AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

15CSE312
COMPUTER NETWORKS
3-0-0 3

Chapter 5: Data Link Layer

- Introduction
- Services
- Error Detection and Correction
  - ✓ Parity Check
  - ✓ Cyclic Redundancy check
  - ✓ Internet checksum

Data Link Layer- Introduction

This layer is the protocol layer that **transfers data between nodes** on a network segment across the physical layer. The data link layer provides the functional and procedural means to transfer data between network entities and may also provide the means to detect and possibly correct errors that can occur in the physical layer.

Any device that runs a link-layer protocol as a node. **Nodes** include hosts, routers, switches, and WiFi access points We will also refer to the communication channels that connect adjacent nodes along the communication path as **links**. In order for a datagram to be transferred from source host to destination host, it must be moved over each of the individual links in the end-to-end path.

The basic service of any link layer is to move a datagram from one node to an adjacent node over a single communication link

Analogy- Link Layer

Consider a travel agent who is planning a trip for a tourist traveling from **Princeton, New Jersey, to Lausanne, Switzerland**. The travel agent decides that it is most convenient for the tourist to take a **limousine from Princeton to JFK airport,** then a **plane from JFK airport to Geneva's airport**, and finally a **train from Geneva's airport to Lausanne's train station**. Once the travel agent makes the three reservations, it is the responsibility of the **Princeton limousine company** to get the tourist from Princeton to JFK; it is the responsibility of the **airline company** to To get the tourist from JFK to Geneva; and it is the responsibility of the **Swiss train service** to get the tourist from Geneva to Lausanne

. Each of the three segments of the trip is "direct" between two "adjacent" locations. Note that the three transportation segments are managed by different companies and use entirely different transportation modes (limousine, plane, and train). Although the transportation modes are different, they each provide the basic service of moving passengers from one location to an adjacent location. **In this transportation analogy, the tourist is a datagram, each transportation segment is a link, the transportation mode is a link-layer protocol, and the travel agent is a routing protocol.**

