クラウドコンピューティング 基礎論 第8回

創造情報•小林克志

ikob@acm.org

Outline

- Administravia
- Homework review
- User eXperience (UX) and Internet
- TCP

Course Outline

- Administrivia
- Cloud computing
- Service reliability
- Scale-up / Scale-out
- Distributed data stores
- Global services
- Datacenter networkings (1)
- Datacenter networkings (2)
- <u>Network performance</u>
- User experiences
- Network latencies
- Advanced topics

For hands-on exercise : Install two softwares

1. Wireshark : A packet capture and analyzer
    - Just install package from http://www.wireshark.org

2. NS2 : Network simulator.
   Three options are there:
    A. Docker
    - Install docker software and container:
        - https://github.com/ekiourk/docker-ns2
        - X-window server is required. It depends on OS.

    B. Native applicationIf you are using Linux, use this option.
        - Install ns-allinone-2.35 from source because NS-2 package.
          Note that some distribution may not work.
        - X-window, perl, gnuplot are also required.

    C. Virtual Machine (VM)
        - Install Hypervisor Software.
            - Oracle VirtualBoX is free.
              vmware or others are also welcome.
        - Linux VM image with NS2 software will be available from the course Web.

演習に向けて：２つのソフトウェアをインストールする

1.パケットアナライザ wireshark

- http://www.wireshark.org を参考にインストールすること。

2.ネットワークシミュレータ NS2

A. Docker

- Docker をインストールし、コンテナを使用する。
  - https://github.com/ekiourk/docker-ns2
  - X-window の設定は OS に依存する。

B. Native

- ･Linux を利用している場合は、この方法を使う。
  - ns-allinone-2.35 を install すること。
    - X-window, perl, gnuplot なども必要となる。

C. VM で動作

- ハイパーバイザの導入
  - Oracle VirtualBoX であれば無償
    vmware 他でもかまわない
- NS2 付きの Linux 仮想マシンイメージを講義ページで配布する

Today’s Assignment

- You should design a DC network with 16,384 (2^14) servers.

1. Tell the number of switches with the following conditions:
    - Every SW has 64 ethernet ports
    - Every server has 4 ethernet ports
    - Network topology is

      2^16 = 65,536 ports required in total.
    - Link Aggregation Group (LAG) can be used both on SW and server
2. Draw the network topology including all SWs and servers.
3. Note what you take into consideration in your design.

- Submit your answers either in Japanese or in English via the course web.

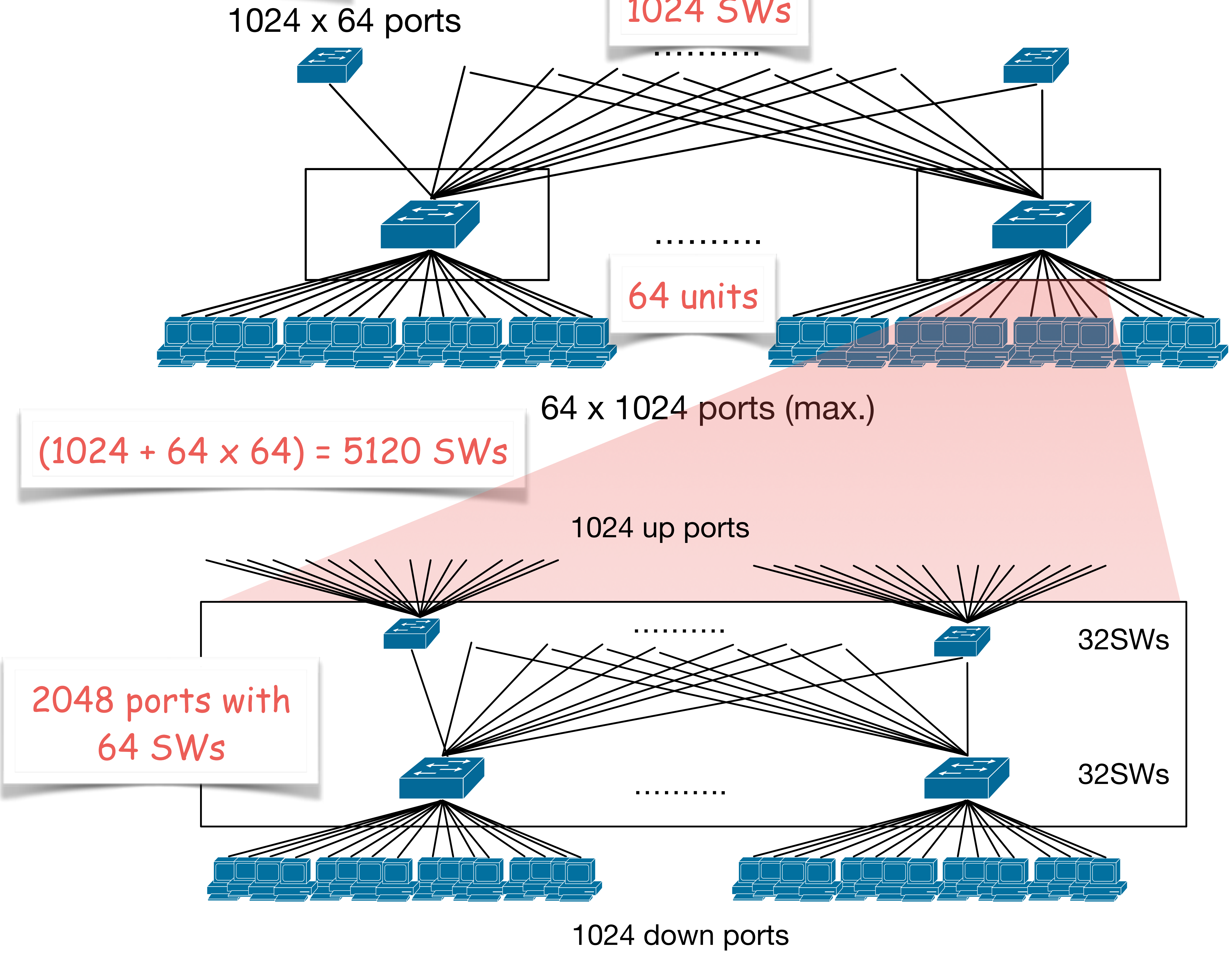
本日の課題

- 16,384 (2^14) 台の server で DC ネットワークを構成したい。

1. 以下の条件で必要な SW 台数を示せ
    - SW 側 Ethernet ポートは 64 ポート
    - Server 側 Ethernet ポートは 4 ポート
    - Folded Clos トポロジ Over subscription なし
    - Link Aggregation Group (LAG) は、server, SW とも利用可能
2. サーバおよびSW の接続トポロジを図示せよ
3. 上のデザインで考慮した点を示せ

- 講義 Web から提出すること

Three tiers Clos network



1024 x 64 ports

1024 SWs

64 units

64 x 1024 ports (max.)

(1024 + 64 x 64) = 5120 SWs

1024 up ports

32SWs

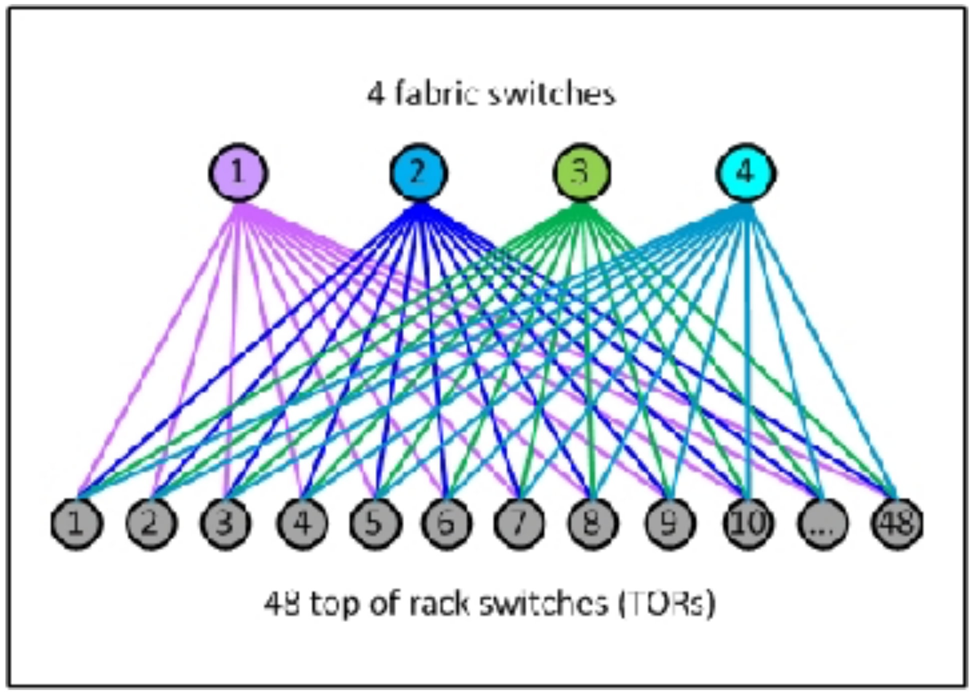
2048 ports with 64 SWs

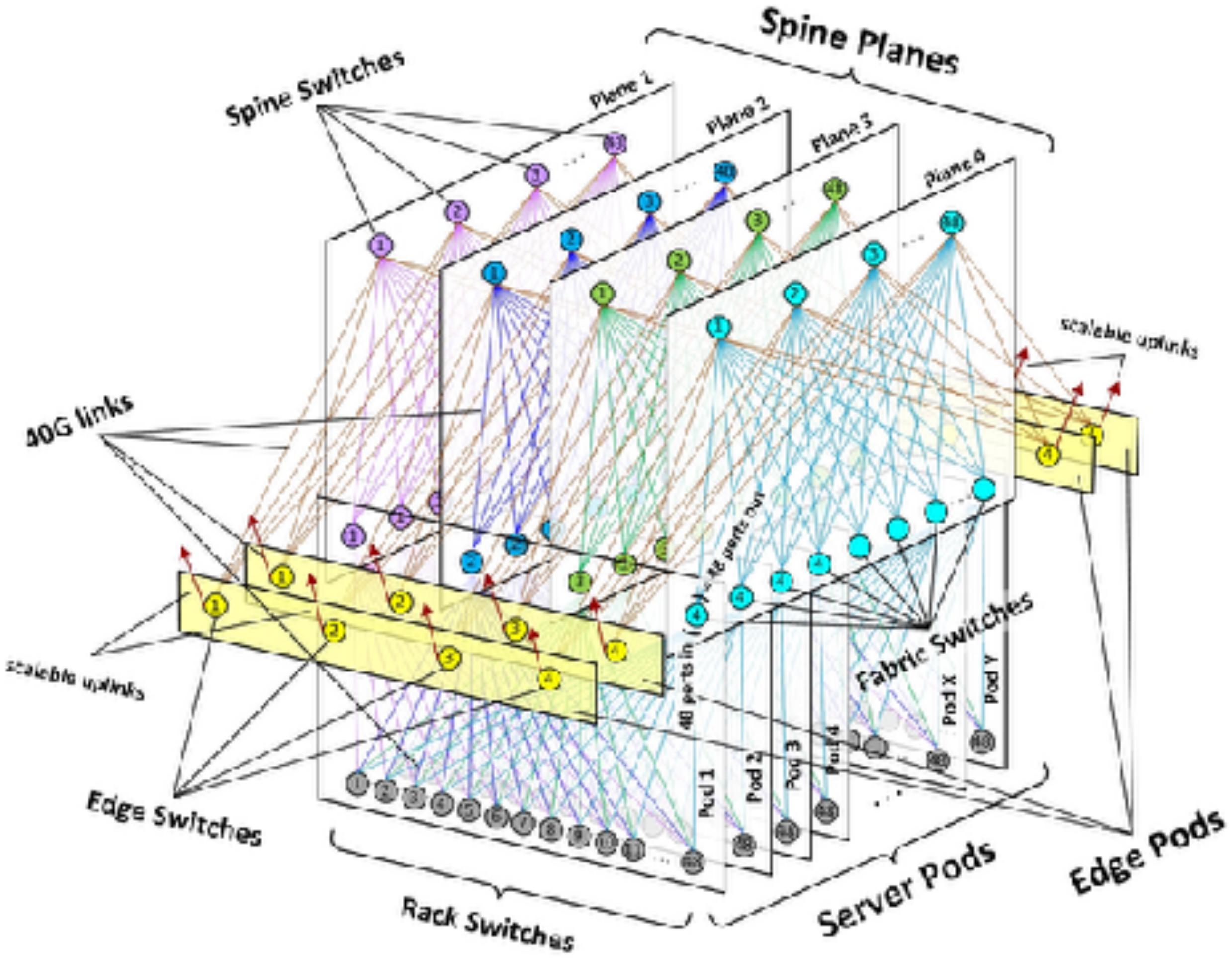
32SWs

1024 down ports

Example: Facebook DC network

Applicatio
Presentati
Session
Transport
Network
Datalink
Physical



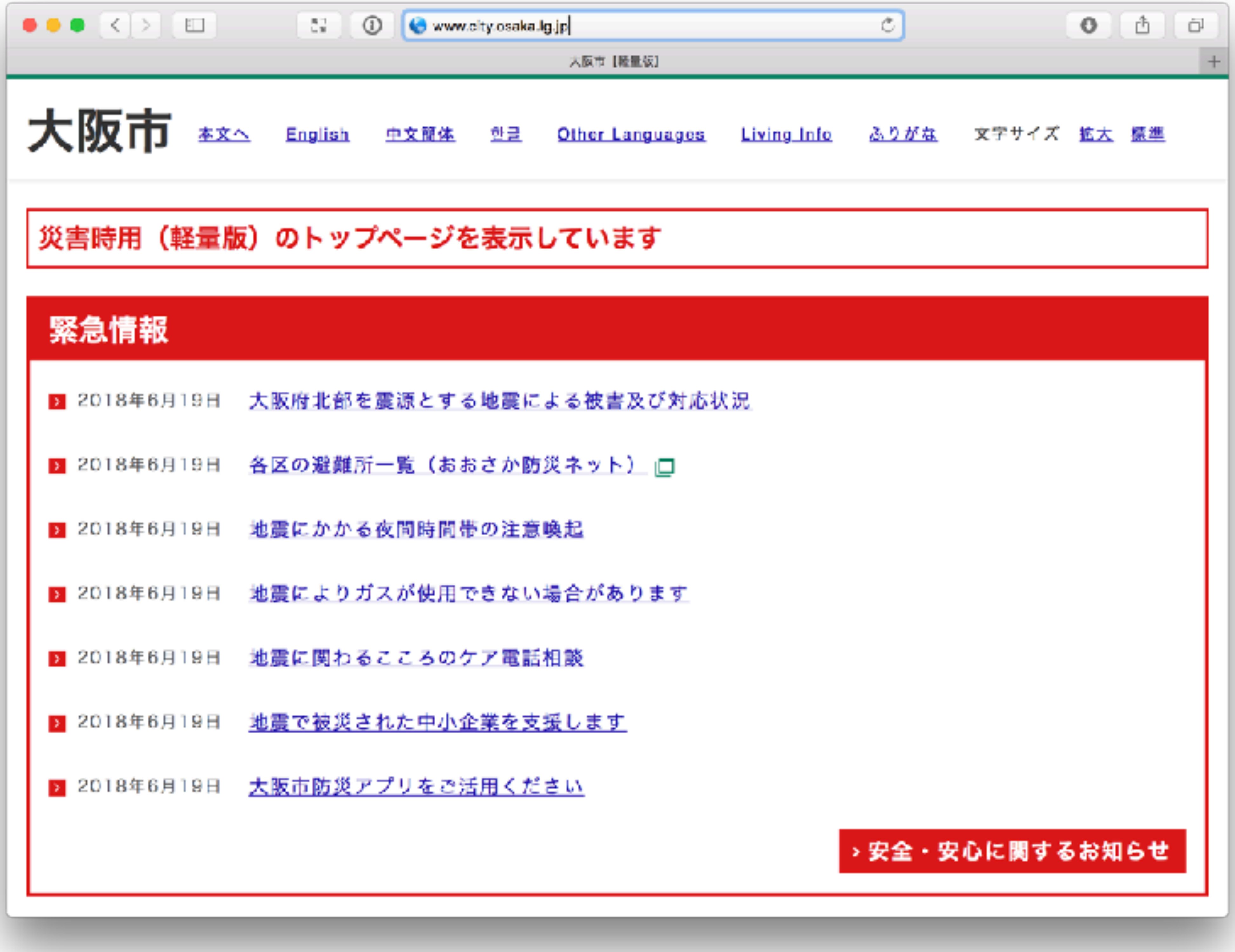


nter fabric, the next-generation Facebook data center network”, Nov. 2014

Outline

- Administravia
- Homework review
- User eXperience (UX) and Internet
- TCP

Have you seen such page?



