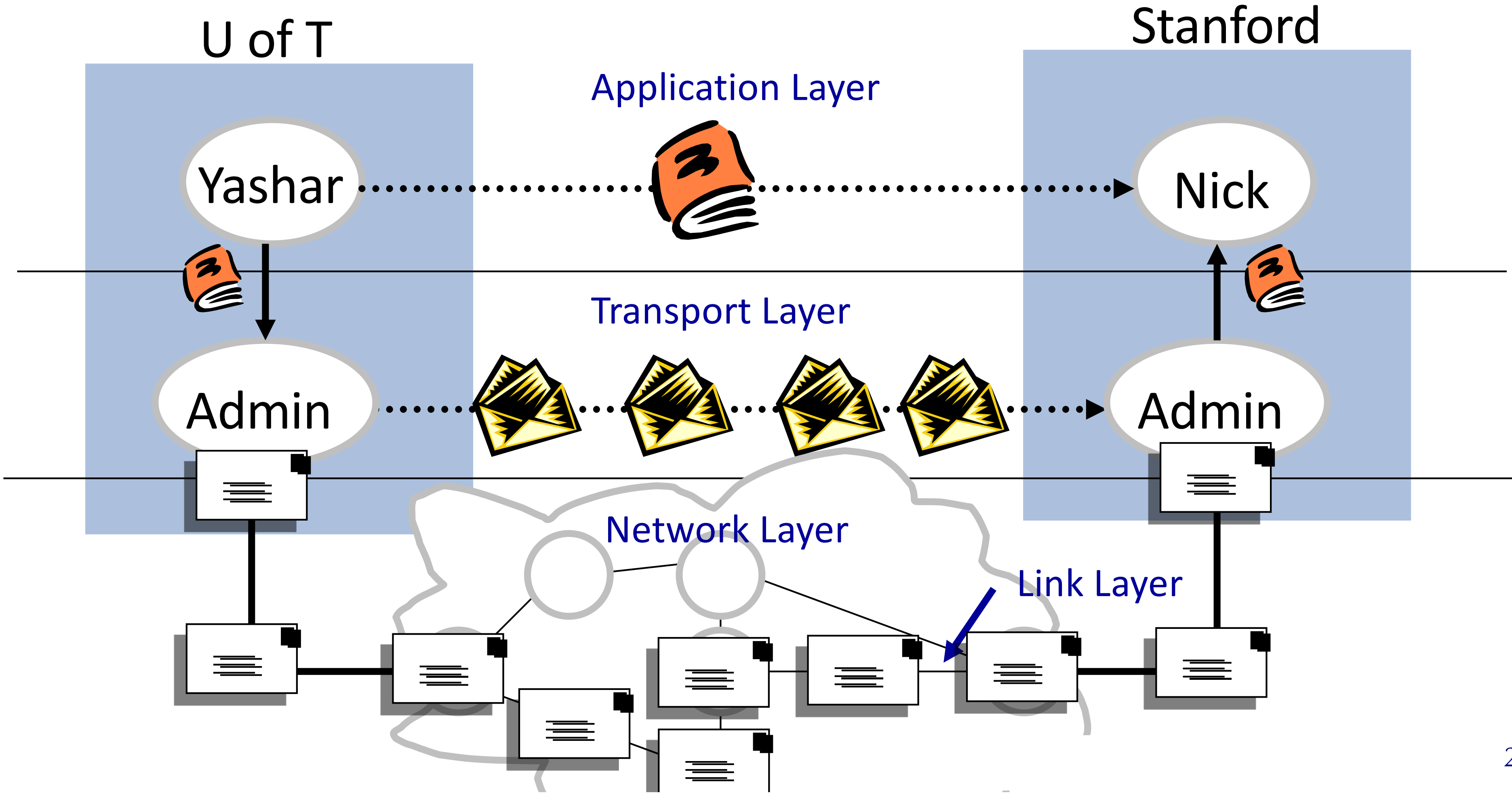
An Introduction to the Mail System



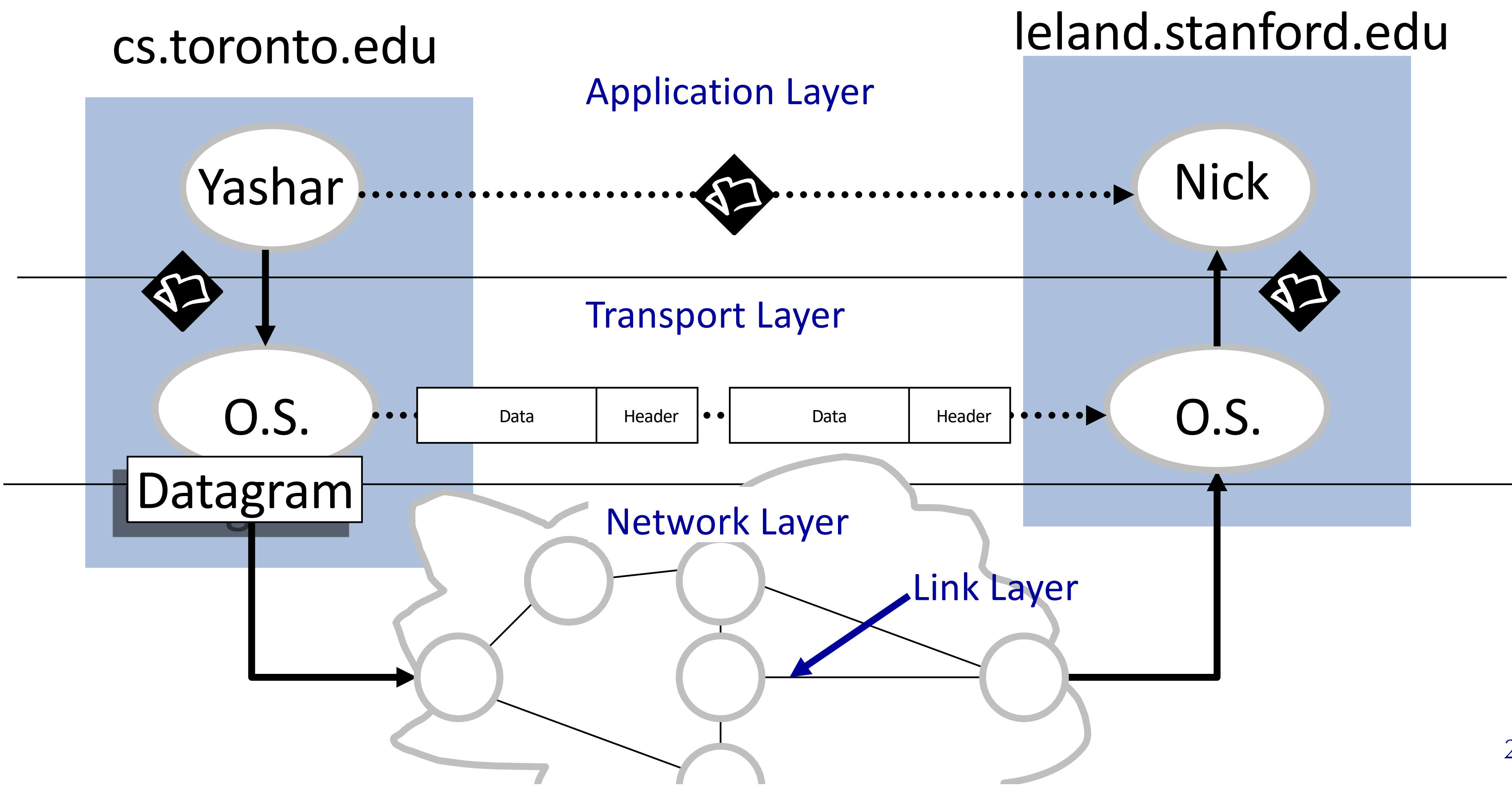
Characteristics of the Mail System

- Each envelope is individually routed.
- No time guarantee for delivery.
- No guarantee of delivery in sequence.
- No guarantee of delivery at all!
  - Things get lost
  - How can we acknowledge delivery?
  - Retransmission
    - How to determine when to retransmit? Timeout?
    - Need local copies of contents of each envelope.
    - How long to keep each copy.
    - What if an acknowledgement is lost?

An Introduction to the Mail System



An Introduction to the Internet



Characteristics of the Internet

- Each packet is individually routed.
- No time guarantee for delivery.
- No guarantee of delivery in sequence.
- No guarantee of delivery at all!
  - Things get lost
  - Acknowledgements
  - Retransmission
    - How to determine when to retransmit? Timeout?
    - Need local copies of contents of each packet.
    - How long to keep each copy?
    - What if an acknowledgement is lost?

