NWEN 243

Networked Applications

Transport layer and application layer

Topic

- TCP flow control
- TCP congestion control
- The Application Layer

Fast Retransmit

- Time-out period often relatively long:
  - long delay before resending lost packet
- Detect lost segments via duplicate ACKs.
  - Sender often sends many segments back-to-back
  - If segment is lost, there will likely be many duplicate ACKs.

- If sender receives 3 ACKs for the same data, it supposes that segment after ACKed data was lost:
  - fast retransmit: resend segment before timer expires

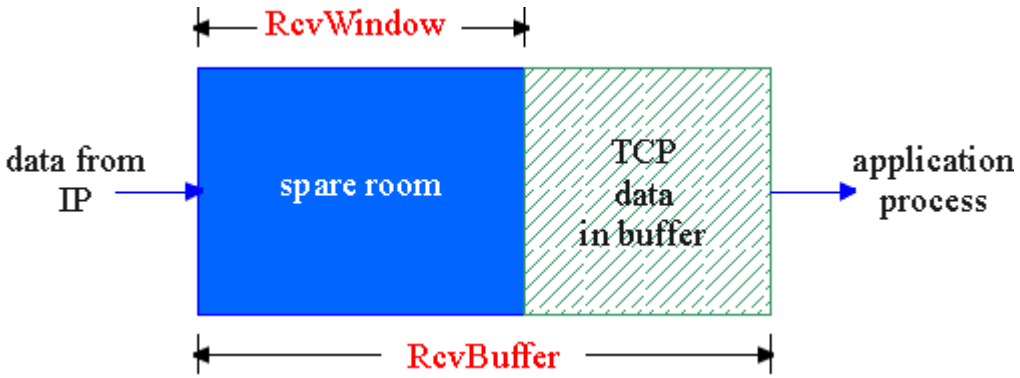
TCP Flow Control

flow control

sender won't overrun Receiver's buffers by transmitting too much, too fast

RcvBuffer = size or TCP Receive Buffer

RcvWindow = amount of spare room in Buffer



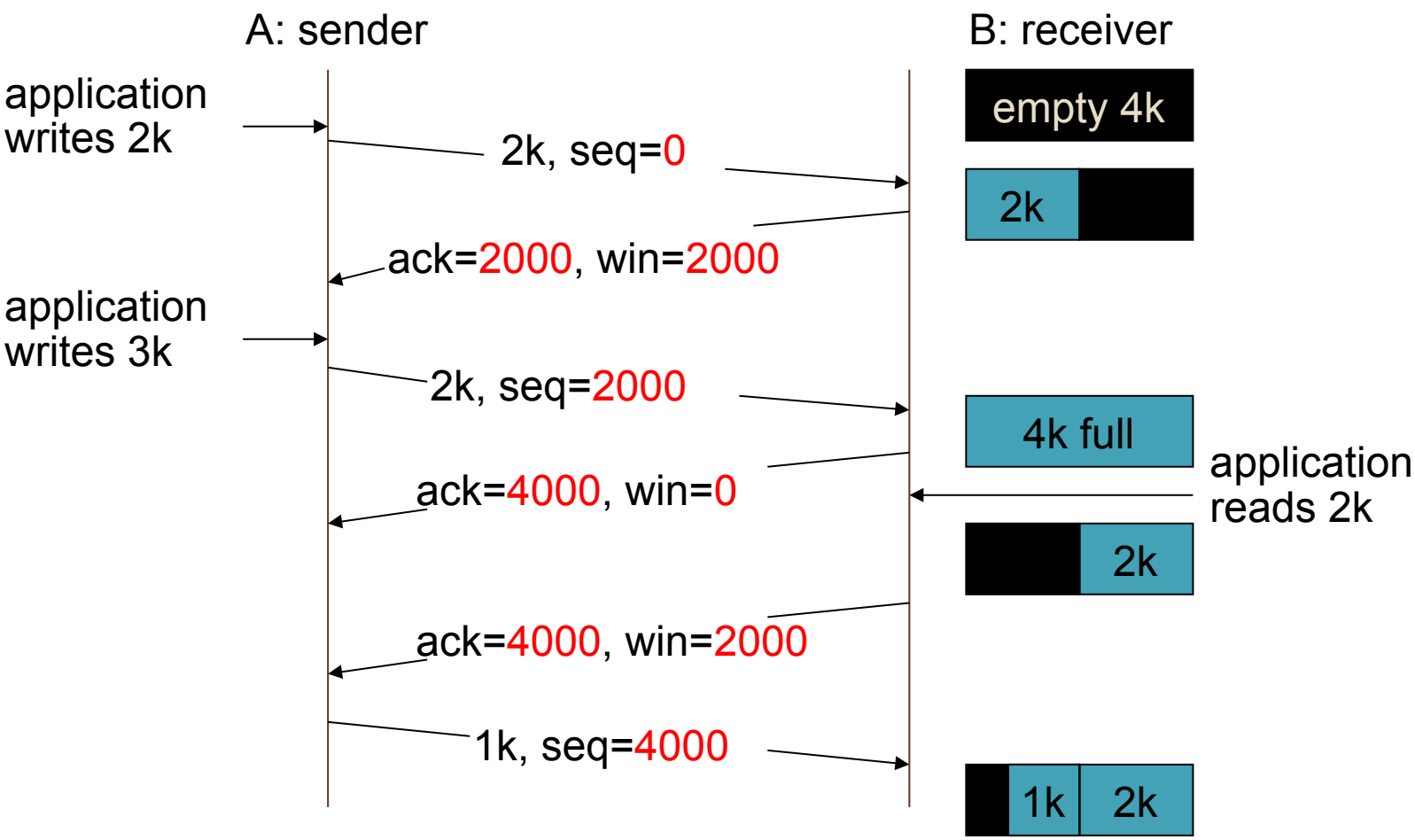
receiver buffering

receiver: explicitly informs sender of (dynamically changing) amount of free buffer space