大阪市 本文へ English 中文簡体 한글 Other Languages Living Info ふりがな 文字サイズ 拡大 標準

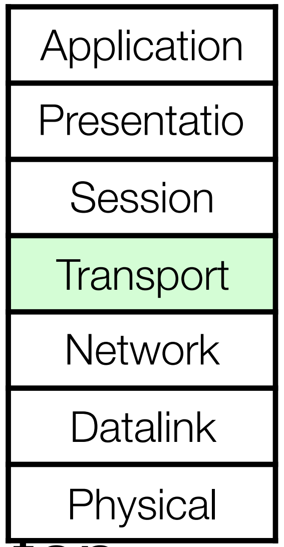
災害時用（軽量版）のトップページを表示しています

緊急情報

- 2018年6月19日 大阪府北部を震源とする地震による被害及び対応状況
- 2018年6月19日 各区の避難所一覧（おおさか防災ネット）
- 2018年6月19日 地震にかかる夜間時間帯の注意喚起
- 2018年6月19日 地震によりガスが使用できない場合があります
- 2018年6月19日 地震に関わるこころのケア電話相談
- 2018年6月19日 地震で被災された中小企業を支援します
- 2018年6月19日 大阪市防災アプリをご活用ください

› 安全・安心に関するお知らせ

Transmission Control Protocol (TCP)



Application

Presentatio

Session

Transport

Network

Datalink

Physical

- A transport protocol that provides end-to-end connections on the top of packet switched networks.
  TCP provides :