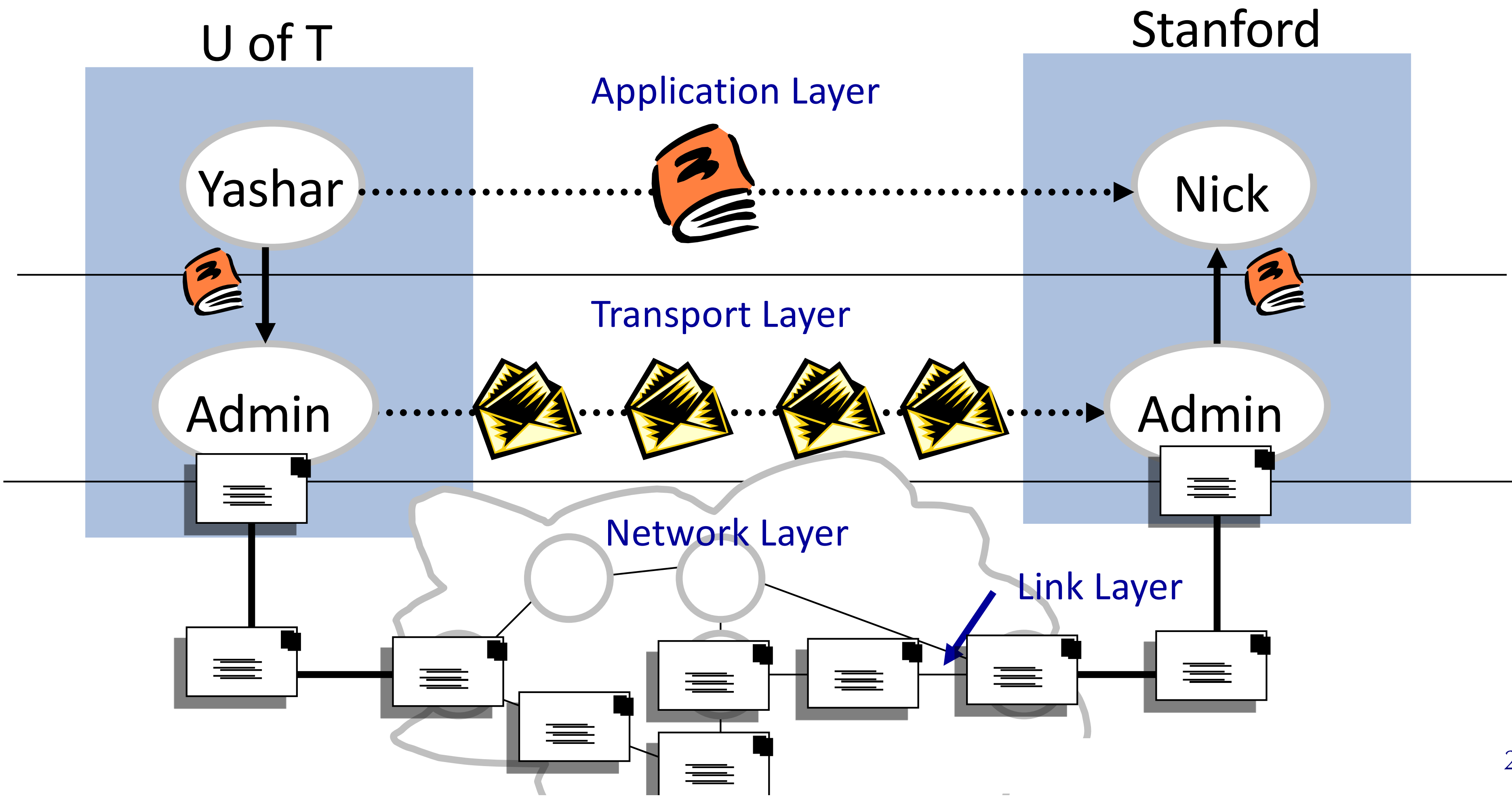
Characteristics of the Internet – Cont'd

- No guarantee of integrity of data.
- Packets can be fragmented.
- Packets may be duplicated.

Layering in the Internet

- Transport Layer
  - Provides reliable, in-sequence delivery of data from end-to-end on behalf of application.
- Network Layer
  - Provides “best-effort”, but unreliable, delivery of datagrams.
- Link Layer
  - Carries data over (usually) point-to-point links between hosts and routers; or between routers and routers.

An Introduction to the Mail System



Some Questions About the Mail System

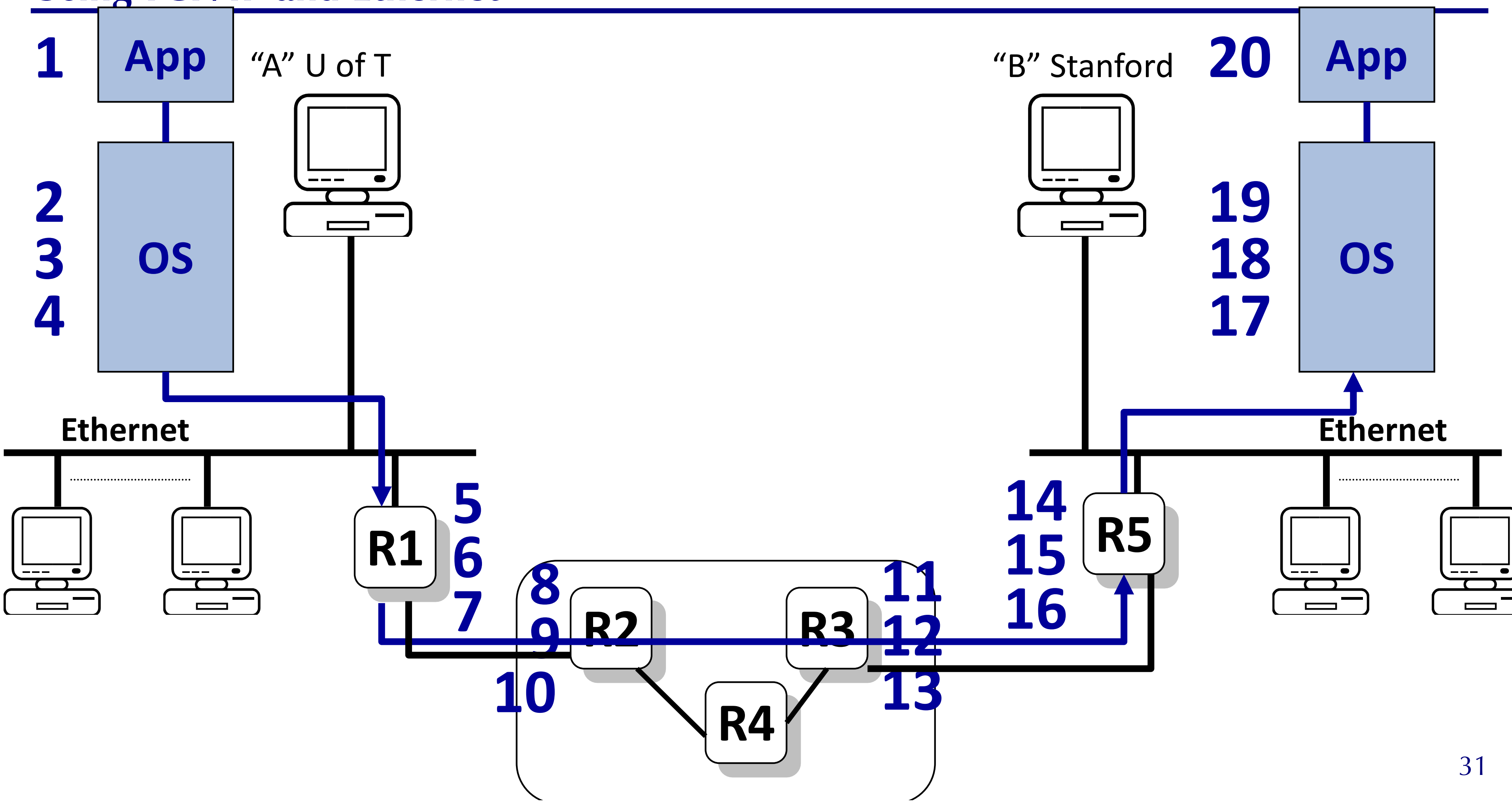
- How many sorting offices are needed and where should they be located?
- How much sorting capacity is needed?
  - Should we allocate for Mother's Day?
- How can we guarantee timely delivery?
  - What prevents delay guarantees?
  - Or delay variation guarantees?
- How can we build an infra-structure for overnight deliveries?
  - What are the challenges for extremely tight deadlines?
- How do we protect against fraudulent mail deliverers, or fraudulent senders?

Outline – Foundations & Basic Concepts

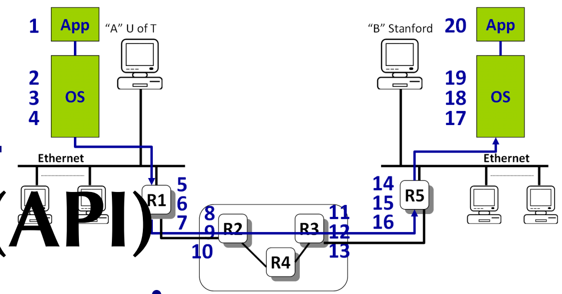
- A detailed FTP example
- Layering
- Packet switching and circuit switching

Example: File Transfer over the Internet

Using TCP/IP and Ethernet



In the Sending Host

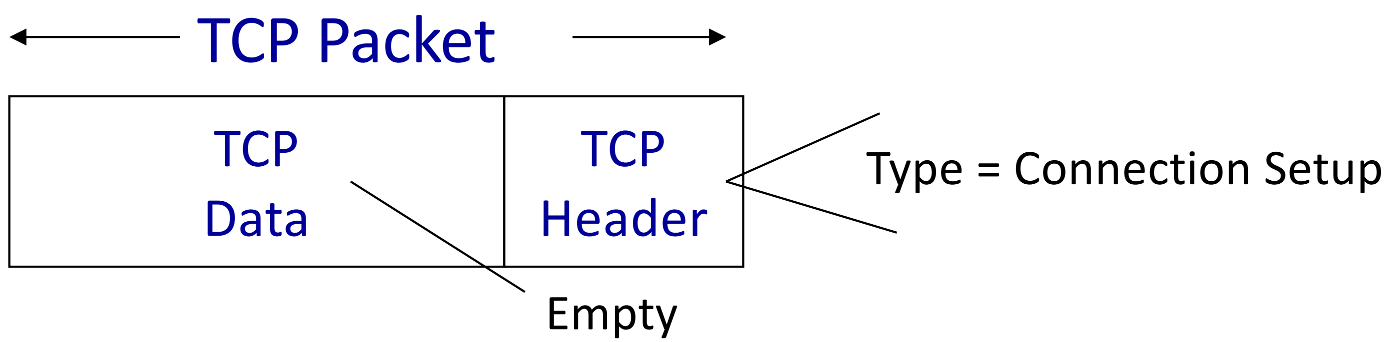


1. Application-Programming Interface (API)

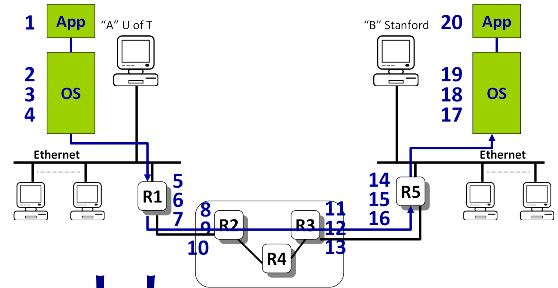
- Application requests TCP connection with "B"

