IERG 4180
Network Software Design and Programming

Application Protocol Design

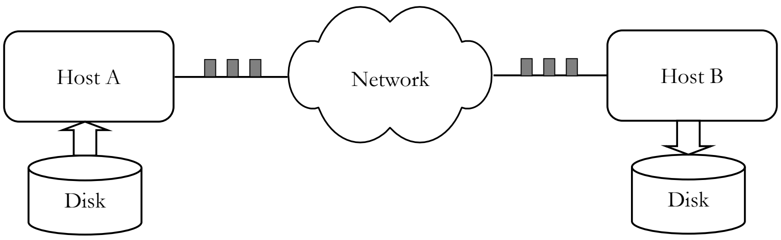
Part II – Performance Issues

Contents

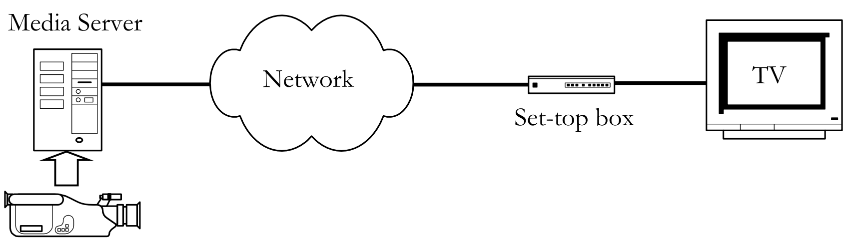
- 1. Introduction
- 2. What do the users want?
- 3. What does the application need to do?
- 4. Application Traffic Characteristics
- 5. Interaction with Lower-Layer Protocols
- 6. Interaction with the Network
- 7. References

1. Introduction

- Illustrative Applications
  - File Transfer



  - Live Media Broadcasting



2. What do the users want?

- Data
  - I.e., the 1's and 0's.
  - Storable or not?
  - Is absolute correctness required? / Are omissions allowed?
  - Should multiple data objects be delivered to the same receiver in sequence, or in parallel? In what order, if any?
- Information
  - I.e., the interpretation of the 1's and 0's.
  - Are there timing information in the data?
  - Does the value of the information depend on the delivery time?
  - Does the value of the information depend on the proportion of data delivered?

Exercise: Compare the two illustrative applications on what are being delivered.

2. What do the users want?

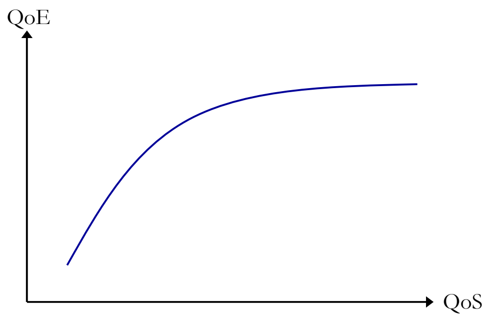
- Quality-of-Service (QoS)
  - Measures performance metrics at the transport layer (or below)
  - Common metrics
    - Bandwidth (average rate and burst size)
    - Delay and delay jitter
    - Loss
- Quality-of-Experience (QoE)
  - Measures performance as perceived by the end user
  - Metrics are application-specific
  - Example:
    - Web browsing - Page loading time
    - Video streaming – startup-delay, rebuffering frequency/duration, video quality, etc.
    - VoIP – delay, voice quality

3. What does the application need to do?

- Issues in Data/Information Delivery
  - Network Traffic Characteristics
    - Elastic v.s. inelastic traffic.
  - Topology
    - Message flows and connectivity.
  - Transmission Model
    - Unicast, multicast, or broadcast.
  - Quality of Service
    - Bandwidth, delay, and loss.
  - Security
    - Authentication, eavesdropping, and privacy.

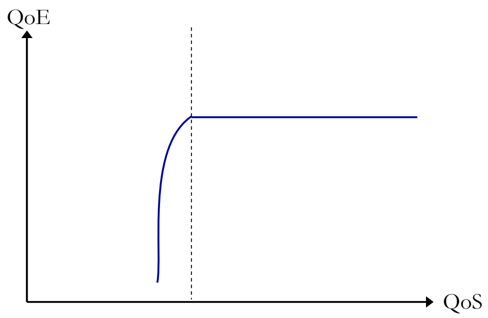
4. Application Traffic Characteristics

- Elastic Traffic
  - Application performance (i.e., QoE) relates to network resources availability (i.e., QoS) in a proportional manner.
  - E.g., the more the resources, the better it performs.



Note the relation is not a straight line.

- Inelastic Traffic
  - Application performance degrades sharply when network resources drop below a threshold.
  - E.g., fails completely when minimum resources are not available.

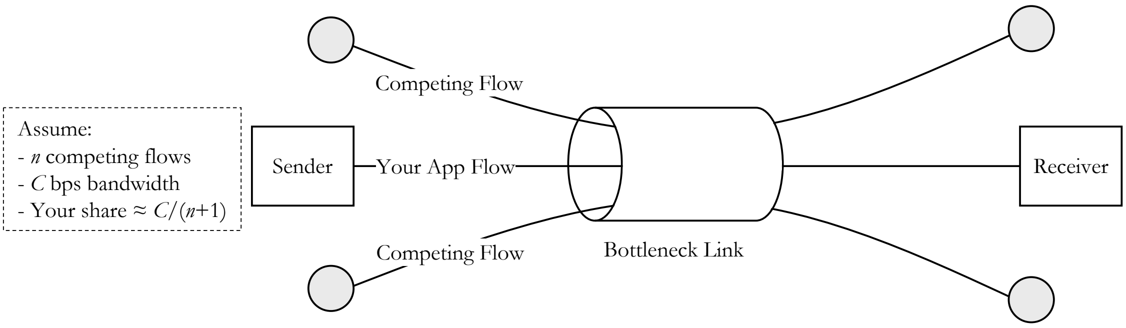


QoE levels off after a critical QoS point.

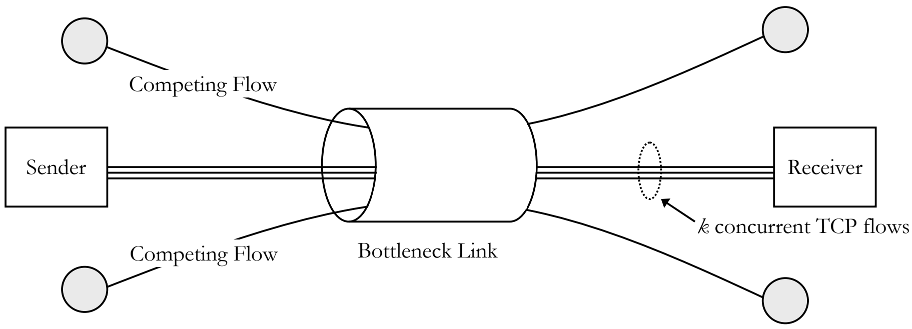
4. Application Traffic Characteristics

- Elastic Traffic
  - Compatible with the current Internet, which provides best-effort service.
  - Service response time is determined by
    - Network delays, bottleneck bandwidth
      - physical/policy limited
      - not directly controllable
    - Transport throughput
      - choice of transport protocol (TCP, UDP, etc.)
      - network programming optimization
    - Application processing delays
      - data retrieval, protocol processing, encoding/decoding, etc.
    - Perceived versus actual response time
      - Progressive rendering, streaming, etc.