□ byte stream type

□ reliable

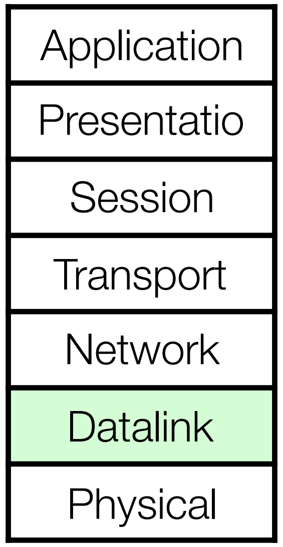
□ flow-controlled

□ multiplex

□ bi-directional

□ congestion controlled

IP Header

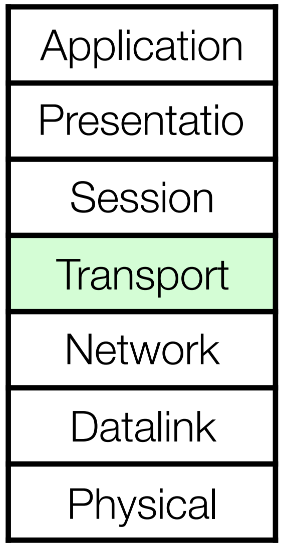


Application
Presentatio
Session
Transport
Network
Datalink
Physical

```
IPv4:
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Version|  IHL  |Type of Service|          Total Length         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Identification        |Flags|      Fragment Offset    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Time to Live |    Protocol   |         Header Checksum       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Source Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Destination Address                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Options                    |    Padding    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
IPv6:
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Version| Traffic Class |           Flow Label                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Payload Length        |  Next Header  |   Hop Limit   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                                                               +
   |                                                               |
   +                         Source Address                        +
   |                                                               |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                                                               +
   |                                                               |
   +                      Destination Address                      +
   |                                                               |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TCP and User Datagram Protocol (UDP) header



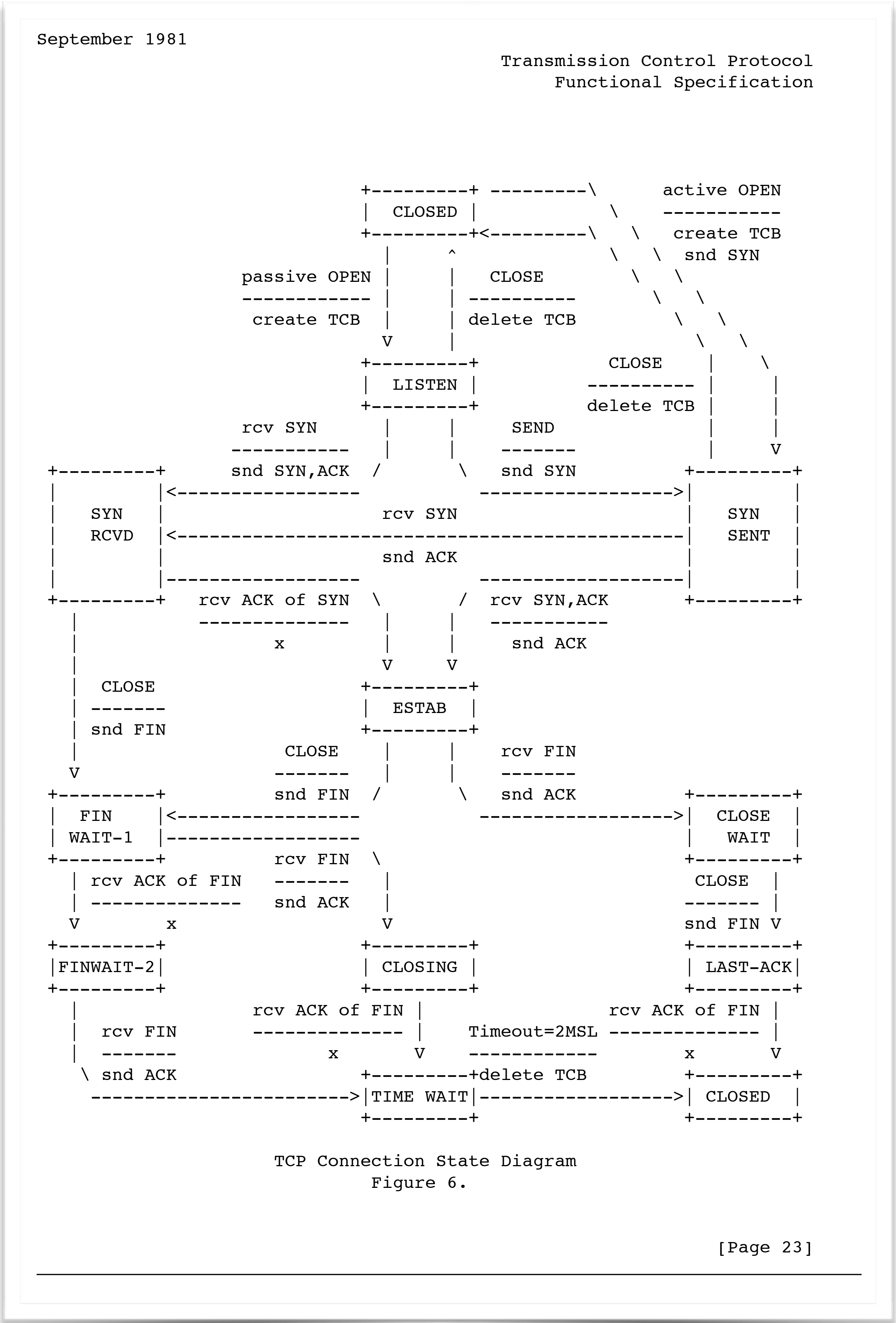
TCP Header:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

UDP Header:

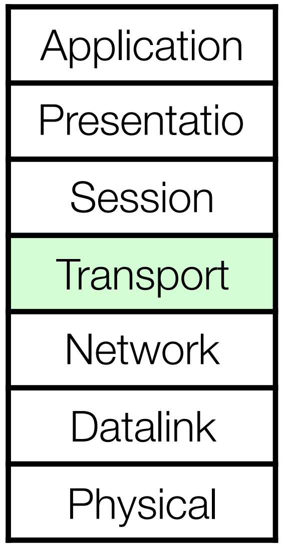
```
 0      7 8     15 16    23 24    31
+--------+--------+--------+--------+
|     Source      |   Destination   |
|      Port       |      Port       |
+--------+--------+--------+--------+
|                 |                 |
|     Length      |    Checksum     |
+--------+--------+--------+--------+
|
|          data octets ...
+---------------- ...
```

J. Postel, ”RFC 793: Transmission control protocol”, 1981
J. Postel, ”RFC 768: User Datagram Protocol”, 1980



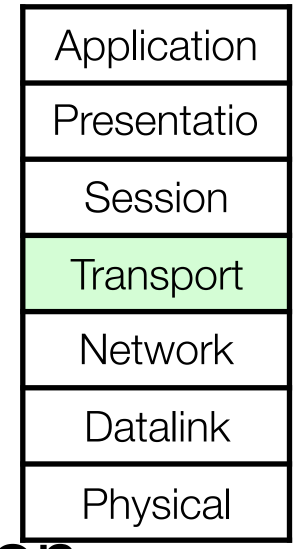
```
                              +---------+ ---------\      active OPEN
                              |  CLOSED |            \    -----------
                              +---------+<---------\   \   create TCB
                                |     ^              \   \  snd SYN
                   passive OPEN |     |   CLOSE        \   \
                   ------------ |     | ----------       \   \
                    create TCB  |     | delete TCB         \   \
                                V     |                      \   \
                              +---------+            CLOSE    |    \
                              |  LISTEN |          ---------- |     |
                              +---------+          delete TCB |     |
                   rcv SYN      |     |     SEND              |     |
                  -----------   |     |    -------            |     V
 +---------+      snd SYN,ACK  /       \   snd SYN          +---------+
 |         |<-----------------           ------------------>|         |
 |   SYN   |                    rcv SYN                     |   SYN   |
 |   RCVD  |<-----------------------------------------------|   SENT  |
 |         |                    snd ACK                     |         |
 |         |------------------           -------------------|         |
 +---------+   rcv ACK of SYN  \       /  rcv SYN,ACK       +---------+
   |           --------------   |     |   -----------
   |                  x         |     |     snd ACK
   |                            V     V
   |  CLOSE                   +---------+
   | -------                  |  ESTAB  |
   | snd FIN                  +---------+
   |                   CLOSE    |     |    rcv FIN
   V                  -------   |     |    -------
 +---------+          snd FIN  /       \   snd ACK          +---------+
 |  FIN    |<-----------------           ------------------>|  CLOSE  |
 | WAIT-1  |------------------                              |   WAIT  |
 +---------+          rcv FIN  \                            +---------+
   | rcv ACK of FIN   -------   |                            CLOSE  |
   | --------------   snd ACK   |                           ------- |
   V        x                   V                           snd FIN V
 +---------+                  +---------+                   +---------+
 |FINWAIT-2|                  | CLOSING |                   | LAST-ACK|
 +---------+                  +---------+                   +---------+
   |                rcv ACK of FIN |                 rcv ACK of FIN |
   |  rcv FIN       -------------- |    Timeout=2MSL -------------- |
   |  -------              x       V    ------------        x       V
    \ snd ACK                 +---------+delete TCB         +---------+
     ------------------------>|TIME WAIT|------------------>| CLOSED  |
                              +---------+                   +---------+
```

TCP Connection State Diagram
Figure 6.



J. Postel, "RFC 793: Transmission control protocol", 1981

Transmission Control Protocol (TCP)


Application
Presentatio
Session
Transport
Network
Datalink
Physical

A transport protocol that provides end-to-end connections on the top of packet switched networks. TCP provides :

□ byte stream type

□ reliable

□ flow-controlled

□ **multiplex**

□ bi-directional

□ congestion controlled

Five tuple : flow/connectio identification

Application
Presentatio
Session
Transport
Network
Datalink
Physical

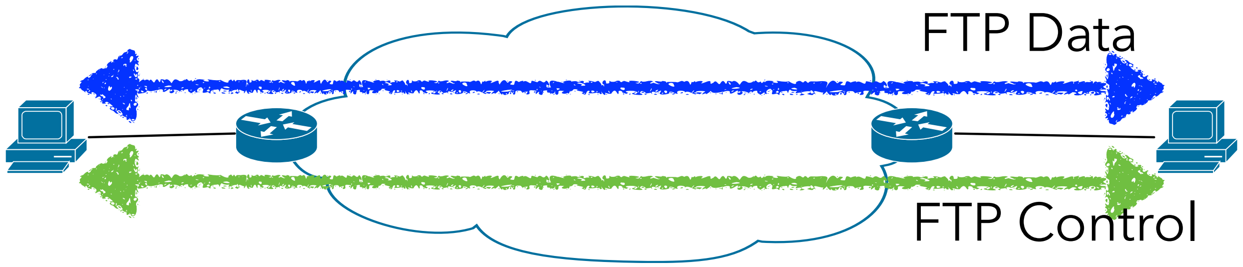
1. Source address

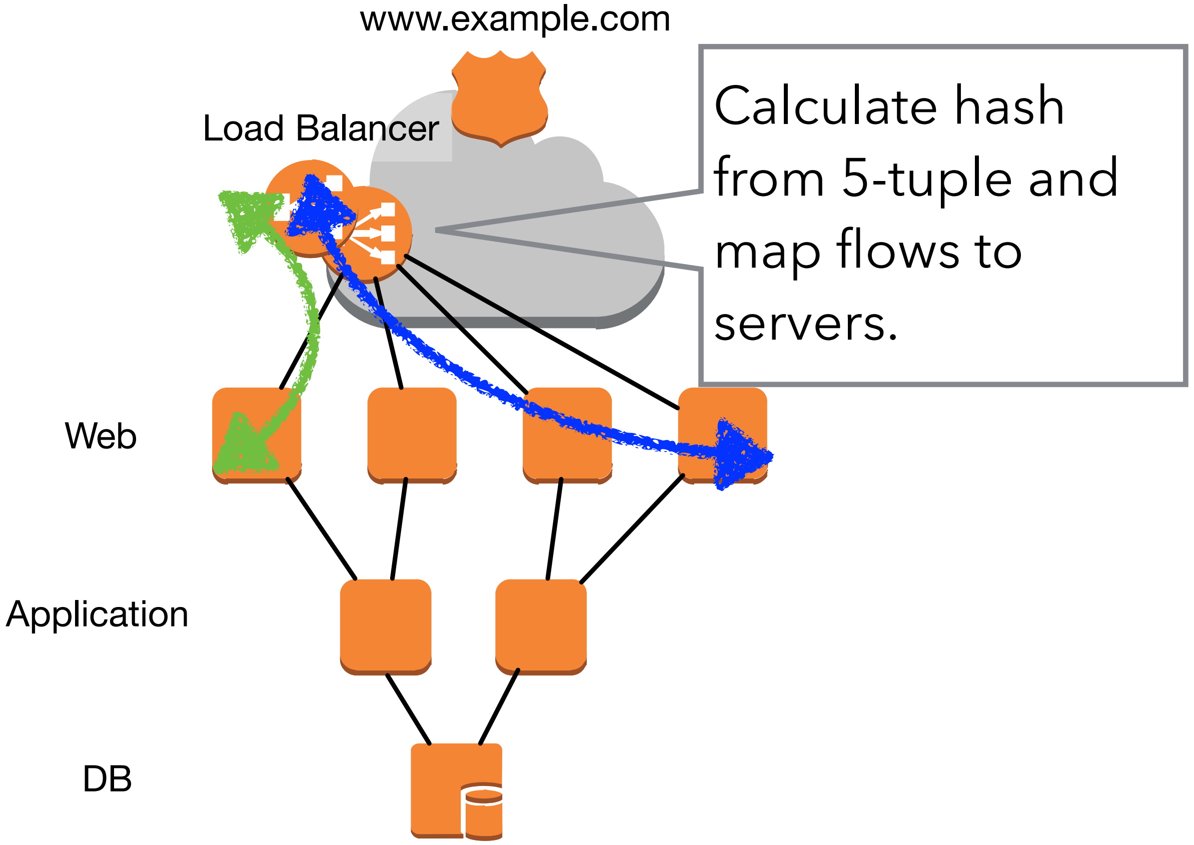
2. Destination address

3. Upper (Transport) Protocol {UDP, TCP}

4. Source port

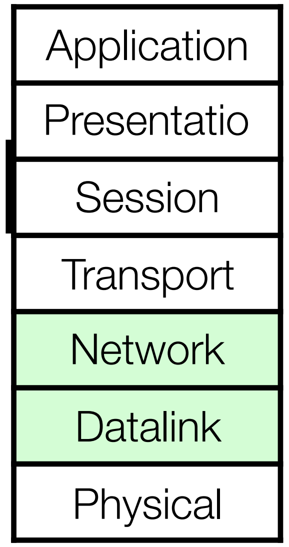
5. Destination port





Calculate hash from 5-tuple and map flows to servers.

Five tuple : flow/connection identification(cont’d)



IP header:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Version|  IHL  |Type of Service|          Total Length         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Identification        |Flags|      Fragment Offset    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Time to Live |    Protocol   |         Header Checksum       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Source Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Destination Address                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Options                    |    Padding    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

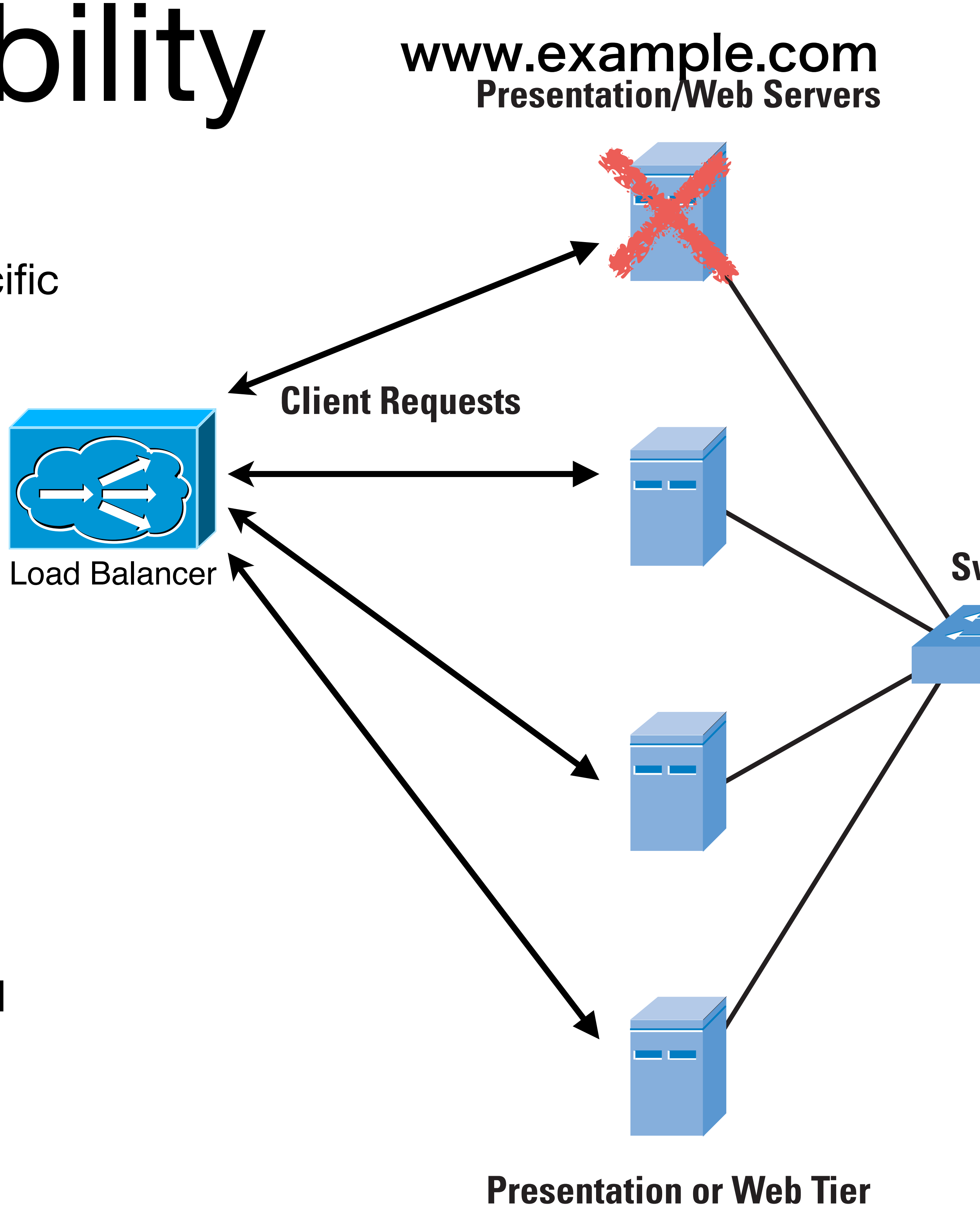
TCP header:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Source Port          |       Destination Port        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Sequence Number                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Acknowledgment Number                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Data |           |U|A|P|R|S|F|                               |
   | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
   |       |           |G|K|H|T|N|N|                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Checksum            |         Urgent Pointer        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Options                    |    Padding    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             data                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

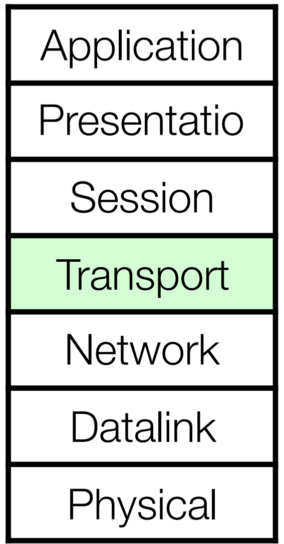
J. Postel, “RFC791: Internet Protocol”, 1981

Load balancing policy and availability

- Policy
  - Hash: based IP address or other client-specific info.
  - Least connections: assign most least connection server
  - Round robin: new connection to the next server with RR
  - Weighted ver. of above: considering server condition both static and dynamic
- Health check
  - monitoring servers and update list in several levels (layers), e.g., ICMP(ping), application layer polling, server load, manual…



Transmission Control Protocol (TCP)



A transport protocol that provides end-to-end connections on the top of packet switched networks. TCP provides :

□ byte stream type

□ reliable

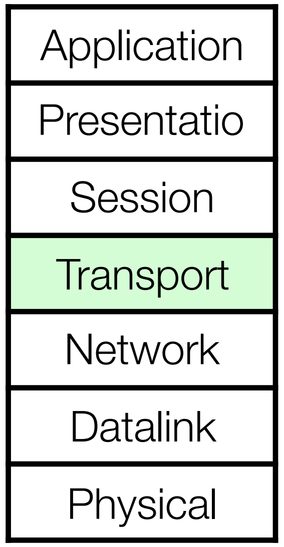
□ flow-controlled

☑ multiplex

□ **bi-directional**

□ congestion controlled

TCP header



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

J. Postel, ”RFC 793: Transmission control protocol”, 1981

Application
Presentatio
Session
Transport
Network
Datalink
Physical

Transmission Control Protocol (TCP)

A transport protocol that provides end-to-end connections on the top of packet switched networks. TCP provides :

□ **byte stream type**

□ **reliable**

□ flow-controlled

☑ multiplex

☑ bi-directional

□ congestion controlled

Reliability on packet switched networks

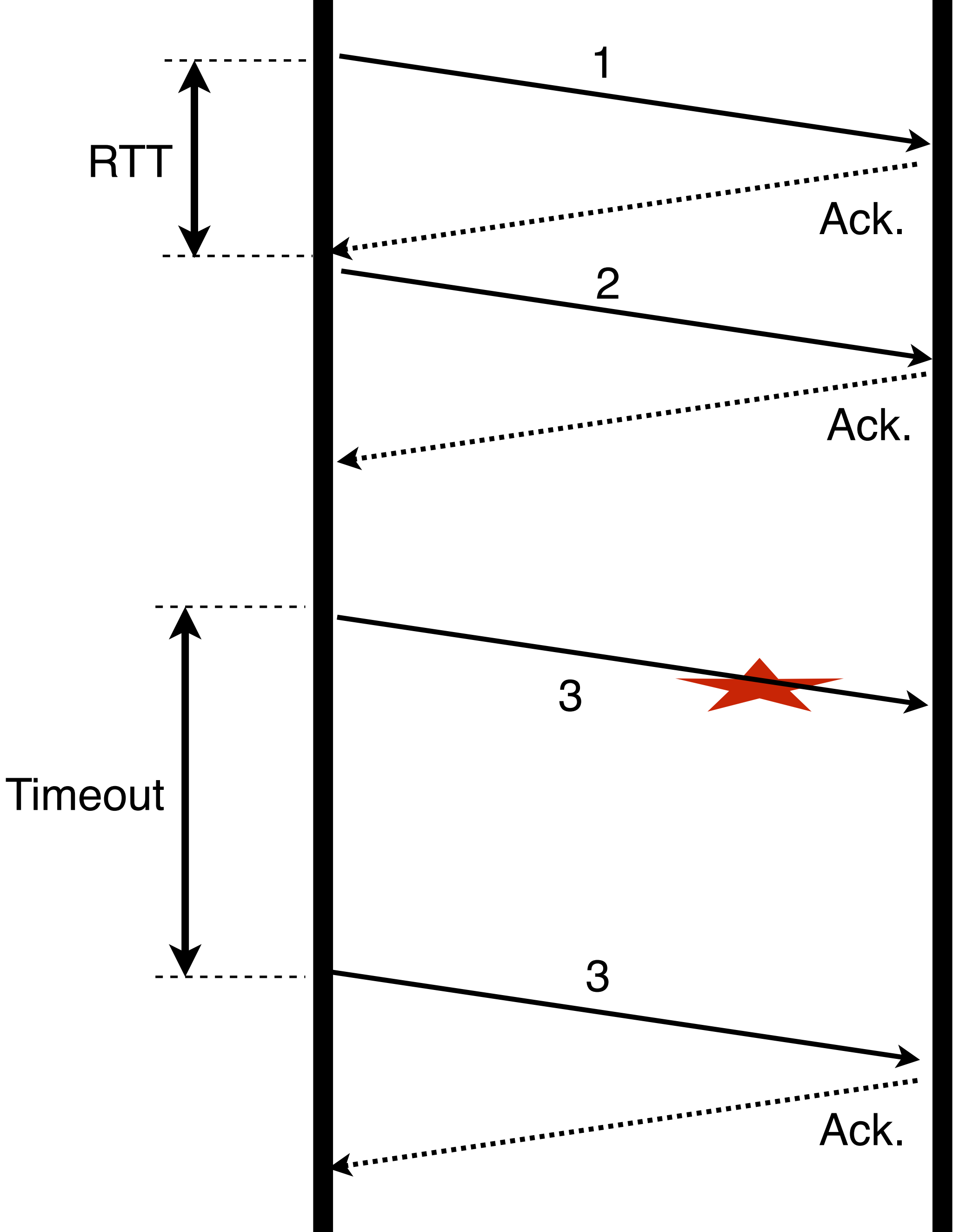
- Requirements:
  - Ordered sequence of segments
  - Resiliency against segment losses.
- Provided with Automatic Repeat reQuest (ARQ)
  - Acknowledge based approaches.
    Throughputs are bounded by Round Trip Time (RTT).
    - Stop-and-Wait
    - Sliding window
      - Go-back-N, Selective repeat

G. Fairhurst and L. Wood, “Advice to link designers on link Automatic Repeat reQuest (ARQ)” RFC3366(2002)

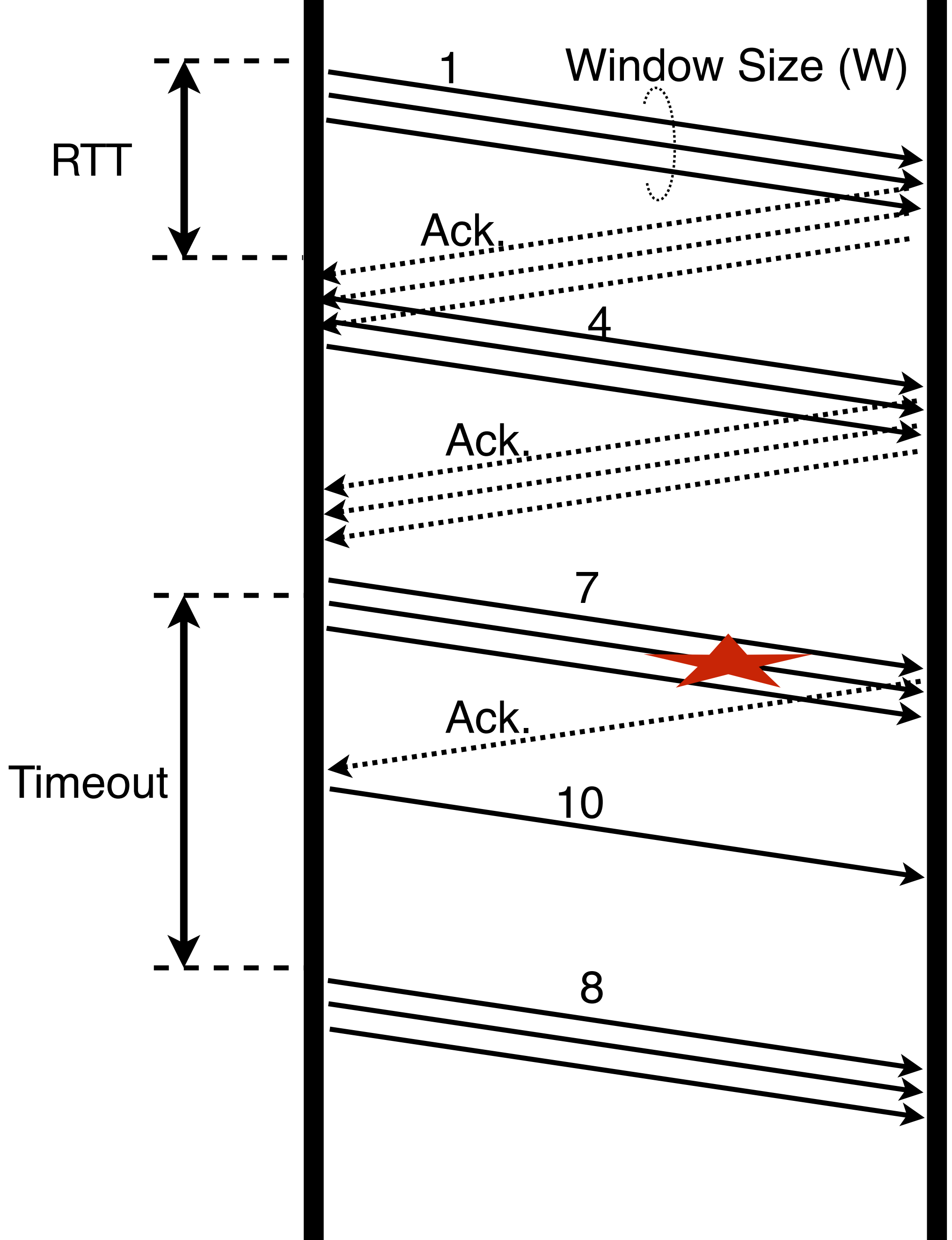
Automatic Repeat reQuest (ARQ)

- Application
- Presentatio
- Session
- Transport
- Network
- Datalink
- Physical

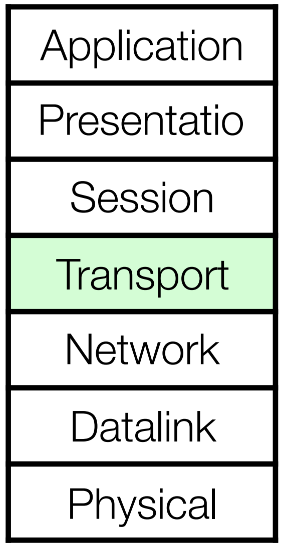
Stop-and-Wait
BW < Psize / RTT



Go-back-N
BW < W x Psize / RTT



TCP header

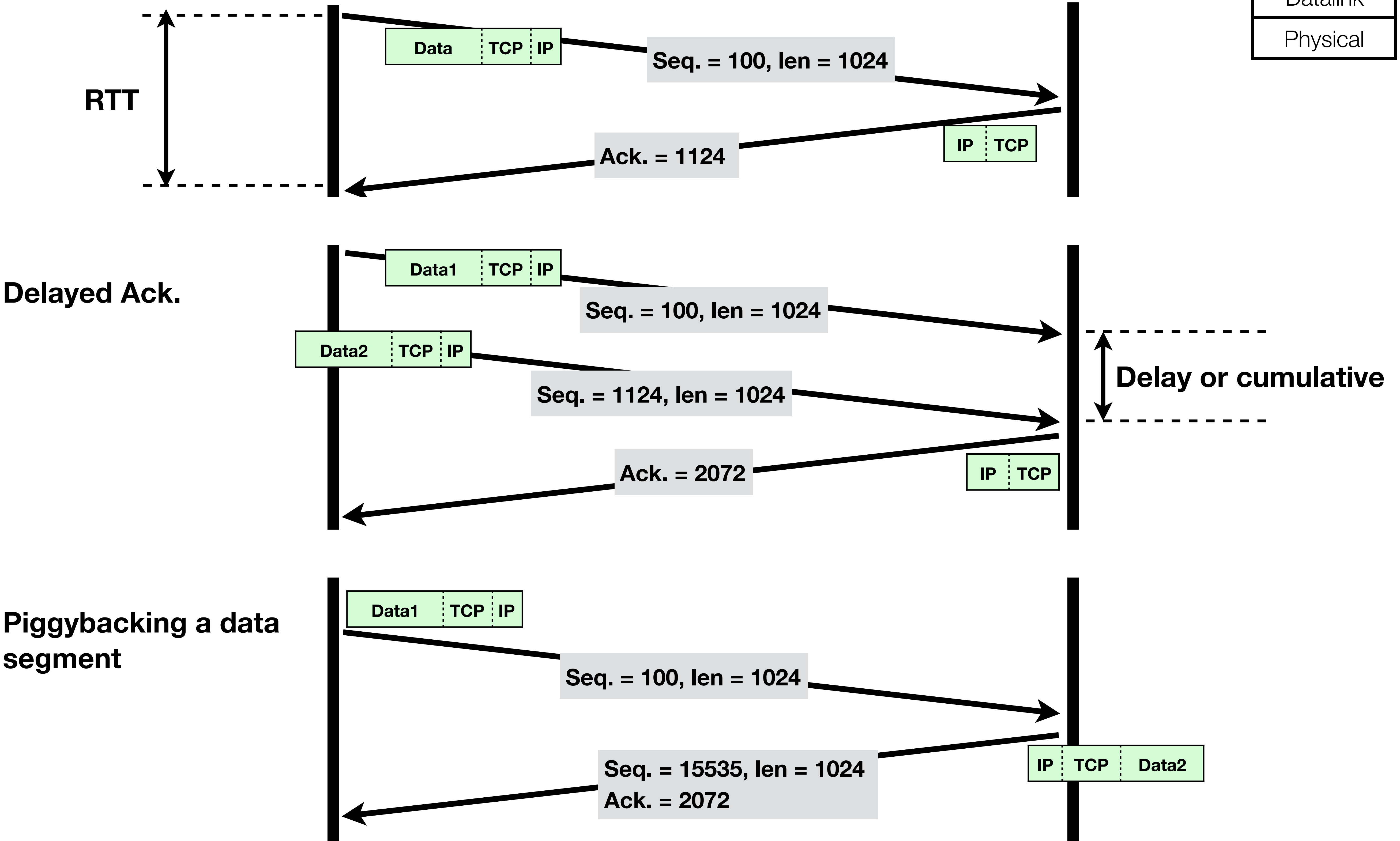