2. Transmission Control Protocol (TCP)

- Creates TCP "Connection setup" packet
- TCP requests IP packet to be sent to "B"

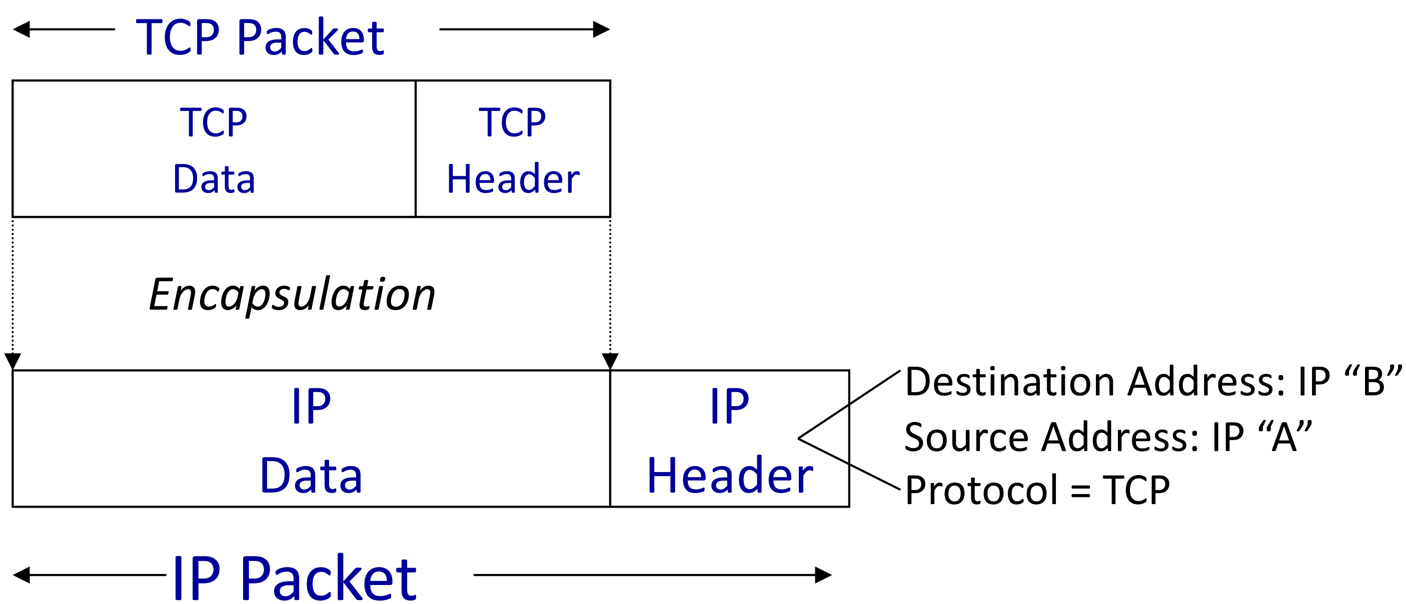


In the Sending Host – Cont'd

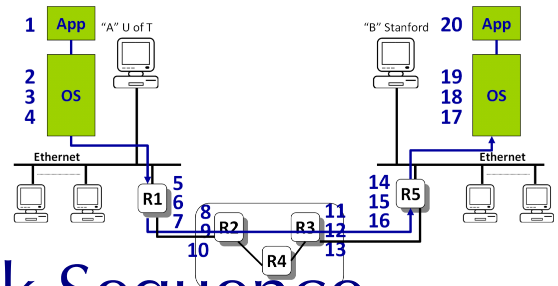


3. Internet Protocol (IP)

- Creates IP packet with correct addresses.
- IP requests packet to be sent to router.

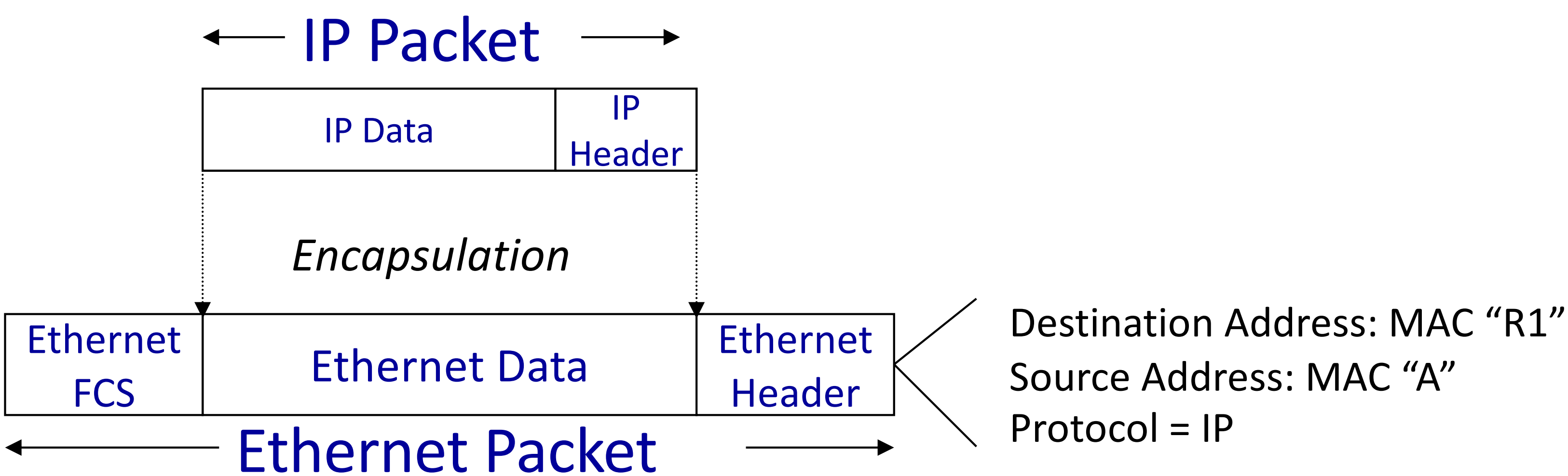


In the Sending Host – Cont’d

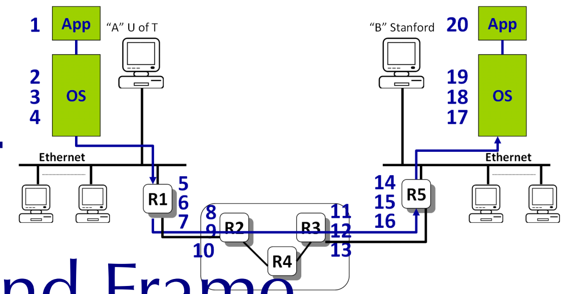


4. Link (“MAC” or Ethernet) Protocol

- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.

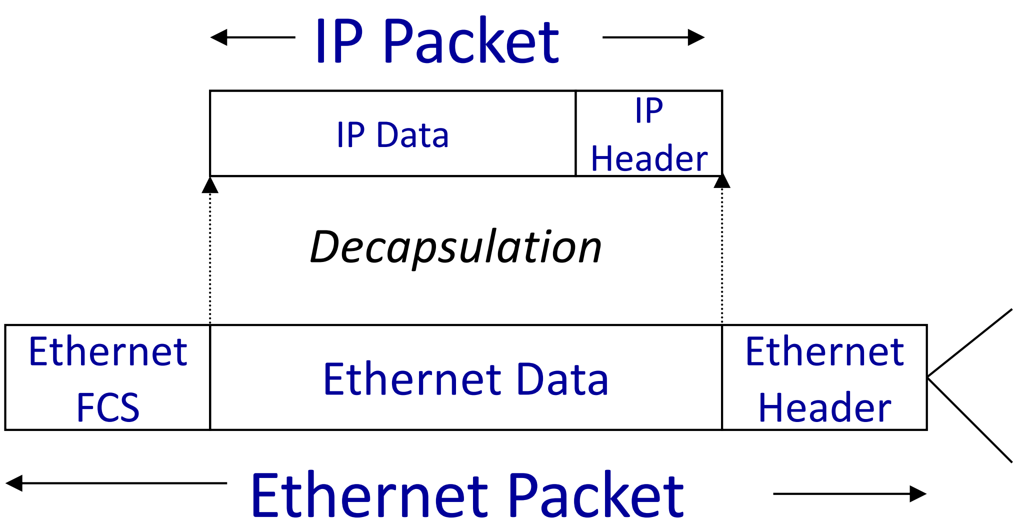


In Router R1



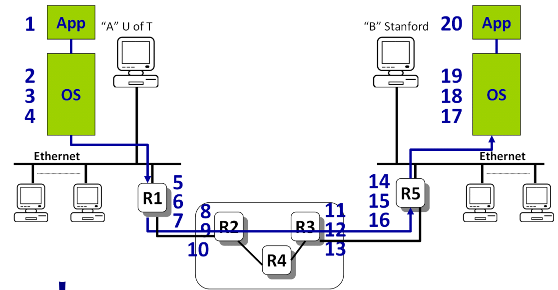
5. Link ("MAC" or Ethernet) Protocol

- Accept MAC frame, check address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.



