Jaringan Komputer

Pertemuan 2

Prodi Informatika

1

Outlines

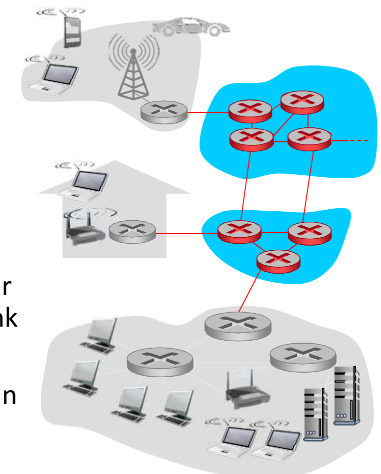
- Network Core (Packet Switching)
- Struktur Internet
- Delay, Lost dan Througput di Jaringan
- Protocol (Layer)
- Application Layer

Prodi Informatika

2

The network core

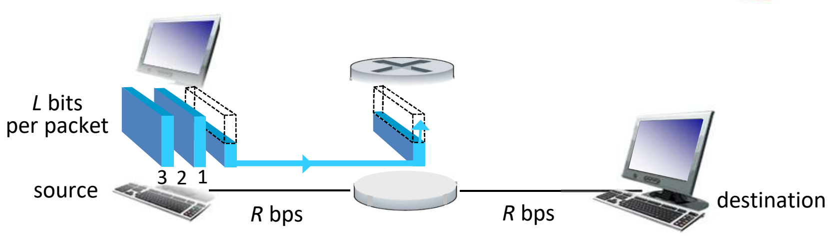
- mesh network dari router yang saling terhubung
- packet-switching: host memecah pesan layer aplikasi ke dalam paket
  - Meneruskan paket dari satu router ke router berikutnya, melintasi Link di jalur dari sumber ke tujuan
  - setiap paket ditransmisikan dengan kapasitas link penuh



Introduction 1-3
Prodi Informatika

3

Packet-switching: store-and-forward



- membutuhkan L / R detik untuk mengirimkan (mendorong keluar) paket L-bit ke dalam link di R bps
- *store and forward:* seluruh paket harus tiba di router sebelum dapat dikirim pada link berikutnya
- end-end delay = $2L/R$ (assuming zero propagation delay)

one-hop numerical example:

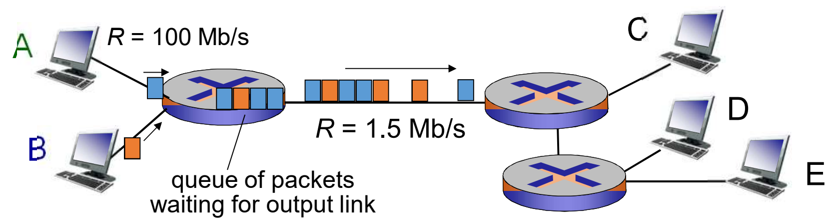
- L = 7.5 Mbits
- R = 1.5 Mbps
- one-hop transmission delay = 5 sec

more on delay shortly …

Introduction 1-4
Prodi Informatika

4

Packet Switching: queueing delay, loss



queuing and loss:

- jika tingkat kedatangan (dalam bit) untuk menghubungkan melebihi tingkat transmisi tautan untuk suatu periode waktu:
  - paket akan mengantri, menungg untuk dikirim pada link
  - paket dapat didrop (hilang) jika memori (buffer) penuh

Introduction 1-5
Prodi Informatika

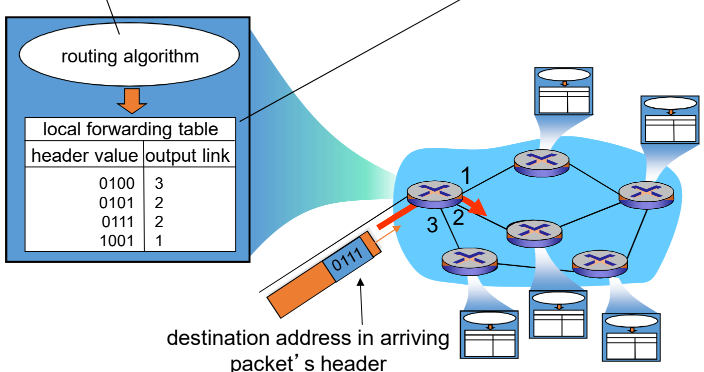
5

Two key network-core functions

routing: determines source-destination route taken by packets

- *routing algorithms*

forwarding: move packets from router's input to appropriate router output



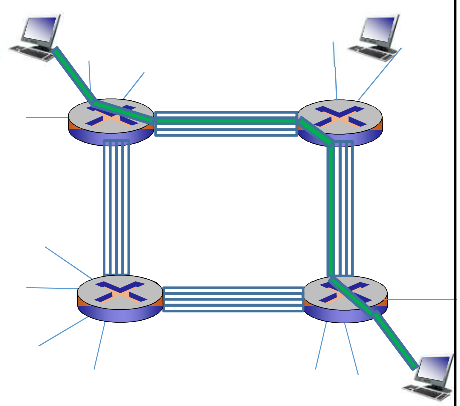
Introduction 2-6
Prodi Informatika

6

Alternative core: circuit switching

end-end resources allocated to, reserved for "call" between source & dest:

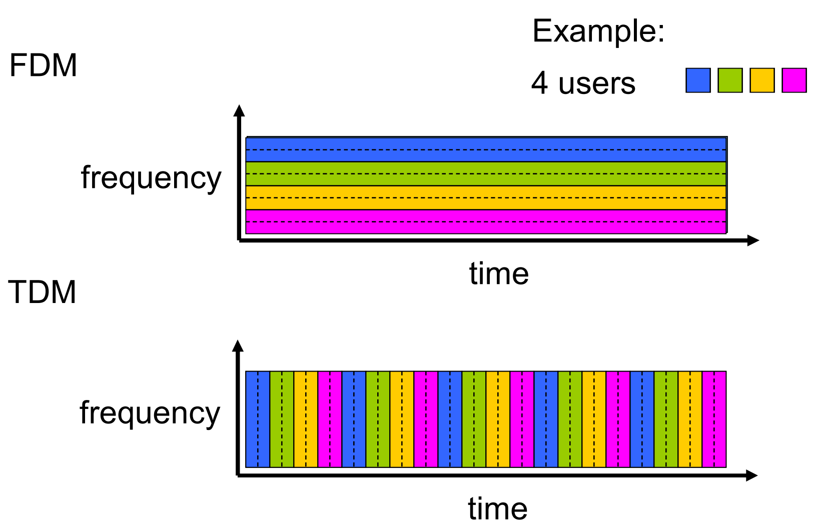
- in diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call *(no sharing)*
- commonly used in traditional telephone networks



Introduction 1-7 Prodi Informatika

7

Circuit switching: FDM versus TDM



Introduction 1-8 Prodi Informatika

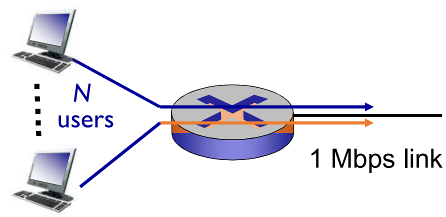
8

Packet switching versus circuit switching

packet switching memungkinkan lebih banyak pengguna menggunakan jaringan!

example:

- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time
- *circuit-switching:*
  - 10 users
- *packet switching:*
  - with 35 users, probability > 10 active at same time is less than .0004 *



Q: how did we get value 0.0004?

Q: what happens if > 35 users ?

Introduction 1-9
Prodi Informatika

9

Packet switching versus circuit switching

is packet switching a "slam dunk winner?"

- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive congestion possible: packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- *Q:* How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps

Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

Introduction 1-10
Prodi Informatika

10

Internet structure: network of networks

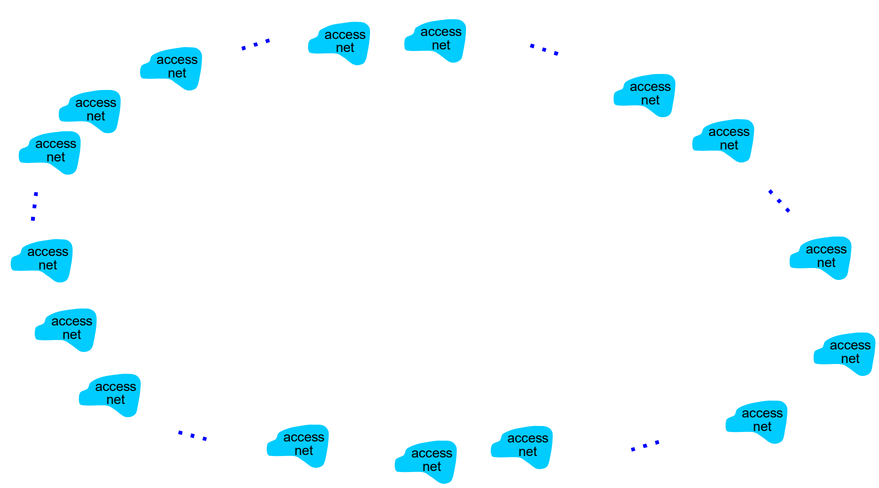
- End systems connect to Internet via access ISPs (Internet Service Providers)
  - residential, company and university ISPs
- Access ISPs in turn must be interconnected.
  - so that any two hosts can send packets to each other
- Resulting network of networks is very complex
  - evolution was driven by economics and national policies
- Let's take a stepwise approach to describe current Internet structure

Introduction 1-11 Prodi Informatika

11

Internet structure: network of networks

Question: given *millions* of access ISPs, how to connect them together?

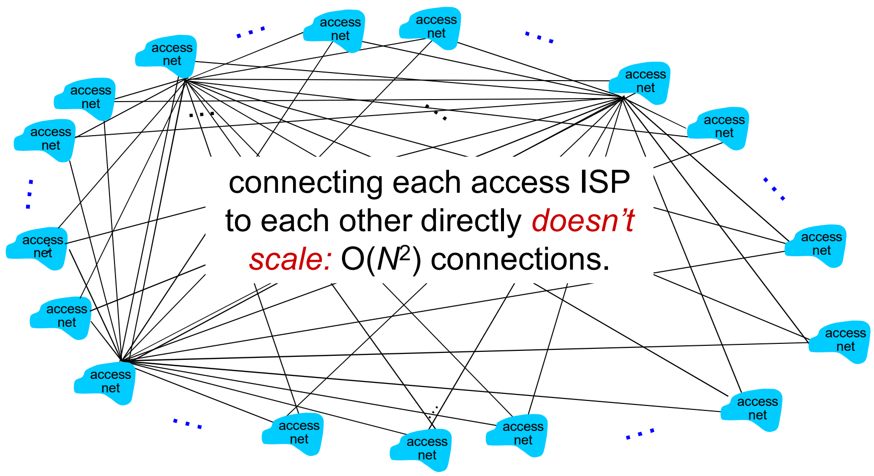


Introduction 1-12 Prodi Informatika

12

Internet structure: network of networks

Option: connect each access ISP to every other access ISP?