```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

J. Postel, ”RFC 793: Transmission control protocol”, 1981

- Application
- Presentatio
- Session
- Transport
- Network
- Datalink
- Physical

TCP acknowledge

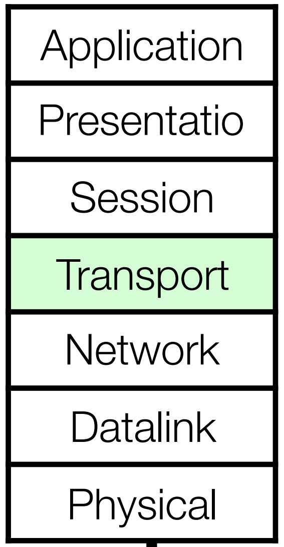


RTT

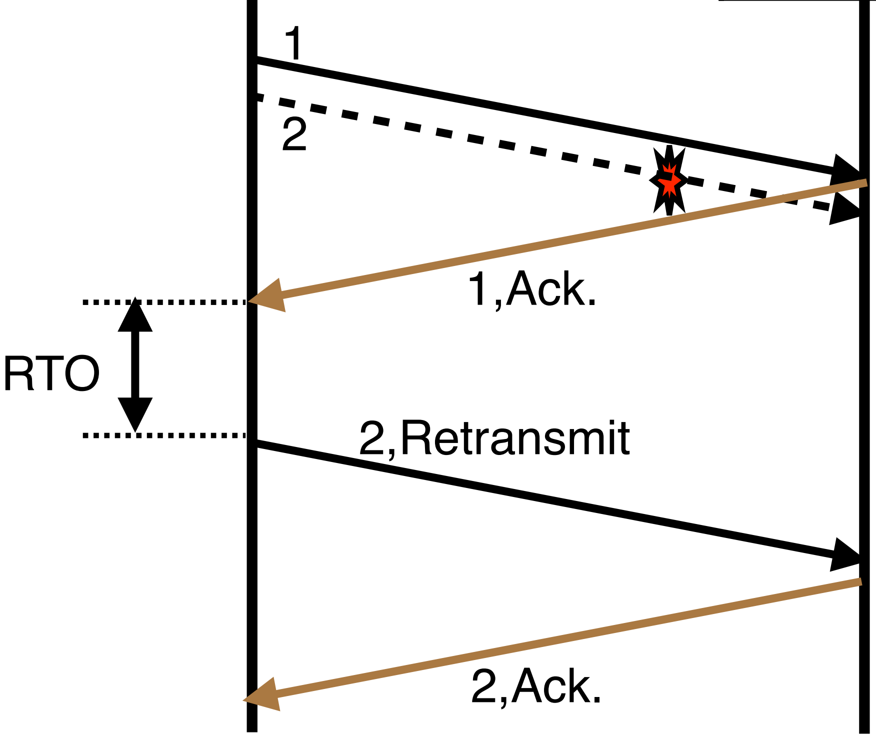
Delayed Ack.

Piggybacking a data segment

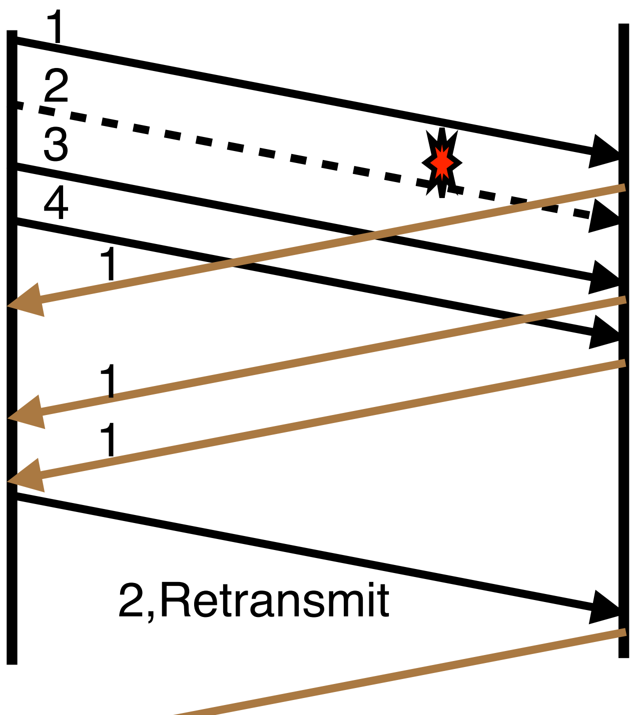
Loss detection and retransmission in TCP



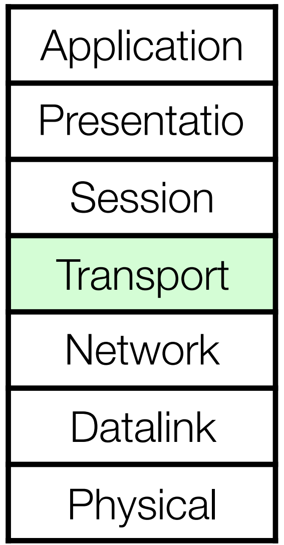
- Retransmit Time Out (RTO)
  When Data Ack. is not received until RTO*,
  - retransmit the segment that is regarded as loss



- Fast Retransmit / FastRecovery
  - If receiving three same ack., then the consecutive packet is considered as a loss.



Transmission Control Protocol (TCP)



- A transport protocol that provides **end-to-end connections** on the top of packet switched networks.
  TCP provides :