- Elastic Traffic
  - Interaction with Transport Protocols
    - Transmission Control Protocol (TCP)
      - No control over transmission rate, send at maximum rate TCP will allow.
      - Share bottleneck bandwidth fairly with other competing TCP flows.

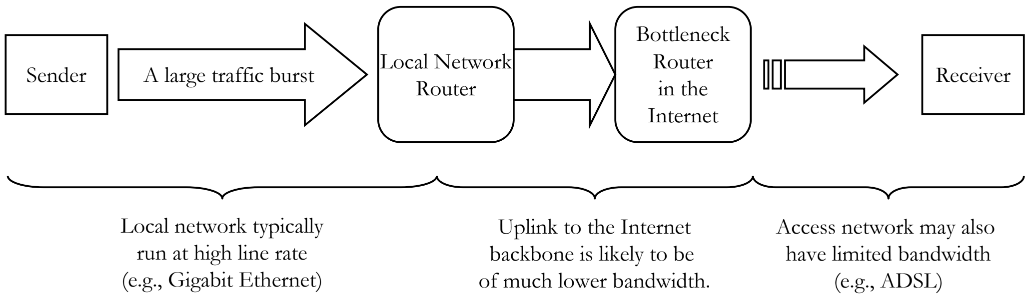


- Elastic Traffic
  - Interaction with Transport Protocols
    - Transmission Control Protocol (TCP)
      - Aggressive tactic:



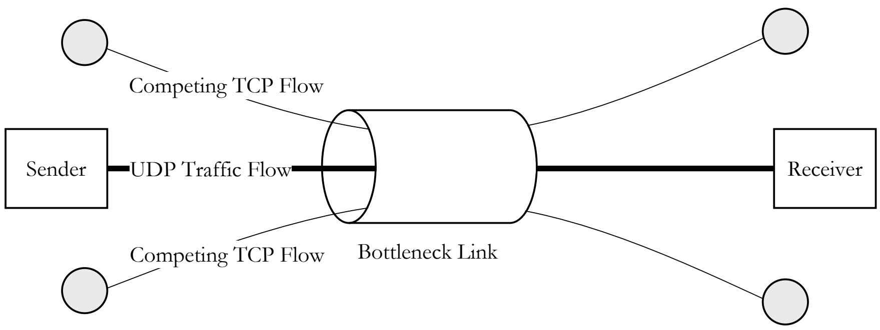
Assume:
- n competing flows
- C bps bandwidth
- k concurrent TCP flows for your application
- Your share ≈ $kC/(n+k)$

- Elastic Traffic
  - Interaction with Transport Protocols
    - User Datagram Protocol (UDP)
      - No limit on transmission rate, controlled by your application.
      - Careful control of the outgoing rate is still needed to avoid network congestion.



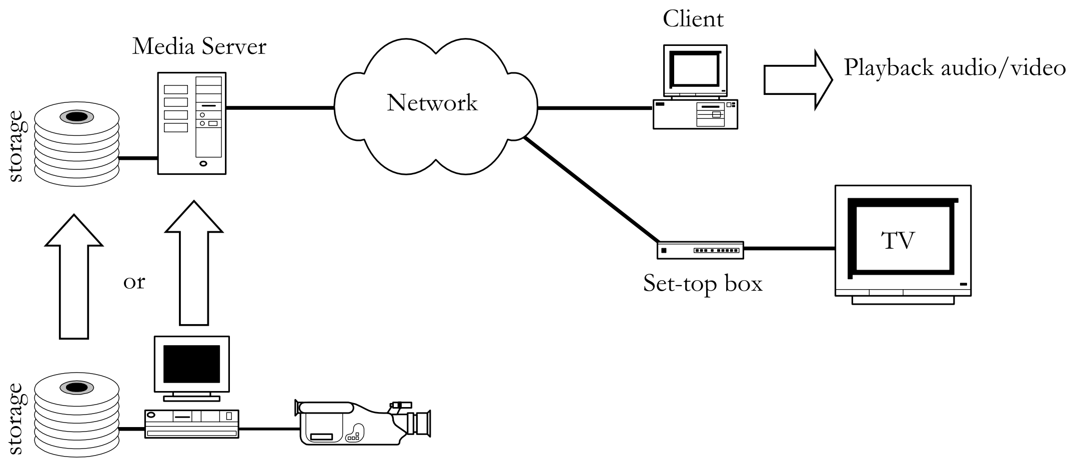
Caveats:
- The Win32 API `TransmitFile()` can send datagram at extremely high data rate.
- Some form of dynamic transmission rate control and receiver feedbacks will be needed.

- Elastic Traffic
  - Interaction with Transport Protocols
    - User Datagram Protocol (UDP)
      - Does not share bandwidth with competing traffic flows fairly.
      - Grab all it can get, other TCP flows may suffer.



Caveats:
- Some ISP implements active queue management (AQM) to enforce fair bandwidth sharing among UDP/TCP flows.
- Some ISP restricts flows on certain port numbers (e.g., P2P applications).

- Inelastic Traffic
  - Example: Continuous Media Streaming
  - Continuous Media
    - Media data that has explicit timing information for correct presentation, e.g., audio, video, etc.