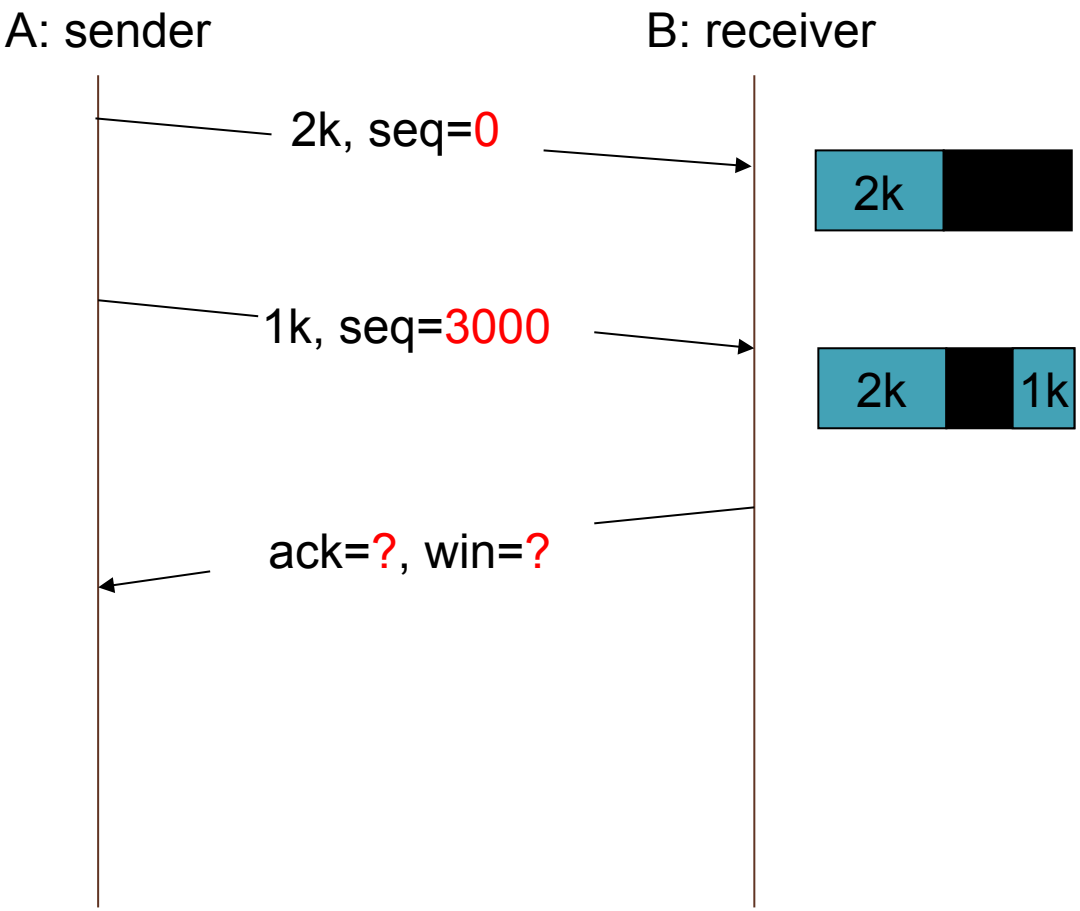
- **RcvWindow field** in TCP segment

sender: keeps the amount of **transmitted**, **unACKed** data less than most recently received **RcvWindow**

Receive Window (Example)