☑ byte stream type

☑ reliable

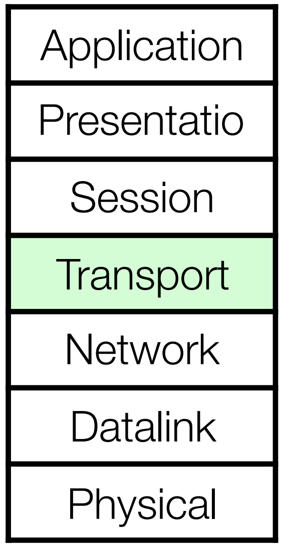
□ flow-controlled

□ multiplex

□ bi-directional

□ congestion controlled

TCP header



Application

Presentatio

Session

Transport

Network

Datalink

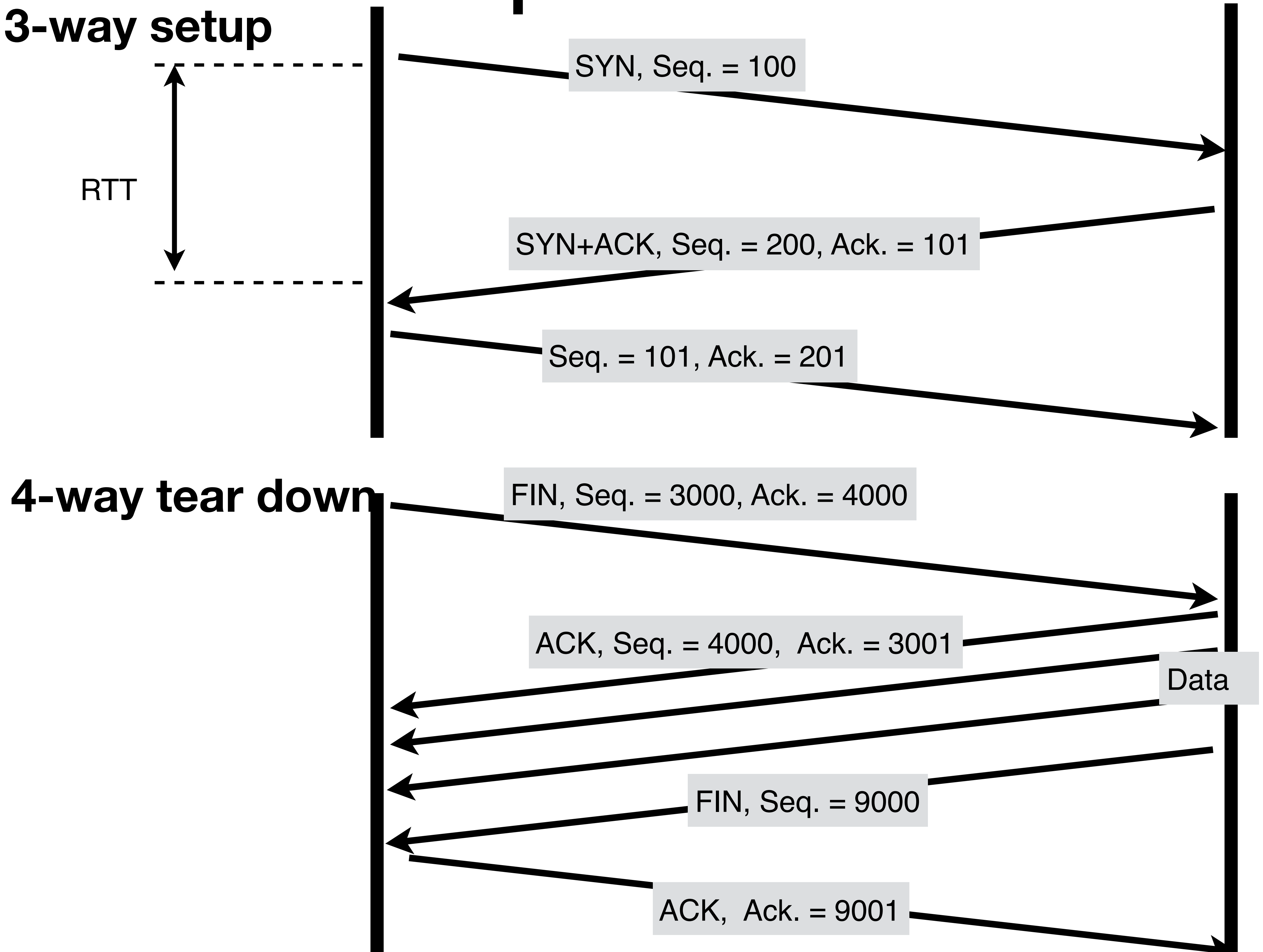
Physical

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

J. Postel, ”RFC 793: Transmission control protocol”, 1981

Application
Presentatio
Session
Transport
Network
Datalink
Physical

TCP connection set-up and tear-down



3-way setup

RTT

SYN, Seq. = 100

SYN+ACK, Seq. = 200, Ack. = 101

Seq. = 101, Ack. = 201

4-way tear down

FIN, Seq. = 3000, Ack. = 4000

ACK, Seq. = 4000, Ack. = 3001

Data

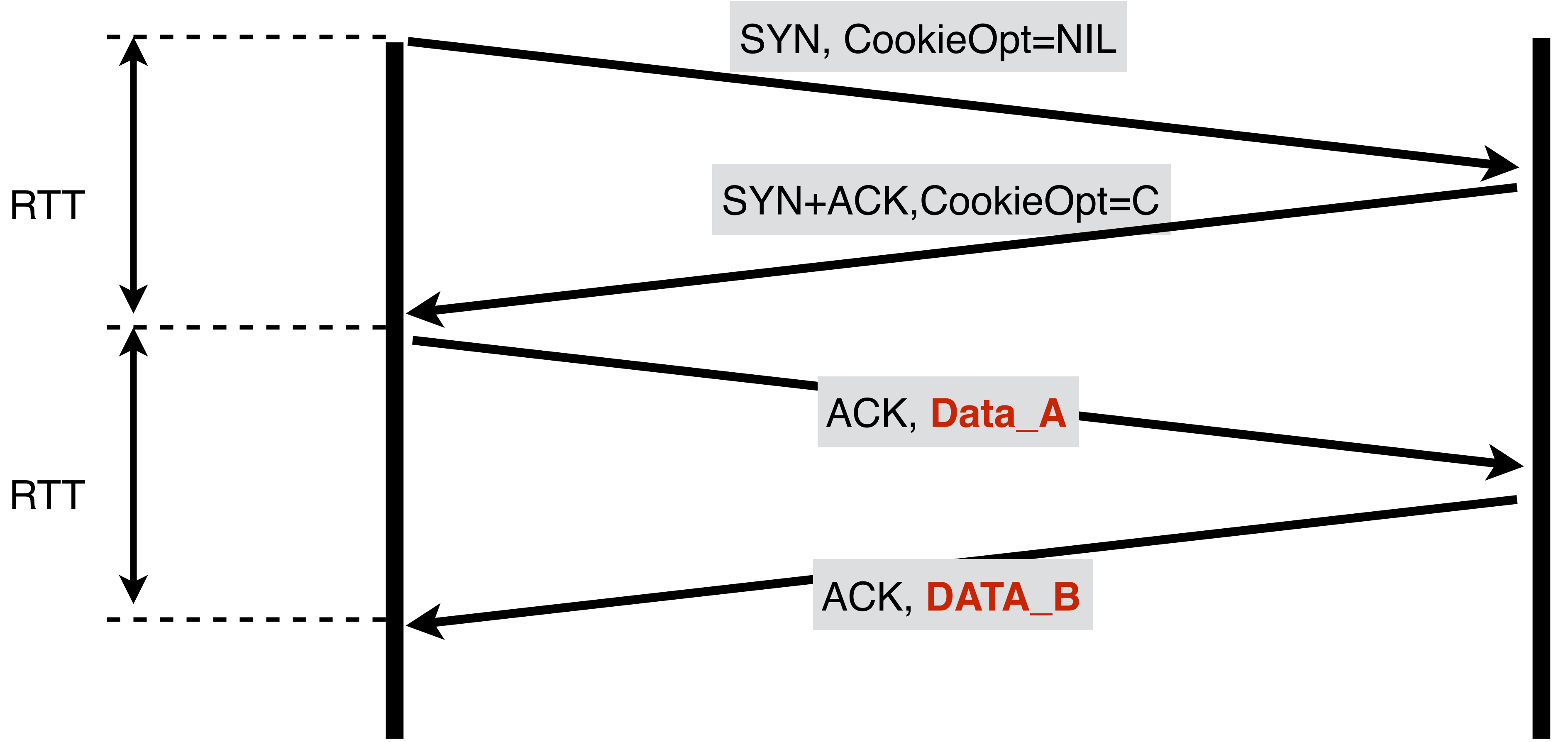
FIN, Seq. = 9000

ACK, Ack. = 9001

Application
Presentatio
Session
Transport
Network
Datalink
Physical

TCP Fast Open (RFC7413

1st connection



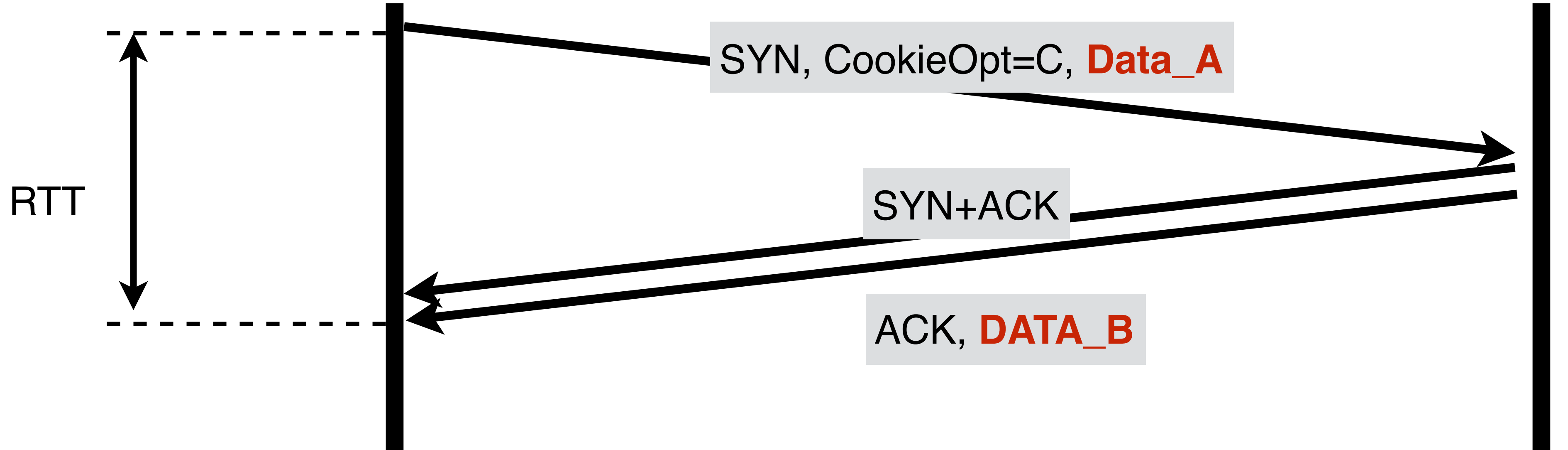
SYN, CookieOpt=NIL

SYN+ACK,CookieOpt=C

ACK, **Data_A**

ACK, **DATA_B**

2nd connection



SYN, CookieOpt=C, **Data_A**

SYN+ACK

ACK, **DATA_B**

Today’s Assignment

- In the class, TCP Fast Open (RFC7413) was introduced as an approach to reduce the latency of TCP connection set-up. Another option T/TCP - TCP extensions for Transactions (RFC1644) which shares the same goal had been standardized for 20 years. In addition, T/TCP reduces the TCP set-up latency not only from the second or later connections, but from the first one. However, now T/TCP standard has been obsoleted.

- Read RFC7413, RFC1644 and related documents. Discuss why T/TCP has been obsoleted.

- Submit your answers either in Japanese or in English via the course web.

本日の課題

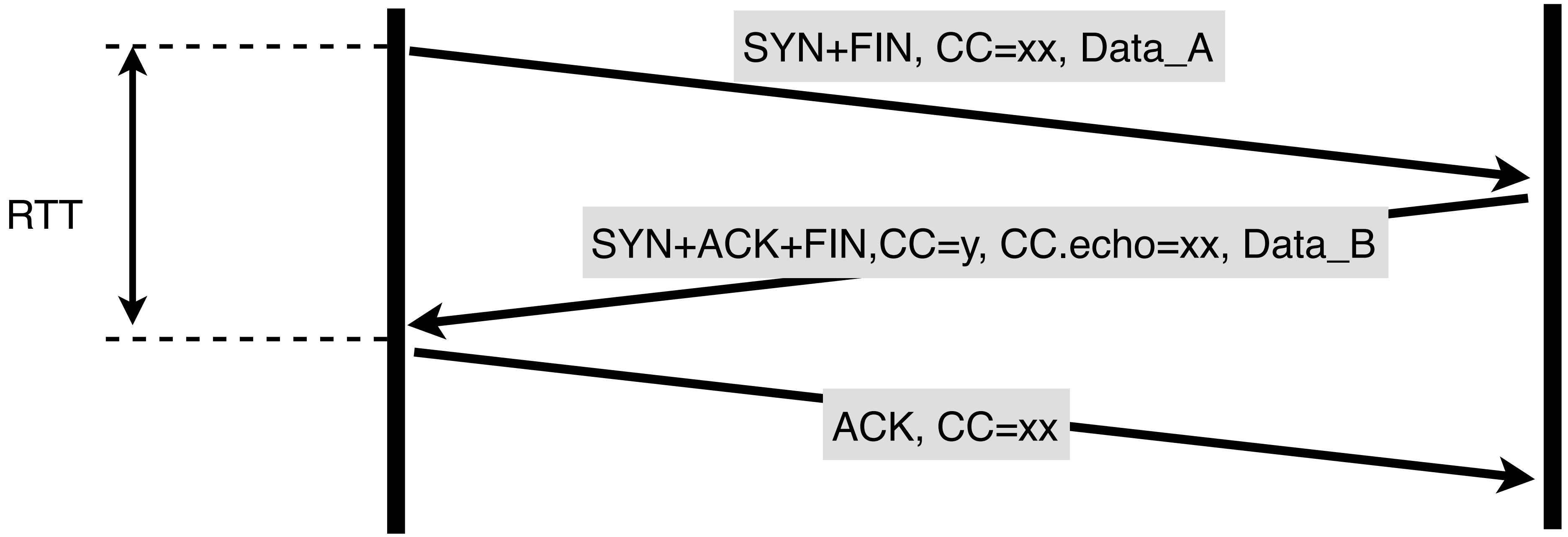
- 講義では、TCP コネクションセットアップの遅延を抑える手法として TCP Fast Open (RFC7413) を取り上げた。TCP Fast Open と同じ目的で　- TCP extensions for Transactions (RFC1644) が 20 年前に標準化されている。さらに、T/TCP ではセットアップ遅延を 2 つめ以降のコネクションだけではなく、最初のコネクションから抑えることができる。しかしながら、T/TCP 標準は廃止された。

- RFC7413, RFC1644 および関連文書を読み、T/TCP が廃止された理由を考察せよ。

- 講義 Web ページから回答すること。

T/TCP - TCP Extensions fo Transactions (RFC1644)

Application
Presentatio
Session
Transport
Network
Datalink
Physical



SYN+FIN, CC=xx, Data_A

RTT

SYN+ACK+FIN,CC=y, CC.echo=xx, Data_B

ACK, CC=xx

Application
Presentatio
Session
Transport
Network
Datalink
Physical

Transmission Control Protocol (TCP)

A transport protocol that provides end-to-end connections on the top of packet switched networks. TCP provides :

☑ byte stream type

☑ reliable

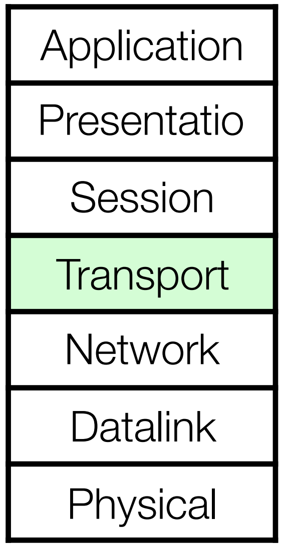
□ **<u>flow-controlled</u>**

☑ multiplex

☑ bi-directional

□ congestion controlled

TCP header



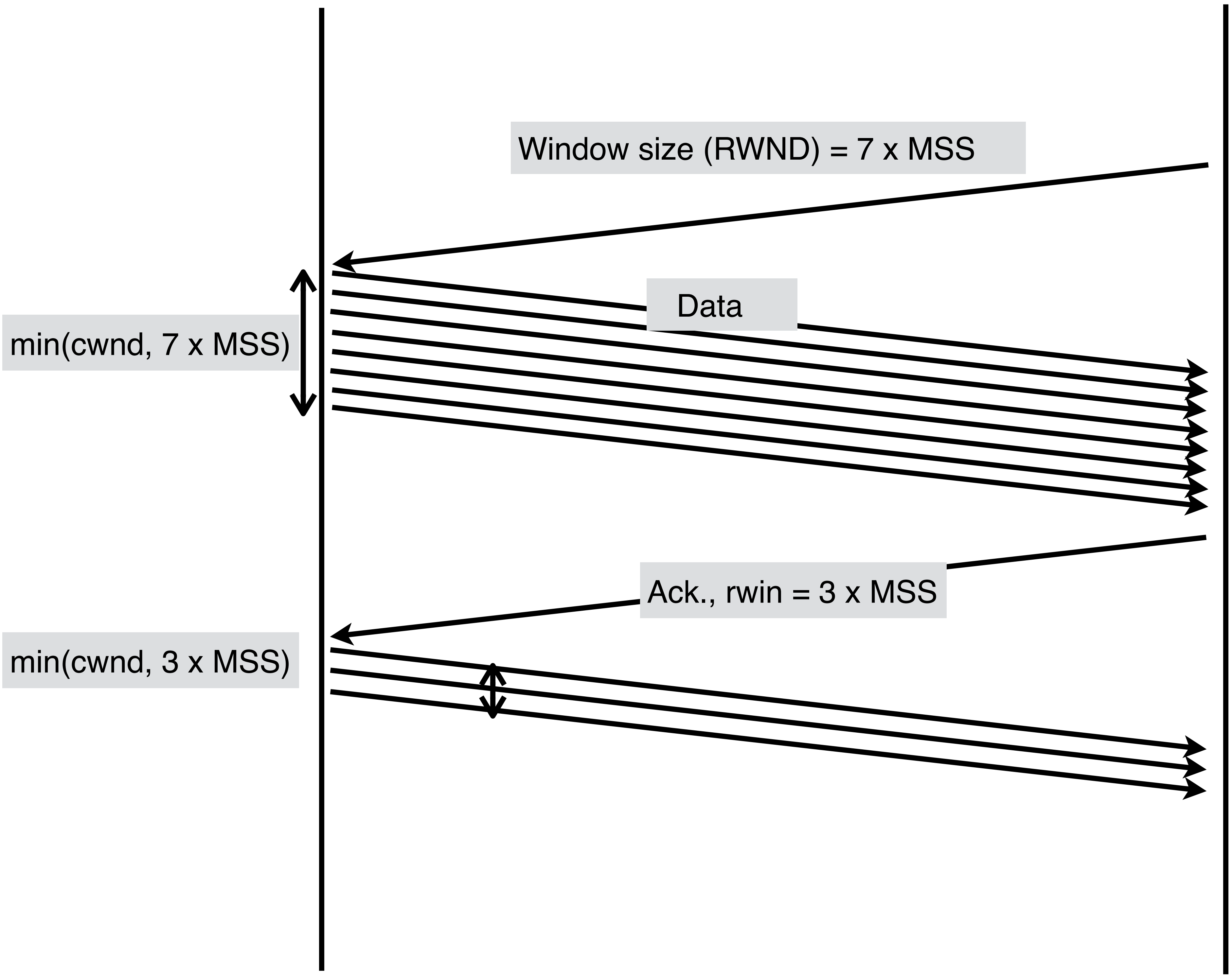
Application
Presentatio
Session
Transport
Network
Datalink
Physical

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

J. Postel, ”RFC 793: Transmission control protocol”, 1981

Application
Presentatio
Session
Transport
Network
Datalink
Physical

Flow control in TCP



Window size (RWND) = 7 x MSS

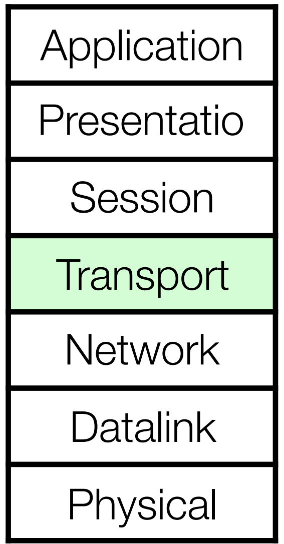
Data

min(cwnd, 7 x MSS)

Ack., rwin = 3 x MSS

min(cwnd, 3 x MSS)

Transmission Control Protocol (TCP)


Application
Presentatio
Session
Transport
Network
Datalink
Physical

A transport protocol that provides end-to-end connections on the top of packet switched networks. TCP provides :

☑ byte stream type

☑ reliable

☑ flow-controlled

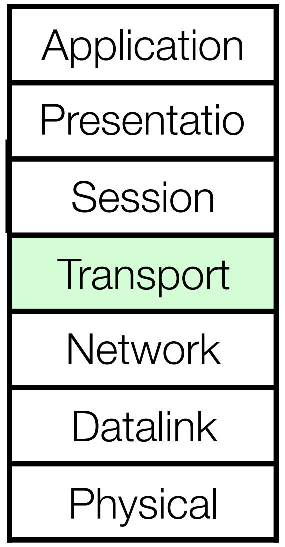
☑ multiplex

☑ bi-directional

□ **congestion controlled**

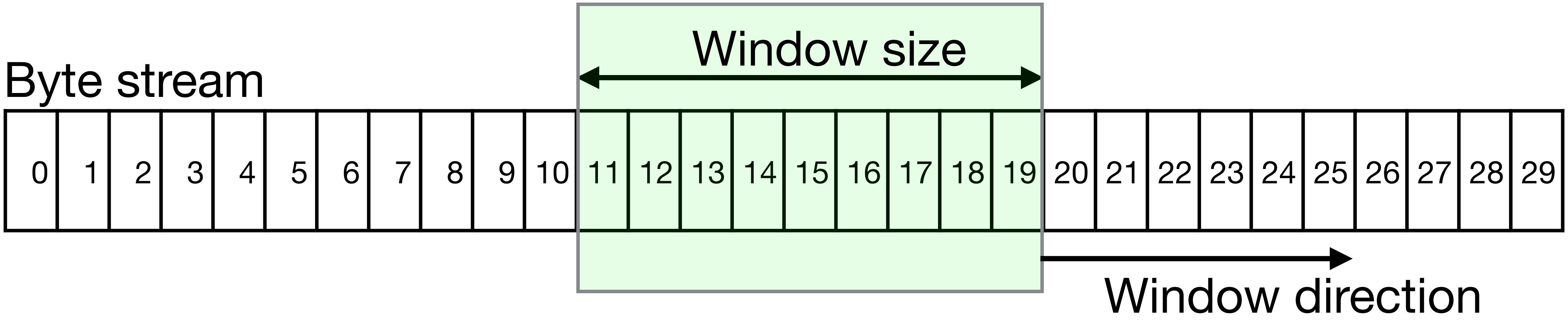
TCP Congestion control implementations/algorithm

- TCP NewReno
  - So-called standard TCP
- CUBIC TCP (Linux), Compound TCP (Windows)
  - Improve throughput for Large Bandwidth Delay Product (BDP) with aggressive approaches.
  - Almost compatible with TCP NewReno on small BDP.
- TCB Bottleneck Bandwidth and Round-trip propagation time (BBR) (by Google)



Sliding Window Congestio Control

- Senders move window, when ack. received.
- If a sender is aware congestion, the sender shrinks window as a result the sending rate decreases.
- If a sender is aware more capacity to the receiver, the sender expand the window.
- Bandwidth throughput : window size x RTT



TCP NewReno Congestion Control

- Packet loss is regarded as a congestion signal. cwnd increases until packet loss.
- Two cwnd control phases:
  - Slow-Start :
    cwnd = cwnd + MSS (per ack.) when cwnd < ssthresh.