Introduction 1-13

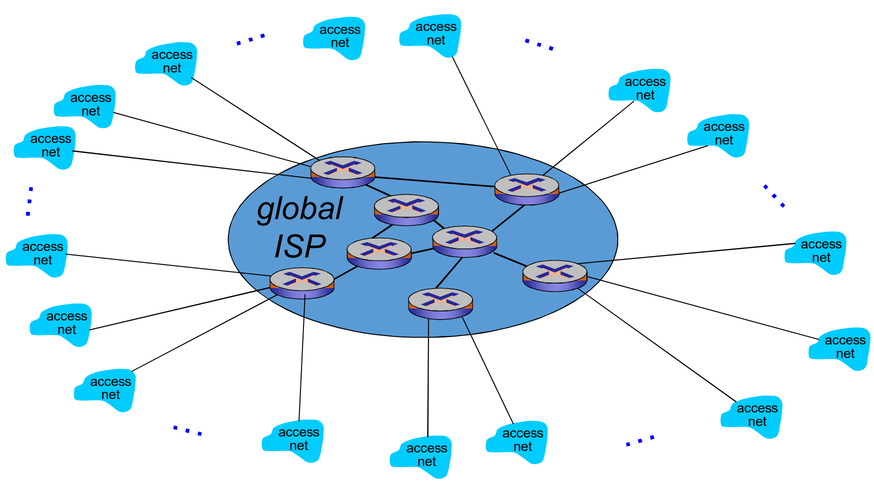
Prodi Informatika

13

Internet structure: network of networks

Option: connect each access ISP to one global transit ISP?

Customer and provider ISPs have economic agreement.



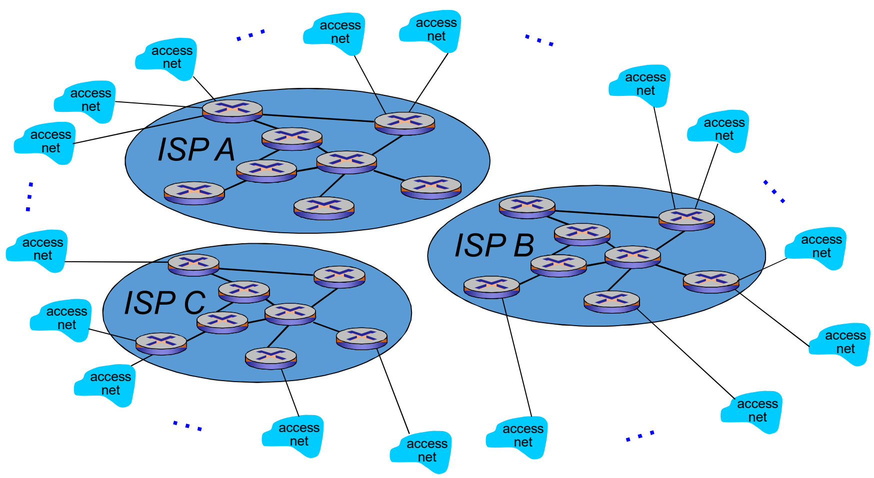
Introduction 1-14

Prodi Informatika

14

Internet structure: network of networks

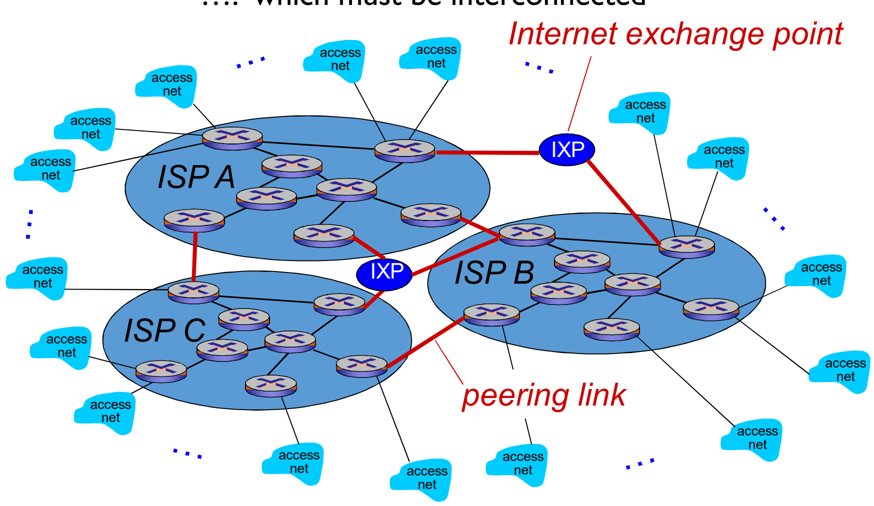
But if one global ISP is viable business, there will be competitors ….



