CIS 457 - Data Communications

Nathan Bowman

Images taken from Kurose and Ross book

Protocol Layers

The Internet is far too complicated to try to understand all at once

Billions of nodes, various applications and protocols in use, many types of links, switches, etc.

Need to be principled in how we approach studying it

Networks are typically studied in a layered architecture

To see an example of what this means, assume there is a program on your machine that can reliably send ASCII messages to an application of your choosing on another computer

Given that, you could decide what useful application you want to build (such as email)

You would still need to agree on an email protocol with the other computer (does an email need to start with FROM, what are valid characters to use, etc.)

You are considering the network at the level of the email protocol

But what if that program to reliably send messages did not exist?

Instead, you had a program that could send messages to another computer, but unreliably

You could build your reliable sender on top of the unreliable one, but you would need to decide on a protocol to detect errors in messages and request that they be sent again

At this point, you do not care whether your reliable sender is going to be used for email, serving web pages, You could keep going on in this way and assume that you did not have an unreliable sender, but instead had something else that offered even less

Each time you do this, you are focusing on a particular *layer*

In networking, we do not layer programs, but rather protocols

Layers

- are hierarchical. Each layer is strictly above or below the others
- provide services to higher layers, such as reliable communication
- expect services from lower layers, such as unreliable communication

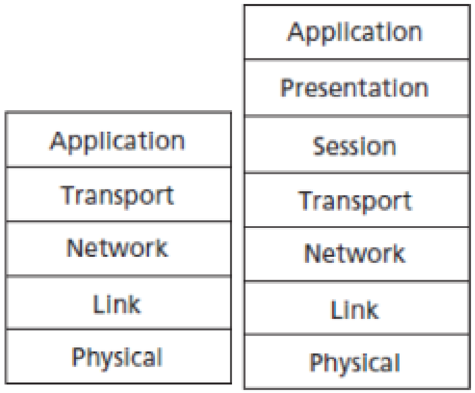
Not only do layers make networks easier to study, but they are good for modularity

Since no layer knows how another works, layers can be reimplimented as long as they provide the same services to the next layer and use the same services from the previous layer

Protocol layers can be implemented in either hardware or software

Higher-layer protocols tend to be implemented in software whereas lower-layer protocols are implemented in hardware

The set of all protocols used is the **protocol stack**



a. Five-layer Internet protocol stack

b. Seven-layer ISO OSI reference model

Application layer

Network applications and related protocols

Includes HTTP (web traffic), SMTP (email), FTP, and others

Creating protocols at this layer is relatively straightforward

Information exchanged at the application layer will be referred to as a **message**

Transport layer

Exists to transport application-layer messages between applications

(Notice: not just between two machines, but between two specific programs)

Two protocols used in Internet

TCP provides guaranteed delivery and congestion control

UDP is a simpler protocol that provides neither

Transport-layer packet is called a **segment**

Network layer

Moves packets from one host to another

Receives segment from transport layer, sends to receiving host where segment is sent back up to transport layer for delivery to application

Only one protocol at this layer: IP protocol

Every Internet component connected to network layer must run IP protocol

Packets at network layer are **datagrams**

Link layer

Network layer focuses on routing datagram between sending and receiving hosts

Link layer handles movement of packet from one node to next

Network layer would not be able to move datagram even one "hop" without link layer

Many link-layer protocols exist, providing different services, sometimes including reliable delivery *over a single link*

Link layer

Packet may go through many different link-layer protocols on its way from sender to receiver

Each link could have a different protocol

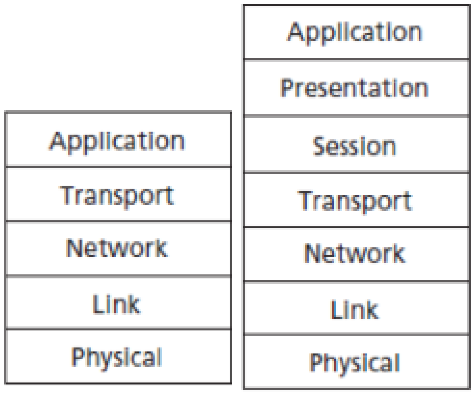
Ethernet, WiFi, and DOCSIS are just a few link layer protocols