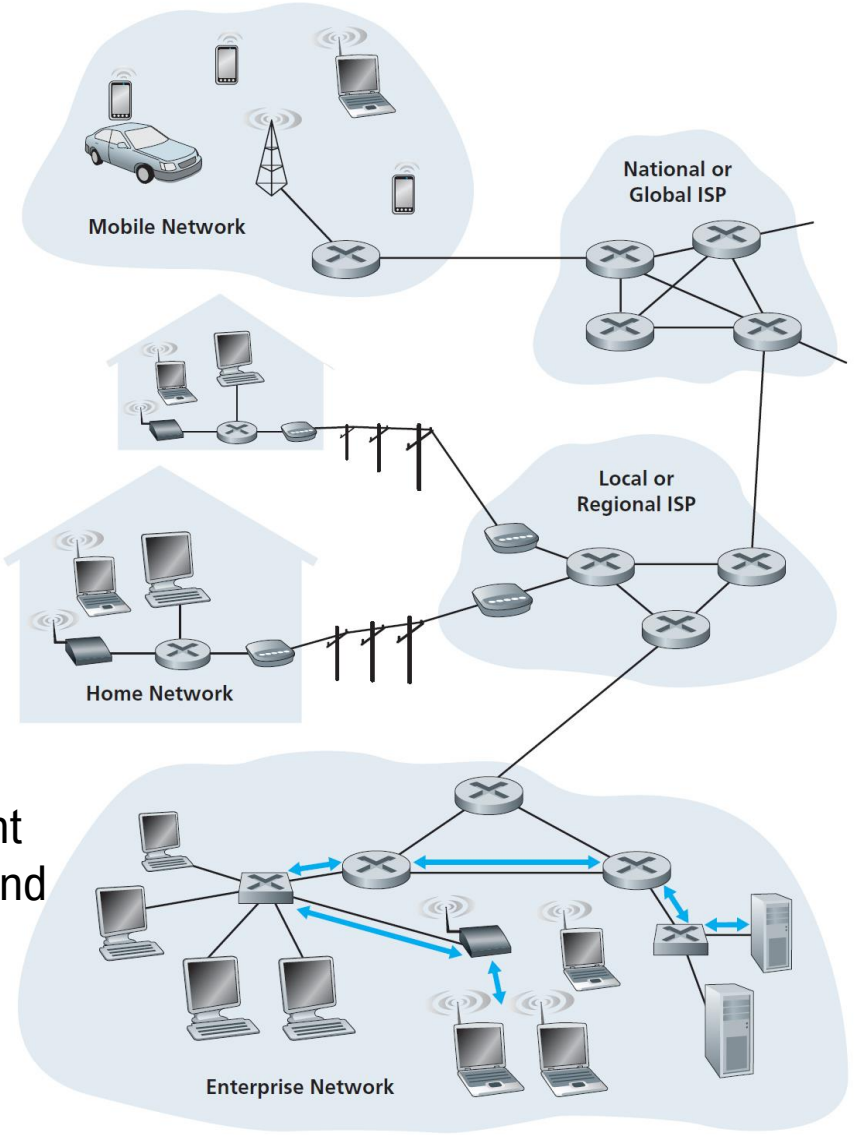


Figure 5.1 ♦ Six link-layer hops between wireless host and server

Chapter 5: Link layer

our goals:

- understand principles behind link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - local area networks: Ethernet, VLANs
- instantiation, implementation of various link layer technologies

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

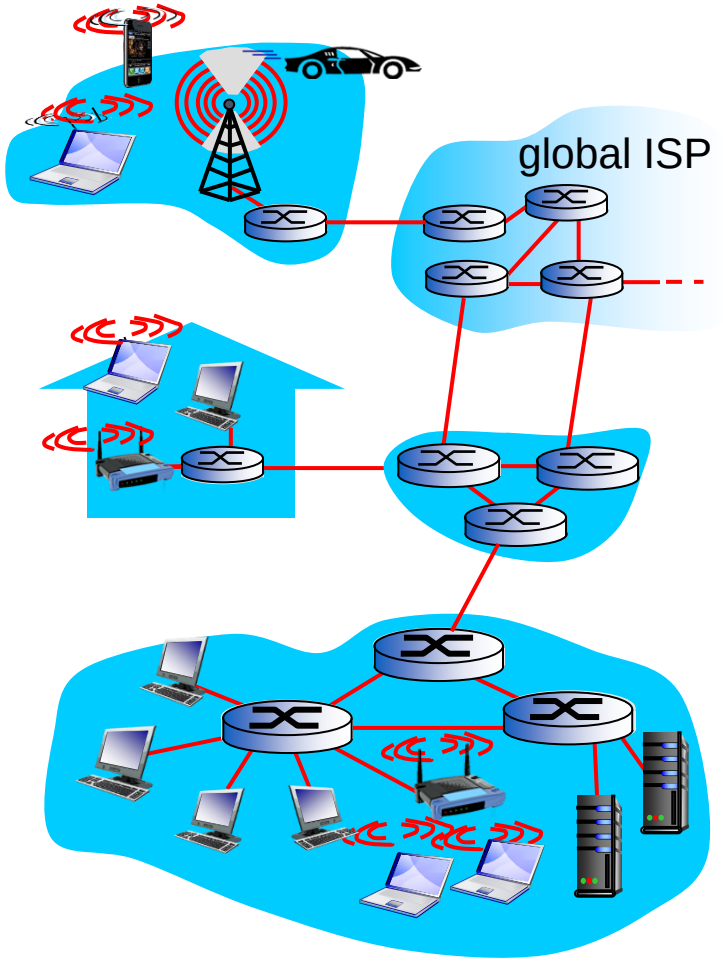
- addressing, ARP
- Ethernet
- switches
- VLANS

Link layer: introduction

terminology:

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired links
  - wireless links
  - LANs
- layer-2 packet: frame, encapsulates datagram

data-link layer has responsibility of transferring datagram from one node to *physically adjacent* node over a link



Link layer: context

- datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Each link protocol provides different services
  - e.g., may or may not provide rdt over link

transportation analogy:

- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

Link layer services

- *framing, link access:*
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, dest
    - different from IP address!
- *reliable delivery between adjacent nodes*
  - we learned how to do this already (chapter 3)!
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates
    - *Q:* why both link-level and end-end reliability?

A: correcting errors locally, rather than end-end

Link layer services (more)

- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error detection*:
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- *error correction:*
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- *half-duplex and full-duplex*
  - with half duplex, nodes at both ends of link can transmit, but not at same time

Where is link layer implemented?

The link layer is implemented in a network adapter, also sometimes known as a **network interface card (NIC).** At the heart of the network adapter is the link-layer controller, usually a single, special-purpose chip that implements many of the link-layer services (framing, link access, error detection, and so on). Thus, much of a **link-layer controller's functionality is implemented in hardware.**

On the **sending side**, the controller takes a **datagram** that has been created and stored in host memory by the higher layers of the protocol stack, **encapsulates** the datagram in a link-layer frame (filling in the frame's various fields), and then **transmits** the frame into the communication link, following the link-access protocol.