15

Internet structure: network of networks

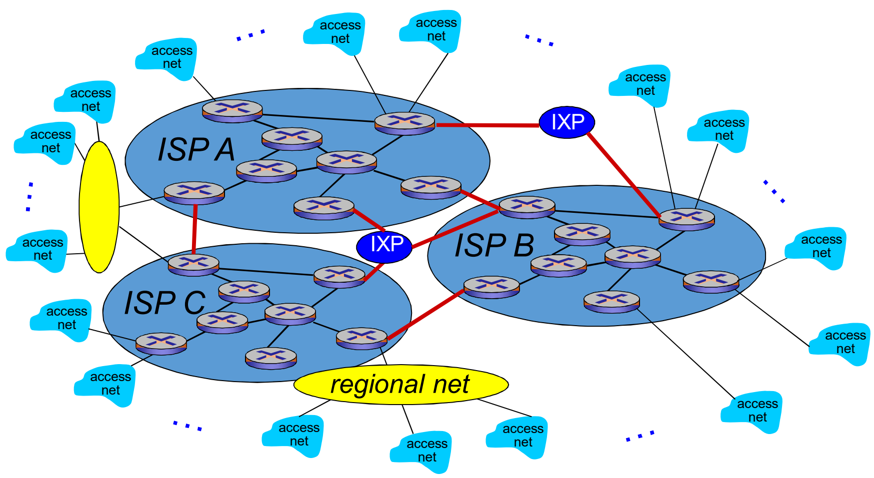
But if one global ISP is viable business, there will be competitors …. which must be interconnected



16

Internet structure: network of networks

… and regional networks may arise to connect access nets to ISPs

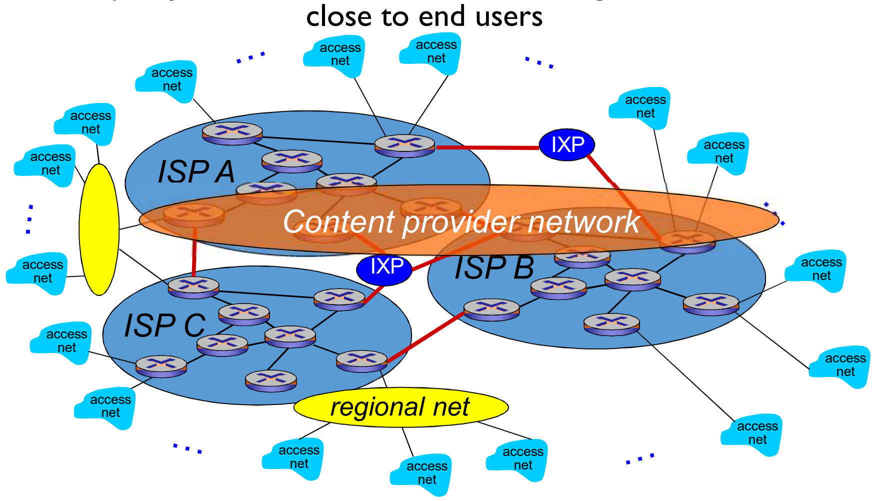


Introduction 1-17
Prodi Informatika

17

Internet structure: network of networks

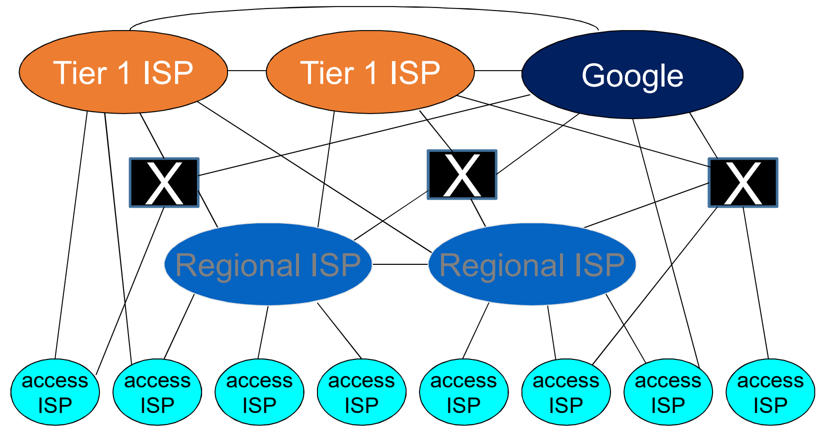
… and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users



Introduction 1-18
Prodi Informatika

18

Internet structure: network of networks

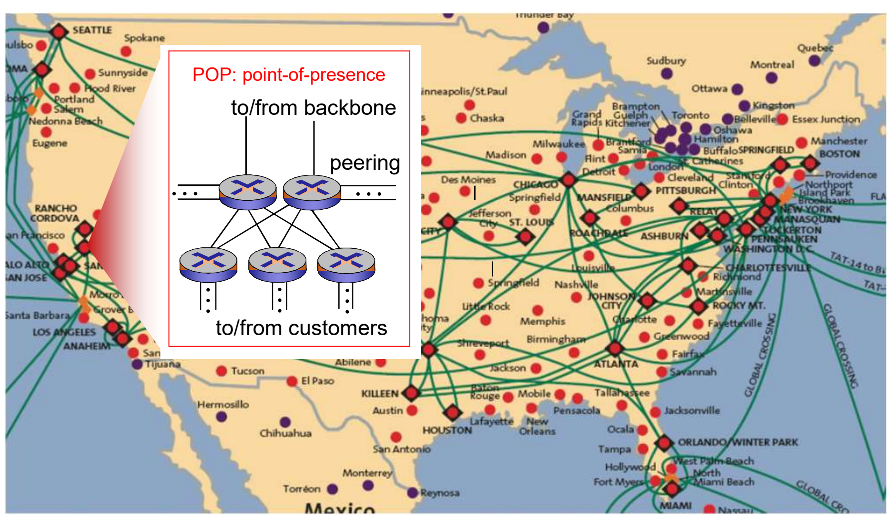


- at center: small # of well-connected large networks
  - "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
  - content provider network (e.g., Google): private network that connects it data centers to Internet, often bypassing tier-1, regional ISPs