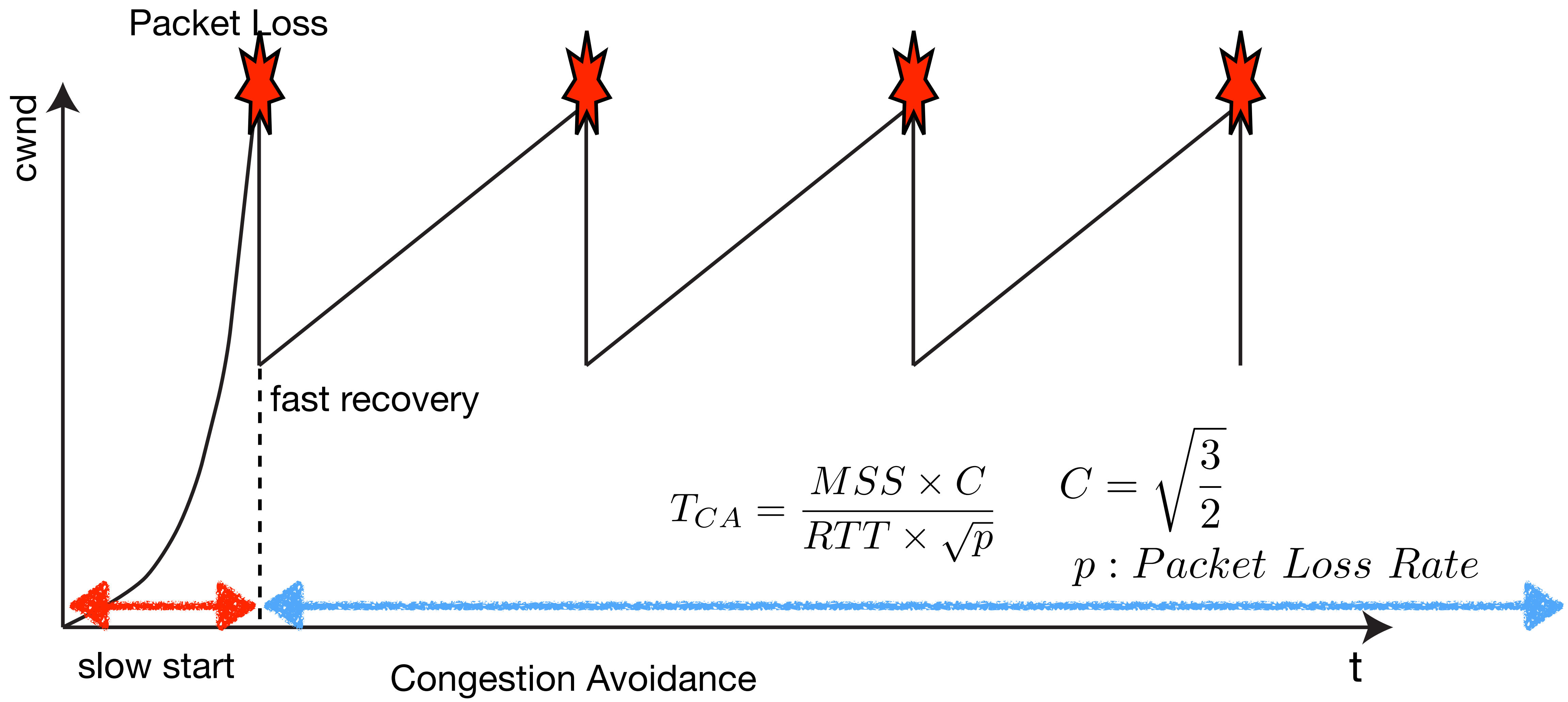
    In fact, not slow but exponential cwnd growth.
  - Congestion Avoidance :
    cwnd = cwnd + MSS /cwnd (per Ack.) when cwnd < ssthresh.

    Additive Increase/Multiplicative Decrease(AIMD))

cwnd: congestion window size
ssthresh: slow-start threshold (= max_cwnd, when connection start)
MSS: Maximum Segment Size (Typ. MSS : 1460 bytes)

TCP window behavior on NewReno

Application
Presentatio
Session
Transport
Network
Datalink
Physical

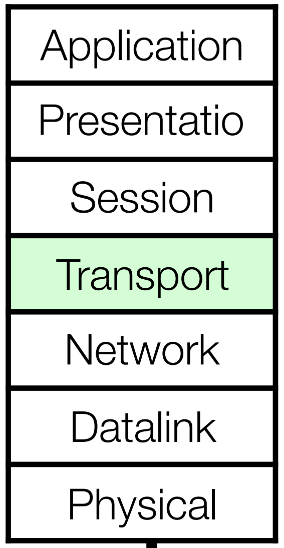


$$T_{CA} = \frac{MSS \times C}{RTT \times \sqrt{p}} \qquad C = \sqrt{\frac{3}{2}}$$

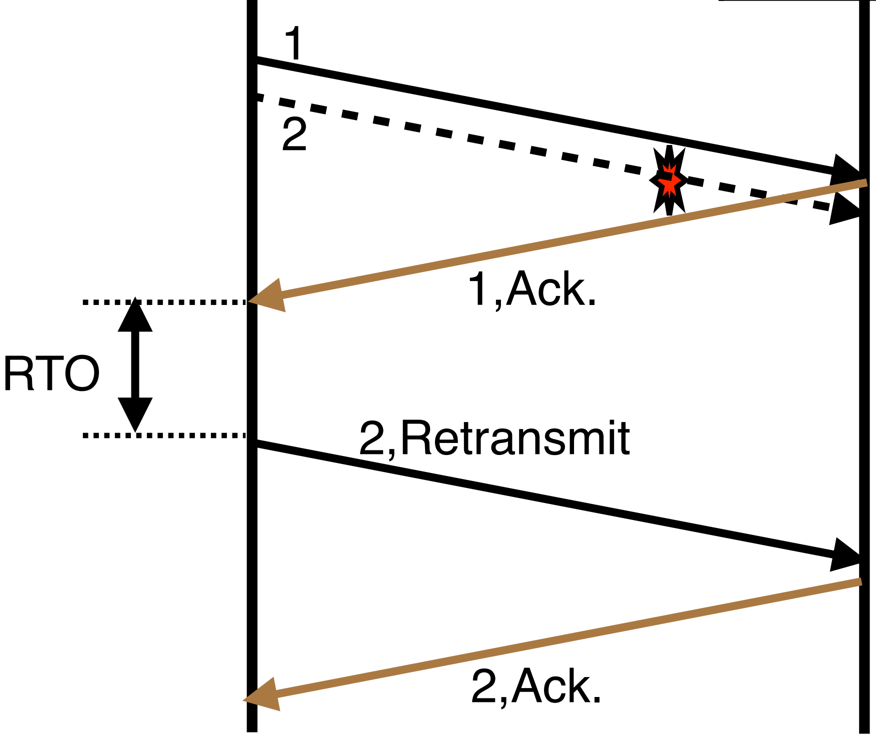
$$p : Packet\ Loss\ Rate$$

Mathis, Matthew, et al. "The macroscopic behavior of the TCP congestion avoidance algorithm." *ACM SIGCOMM Computer Communication Review* 27.3 (1997): 67-82.

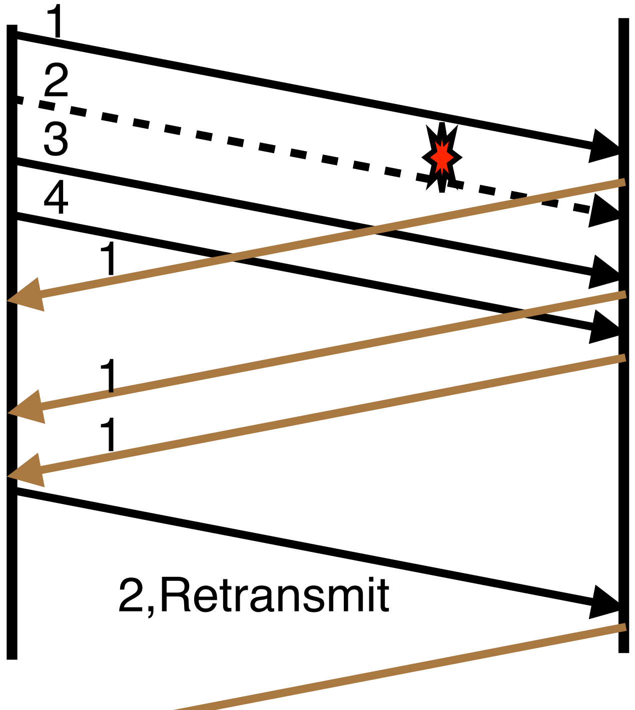
Loss detection and retransmission in TCP



- Retransmit Time Out (RTO)
  When Data Ack. is not received until RTO*,
  - retransmit the segment that is regarded as loss
  - ssthresh = cwnd / 2, cwnd = min_cwnd.
  - RTO is derived by measured RTT.
    min_RTO :
    - 200msec on Linux default
    - 10msec on Google DC intra-traffic



- Fast Retransmit / FastRecovery
  - If receiving three same ack., then the consecutive packet is considered as a loss.
    If retransmit is success,
     ssthresh = cwnd / 2, cwnd = ssthresh + 3 * MSS



Queue / packet buffer at router interface

Application
Presentatio
Session
Transport
Network
Datalink
Physical

- Absorb burst traffic caused by different rate links.
    - In case of small sizes: Unable to absorb large burst.
    - Large: longer queuing delays. Buffer space does not overflow as quickly but the buffers become full due to (greedy) TCP's behavior
    - C = BW x RTT (C: Optimal buffer capacity in case of single TCP flow)

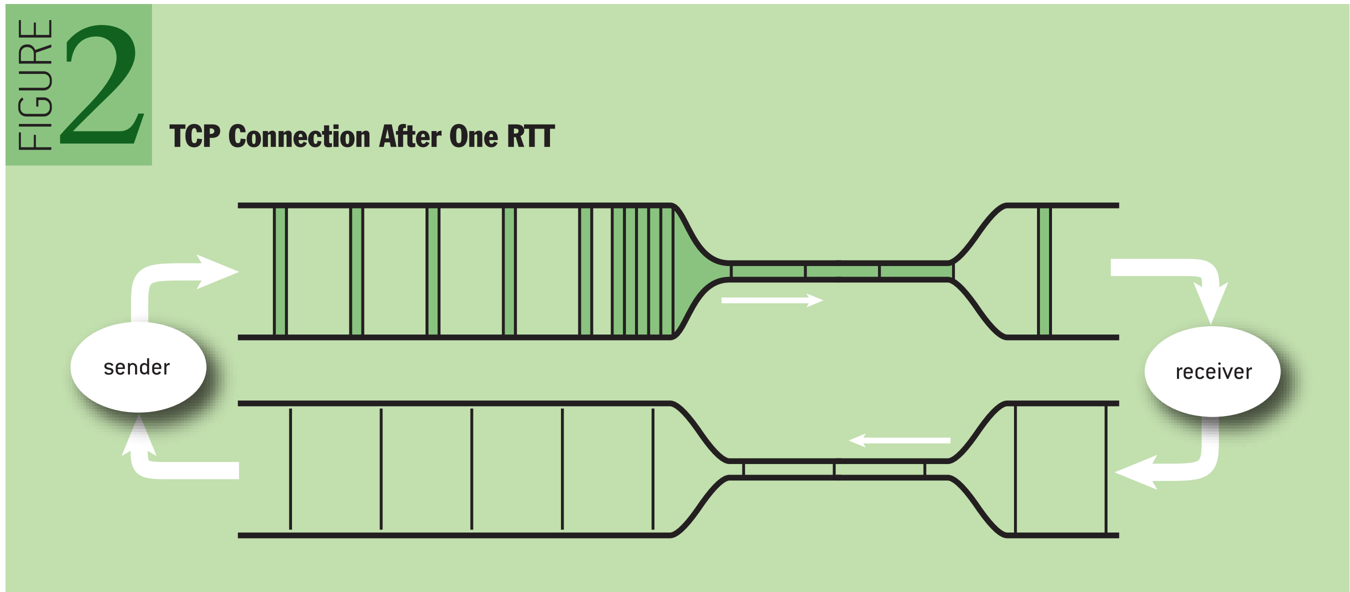
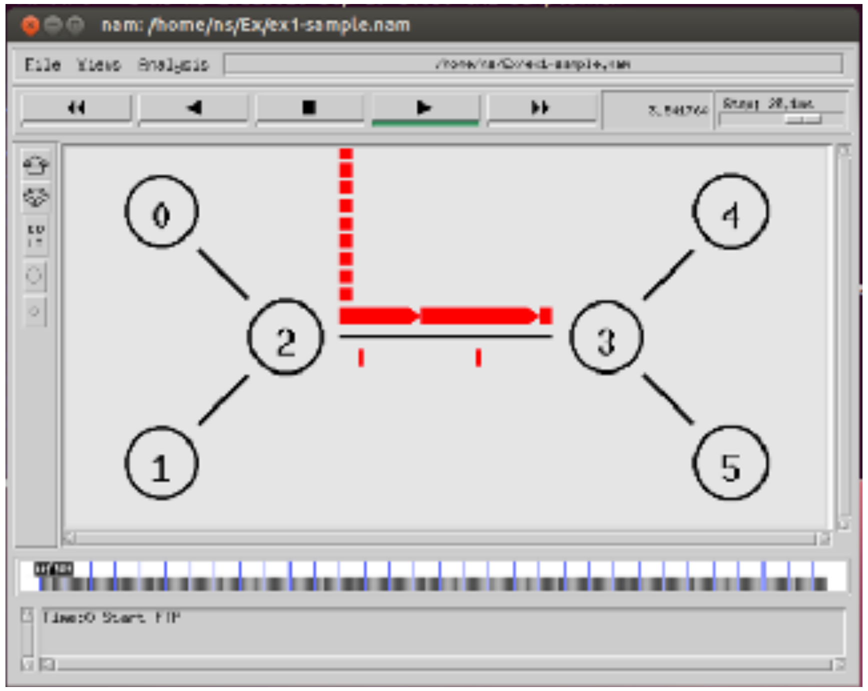