On the **receiving side**, a controller **receives** the entire frame, and **extracts the network-layer datagram**. If the link layer performs **error detection**, then it is the sending controller that **sets the error-detection** bits in the frame header and it is the receiving controller that performs error detection

Where is the link layer implemented?

- in each and every host
- link layer implemented in "adaptor" (aka *network interface card* NIC) or on a chip
  - Ethernet card, 802.11 card; Ethernet chipset
    - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware

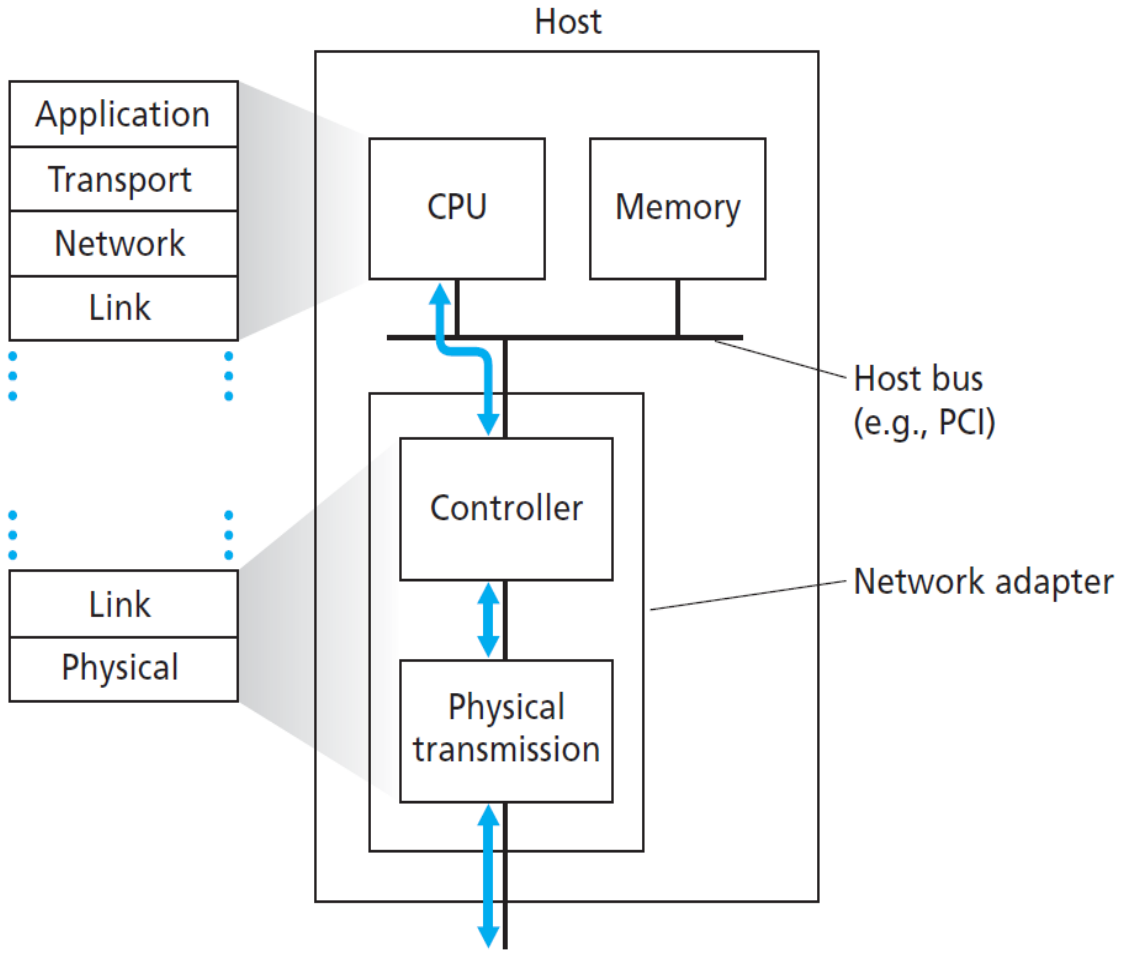
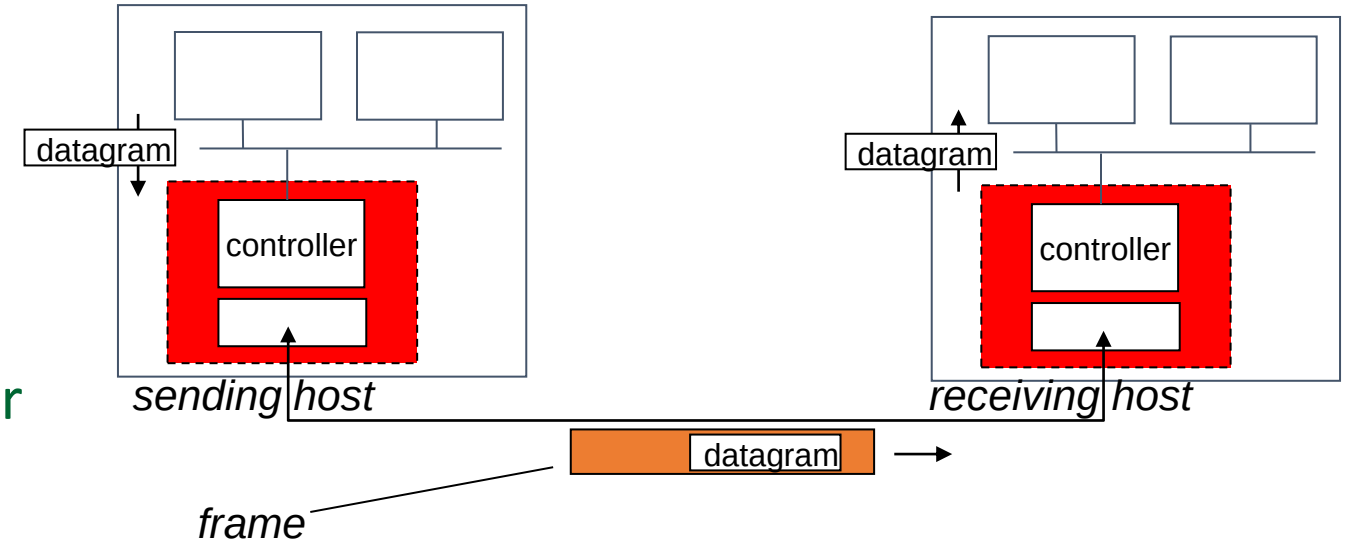