Introduction 1-19
Prodi Informatika

19

Tier-1 ISP: e.g., Sprint



Introduction 1-20
Prodi Informatika

20

Delay, loss, throughput in networks

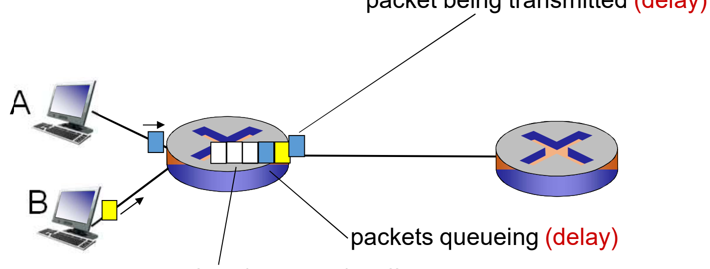
Introduction 1-21
Prodi Informatika

21

How do loss and delay occur?

paket mengantri di buffer router

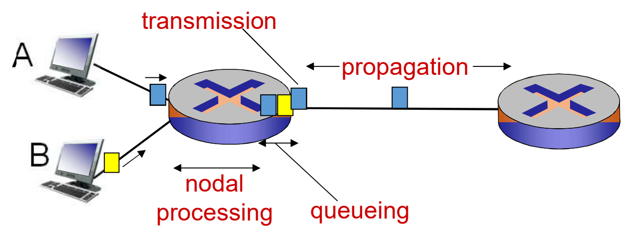
- tingkat kedatangan paket untuk link (sementara) melebihi kapasitas link keluaran
- paket antrian, tunggu giliran



Introduction 1-22
Prodi Informatika

22

Four sources of packet delay



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{proc}: nodal processing

- check bit errors
- determine output link
- typically < msec

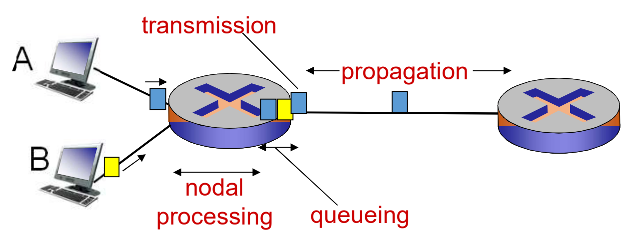
d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

Introduction 1-23
Prodi Informatika

23

Four sources of packet delay



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{trans}: transmission delay:

- *L*: packet length (bits)
- *R*: link *bandwidth (bps)*
- $d_{trans} = L/R$

d_{trans} and d_{prop} *very* different

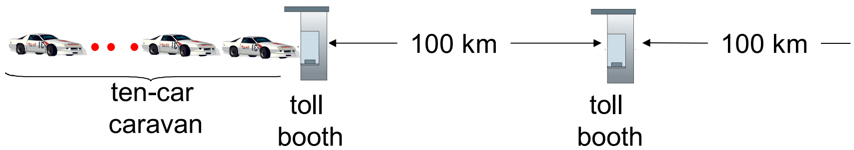
d_{prop}: propagation delay:

- *d*: length of physical link
- *s*: propagation speed (~2x10^8 m/sec)
- $d_{prop} = d/s$

Introduction 1-24
Prodi Informatika

24

Caravan analogy

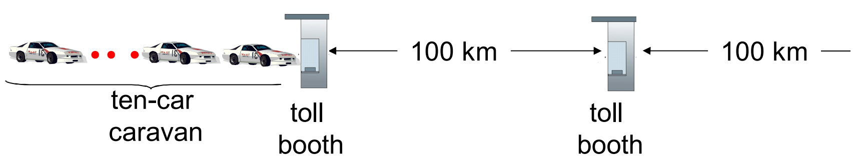


- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- car ~ bit; caravan ~ packet
- *Q:* How long until caravan is lined up before 2nd toll booth?
- time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- *A:* 62 minutes

Introduction 1-25
Prodi Informatika

25

Caravan analogy (more)



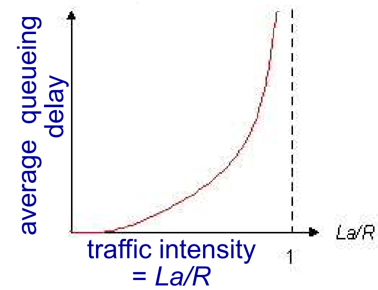
- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?
  - *A: Yes!* after 7 min, first car arrives at second booth; three cars still at first booth

Introduction 1-26
Prodi Informatika

26

Queueing delay (revisited)

- *R:* link bandwidth (bps)
- *L:* packet length (bits)
- a: average packet arrival rate



- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving than can be serviced, average delay infinite!

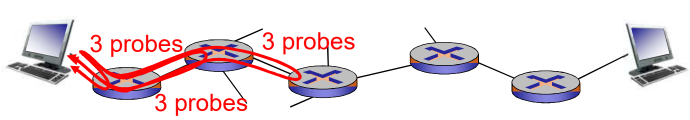
* Check online interactive animation on queuing and loss

Introduction 1-27 Prodi Informatika

27

"Real" Internet delays and routes

- what do "real" Internet delay & loss look like?
- **traceroute** program: provides delay measurement from source to router along end-end Internet path towards destination. For all *i:*
  - sends three packets that will reach router *i* on path towards destination
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.