Link-layer packets are **frames**

Physical layer

Moves individual bits from one node to the next

Which protocol is used depends on physical medium, e.g., coaxial cable



a. Five-layer Internet protocol stack

b. Seven-layer ISO OSI reference model

OSI Model

Other protocol stacks besides Internet protocol stack exist

OSI model often taught as example, but otherwise not of wide interest

Has two additional layers

OSI Model

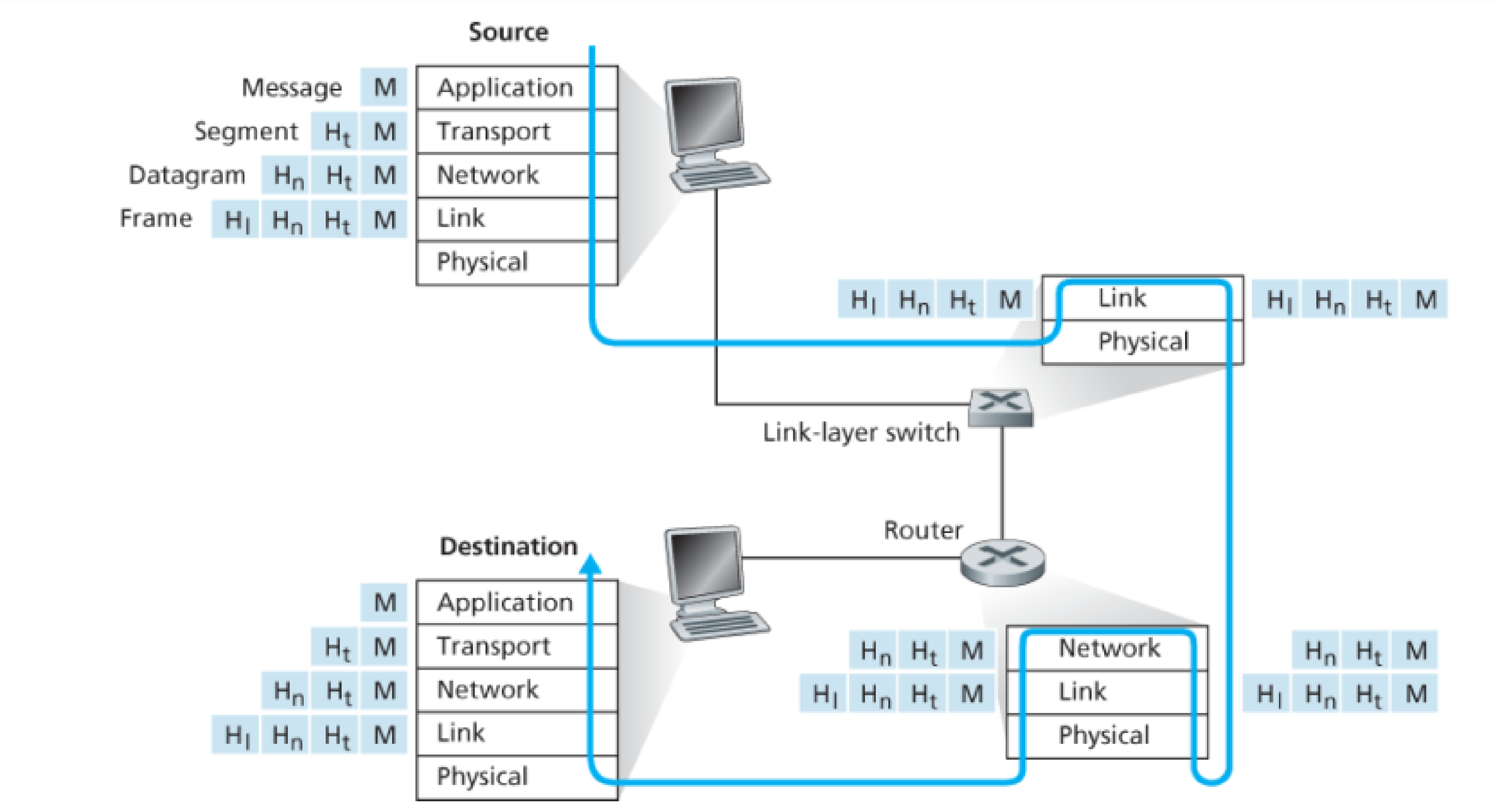
Presentation layer provides services to allow applications to interpret meaning of data, such as compression and encryption

Session layer provides support for checkpointing and recovery of data exchange

In TCP/IP stack, these services are not available and would need to be implemented by the application layer if desired

Encapsulation

As a message goes up and down the protocol stack, header information is added and removed by the various layers


Source
Message
M
Application
Segment
H_t M
Transport
Datagram
H_n H_t M
Network
Frame
H_l H_n H_t M
Link
Physical
H_l H_n H_t M
Link
Physical
H_l H_n H_t M
Link-layer switch
Router
Destination
M
Application
H_t M
Transport
H_n H_t M
Network
H_l H_n H_t M
Link
Physical
H_n H_t M
H_l H_n H_t M
Network
Link
Physical
H_n H_t M
H_l H_n H_t M

This header information is necessary to allow lower layers to do their job, such as providing error detection

However, since the application layer does not know about the layers below it or how they are implemented, extra headers must be removed before message is returned to application layer

This is true of all layers: header information is added when going down, and removed before the packet is sent back up

Header information from one layer can never be removed by another because the various layers do not know what each other's headers contain

Real-life example of encapsulation (stolen from your book) would be an interoffice memo

Alice wants to send message to Bob, who works at different branch of same company