FIGURE 2 TCP Connection After One RTT


nam: /home/ns/Ex/ex1-sample.nam

Time:0 Start FTP

TCP Self/Ack. clocking

Application
Presentatio
Session
Transport
Network
Datalink
Physical

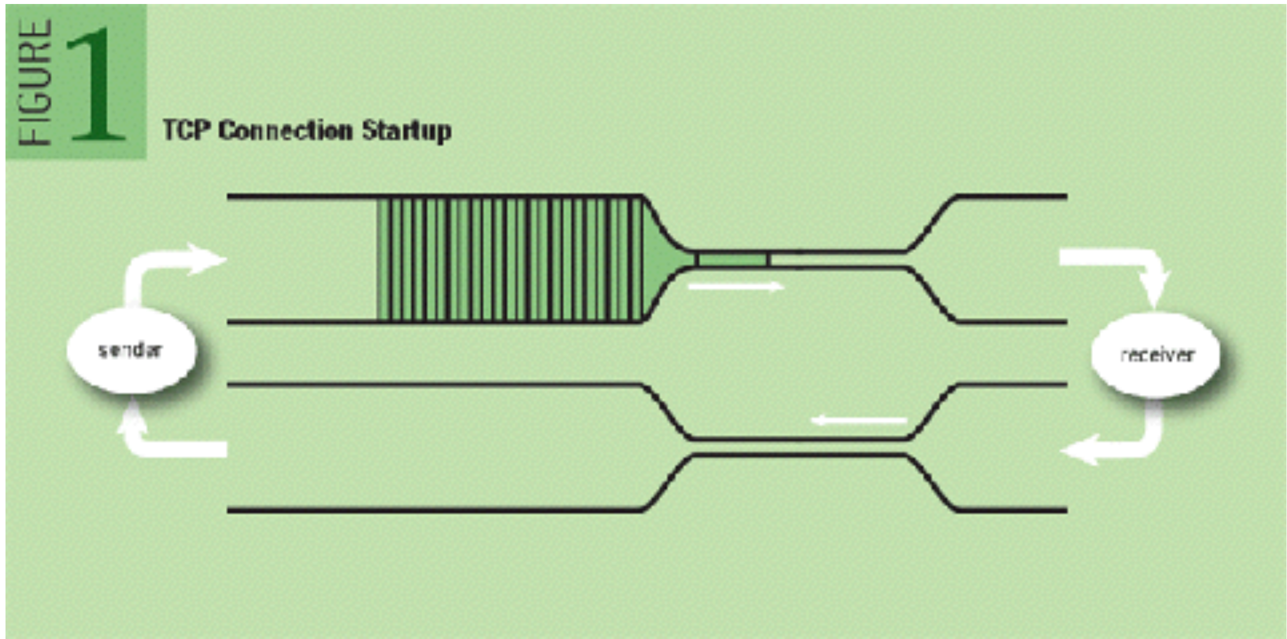


FIGURE 1 TCP Connection Startup

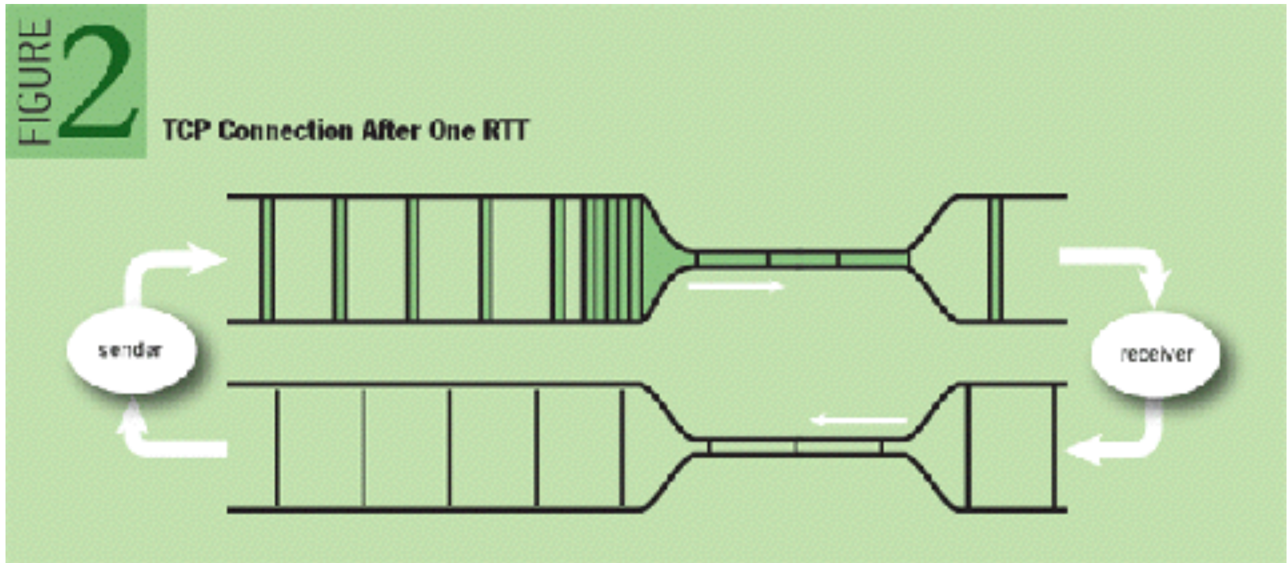
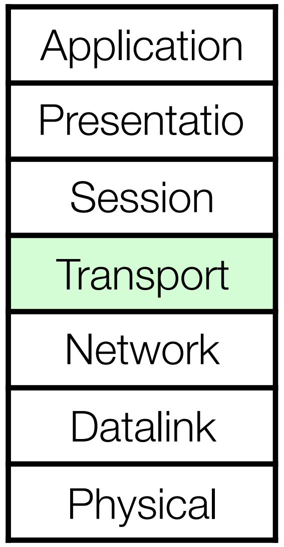


FIGURE 2 TCP Connection After One RTT

Jacobson, Van. "Congestion avoidance and control." *ACM SIGCOMM Computer Communication Review*. Vol. 18. No. 4. ACM, 1988.
Nichols, K., and V. Jacobson. A modern aqm is just one piece of the solution to bufferbloat. Tech. rep., 2012.

TCP window behavior on NewReno


Application

Presentatio

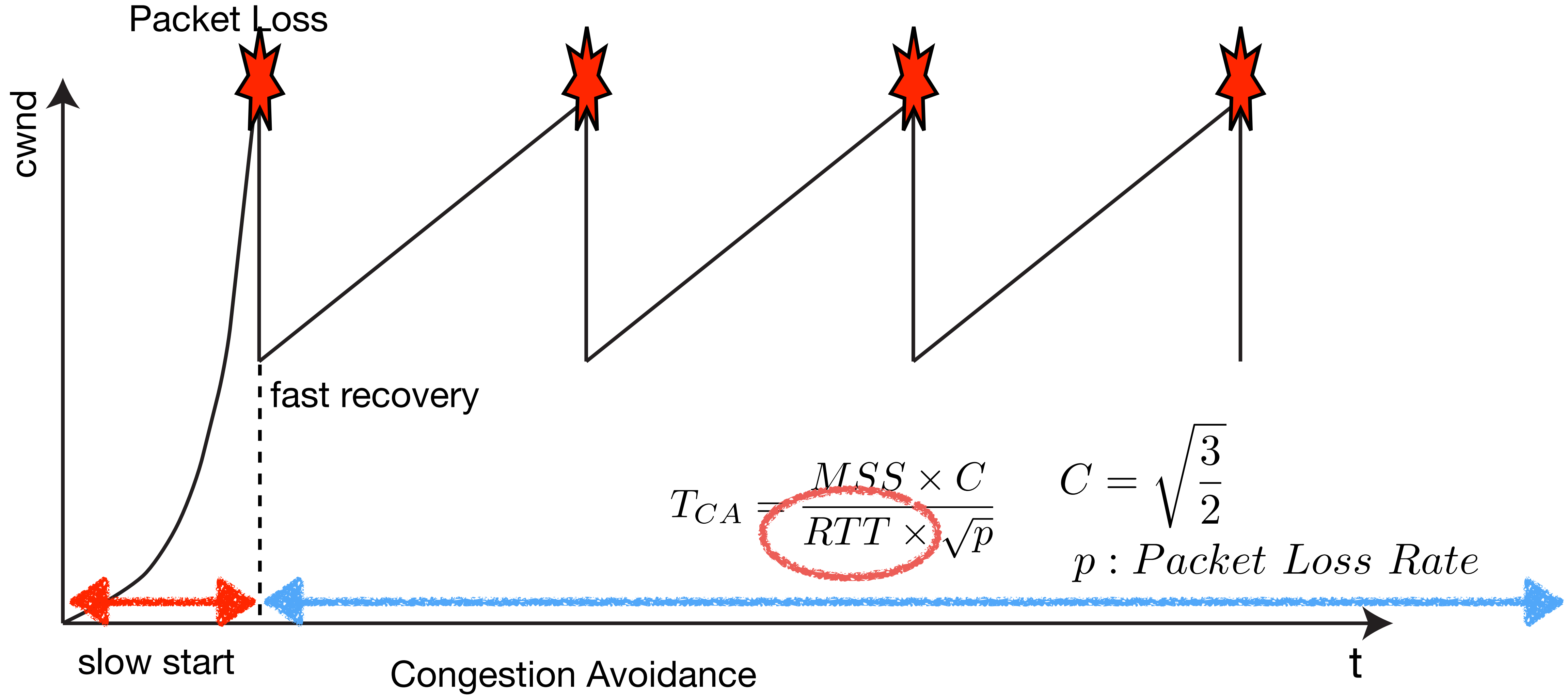
Session

Transport

Network

Datalink

Physical

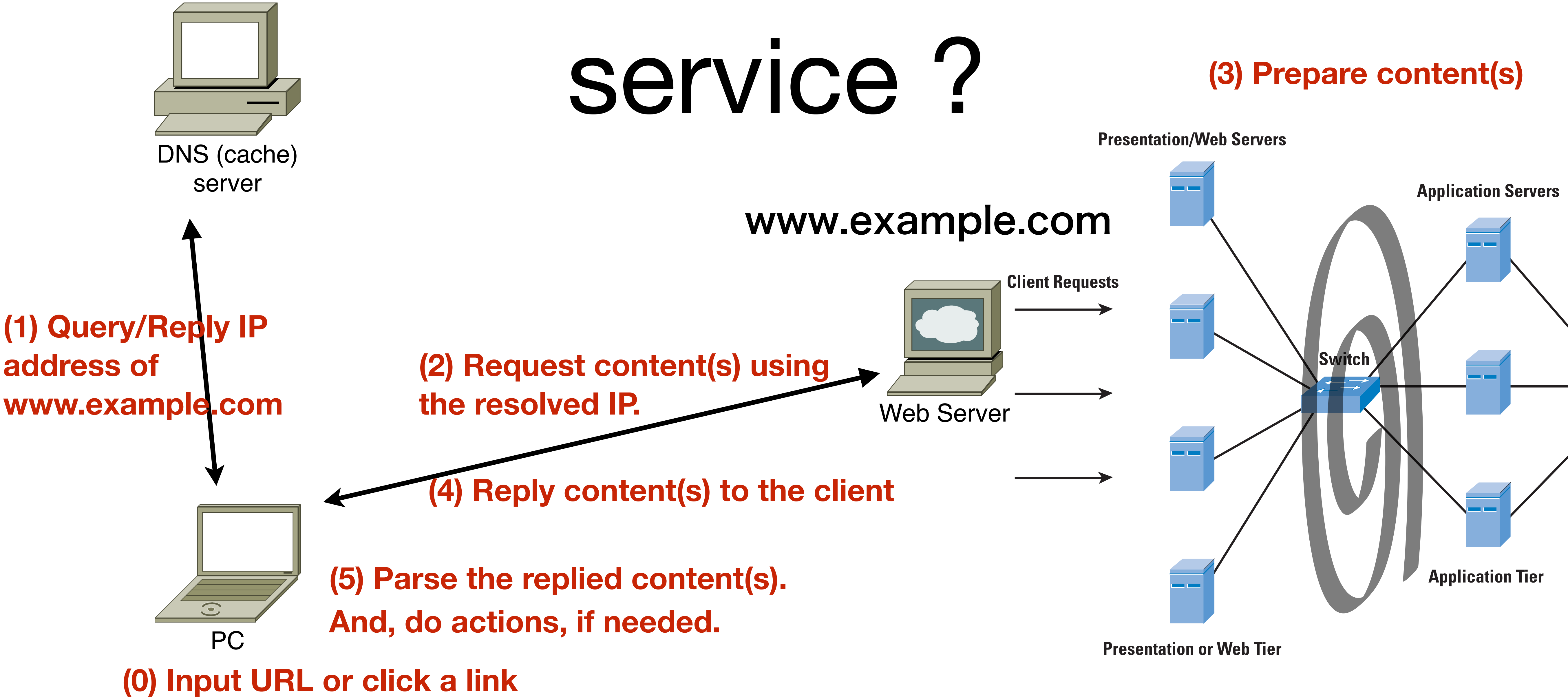


$$T_{CA} = \frac{MSS \times C}{RTT \times \sqrt{p}} \qquad C = \sqrt{\frac{3}{2}}$$

$p : Packet\ Loss\ Rate$

Mathis, Matthew, et al. "The macroscopic behavior of the TCP congestion avoidance algorithm." *ACM SIGCOMM Computer Communication Review* 27.3 (1997): 67-82.

How client accesses Web service ?



(0) Input URL or click a link

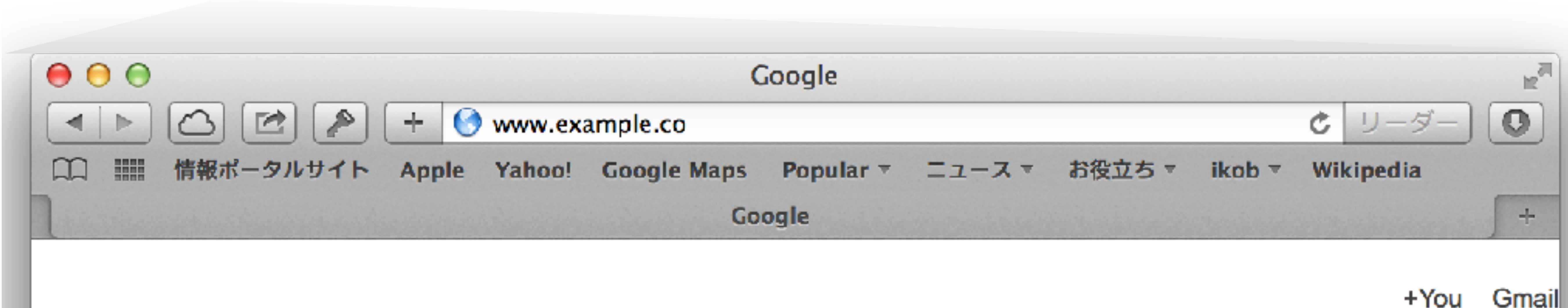
(1) Query/Reply IP address of www.example.com

(2) Request content(s) using the resolved IP.

(3) Prepare content(s)

(4) Reply content(s) to the client

(5) Parse the replied content(s). And, do actions, if needed.

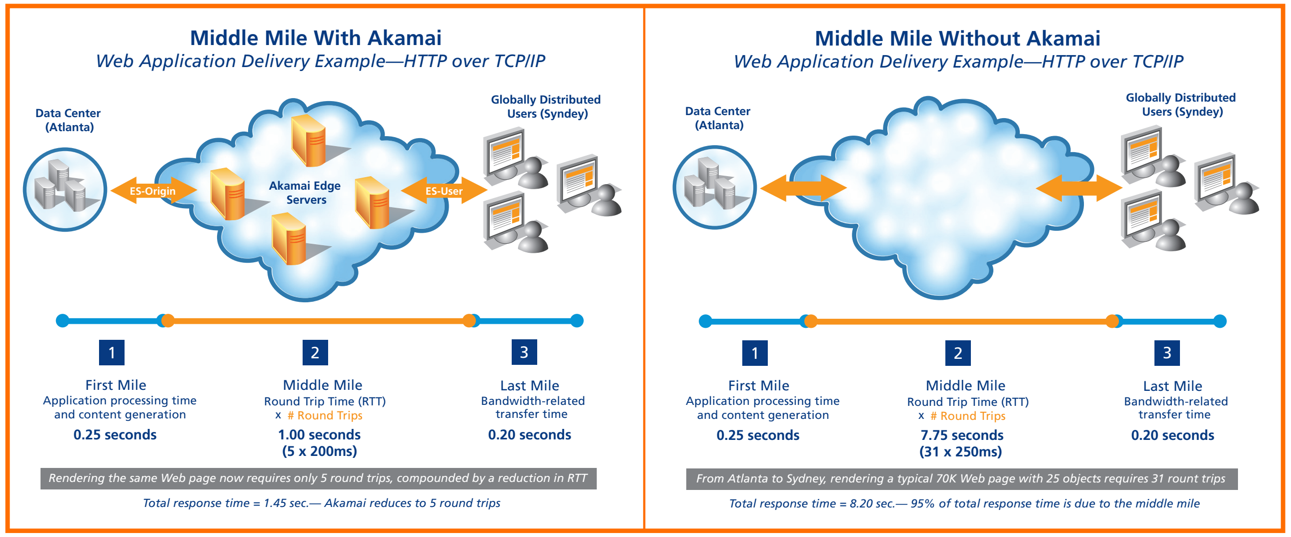


Google

www.example.co

Contents Distribution Network (CDN) and RTT

- CDN is developed to serve content to Internet users with optimized availability and performance. A CDN uses servers that are geographically distributed, helping to accelerate the delivery of content by caching it in multiple locations and then using the closest server to fulfill a request for content from each particular user.
  Many CDN providers compete Akamai, Cloudflare, AWS CloudFront compete with each other.
  - CDN offers “Best” server using metrics such as:
    - Small RTT / Bandwidth Capacity / Stable connectivity each other.
  - Akamai deploys more than 100,000 edge servers in order to improve UX incl. to reduce RTT.