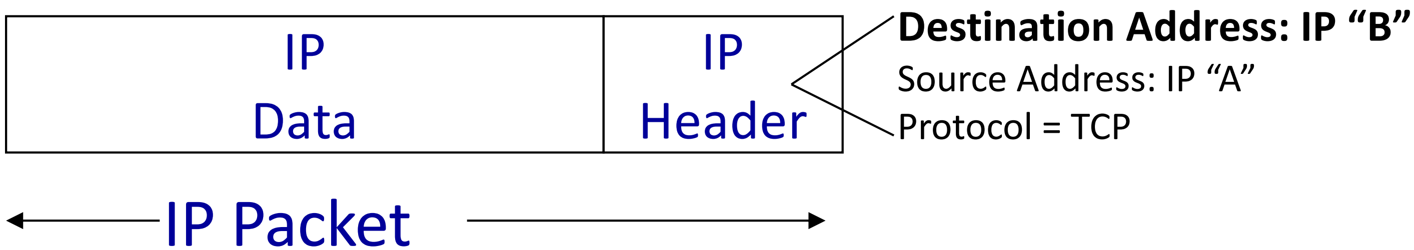
Destination Address: MAC "R1"
Source Address: MAC "A"
Protocol = IP

In Router R1



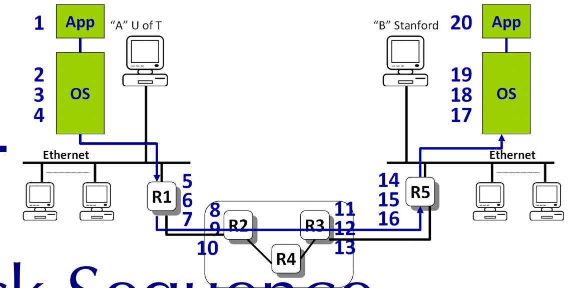
6. Internet Protocol (IP)

- Use IP destination address to decide where to send packet next ("next-hop routing").
- Request Link Protocol to transmit packet.

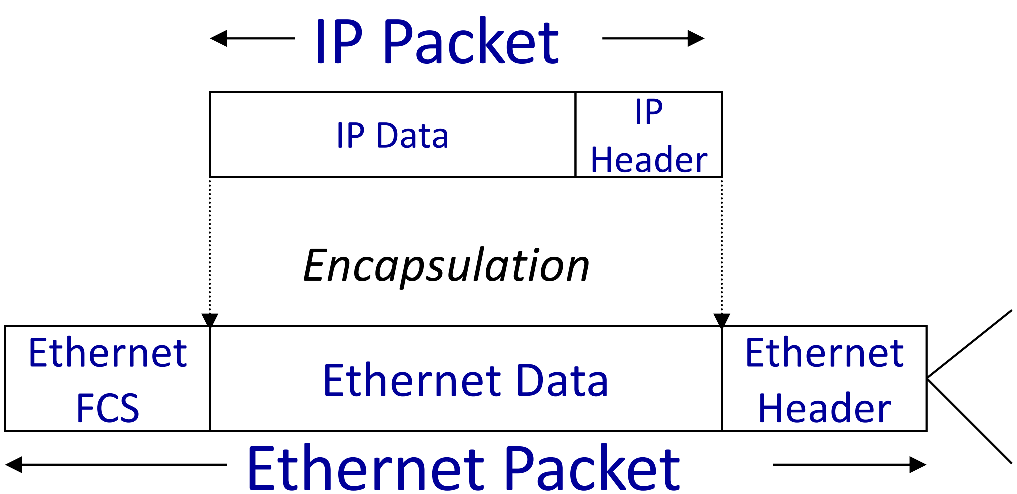


In Router R1

7. Link (“MAC” or Ethernet) Protocol

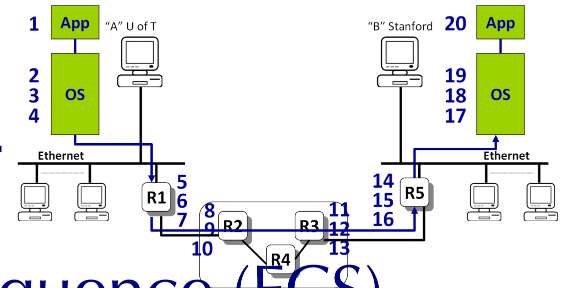


- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.



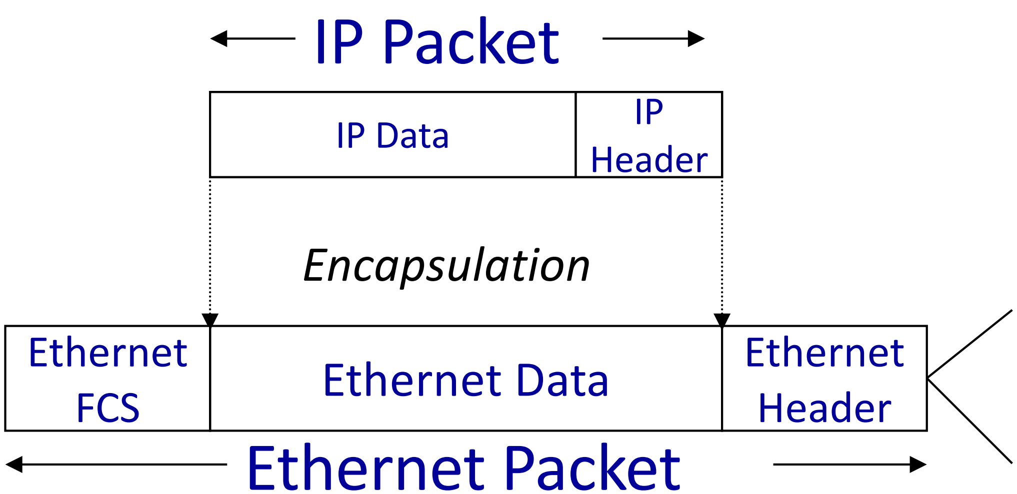
Destination Address: MAC “R2”
Source Address: MAC “R1”
Protocol = IP

In Router R5



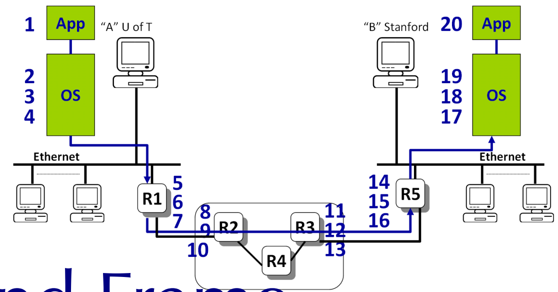
16. Link ("MAC" or Ethernet) Protocol

- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.



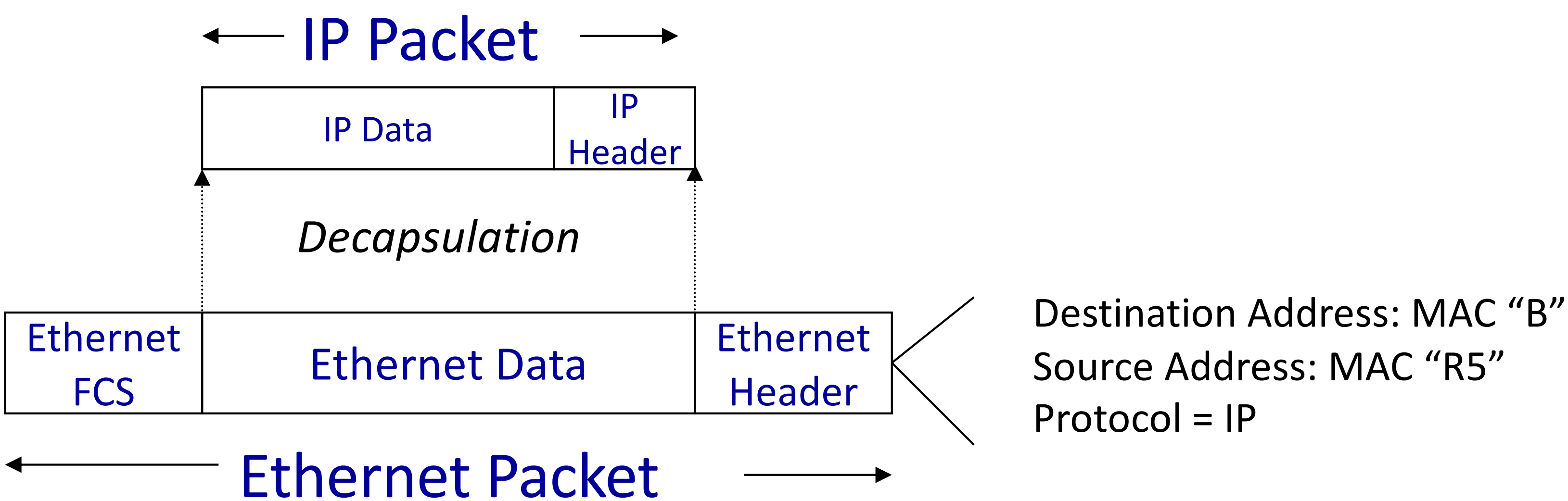
Destination Address: MAC "B"
Source Address: MAC "R5"
Protocol = IP

In the Receiving Host

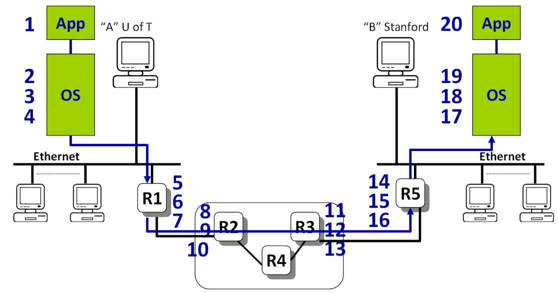


17. Link ("MAC" or Ethernet) Protocol

- Accept MAC frame, check address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.