Introduction 1-28 Prodi Informatika

28

"Real" Internet delays, routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

3 delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 ms 18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms 109 ms 106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms 102 ms 104 ms
10  de.fr1.fr.geant.net (62.40.96.50)  113 ms 121 ms 114 ms
11  renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12  nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13  nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14  r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15  eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16  194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17  * * *
18  * * *
19  fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136 ms
```

trans-oceanic link

* means no response (probe lost, router not replying)

* Do some traceroutes from exotic countries at www.traceroute.org

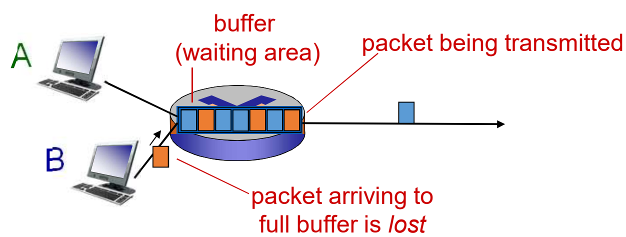
Introduction 1-29

Prodi Informatika

29

Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



* Check out the Java applet for an interactive animation on queuing and loss

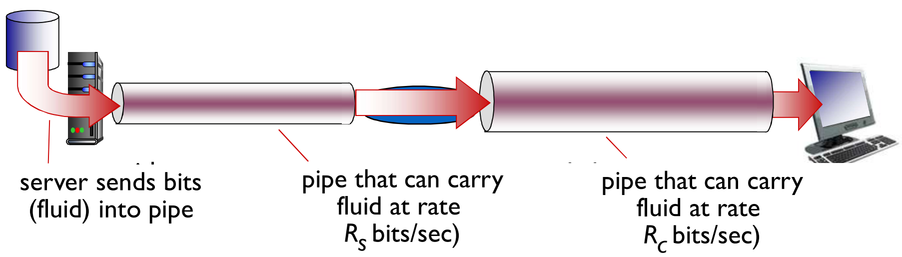
Introduction 1-30

Prodi Informatika

30

Throughput

- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

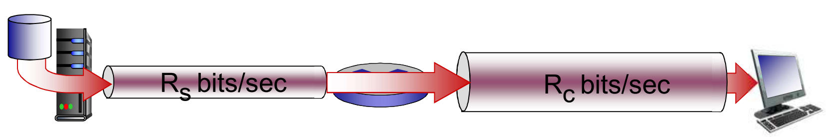


Introduction 1-31 Prodi Informatika

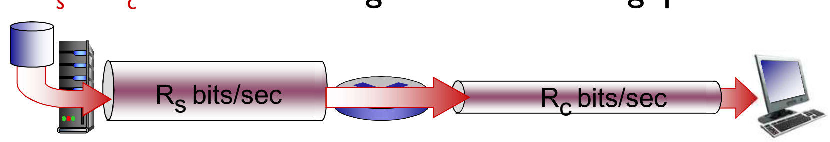
31

Throughput (more)

- $R_s < R_c$ What is average end-end throughput?



- $R_s > R_c$ What is average end-end throughput?



bottleneck link

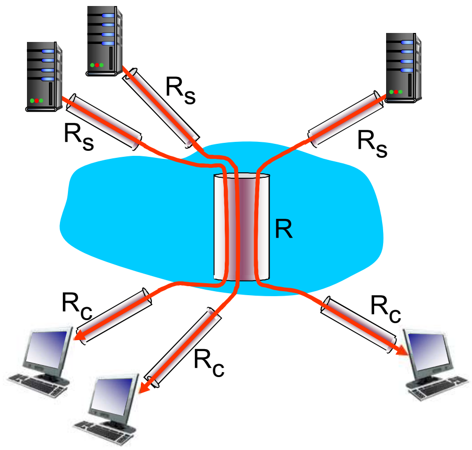
link on end-end path that constrains end-end throughput

Introduction 1-32 Prodi Informatika

32

Throughput: Internet scenario

- per-connection end-end throughput: $min(R_c,R_s,R/10)$
- in practice: R_c or R_s is often bottleneck



10 connections (fairly) share backbone bottleneck link R bits/sec

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Introduction 1-33 Prodi Informatika

33

Protocol "layers"

Networks are complex, with many "pieces":

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

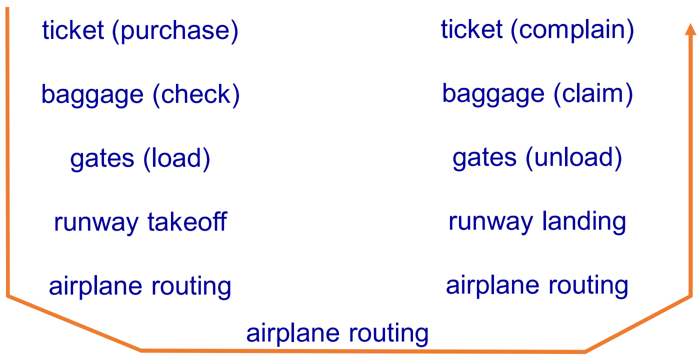
is there any hope of *organizing* structure of network?

…. or at least our discussion of networks?

Introduction 1-34 Prodi Informatika

34

Organization of air travel

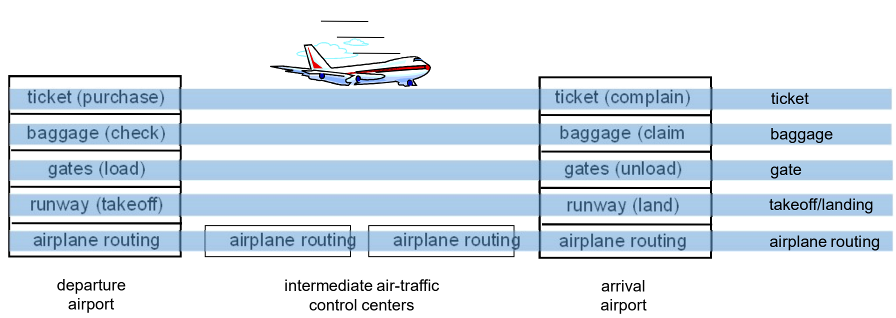


- a series of steps

Introduction 1-35 Prodi Informatika

35

Layering of airline functionality



layers: setiap layer mengimplementasikan sebuah layanan

- via its own internal-layer actions
- relying on services provided by layer below

Introduction 1-36 Prodi Informatika

36

Why layering?

dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?

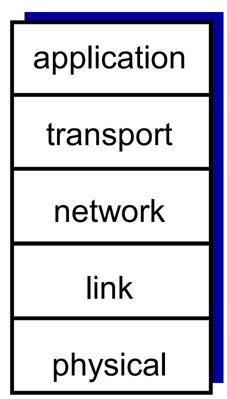
Introduction 1-37

Prodi Informatika

37

Internet protocol stack

- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- *physical:* bits "on the wire"



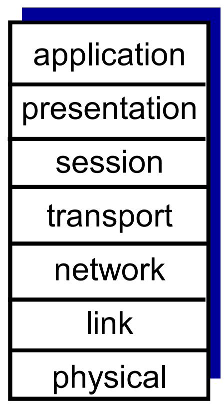
Introduction 1-38

Prodi Informatika

38

ISO/OSI reference model

- *presentation:* allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session:* synchronization, checkpointing, recovery of data exchange