Figure 5.2 ♦ Network adapter: its relationship to other host components and to protocol stack functionality

Adaptors communicating

- sending side:
  - encapsulates datagram in frame
  - adds error checking bits, rdt, flow control, etc.
- receiving side
  - looks for errors, rdt, flow control, etc
  - extracts datagram, passes to upper layer at receiving side



The **software components** of the link layer implement higher-level link layer functionality such as **assembling link-layer addressing information** and **activating the controller hardware**. On the receiving side, link-layer software **responds to controller interrupts** (e.g., due to the receipt of one or more frames), handling **error conditions and passing a datagram** up to the network layer

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

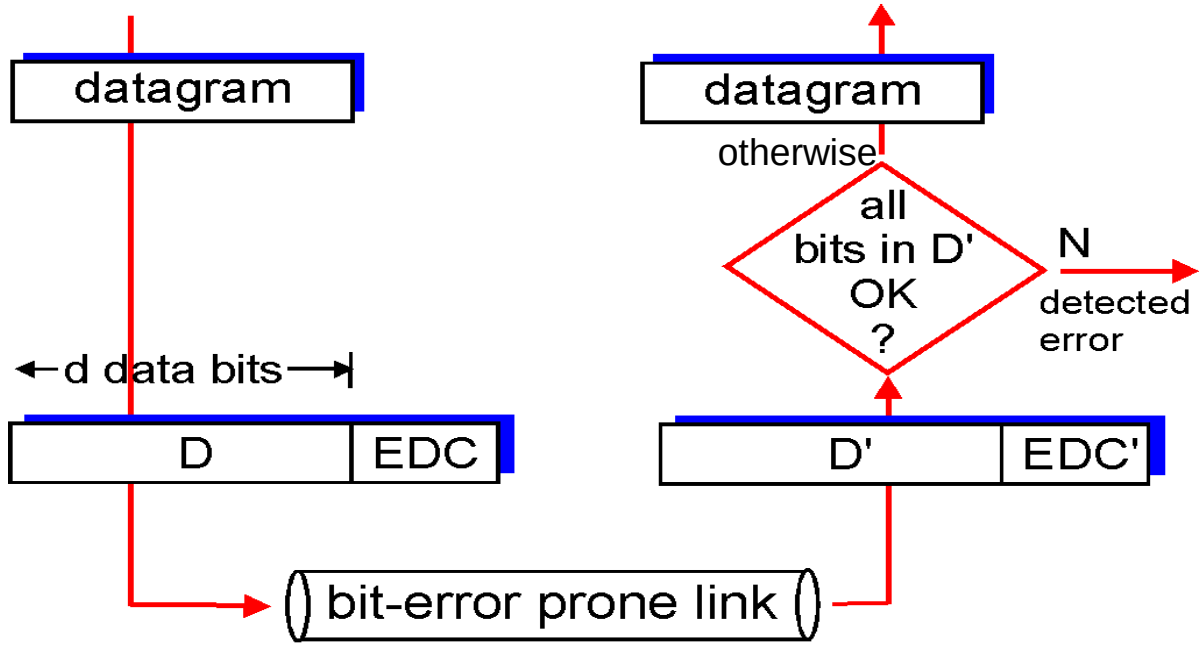
5.6 data center networking

5.7 a day in the life of a web request

Error detection

EDC= Error Detection and Correction bits (redundancy)
D = Data protected by error checking, may include header fields

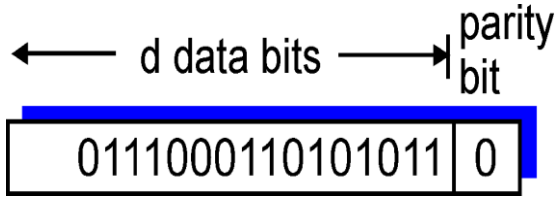
- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



Parity checking

single bit parity:

- *d*etect single bit errors

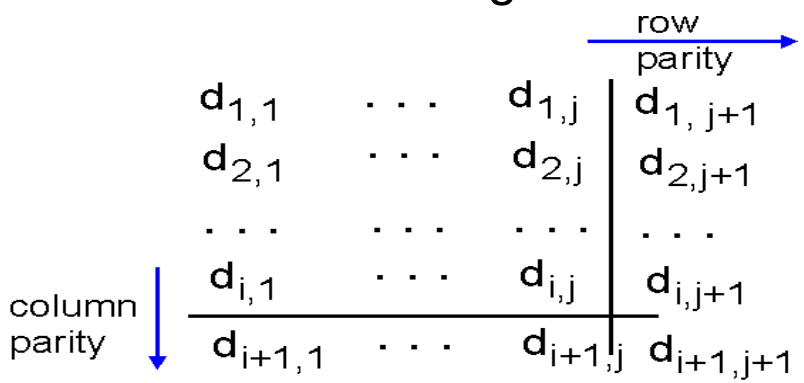


Errors rather than occurring independently, are often clustered together in "bursts."

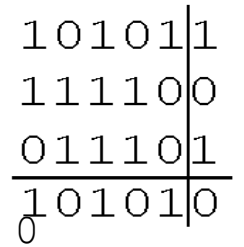
Hence a more robust method required→