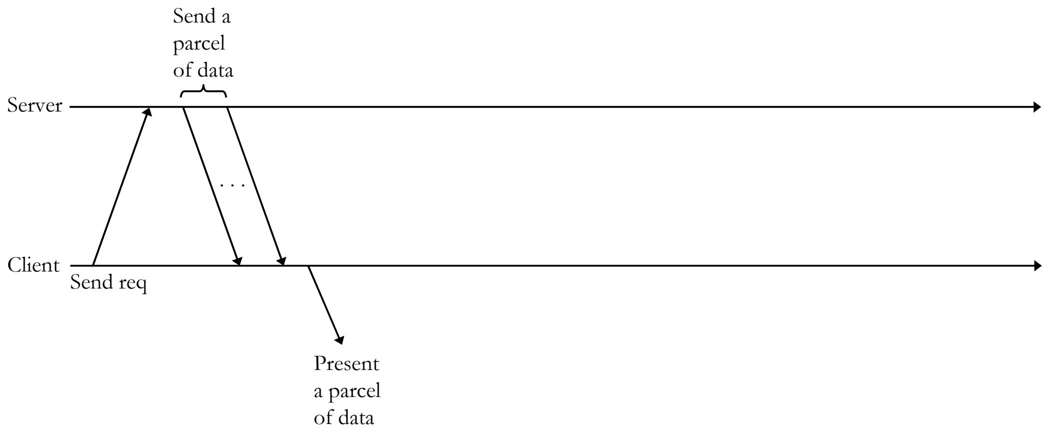


4. Application Traffic Characteristics

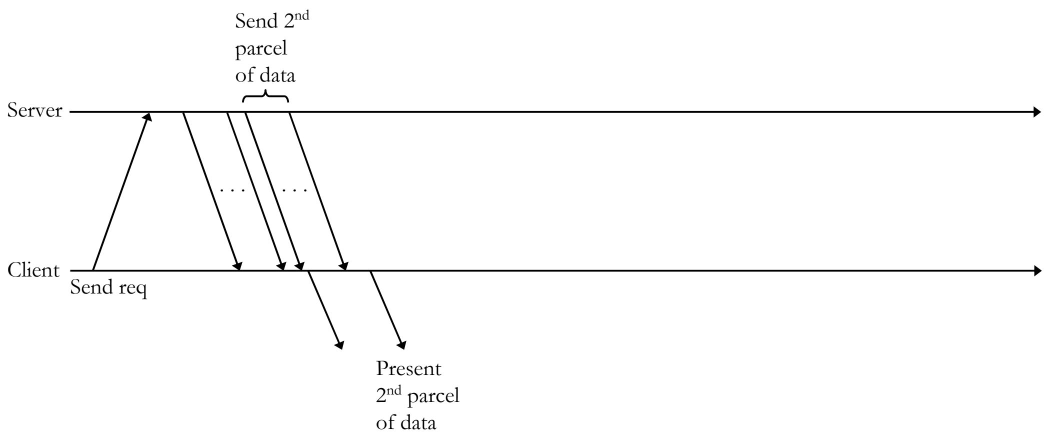
- Inelastic Traffic
  - Requirements for Continuous Media Delivery
    - Bandwidth
      - The available bandwidth must not be substantially lower than the media data rate.
    - Delay or response time
      - Real-time applications (e.g., 150ms one-way delay for VoIP)
      - Soft-real-time applications (e.g., a few seconds for VoD)
    - Data integrity
      - Not necessarily requires complete integrity, especially for real-time applications.
    - Playback Continuity
      - Not necessarily requires complete continuity, especially for real-time applications.

4. Application Traffic Characteristics

- Inelastic Traffic
  - Continuous Media Streaming (or Play-while-Download)

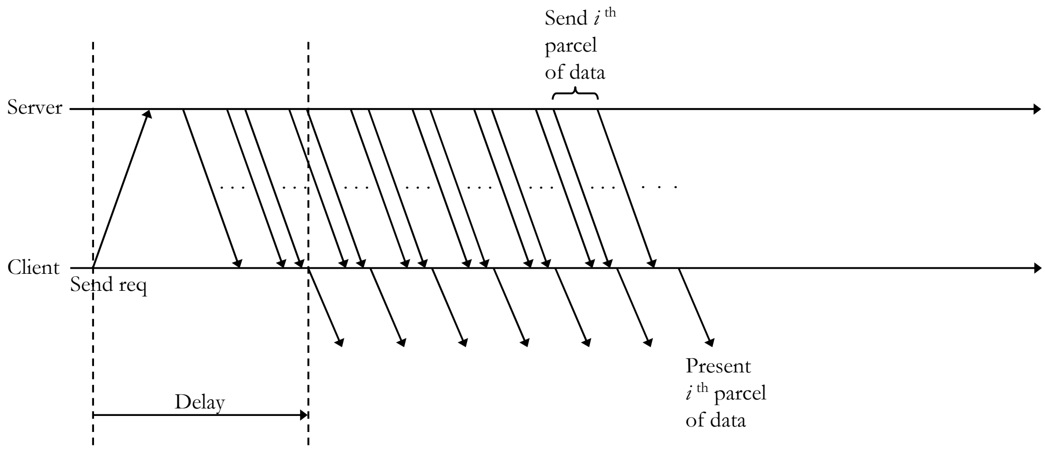


- Inelastic Traffic
  - Continuous Media Streaming



4. Application Traffic Characteristics

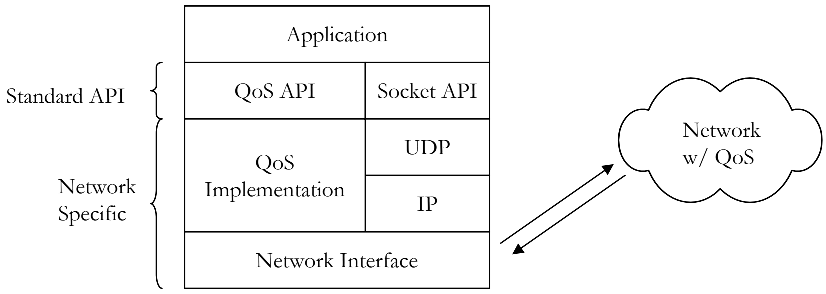
- Inelastic Traffic
  - Continuous Media Streaming



Question: What are the prerequisites for streaming to work?

4. Application Traffic Characteristics

- Inelastic Traffic
  - Types of Network
    - Best-effort delivery without performance guarantees
      - E.g., the Internet.
    - Private network with capacity dimensioning
      - E.g., broadband residential networks.
    - With resource reservation and performance guarantees
      - E.g., ATM, RSVP, etc.
      - Winsock 2 has defined a set of QoS APIs for this purpose.



4. Application Traffic Characteristics

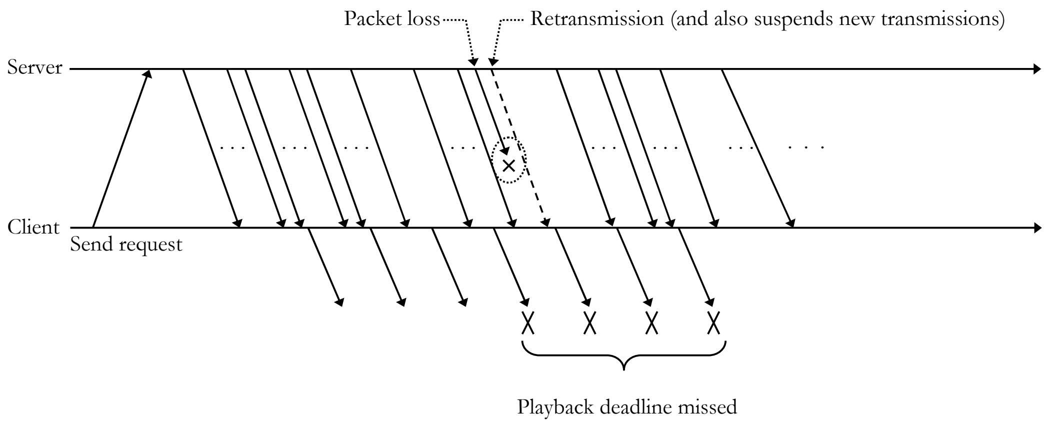
- Inelastic Traffic
  - Resource Reservation API
    - Example: Winsock 2 QoS API

```
int WSAAPI
WSAConnect (
     SOCKET                        s,
     const struct sockaddr FAR     *name,
     int                           namelen,
     LPWSABUF                      lpCallerData,
     LPWSABUF                      lpCalleeData,
     LPQOS                         lpSQOS,
     LPQOS                         lpGQOS);

typedef struct _flowspec
{
    uint32      TokenRate;              /* In Bytes/sec */
    uint32      TokenBucketSize;        /* In Bytes */
    uint32      PeakBandwidth;          /* In Bytes/sec */
    uint32      Latency;                /* In microseconds */
    uint32      DelayVariation;         /* In microseconds */
    SERVICETYPE ServiceType;
    uint32      MaxSduSize;             /* In Bytes */
    uint32      MinimumPolicedSize;     /* In Bytes */

} FLOWSPEC, *PFLOWSPEC, FAR * LPFLOWSPEC;
```

Source Traffic Characteristics: TokenRate, TokenBucketSize, PeakBandwidth

Desired QoS: Latency, DelayVariation, ServiceType, MaxSduSize, MinimumPolicedSize