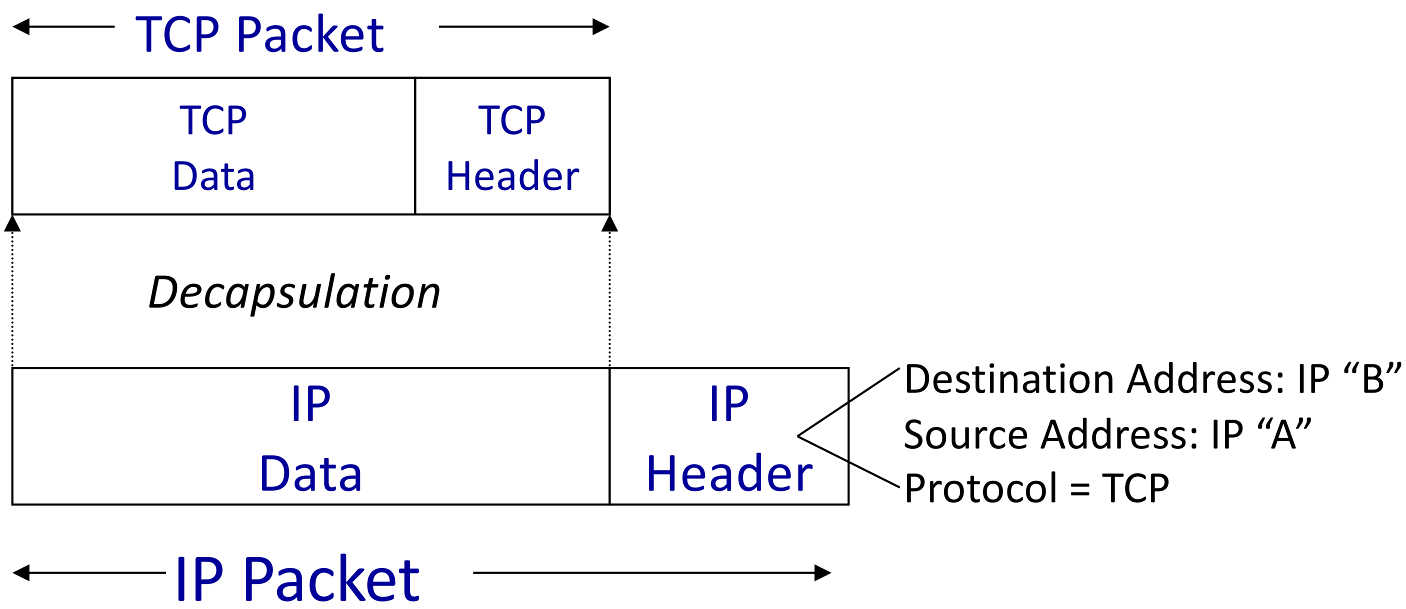


In the Receiving Host - Cont’d

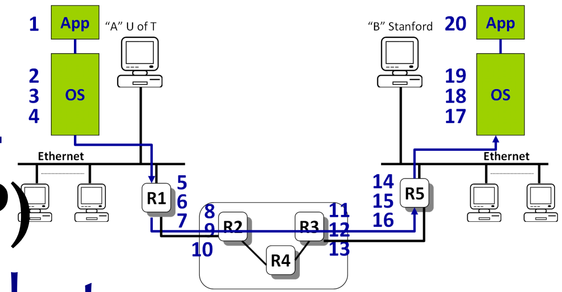


18. Internet Protocol (IP)

- Verify IP address.
- Extract/decapsulate TCP packet from IP packet.
- Pass TCP packet to TCP Protocol.



In the Receiving Host - Cont’d

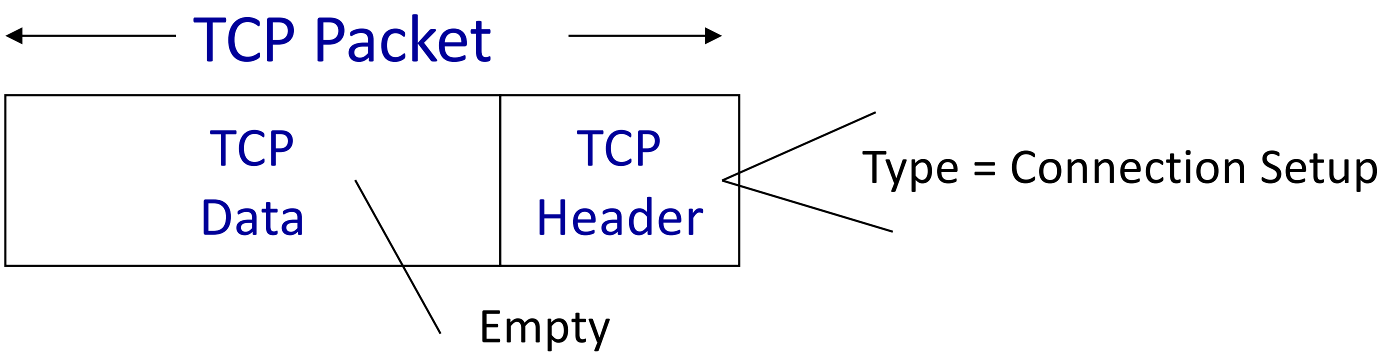


19. Transmission Control Protocol (TCP)

- Accepts TCP “Connection setup” packet
- Establishes connection by sending “Ack”.

20. Application-Programming Interface (API)

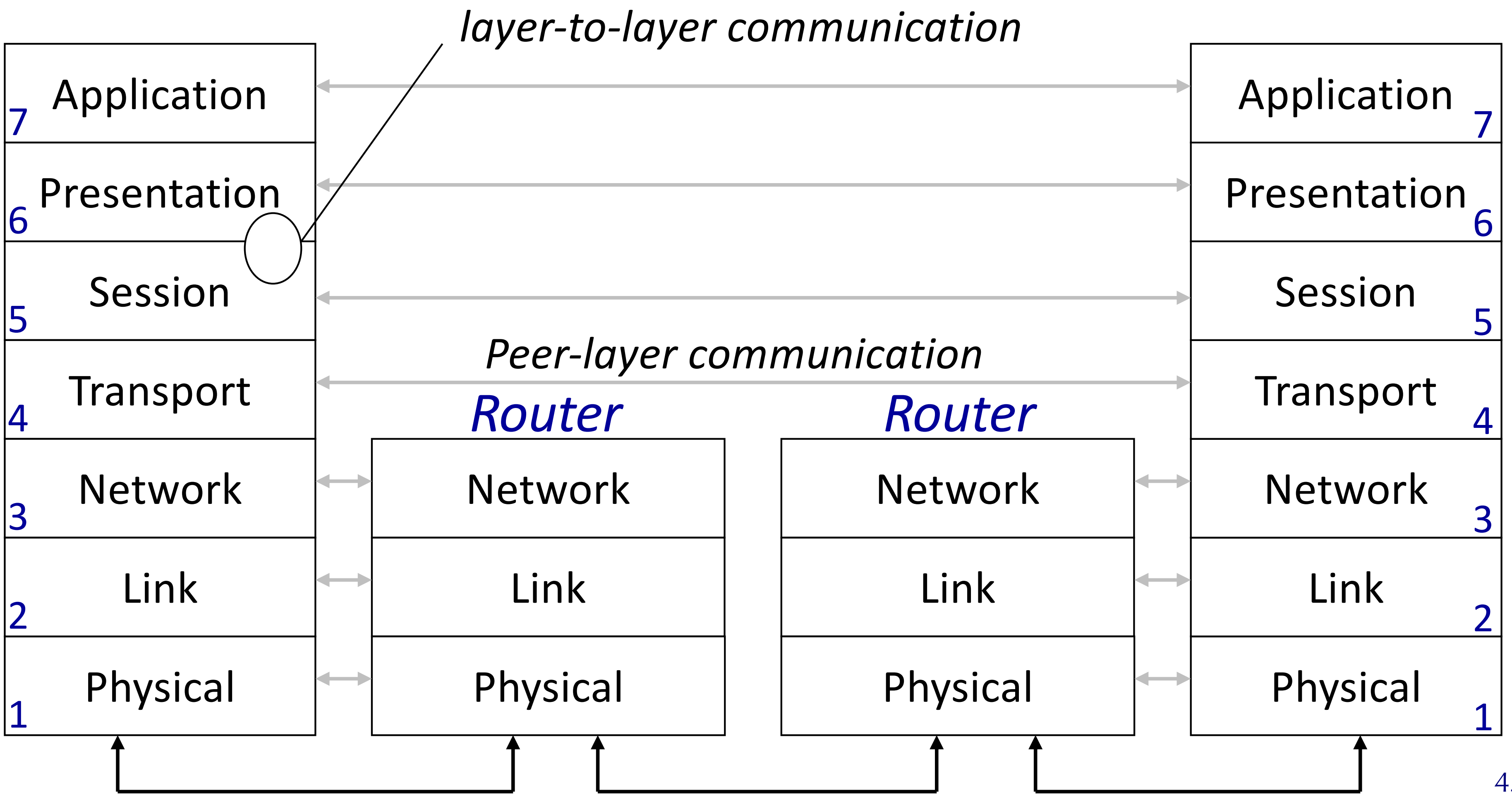
- Application receives request for TCP connection with “A”.