Alice writes the message, which is the application-layer message

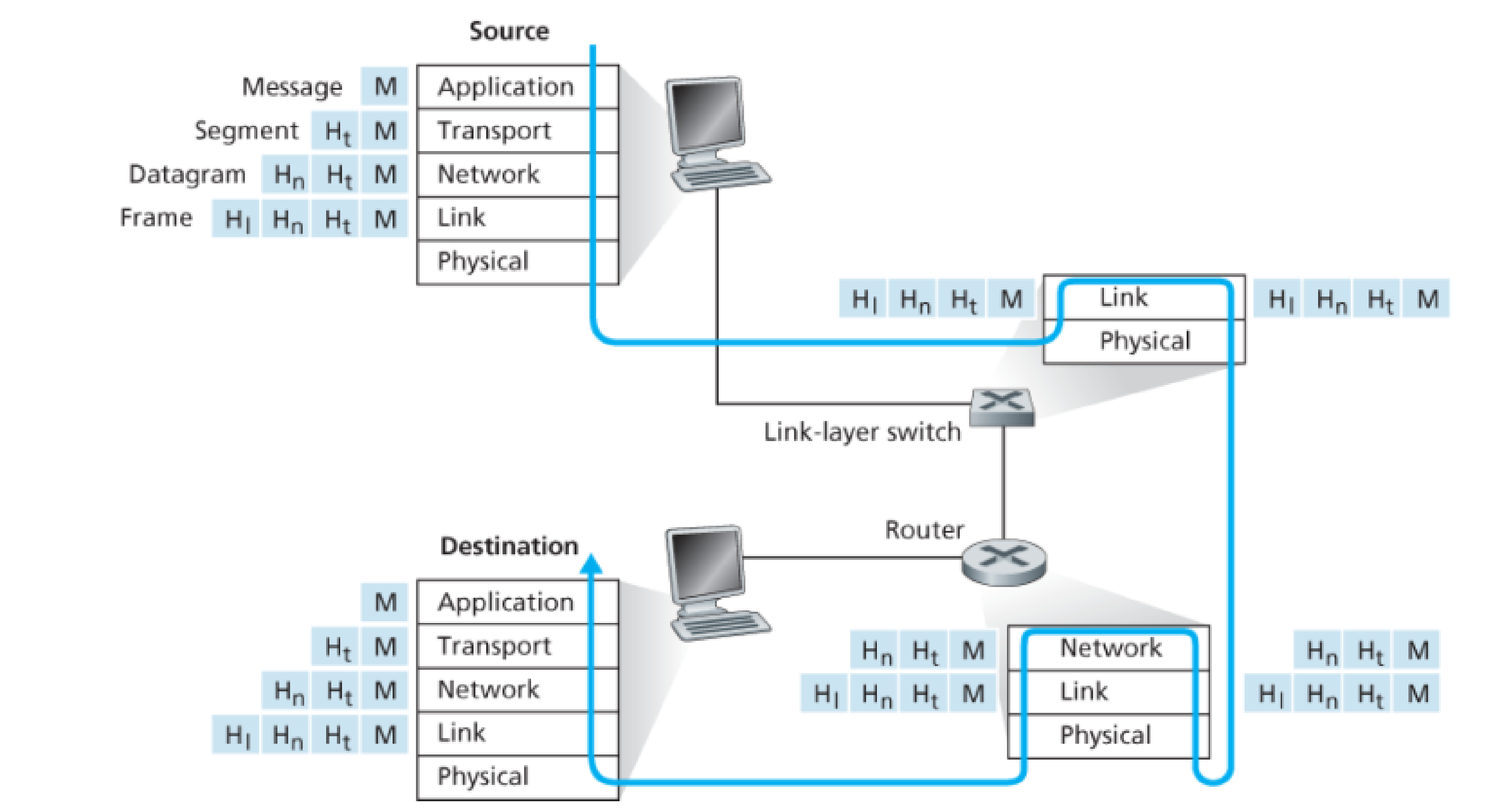
She puts it an interoffice envelope and adds Bob's employee ID (header information). Envelope is at transport layer and contains transport-layer header and original message

Envelope is sent to mail room where it is sealed in a USPS envelope with Bob's branch office's address

We are now at network layer, with two layers' worth of headers and the original message

When Bob's office gets the letter, they remove the outer envelope (network-layer header) and pass the inner envelope to intraoffice mail (transport layer)

Finally, header is removed and envelope delivered safely to Bob, who only needs to see the actual memo


Source
Message
M
Application
Segment
H_t M
Transport
Datagram
H_n H_t M
Network
Frame
H_l H_n H_t M
Link
Physical
H_l H_n H_t M
Link
Physical
H_l H_n H_t M
Link-layer switch
Router
Destination
M
Application
H_t M
Transport
H_n H_t M
Network
H_l H_n H_t M
Link
Physical
H_n H_t M
H_l H_n H_t M
Network
Link
Physical
H_n H_t M
H_l H_n H_t M