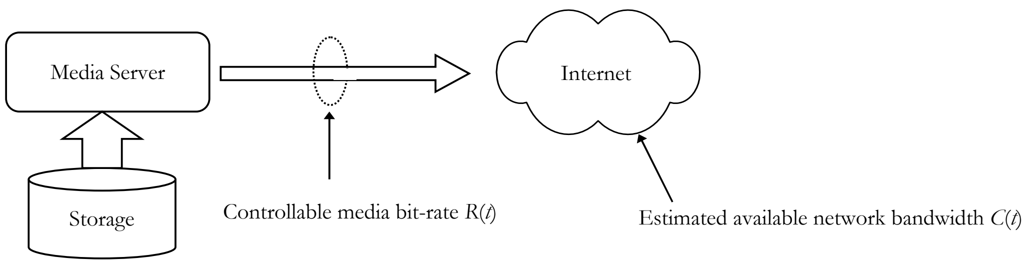
- Inelastic Traffic
  - Interaction with Transport Protocols
    - Streaming over TCP
      - Packet loss results in aggressive congestion avoidance.



4. Application Traffic Characteristics

- Inelastic Traffic
  - Interaction with Transport Protocols
    - Streaming over UDP
      - Application can control the transmission rate directly.
      - Explicit flow control may be needed.
      - Subject to rate limiting by the ISP.
      - Other competing TCP flows may suffer with less bandwidth.
      - Explicit congestion avoidance may be needed.
      - Suitable for
        - » real-time applications where timely delivery is more important than complete delivery.
        - » private network with prior bandwidth provisioning or network supporting resource reservation.

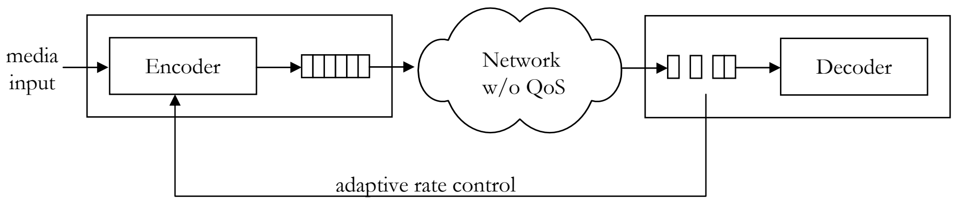
- Semi-elastic Traffic
  - Adaptive Continuous Media Streaming
    - Media content data rate can be controlled to some extent.



Goal: Adjust *R*(*t*) according to *C*(t) to optimize performance
(e.g., reduce playback jitter, improve visual quality).

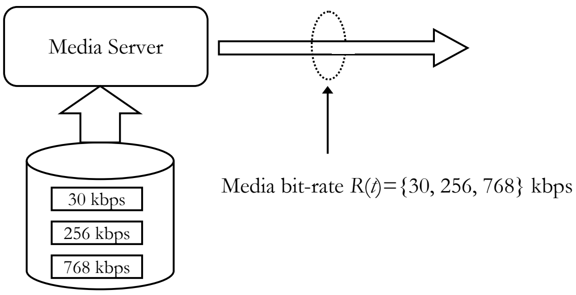
4. Application Traffic Characteristics

- Semi-elastic Traffic
  - Adaptive Continuous Media Streaming
    - On-the-fly Encoded Media Streams
      - E.g., voice/video phone/conferencing.



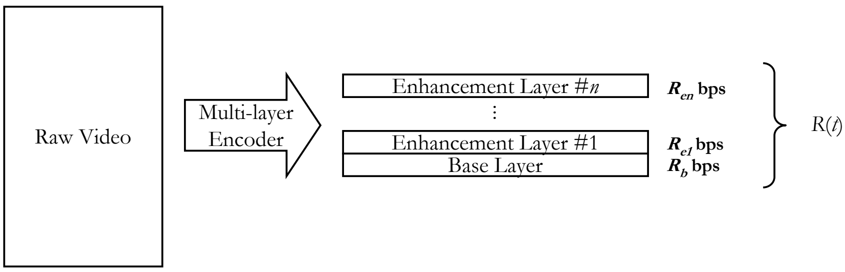
Question: What are the disadvantages or limitations of this approach?

- Semi-elastic Traffic
  - Adaptive Continuous Media Streaming
    - Encode and store multiple versions of the media content.
    - Select the best version in accordance with bandwidth availability.



The most widely-used solution in the industry, e.g.,
- Apple's HTTP Live Streaming (m3u8/MPEG-TS)
- Microsoft's Smooth Streaming
- MPEG-DASH

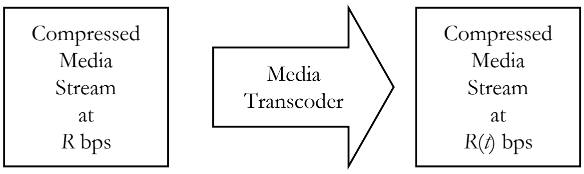
- Semi-elastic Traffic
  - Adaptive Continuous Media Streaming
    - Use multi-layer codec.



$$R(t) = \left\{R_b, R_b + R_{e1}, ..., R_b + R_{e1} + ... + R_{en}\right\}$$

Question: What are the disadvantages or limitations of this approach?

- Semi-elastic Traffic
  - Adaptive Continuous Media Streaming
    - Use online real-time media transcoding:



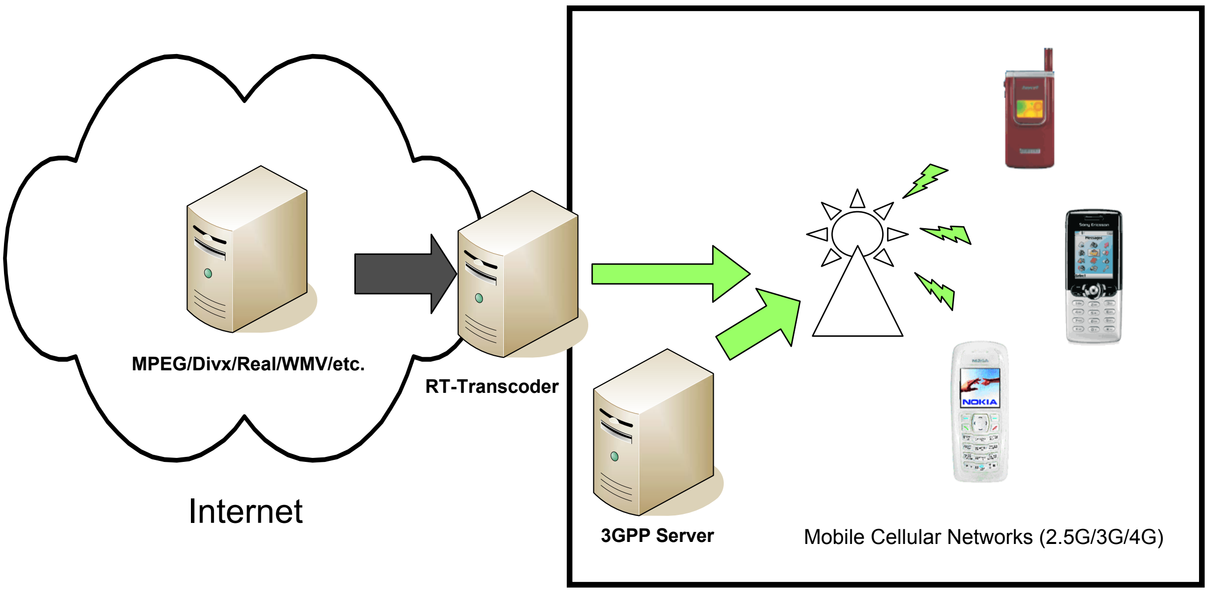
$$R \geq R(t) > R_{min}$$

<u>Characteristics:</u>

1. Bit-rate is continuously adjustable.
2. Only one version of the media is needed.
3. Standard media encoder can be used.
4. Transcoded media bit-stream conforms to the original encoding standard/format.

