

# Web



# Contents

Web Browsing

Protocols

IETF

# Web Browsing

Example

Locating Resource: DNS

Connection

End-to-end

Packets

Bits

Points to remember

## Web: Example

Click Link or URL

Get content from local or remote computer

URL: <http://www.google.com/string>

Specifies

Protocol: http

Computer: www.google.com

String

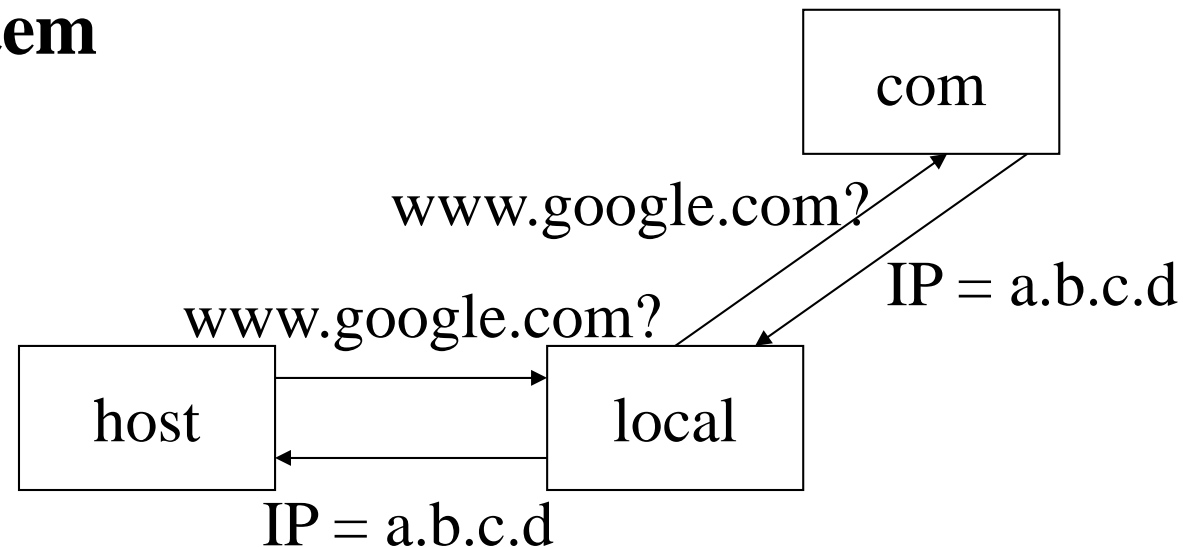
Computer (server) selects contents based on string