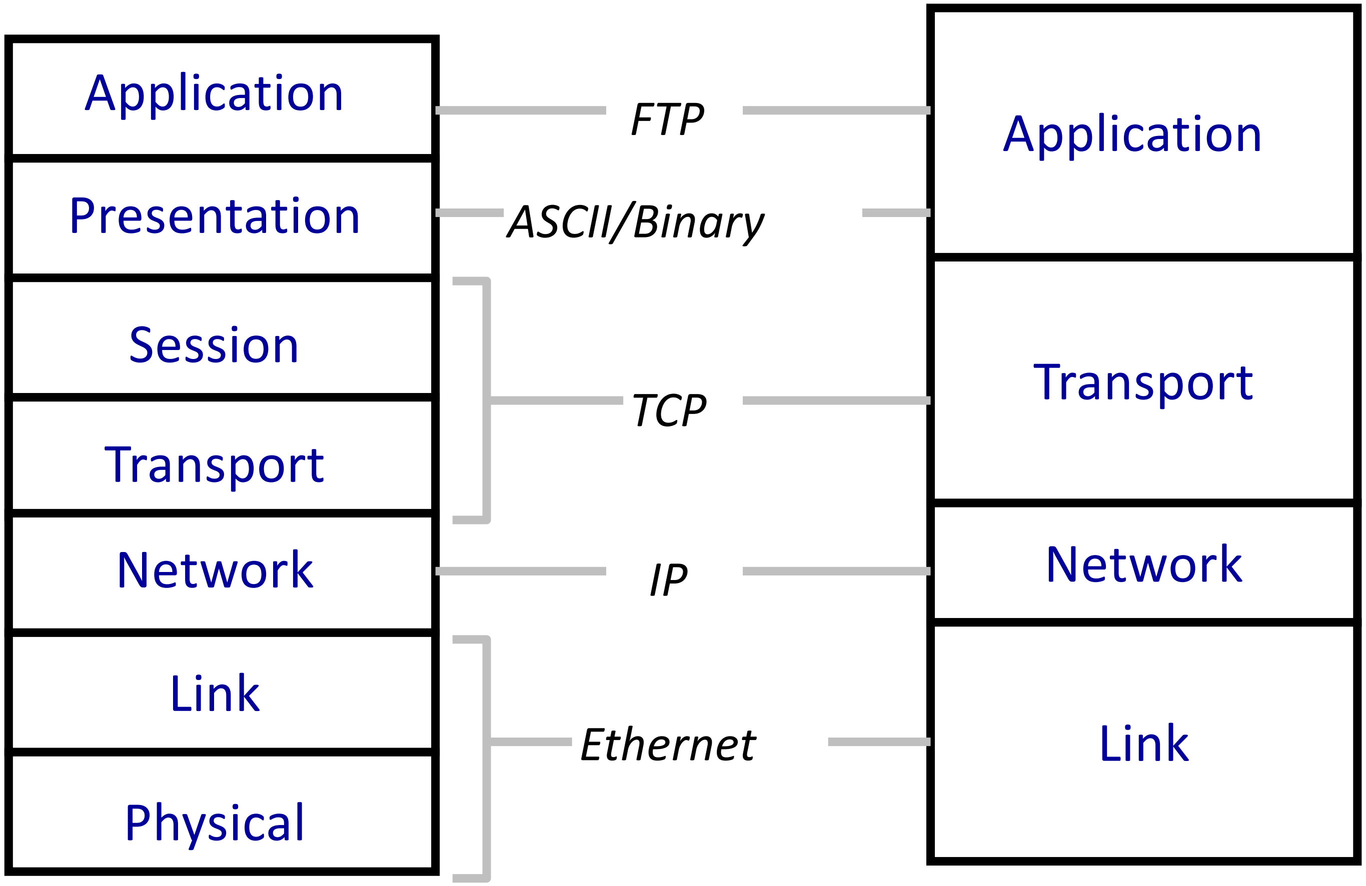
Outline – Foundations & Basic Concepts

- A detailed FTP example
- Layering
- Packet switching and circuit switching

Layering – *The OSI Model*



Layering – *Our File Transfer Example*



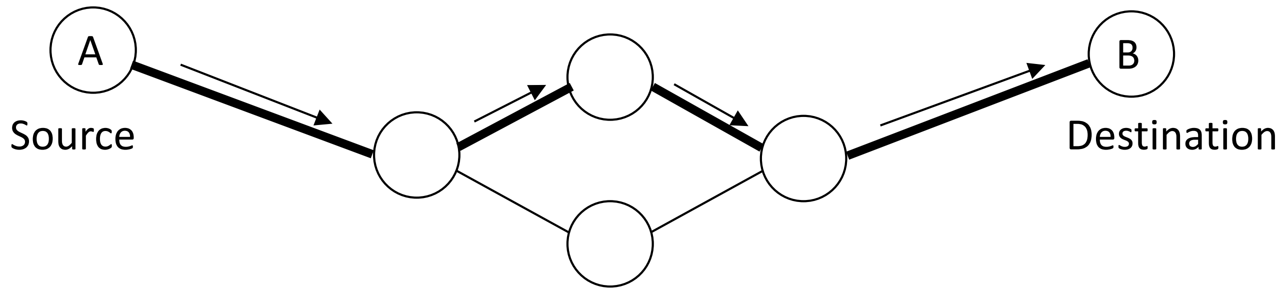
The 7-layer OSI Model

The 4-layer Internet model

Outline – Foundations & Basic Concepts

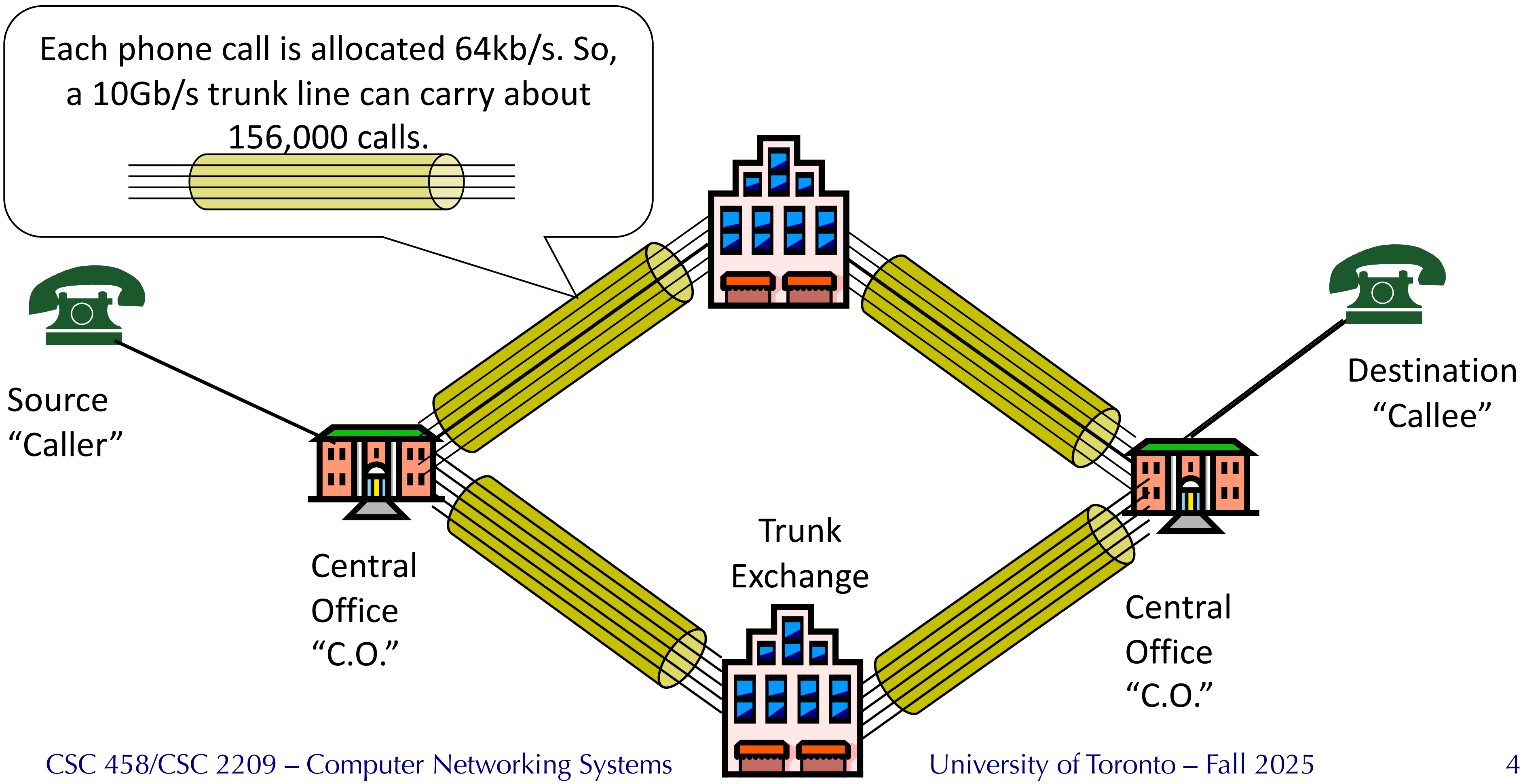
- A detailed FTP example
- Layering
- Packet switching and circuit switching

Circuit Switching

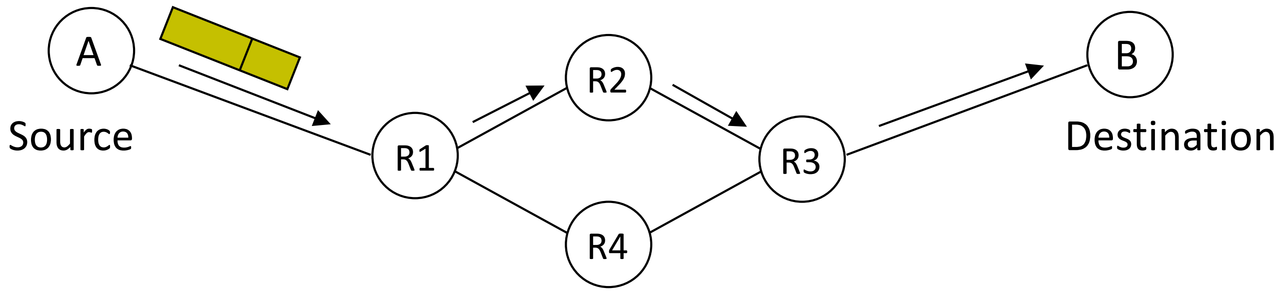


- It's the method used by the telephone network.
- A call has three phases:
  - Establish circuit from end-to-end ("dialing"),
  - Communicate,
  - Close circuit ("tear down").
- Originally, a circuit was an end-to-end physical wire.
- Nowadays, a circuit is like a virtual private wire: each call has its own private, guaranteed data rate from end-to-end.