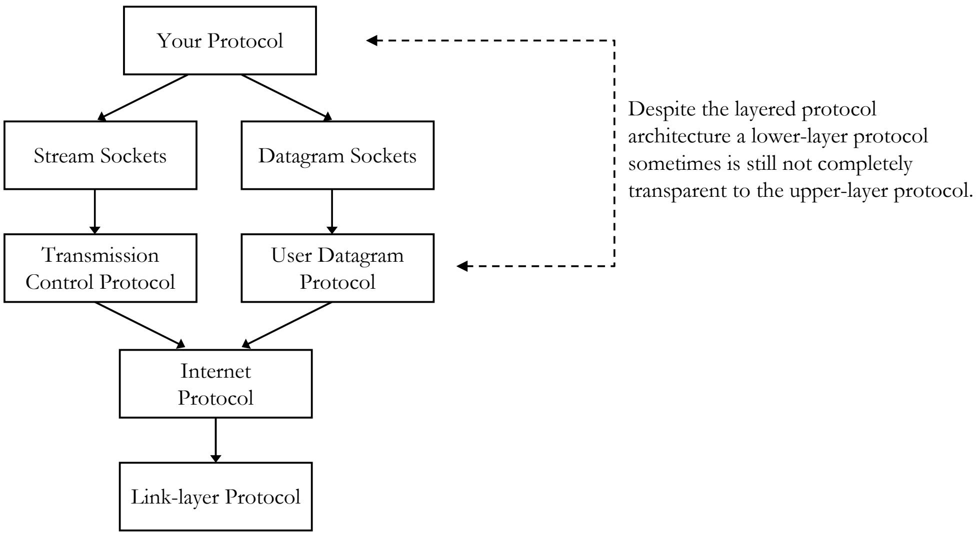
Question: What are the disadvantages or limitations of this approach?

- Semi-elastic Traffic
  - Adaptive Continuous Media Streaming
    - Enabling Ubiquitous Access to Media Contents

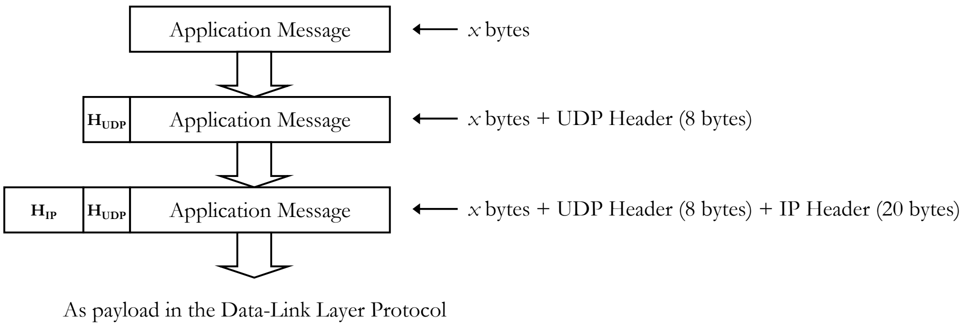


5. Interaction with Lower-Layer Protocols

- Protocol Layers in the Internet



- User Datagram Protocol (UDP)
  - Datagram Sizing



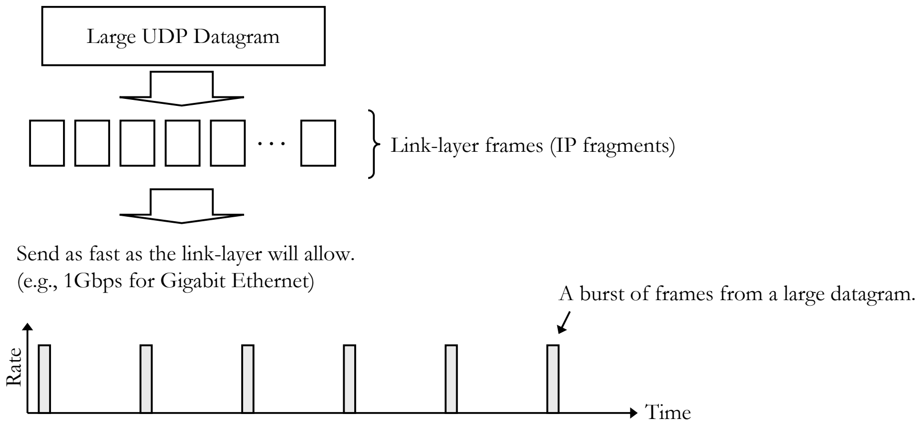
Max payload size of an Ethernet frame = 1,500 bytes.

So if x > (1,500-20-8) then the IP packet is too large to deliver over an Ethernet frame.

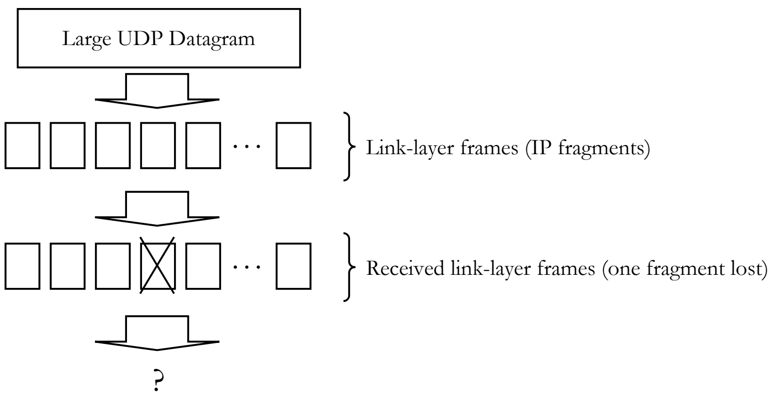
IP can fragment the packet automatically, and automatically reassemble it at the receiver.

So what are the implications?

- User Datagram Protocol (UDP)
  - Datagram Sizing
    - Scenario #1
      - x is slightly larger than the payload limit, e.g., (1,500-20-8).
    - Scenario #2
      - x is much larger than the link-layer MTU size.



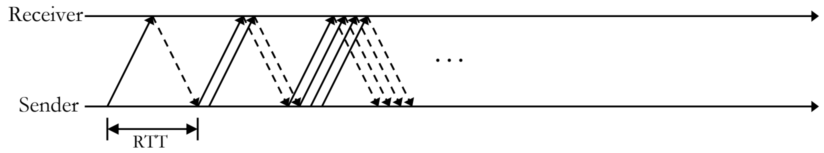
- User Datagram Protocol (UDP)
  - Datagram Sizing
    - Scenario #2
      - x is much larger than the link-layer MTU size.