Quick exercise



- A. ack=2000, win=1000
- B. ack=2000, win=2000
- C. ack=4000, win=1000
- D. ack=4000, win=2000

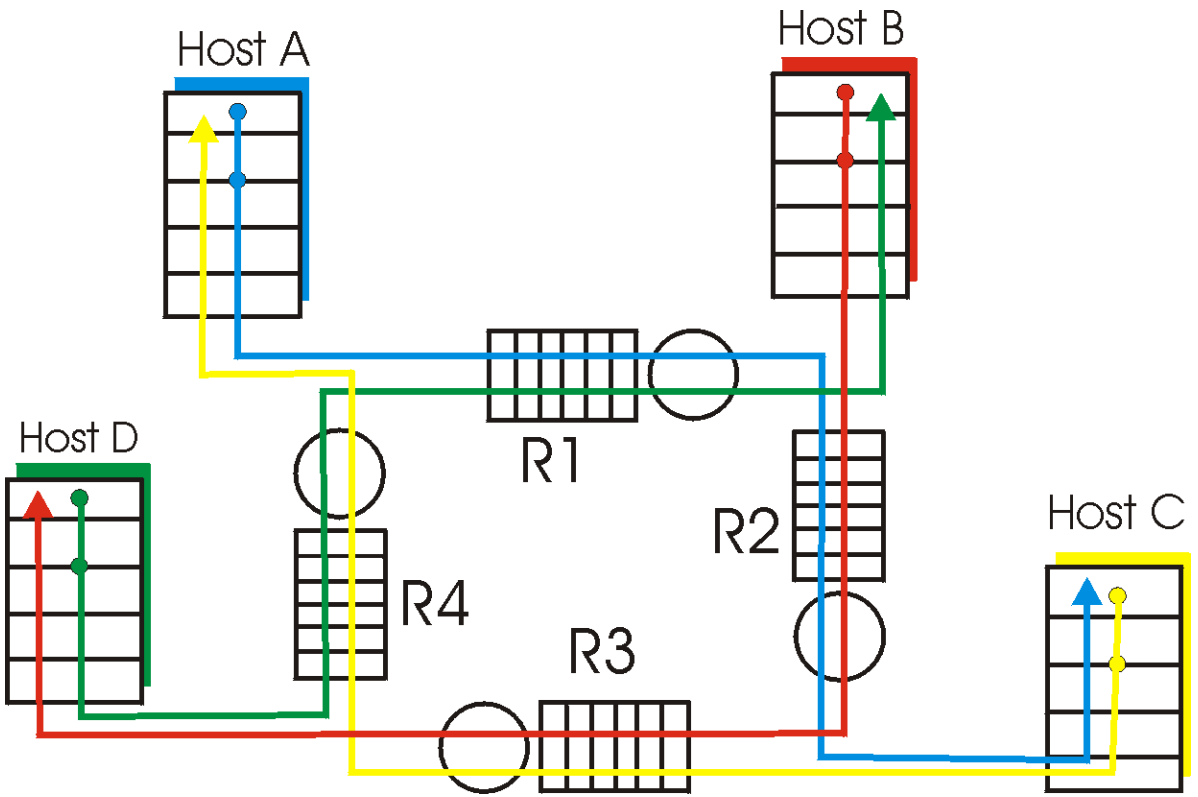
Principles of Congestion Control

Congestion:

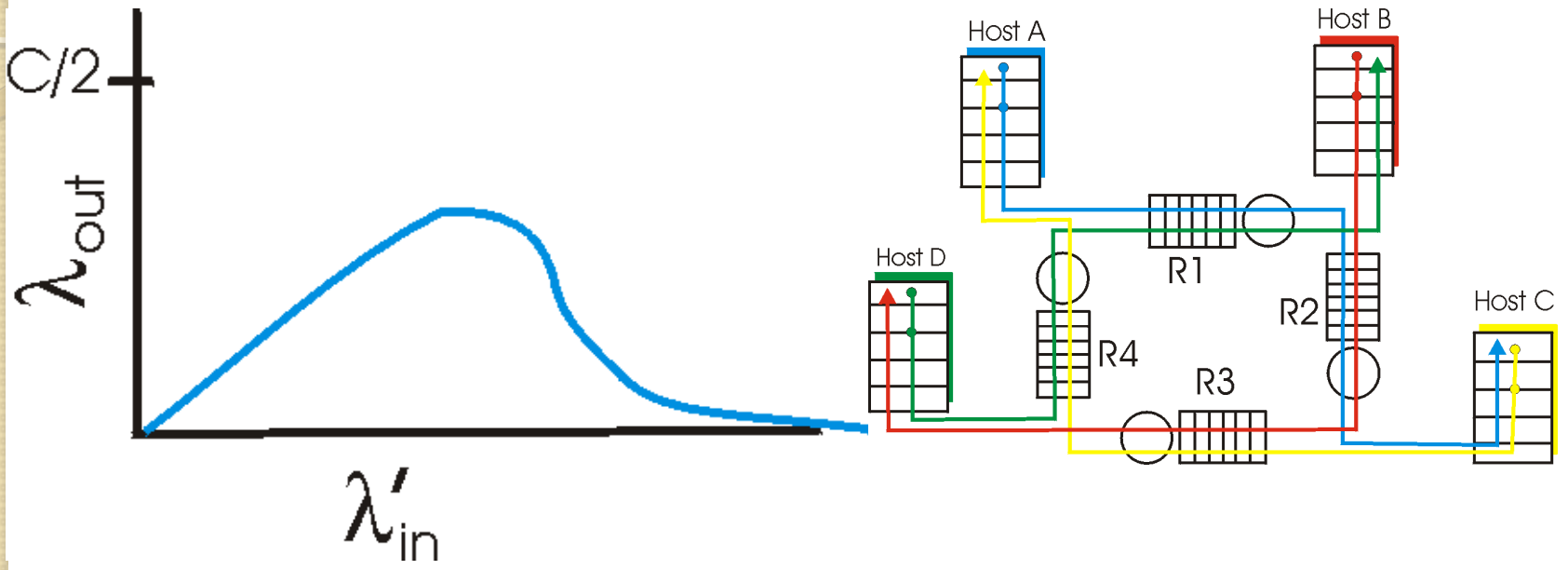
- Informally: "too many sources sending too much data too fast for *network* to handle"
- Different from flow control!
- Manifestations:
  - lost packets (buffer overflow at routers)
  - long delays (queuing in router buffers)