Circuit Switching – *Telephone Network*

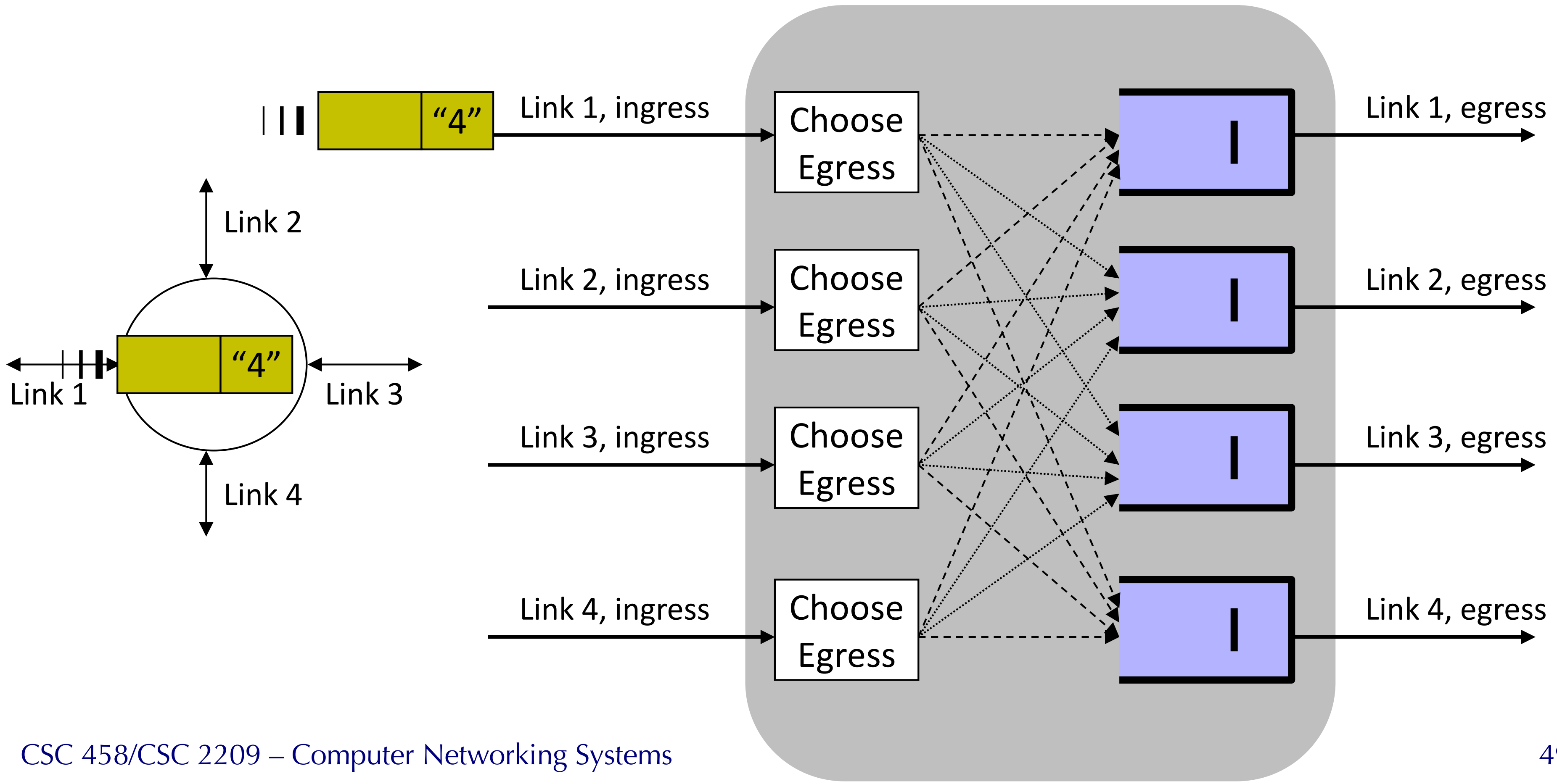


Packet Switching



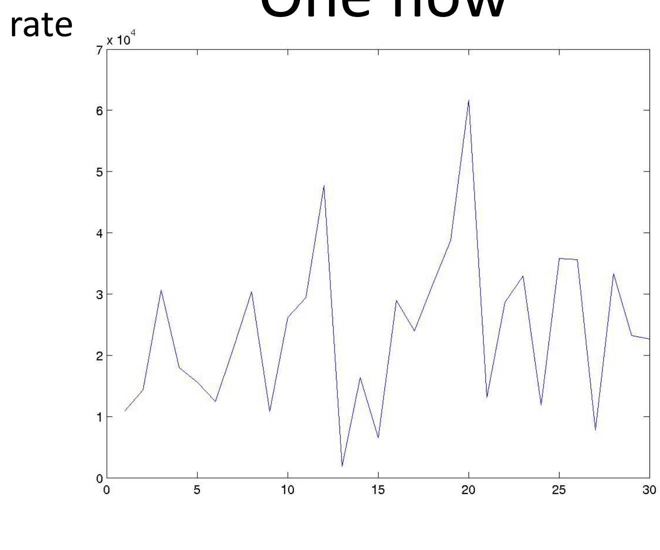
- It's the method used by the Internet.
- Each packet is individually routed packet-by-packet, using the router's local routing table.
- The routers maintain no per-flow state.
- Different packets may take different paths.
- Several packets may arrive for the same output link at the same time. Therefore, a packet switch has buffers.

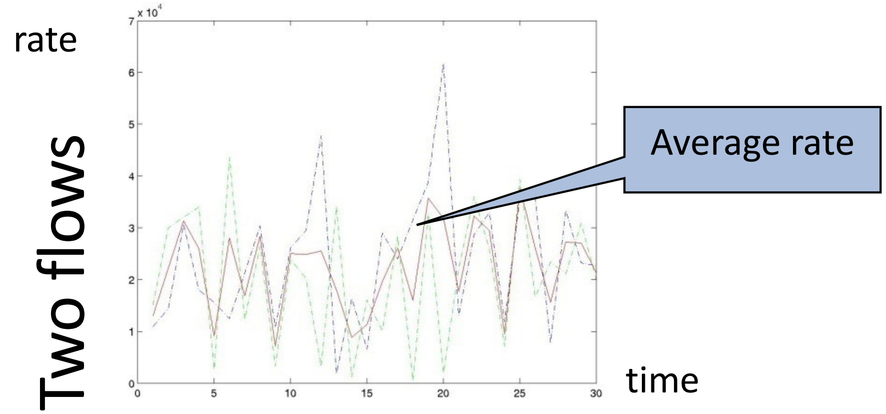
Packet Switching – *Simple Router Model*