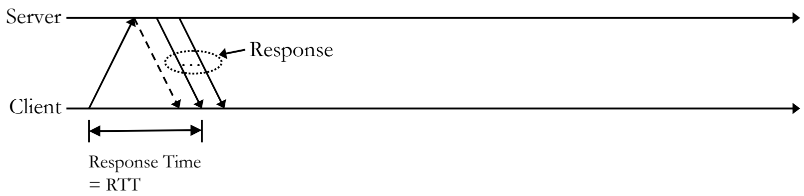
Question: What is the probability of correctly receiving an x-byte datagram assuming MTU of y bytes and packet loss probability of p?

5. Interaction with Lower-Layer Protocols

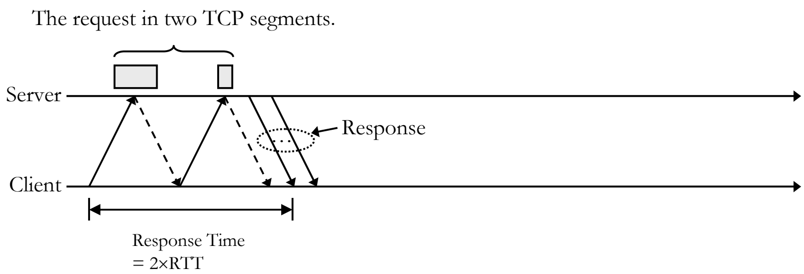
- Transmission Control Protocol (TCP)
  - Slow-Start



  - Request-Response-Type Application Protocol
    - If the request message can be transported in one TCP segment:



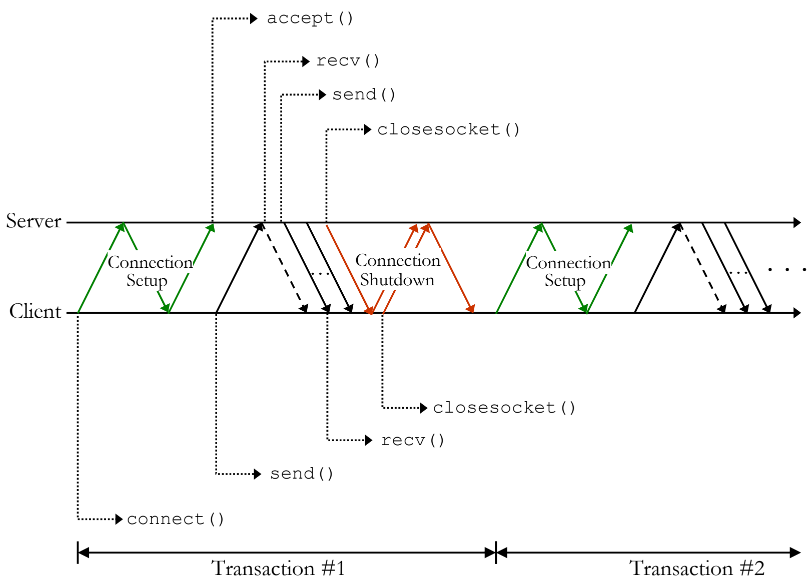
- Transmission Control Protocol (TCP)
  - Request-Response-Type Application Protocol
    - If the request message cannot be transported in one TCP segment:



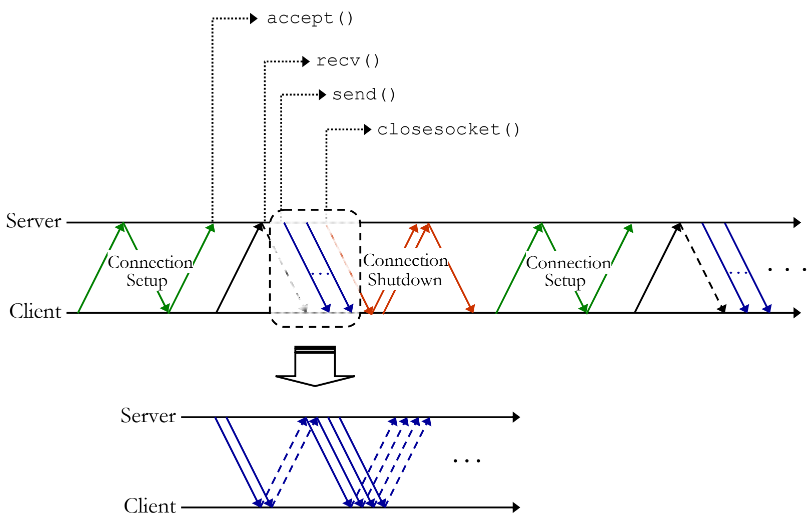
Minimum MTU for IP-compatible link-layer = 576 bytes

So what is the maximum request size to avoid generating two TCP segments?

- Transmission Control Protocol (TCP)
  - ◆ Transaction-based Protocols
    - Using non-persistent TCP Connection (e.g., HTTP 1.0):

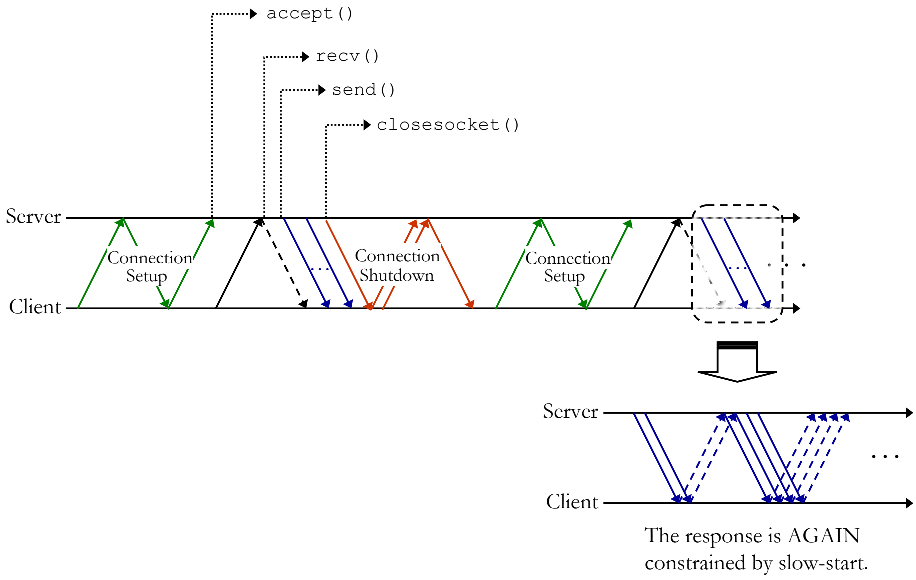


- Transmission Control Protocol (TCP)
  - Transaction-based Protocols
    - Using non-persistent TCP Connection (e.g., HTTP 1.0):

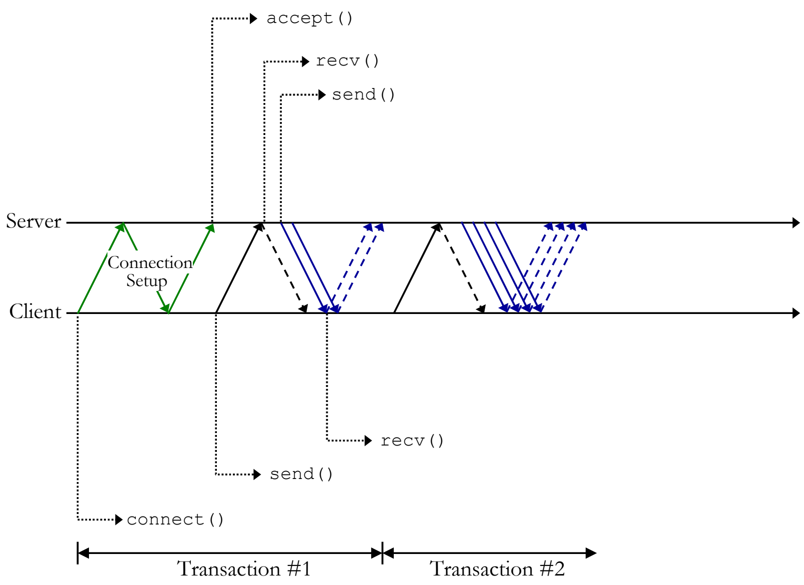


The response is constrained by slow-start.