## Web: Location Resource

www.google.com is the **name** of a computer

Network uses **IP addresses**

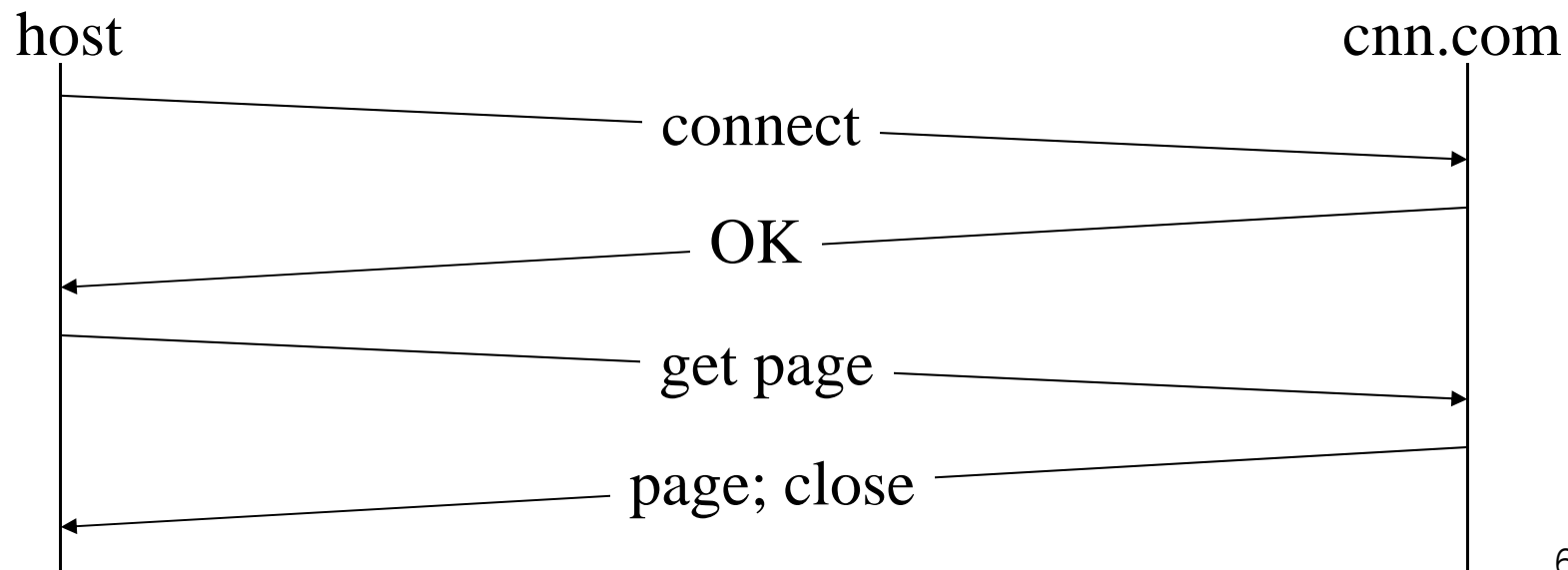
To find the IP address, the application uses a hierarchical directory service called the **Domain Name System**



## Web: Connection

The protocol (http) sets up a **connection** between the host and server (cnn.com) to transfer the page

The connection transfers the page as a byte stream, without errors: **pacing + error control**



## Web: End-to-end

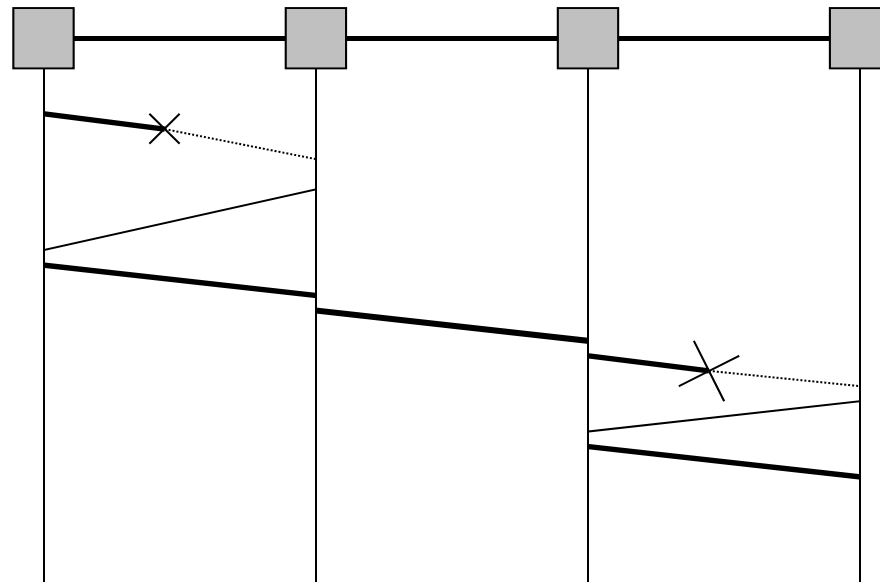
The byte stream flows from end to end across many links and switches: **routing (+ addressing)**

That stream is regulated and controlled by both ends: **retransmission** of erroneous or missing bytes: **flow control**

## Link ARQ vs. End-to-End ARQ

Question: Should every link perform ARQ?