Middle Mile With Akamai

Web Application Delivery Example—HTTP over TCP/IP

Rendering the same Web page now requires only 5 round trips, compounded by a reduction in RTT

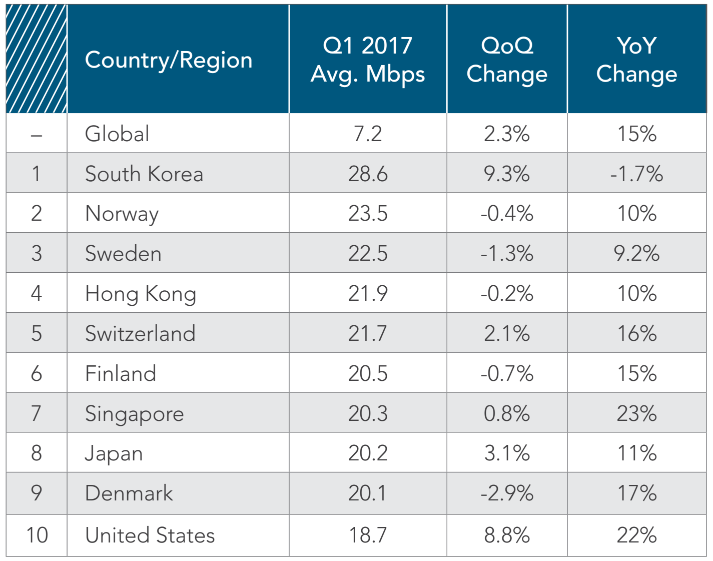
Total response time = 1.45 sec.— Akamai reduces to 5 round trips

Middle Mile Without Akamai

Web Application Delivery Example—HTTP over TCP/IP

From Atlanta to Sydney, rendering a typical 70K Web page with 25 objects requires 31 rount trips

Total response time = 8.20 sec.— 95% of total response time is due to the middle mile



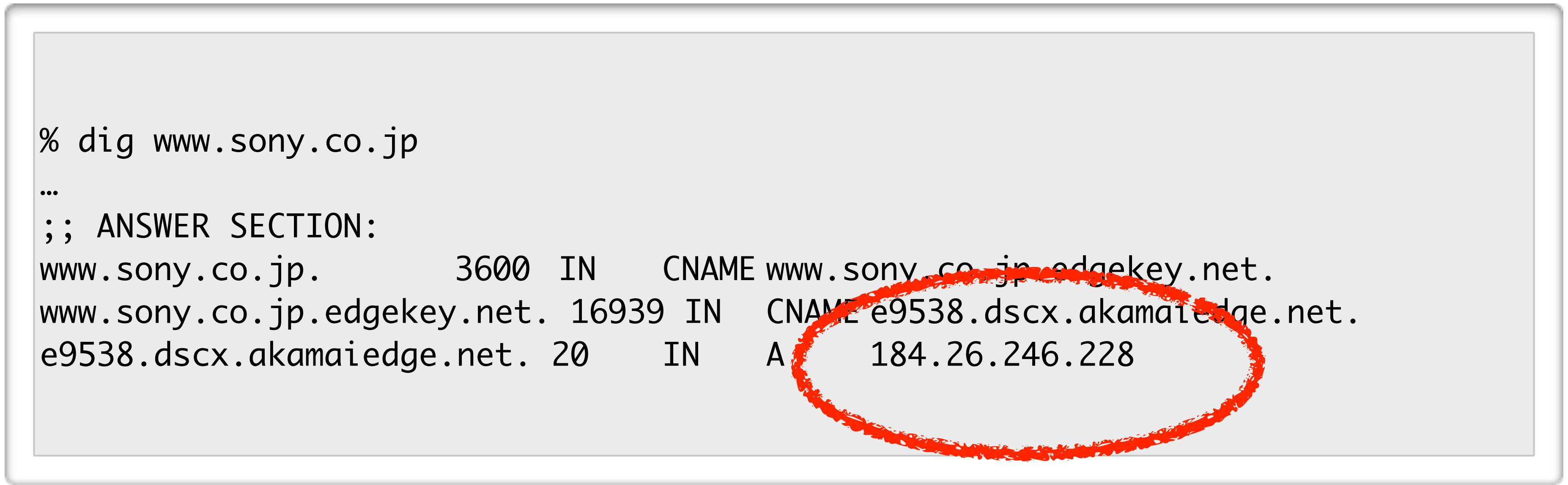
	Country/Region	Q1 2017 Avg. Mbps	QoQ Change	YoY Change
–	Global	7.2	2.3%	15%
1	South Korea	28.6	9.3%	-1.7%
2	Norway	23.5	-0.4%	10%
3	Sweden	22.5	-1.3%	9.2%
4	Hong Kong	21.9	-0.2%	10%
5	Switzerland	21.7	2.1%	16%
6	Finland	20.5	-0.7%	15%
7	Singapore	20.3	0.8%	23%
8	Japan	20.2	3.1%	11%
9	Denmark	20.1	-2.9%	17%
10	United States	18.7	8.8%	22%

Figure 6: Average Connection Speed (IPv4) by Country/Region

Akamai Corp., “Akamai’s EdgePlatform for Application Acceleration”

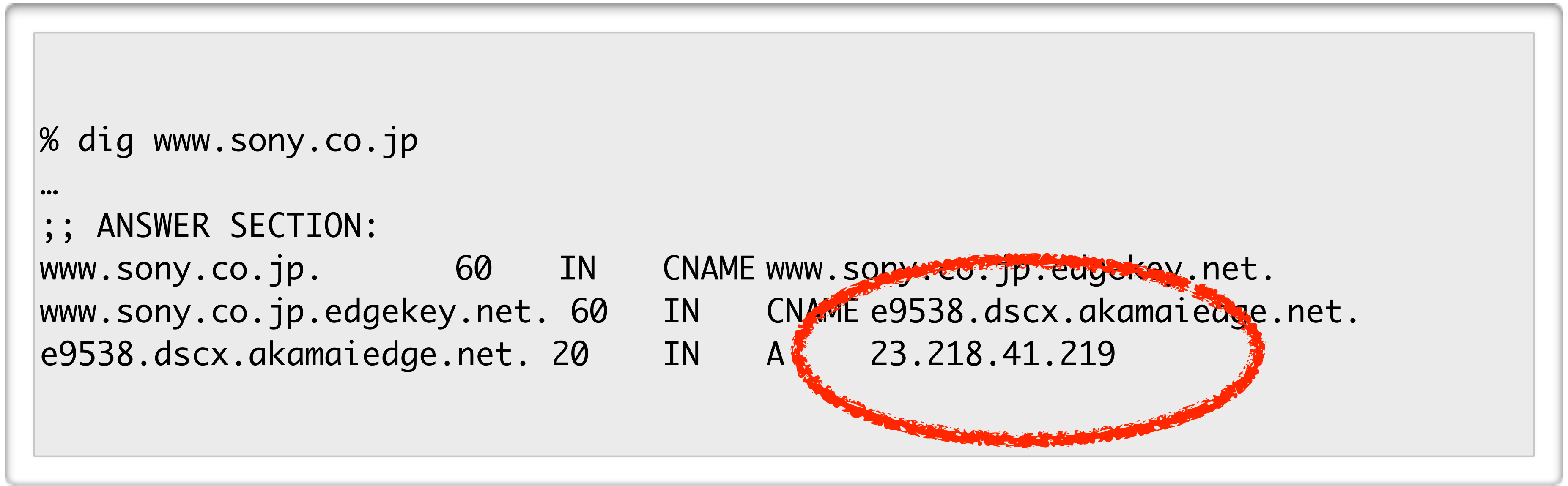
Akamai Corp., “Akamai’s [state of the internet] Q1 2017 report ”

Akamai’s CDN server selection with DNS



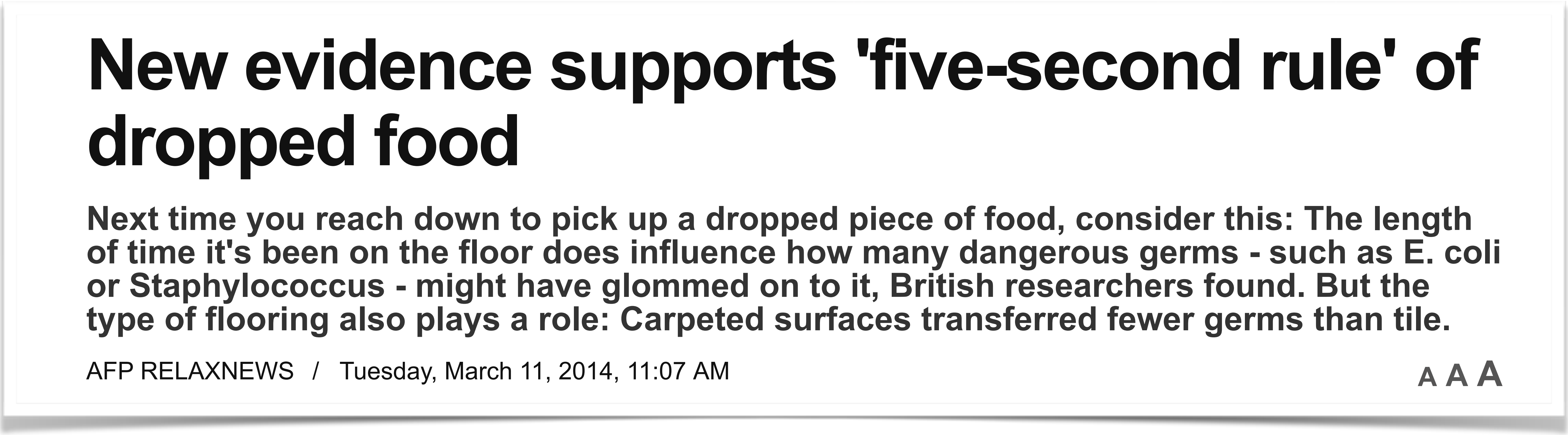
```
% dig www.sony.co.jp
…
;; ANSWER SECTION:
www.sony.co.jp.         3600  IN    CNAME www.sony.co.jp.edgekey.net.
www.sony.co.jp.edgekey.net. 16939 IN    CNAME e9538.dscx.akamaiedge.net.
e9538.dscx.akamaiedge.net. 20    IN    A      184.26.246.228
```

From U-Tokyo



```
% dig www.sony.co.jp
…
;; ANSWER SECTION:
www.sony.co.jp.         60    IN    CNAME www.sony.co.jp.edgekey.net.
www.sony.co.jp.edgekey.net. 60   IN    CNAME e9538.dscx.akamaiedge.net.
e9538.dscx.akamaiedge.net. 20    IN    A      23.218.41.219
```

From Cloud service (AWS us-east-1)

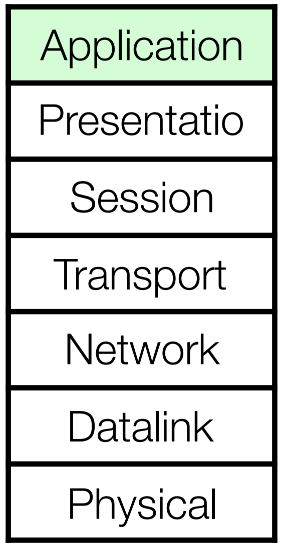


New evidence supports 'five-second rule' of dropped food

Next time you reach down to pick up a dropped piece of food, consider this: The length of time it's been on the floor does influence how many dangerous germs - such as E. coli or Staphylococcus - might have glommed on to it, British researchers found. But the type of flooring also plays a role: Carpeted surfaces transferred fewer germs than tile.

AFP RELAXNEWS / Tuesday, March 11, 2014, 11:07 AM

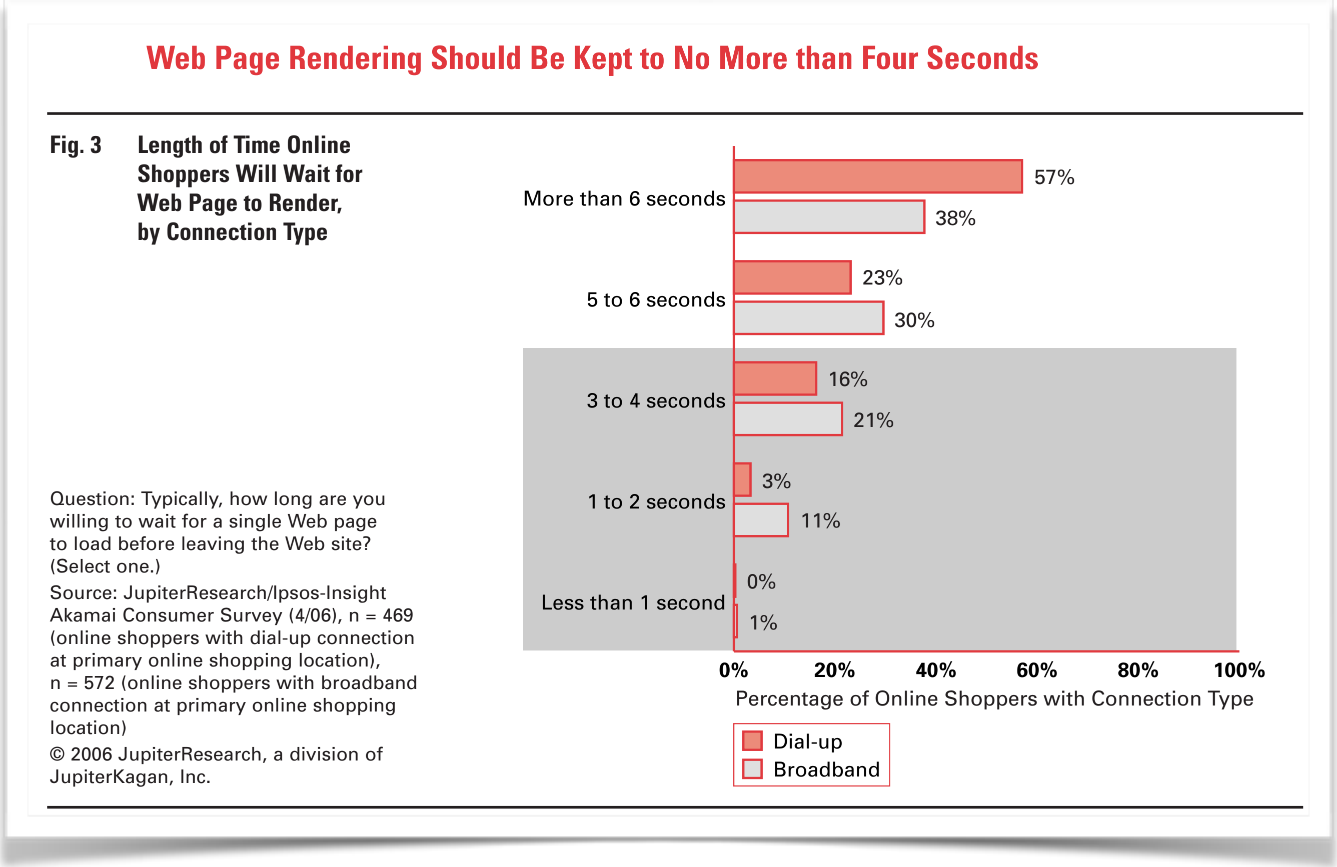
NYDailyNews.com より



Application
Presentatio
Session
Transport
Network
Datalink
Physical

Four/Two second rule in Web services

- < 4 sec. render completion limit to keep customers’ attention @ 2006
    - The limit is decreasing, e.g., 2 sec.@2013
- Poor satisfaction decreases customer loyalty and revisiting.
    - To miss opportunity, and to lost revenue on e-Commerce.



Web Page Rendering Should Be Kept to No More than Four Seconds

Fig. 3 Length of Time Online Shoppers Will Wait for Web Page to Render, by Connection Type

Question: Typically, how long are you willing to wait for a single Web page to load before leaving the Web site? (Select one.)

Source: JupiterResearch/Ipsos-Insight Akamai Consumer Survey (4/06), n = 469 (online shoppers with dial-up connection at primary online shopping location), n = 572 (online shoppers with broadband connection at primary online shopping location)

© 2006 JupiterResearch, a division of JupiterKagan, Inc.

Jupiter Research, “RETAIL WEB SITE PERFORMANCE Consumer Reaction to a Poor Online Shopping Experience”, 2006

Today’s Quiz

1. Show two or more Contents Delivery Network (CDN) hosted sites.

2. Tell the reasons why such sites look like hosted on CDN than own Web server.

- Submit your answers either in Japanese or in English via the course web.

Today’s Quiz

1. Contents Delivery Network (CDN)でホストされている Web サービスを２つ以上示せ。

2. これらのサービス が自身の CDN でホストされている理由を示せ。

- Submit your answers either in Japanese or in English via the course web.

Today’s Assignment

- In the class, TCP Fast Open (RFC7413) was introduced as an approach to reduce the latency of TCP connection set-up. Another option T/TCP - TCP extensions for Transactions (RFC1644) which shares the same goal had been standardized for 20 years. In addition, T/TCP reduces the TCP set-up latency not only from the second or later connections, but from the first one. However, now T/TCP standard has been obsoleted.

- Read RFC7413, RFC1644 and related documents. Discuss why T/TCP has been obsoleted.

- Submit your answers either in Japanese or in English via the course web.

本日の課題

- 講義では、TCP コネクションセットアップの遅延を抑える手法として TCP Fast Open (RFC7413) を取り上げた。TCP Fast Open と同じ目的で - TCP extensions for Transactions (RFC1644) が 20 年前に標準化されている。さらに、T/TCP ではセットアップ遅延を 2 つめ以降のコネクションだけではなく、最初のコネクションから抑えることができる。しかしながら、T/TCP 標準は廃止された。

- RFC7413, RFC1644 および関連文書を読み、T/TCP が廃止された理由を考察せよ。

- 講義 Web ページから回答すること。