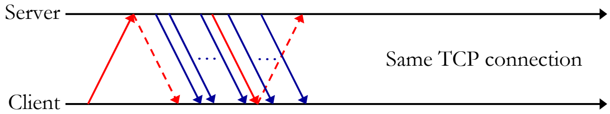
- Transmission Control Protocol (TCP)
  - Transaction-based Protocols
    - Using non-persistent TCP Connection (e.g., HTTP 1.0):



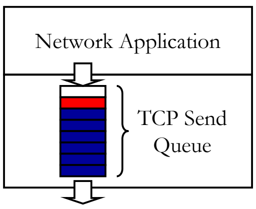
- Transmission Control Protocol (TCP)
  - Transaction-based Protocols
    - Using persistent TCP Connection (e.g., HTTP 1.1):



- Transmission Control Protocol (TCP)
  - Session-based Protocols
    - In-band Control Messages (e.g., persistent HTTP):



- Issues
  - Control message blocking by queued data



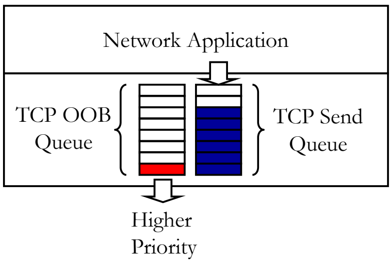
- Transmission Control Protocol (TCP)
  - Session-based Protocols
    - Pseudo Out-of-band Control Messages:

```
int send (SOCKET s, const char* buf, int len, int flags);
```

flags	**Value**	**Meaning**
	`MSG_OOB`	Send out-of-band data (`SOCK_STREAM` only)

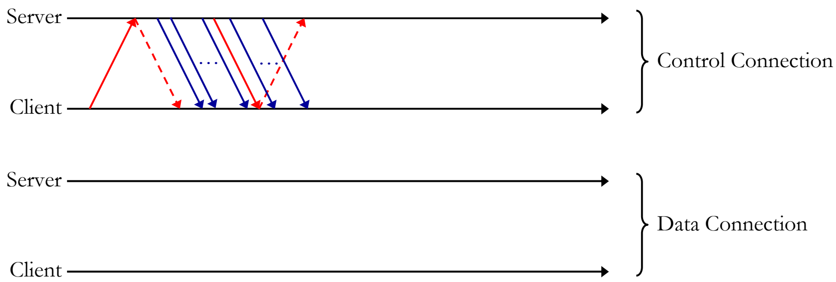
```
int recv (SOCKET s, char* buf, int len, int flags);
```

flags	**Value**	**Meaning**
	`MSG_OOB`	Process out-of-band data.



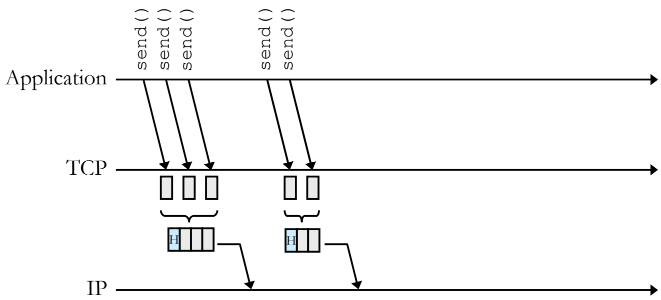
Caveat: OOB data support is not mandatory.

- Transmission Control Protocol (TCP)
  - Session-based Protocols
    - Out-of-band Control Messages (e.g., FTP):
      - Separate connections for control messages and data exchange.

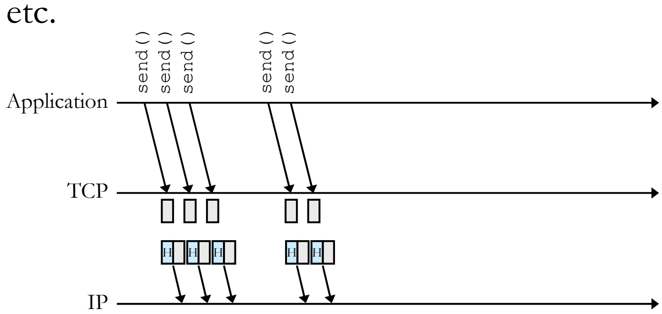


Question: Does this guarantee minimum delay for control messages?

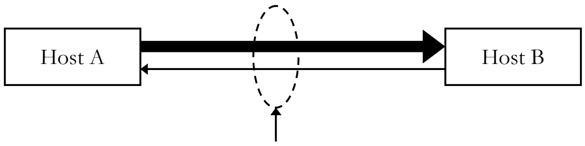
- Protocol-induced Delay
  - Nagle's Algorithm in TCP
    - Delay sending a message until
      - all sent data are acknowledged or
      - the queueing data can fill a link-layer frame or
      - it has an ACK to send.



- Protocol-induced Delay
  - Nagle's Algorithm in TCP
    - Turned on by default in most implementations to improve bandwidth utilization (by reducing header overhead).
    - Can be turned on/off via `setsockopt()` with the option `TCP_NODELAY`.
    - Should NOT be turned off unless short delay is more important than throughput/bandwidth utilization, e.g., real-time applications such as voice-over-IP, protocols for network games, etc.



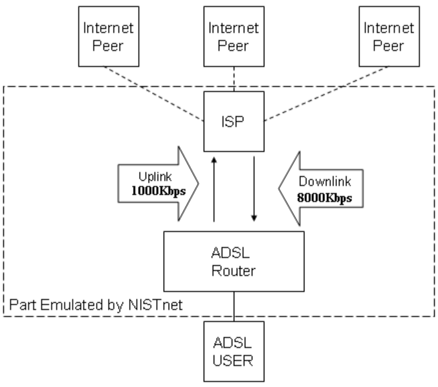
- Asymmetric Network
  - Example: xDSL

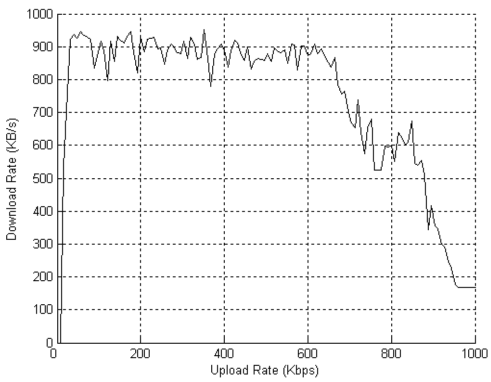


The link from Host A to Host B has significantly more bandwidth (and/or shorter delay).