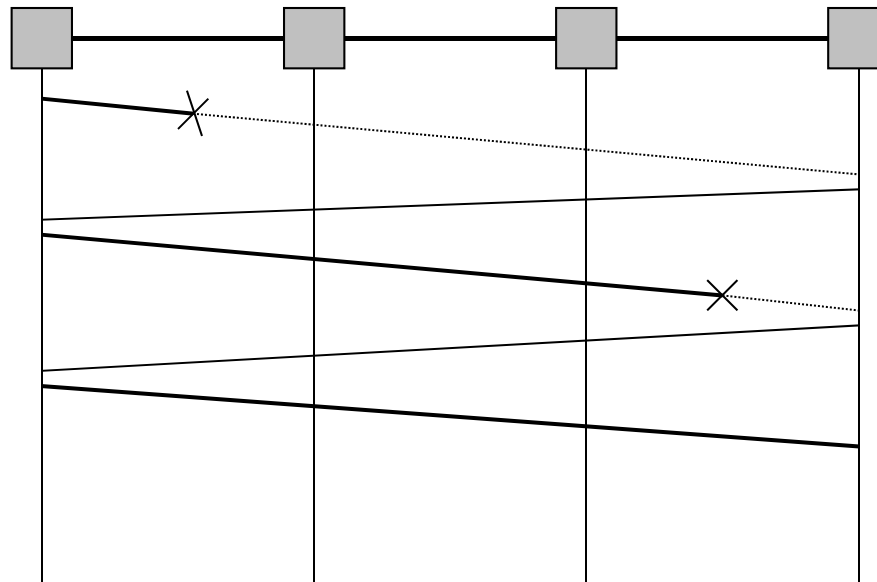
Link ARQ:





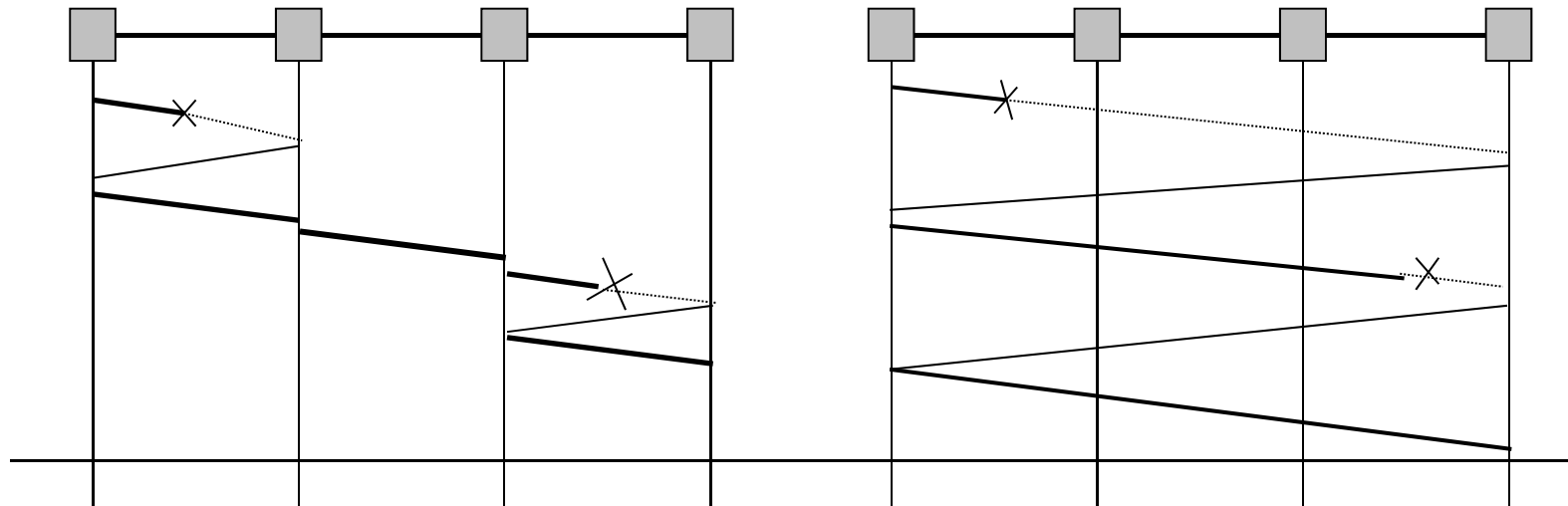
# Link ARQ vs. End-to-End ARQ

End-to-End ARQ:



## Link ARQ vs. End-to-End ARQ

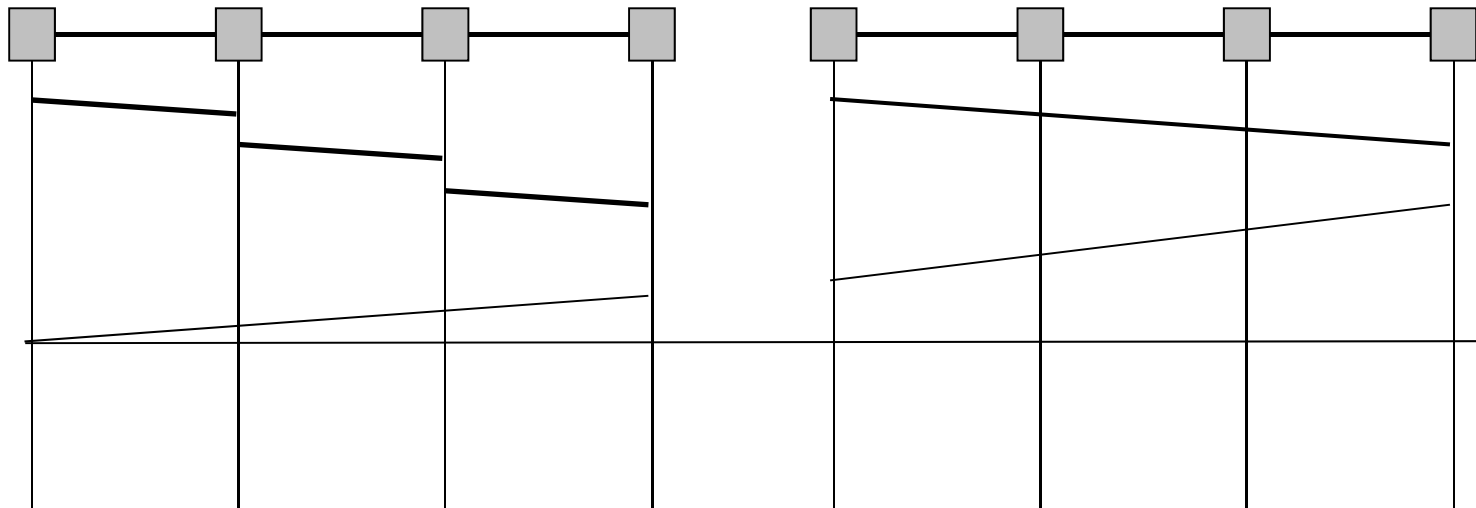
Two Extreme Cases. Case (1): Noisy Links



Link ARQ is faster than E2E ARQ when there are many errors

## Link ARQ vs. End-to-End ARQ

Case (2): Reliable Links



Link ARQ is slower than E2E ARQ when there are few errors

## Web: Packets

The networks transports bytes grouped into **packets**

The packets are “self-contained” and routers handle them **one by one**

The **end hosts** worry about errors and flow control:

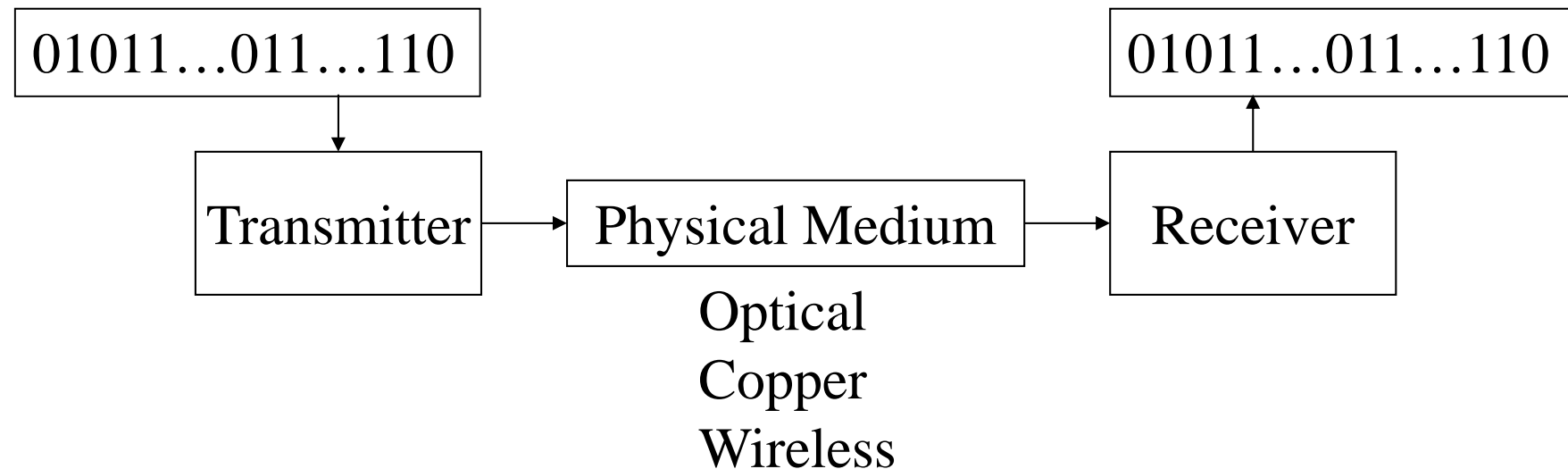
- Destination checks packet for error (using error detection code) and sends ACKs with sequence number

- Source retransmits packets that were not ACKed and adjusts rate of transmissions

## Web: Bits

Equipment in each node sends the packets as a string of bits

That equipment is not aware of the meaning of the bits



## Web: Points to remember

### Separation of tasks

- send bits on a link: transmitter/receiver [clock, modulation, ...]
- send packet on each hop [framing, error detection, ...]
- send packet end to end [addressing, routing]
- pace transmissions [detect congestion]
- retransmit erroneous or missing packets [acks, timeout]
- find destination address from name [DNS]

### Scalability

- routers don't know about connections
- names and addresses are hierarchical

# Protocols

Agreement dictating the form and function of data exchanged between two (or more) parties to effect a communication

Two parts: *syntax* and *semantics*

syntax: where bits go

semantics: what they mean and what to do with them

## Protocol Example

### Internet Protocol (IP)

if you can generate and understand IP, you can be on the  
Internet

media, OS, data rate independent

### TCP and HTTP

if you can do these, you are on the web



## Protocol Standards

New functions require new protocols

Thus there are many (e.g. IP, TCP, UDP, HTTP, RIP, SMTP, SNMP, Telnet, FTP, DNS, NNTP, NFS, NTP, BGP, ICMP, IGMP, PIM, DVMRP, IS-IS, OSPF, ARP)

Specifications do not change frequently

Organizations: IETF, IEEE, ITU

## The IETF

Specifies Internet-related protocols

Produces “RFCs” ([www.rfc-editor.org](http://www.rfc-editor.org))

Quotation from IETF T-shirt:

*We reject kings, presidents and voting.*

*We believe in rough consensus and running code.*

--- David Clark

## Summary

Network uses **IP addresses** for **routing**

Byte stream flows between end points with  
retransmission and **flow/congestion control**

**Flow control** prevents a sender from overrunning the  
capacity of a receiver: end-to-end issue

**Congestion control** prevents too much data from  
being injected into the network: hosts and networks

IP, TCP/UDP, HTTP