Causes/costs of congestion

- Four senders/receivers
- Multi-hop paths
- Timeout/retransmit



Reduced throughput

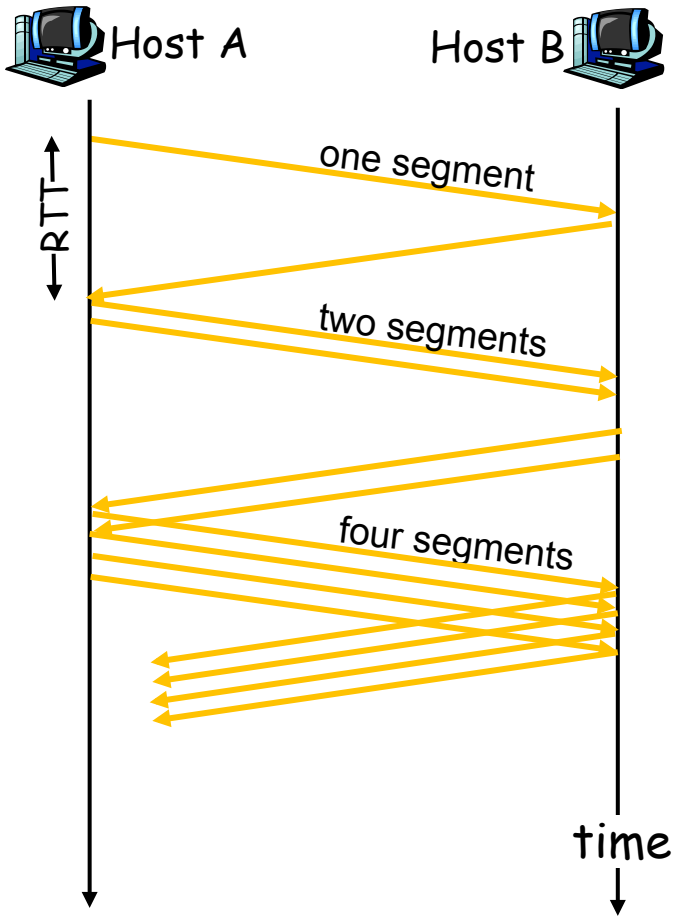


Extra "cost" of congestion:

when packet dropped, any upstream transmission capacity used for that packet was wasted!

TCP Slow Start

- Probe a link's capacity…
- For every round of successful transmission, the congestion window size is doubled.
- First timeout establishes our first guess at the link's capacity.
- This topic is covered in detail in NWEN302.



Quick exercise

- Which of the following are correct?
  - A. In the slow start algorithm, the congestion window increases additively until congestion is detected.
  - B. If congestion is detected by timeout, a new congestion avoidance phase will be started.
  - C. Congestion control aims to regulate the amount of data a source can send before receiving any acknowledgement.
  - D. None of the above

Quick exercise on TCP

- A. TCP supports reliable transport: between sending and receiving process
- B. TCP supports flow control: sender won't overwhelm receiver
- C. TCP supports congestion control: throttle sender when network overloaded
- D. TCP provides timing, minimum bandwidth guarantees

Quick exercise on UDP

- A. Best effort data transfer between sending and receiving process
- B. Does not require connection setup
- C. Does not check data integrity
- D. Does not provide flow control, congestion control, timing, or bandwidth guarantees
- E. Provide maximum freedom to application designers

Application Layer

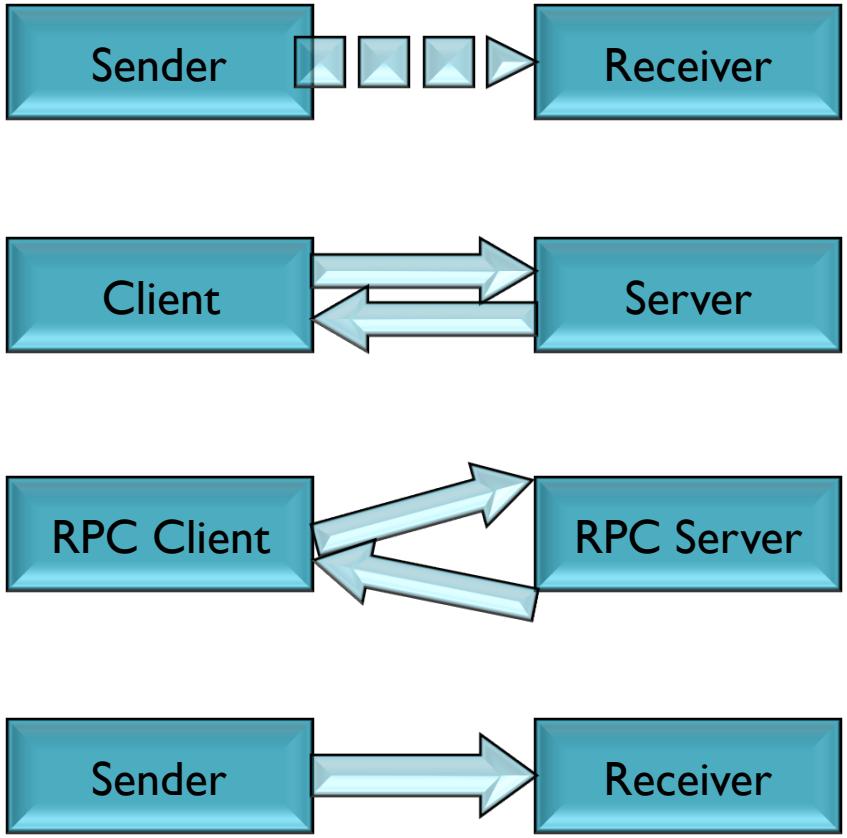
- Application characteristics
- Communication architecture
- HTTP