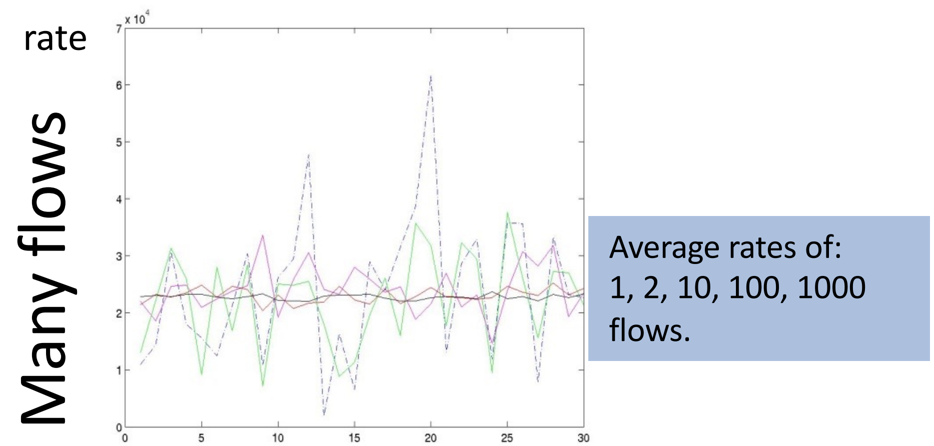


Statistical Multiplexing – *Basic Idea*

One flow

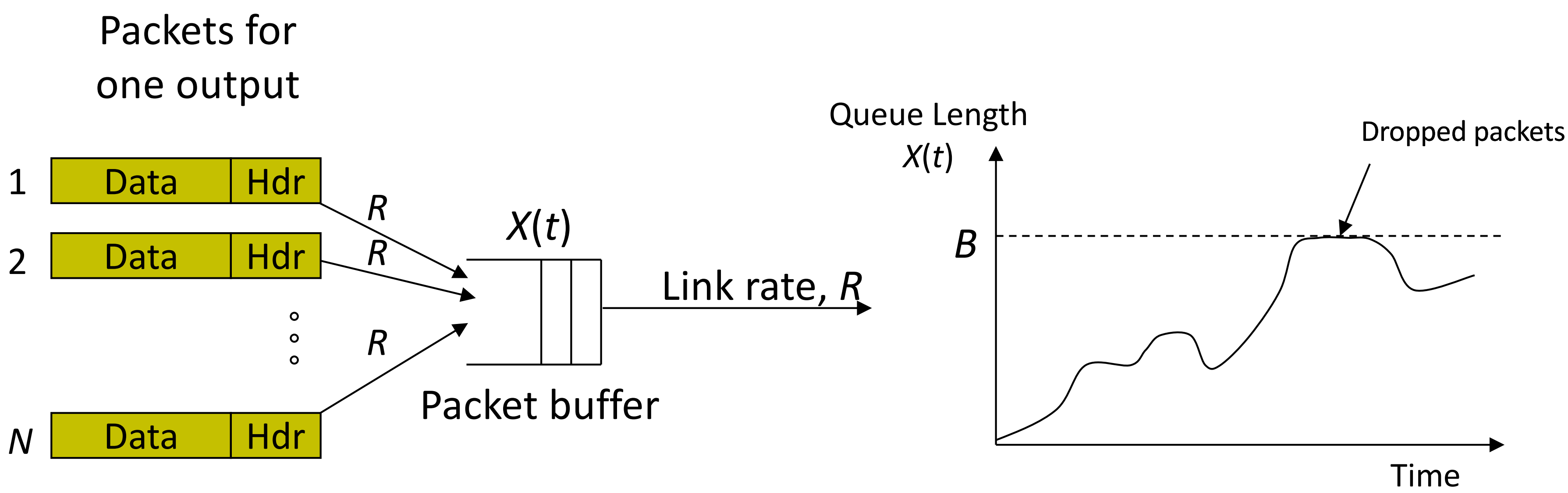






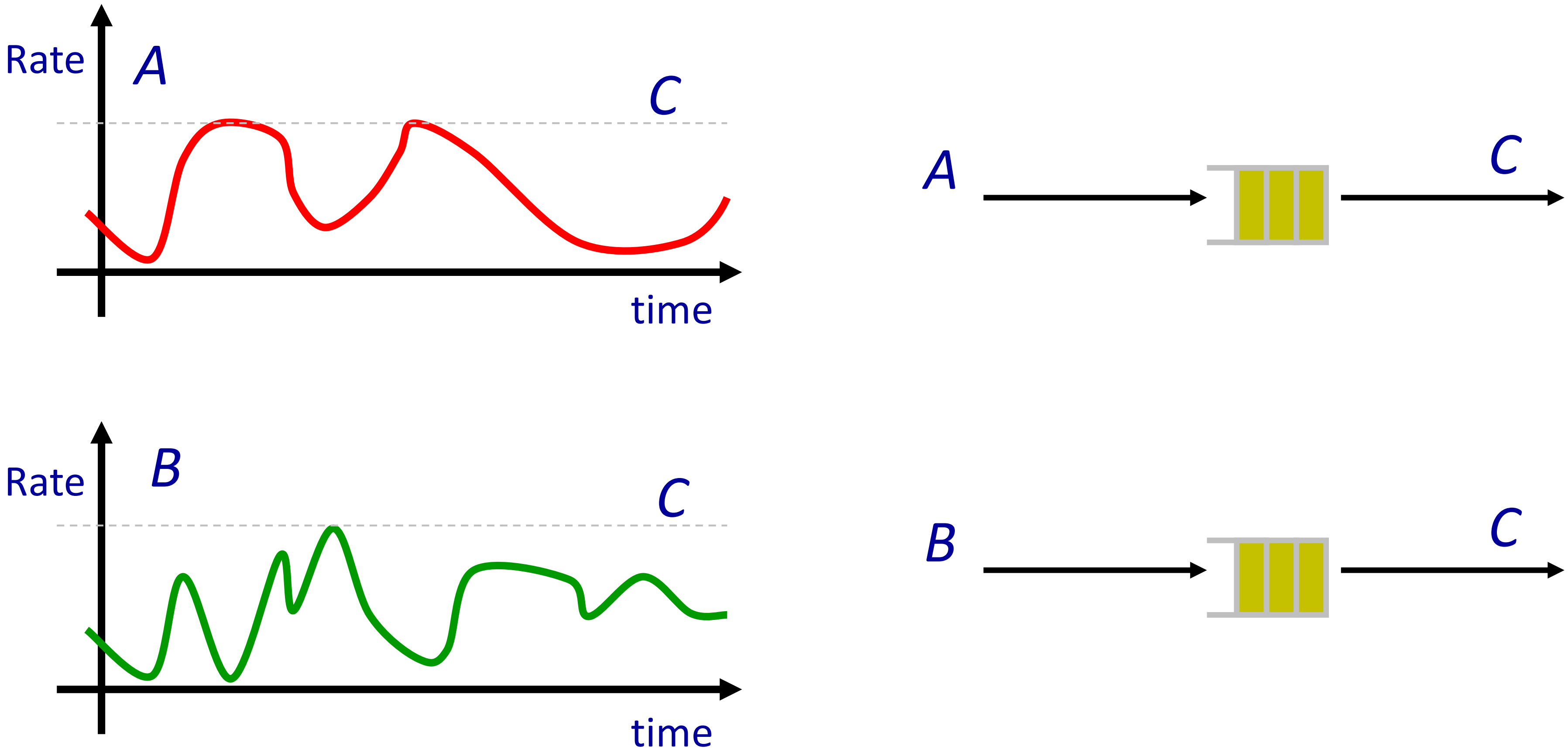
- Network traffic is bursty.
  i.e. the rate changes frequently.
- Peaks from independent flows generally occur at different times.
- Conclusion: The more flows we have, the smoother the traffic.

Packet Switching – *Statistical Multiplexing*

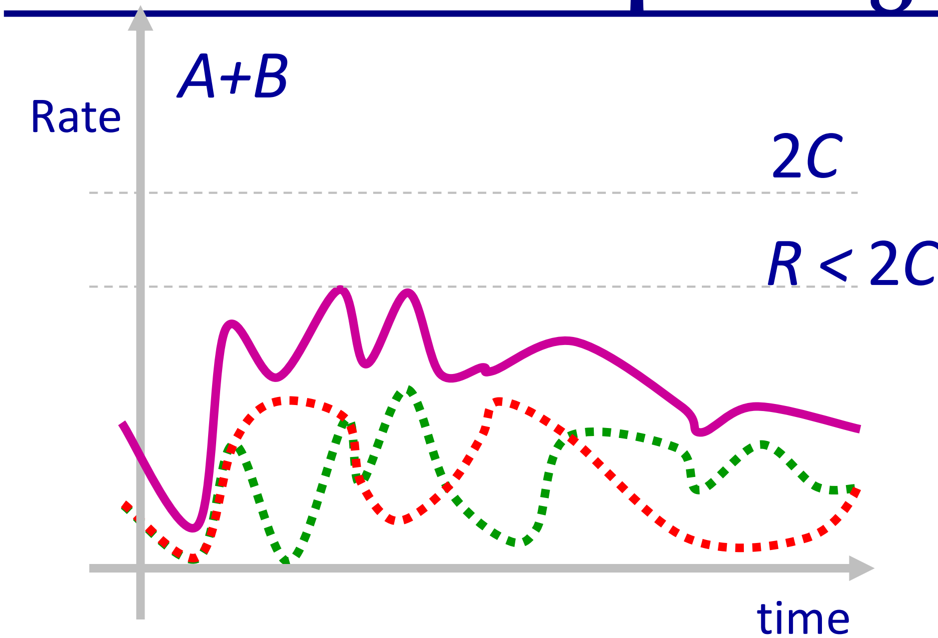


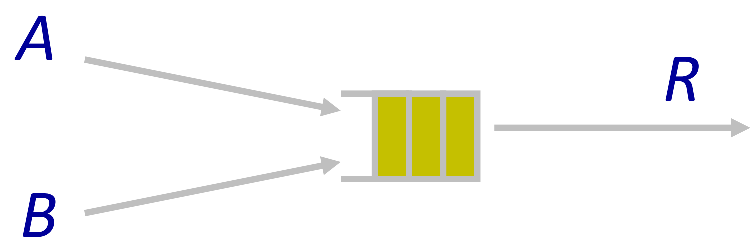
- Because the buffer absorbs temporary bursts, the egress link need not operate at rate $N.R$.
- But the buffer has finite size, B, so losses will occur.

Statistical Multiplexing



Statistical Multiplexing Gain





Statistical multiplexing gain = $2C/R$

Other definitions of SMG: The ratio of rates that give rise to a particular queue occupancy, or particular loss probability.

Why Packet Switching in the Internet?

- Efficient use of expensive links:
  - The links are assumed to be expensive and scarce.
  - Packet switching allows many, bursty flows to share the same link efficiently.
  - "Circuit switching is rarely used for data networks, ... because of very inefficient use of the links" - Gallager
- Resilience to failure of links & routers:
  - "For high reliability, ... [the Internet] was to be a datagram subnet, so if some lines and [routers] were destroyed, messages could be ... rerouted" - Tanenbaum

Final Comments, Discussion

- Is layering the best approach?
  - Simplifies design
  - Yet, limited and inflexible
- Best effort service
  - Made the rapid growth of the Internet possible
  - Makes providing any guarantees very difficult
- Packet switching
  - Enables statistical multiplexing
  - We need extremely fast routers
- Routing
  - How does a router know which output port to send the packet to?