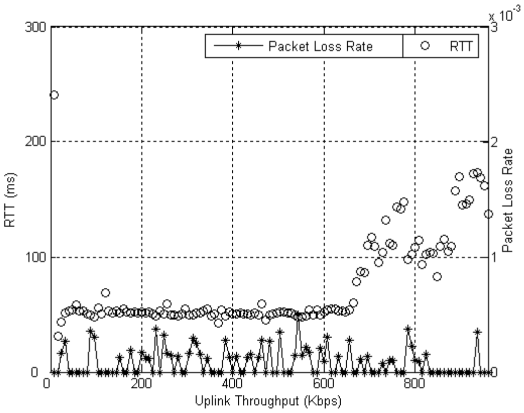
  - Implication to TCP?
  - Implication to network application?

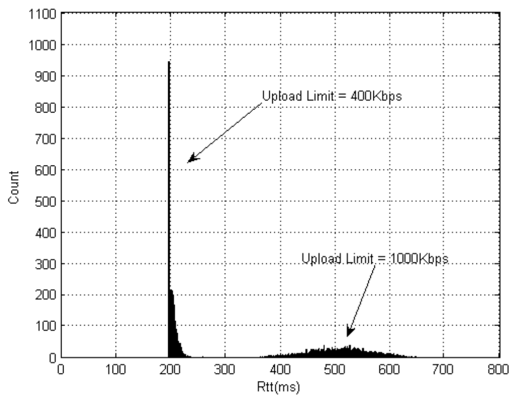
- Asymmetric Network
  - Performance of Bit-torrent over ADSL



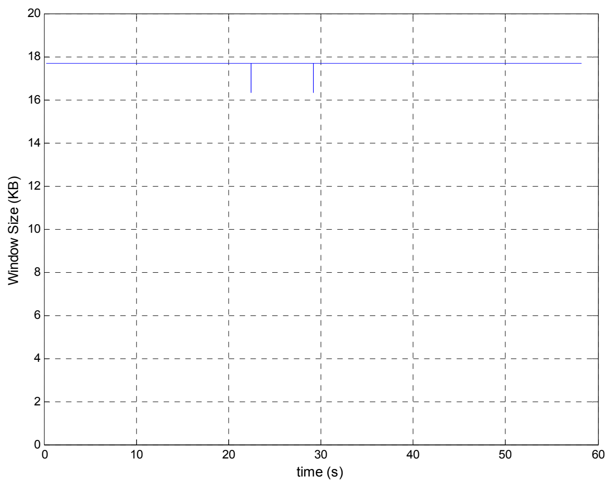


- Asymmetric Network
  - Performance of Bit-torrent over ADSL



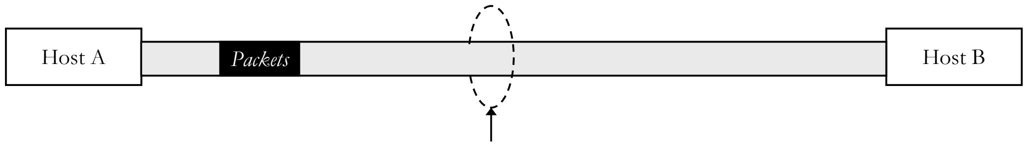


- Network with Large Bandwidth-Delay Product
  - TCP Receiver's Advertised Window Size (AWnd)
    - Small in some platforms (e.g., 17KB ~ 64KB in Windows XP)
    - AWnd is a 16-bit field inside the TCP header
    - Is flow control still a problem today?



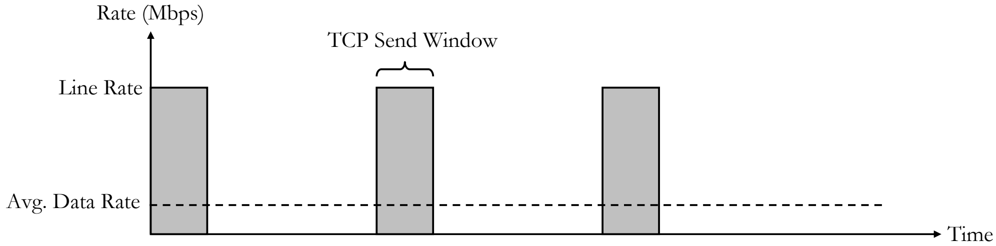
6. Interaction with the Network

- Network with Large Bandwidth-Delay Product
  - Bandwidth-Delay Product



Link with bandwidth C and end-to-end delay D, its bandwidth-delay product = $C \times D$.

E.g., two hosts connected by a fiber-optic link of 10Gbps with 100ms,
its bandwidth delay product will be equal to 10Gbps × 100ms = 1Gb or 125MB.



- Network with Large Bandwidth-Delay Product
  - TCP Large Window Extensions (RFC1323)
    - Adds a Window scale option to multiply Window Size field by 16 bits, thus extending the window to 2^{32}-1.
    - Negotiated at connection setup time.
    - Setting socket buffer size > 64KB may help trigger the RFC1323 negotiation at connection setup.
    - Supported in
      - FreeBSD
      - Linux 2.4 and 2.6
      - Max OS X
      - NetBSD 1.1/1.2
      - Microsoft Windows 2000 and XP

Source: Enabling High Performance Data Transfers, available at http://www.psc.edu/networking/projects/tcptune/

- Mobile and Wireless Networks
  - TCP and Packet Loss
    - TCP assumes packet losses are due to congestion.
    - Packets may be lost due to transmission errors in wireless networks.
    - TCP's congestion control algorithm may be confused by that.
  - Multicast and Broadcast in 802.11 WLAN
    - No link-layer ARQ is performed.
    - Bit-rate is usually limited to lower than the unicast data rate.
  - Dynamic IP Address Assignment
    - In some mobile networks the host's IP address may change from time to time. May also happen in some WLAN and ISP access.
    - Care must be taken to not depend on the IP address being fixed over the lifetime of the application process.

- Two-way Traffic in Multiple Access Network
  - Link Layer with Shared Medium
    - Half-Duplex Ethernet
    - An Ethernet Hub
    - Wireless LAN
  - Bi-directional Protocol Messages
    - TCP is bi-directional due to the ACKs.
    - Could induce more collisions in such networks.
    - Collisions lead to lower throughput and longer delay.
    - Careful use of control messages (e.g., feedbacks or application-layer acknowledgements) can reduce such collisions.

References

- Interaction between HTTP and TCP
  - J. Heidemann, "Performance Interactions Between P-HTTP and TCP Implementations," *Computer Communication Review*, vol.27(2), April 1997, pp.65-73.
  - H. F. Nielsen, "HTTP/1.1 and Nagle's Algorithm," online at http://www.w3.org/Protocols/HTTP/Performance/Nagle/
  - Yuan Pan and Jack Y. B. Lee, "Adaptive Scheduling of Data Transfer in P2P Applications over Asymmetric Networks," *Proc. IEEE ICC 2010*, Cape Town, South Africa, 23-27 May 2010.
- Adaptive Video Streaming
  - Apple HTTP Live Streaming, URL: https://developer.apple.com/resources/http-streaming/
  - I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet", *IEEE Multimedia*, Oct-Nov 2011.