- Internet stack "missing" these layers!
  - these services, *if needed,* must be implemented in application
  - needed?

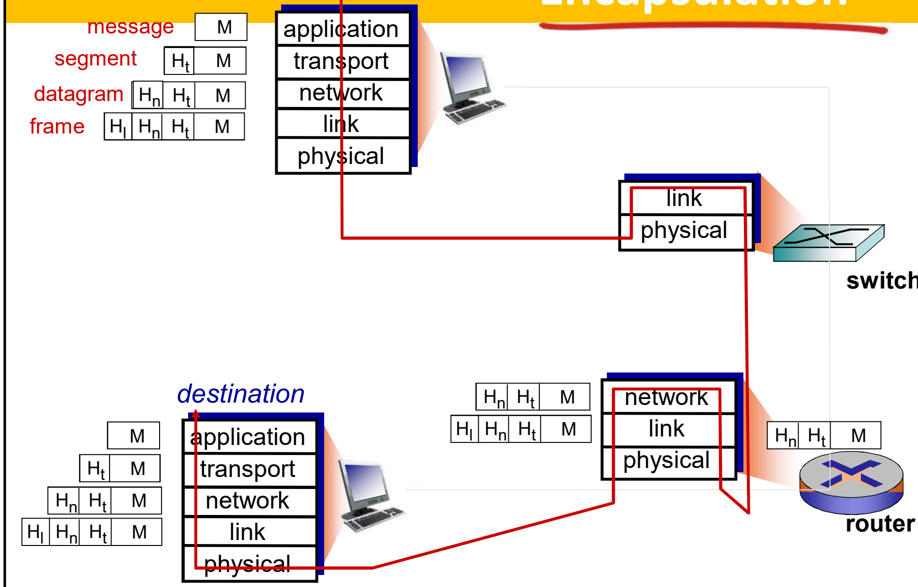


Introduction 1-39

Prodi Informatika

39

Encapsulation



Introduction 1-40

Prodi Informatika

40

Application layer

our goals:

- conceptual, implementation aspects of network application protocols
  - transport-layer service models
  - client-server paradigm
  - peer-to-peer paradigm
  - content distribution networks
- learn about protocols by examining popular application-level protocols
  - HTTP
  - FTP
  - SMTP / POP3 / IMAP
  - DNS
- creating network applications
  - socket API

Application Layer 2-41 Prodi Informatika

41

Some network apps

- e-mail
- web
- text messaging
- remote login
- P2P file sharing
- multi-user network games
- streaming stored video (YouTube, Hulu, Netflix)
- voice over IP (e.g., Skype)
- real-time video conferencing
- social networking
- search
- …
- …

Application Layer 2-42 Prodi Informatika

42

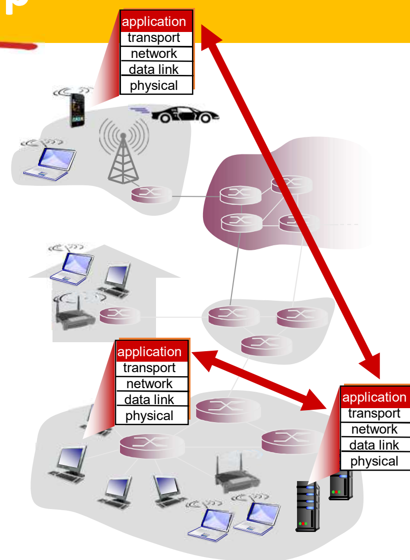
Creating a network app

write programs that:

- run on (different) *end systems*
- communicate over network
- e.g., web server software communicates with browser software

no need to write software for network-core devices

- network-core devices do not run user applications
- applications on end systems allows for rapid app development, propagation



Prodi Informatika

43

Application architectures

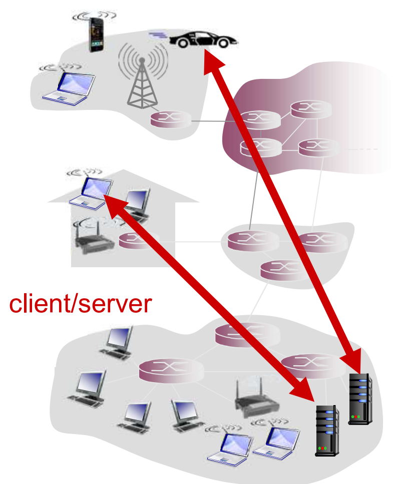
possible structure of applications:

- client-server
- peer-to-peer (P2P)

Prodi Informatika

44

Client-server architecture



server:

- always-on host
- permanent IP address
- data centers for scaling

clients:

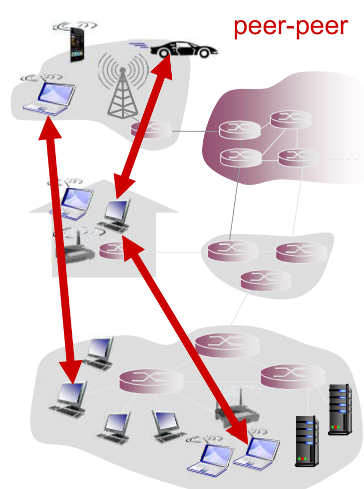
- communicate with server
- may be intermittently connected
- may have dynamic IP addresses
- do not communicate directly with each other

Prodi Informatika

45

P2P architecture

- *no* always-on server
- arbitrary end systems directly communicate
- peers request service from other peers, provide service in return to other peers
  - *self scalability* – new peers bring new service capacity, as well as new service demands
- peers are intermittently connected and change IP addresses
  - complex management



Prodi Informatika

46

Processes communicating

process: program running within a host

- within same host, two processes communicate using inter-process communication (defined by OS)
- processes in different hosts communicate by exchanging messages

clients, servers

client process: process that initiates communication

server process: process that waits to be contacted

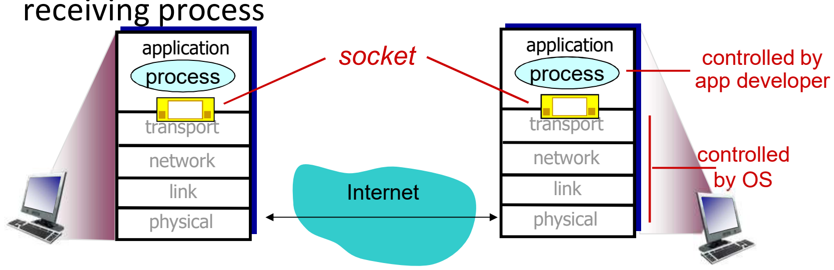
- aside: applications with P2P architectures have client processes & server processes

Application Layer 2-47 Prodi Informatika

47

Sockets

- process sends/receives messages to/from its socket
- socket analogous to door
  - sending process shoves message out door
  - sending process relies on transport infrastructure on other side of door to deliver message to socket at receiving process



Prodi Informatika

48

Addressing processes

- to receive messages, process must have *identifier*