Application-layer protocols

- Just as the Datalink, Network and Transport layers had protocols to communicate
- So too does the Application layer. However these are defined on a per application basis.
- Examples, HTTP, SMTP, Jabber/XMPP etc. There are many protocols, and no one can know them all (many are proprietary)

Communication Architectures

- Messaging
- Request-response
- Do-operation
- Streaming



Application Protocol Characteristics

Data loss

- some apps (e.g., audio) can tolerate some loss
- other apps (e.g., file transfer, telnet) require 100% reliable data transfer

Timing

- some apps (e.g., Internet telephony, interactive games) require low delay to be "effective"

Bandwidth

some apps (e.g., multimedia) require minimum amount of bandwidth to be "effective"

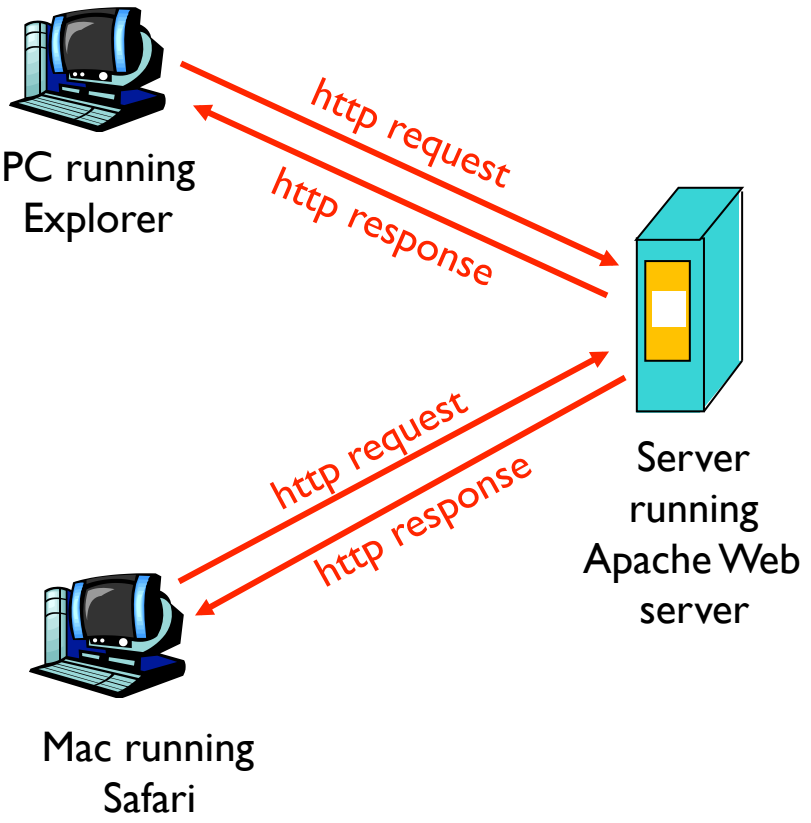
other apps ("elastic apps") make use of whatever bandwidth they get