two-dimensional bit parity:

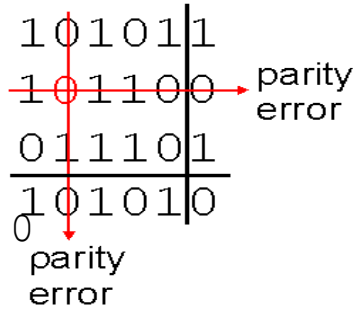
- detect and correct single bit errors



In an even parity scheme, the sender simply includes one additional bit and chooses its value such that the total number of 1s in the *d* + 1 bits (the original information plus a parity bit) is even. For odd parity schemes, the parity bit value is chosen such that there is an odd number of 1s



no errors



correctable single bit error

Receiver operation is also simple with a single parity bit. The receiver need only count the number of 1s in the received d + 1 bits. If an odd number of 1- valued bits are found with an even parity scheme, the receiver knows that at least one bit error has occurred. More precisely, it knows that some odd number of bit errors have occurred.

----→ **But what happens if an even number of bit errors occur?**

Internet checksum (review)

sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

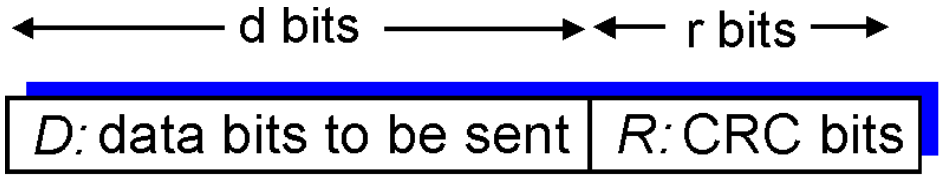
goal: detect "errors" (e.g., flipped bits) in transmitted packet (note: used at transport layer *only*)

receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?*

Cyclic redundancy check

- more powerful error-detection coding
- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi, ATM)



bit pattern

$D * 2^r$ XOR R

mathematical formula

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

5.6 data center networking

5.7 a day in the life of a web request

Namah Shivaya