- host device has unique 32-bit IP address
- Q: does IP address of host on which process runs suffice for identifying the process?
  - A: no, *many* processes can be running on same host

- *identifier* includes both IP address and port numbers associated with process on host.
- example port numbers:
  - HTTP server: 80
  - mail server: 25
- to send HTTP message to gaia.cs.umass.edu web server:
  - IP address: 128.119.245.12
  - port number: 80
- more shortly…

Application Layer 2-49 Prodi Informatika

49

App-layer protocol defines

- types of messages exchanged,
  - e.g., request, response
- message syntax:
  - what fields in messages & how fields are delineated
- message semantics
  - meaning of information in fields
- rules for when and how processes send & respond to messages

open protocols:

- defined in RFCs
- allows for interoperability
- e.g., HTTP, SMTP

proprietary protocols:

- e.g., Skype

Application Layer 2-50 Prodi Informatika

50

What transport service does an app need?

data integrity

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

timing

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

throughput

- some apps (e.g., multimedia) require minimum amount of throughput to be "effective"
- other apps ("elastic apps") make use of whatever throughput they get

security

- encryption, data integrity, …

Application Layer 2-51 Prodi Informatika

51

Transport service requirements: common apps

application	data loss	throughput	time sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5kbps-1Mbps video:10kbps-5Mbps	yes, 100's msec
stored audio/video	loss-tolerant	same as above	
interactive games	loss-tolerant	few kbps up	yes, few secs
text messaging	no loss	elastic	yes, 100's msec yes and no

Application Layer 2-52 Prodi Informatika

52

Internet transport protocols services

TCP service:

- *reliable transport* between sending and receiving process
- *flow control:* sender won't overwhelm receiver
- *congestion control:* throttle sender when network overloaded
- *does not provide:* timing, minimum throughput guarantee, security
- *connection-oriented:* setup required between client and server processes

UDP service:

- *unreliable data transfer* between sending and receiving process
- *does not provide:* reliability, flow control, congestion control, timing, throughput guarantee, security, or connection setup,

Q: why bother? Why is there a UDP?

Application Layer 2-53 Prodi Informatika

53

Internet apps: application, transport protocols

application	application layer protocol	underlying transport protocol
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (e.g., YouTube), RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP, proprietary (e.g., Skype)	TCP or UDP

Application Layer 2-54 Prodi Informatika

54

Securing TCP

TCP & UDP

- no encryption
- cleartext passwds sent into socket traverse Internet in cleartext

SSL

- provides encrypted TCP connection
- data integrity
- end-point authentication

SSL is at app layer

- apps use SSL libraries, that "talk" to TCP

SSL socket API

- cleartext passwords sent into socket traverse Internet encrypted

p

Application Layer 2-55 Prodi Informatika

55