Quick exercise

- Suppose that you are building an interactive computer gaming application, what communication requirements should the gaming protocol support?
  - A. minimum bandwidth
  - B. no data loss
  - C. maximum delay
  - D. connection-oriented delivery

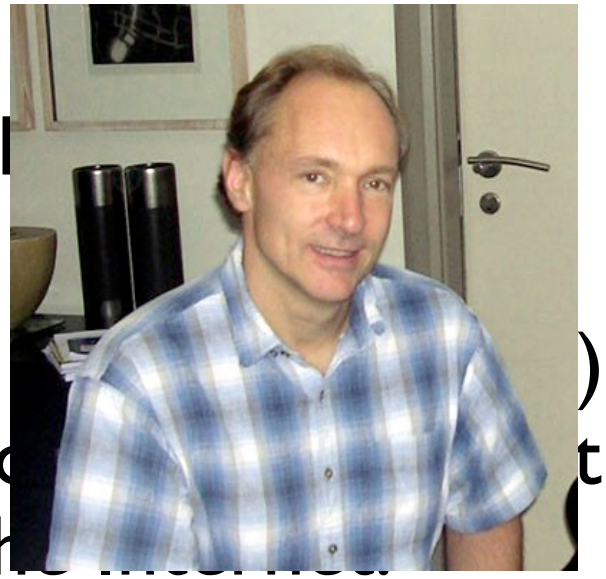
The Web: the HTTP protocol

- Client-server model:
  - client: browser that requests, receives, "renders" Web objects
  - server: Web server sends objects in response to requests



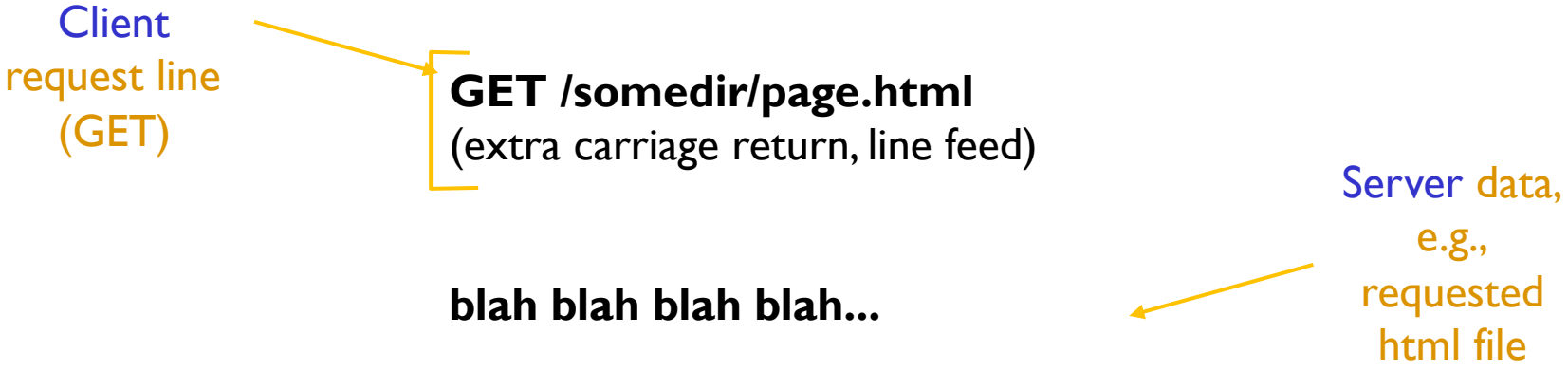
A Little Historical Context

- The `WEB' is composed of a set of linked HTML documents.
- The WWW is a networ
- The HyperText Transfer
  is an application level pr
  'web' documents over th



Growth of a Protocol (HTTP)

- Started with HTTP/0.9 (1990)
  - raw data transfer over the Internet (simple)
  - client issues GET Request
  - server replies with bytes comprising the HTML



Growth of a Protocol (HTTP/1.0)

- Next HTTP/1.0 (1996)
  - sophisticated requests {GET, HEAD and POST}
  - server response {headers + data}
  - better client-server coordination/interaction.
  - coordinates with other protocols {SMTP, NNTP, …}
  - MIME (Multipurpose Internet Mail Extensions)