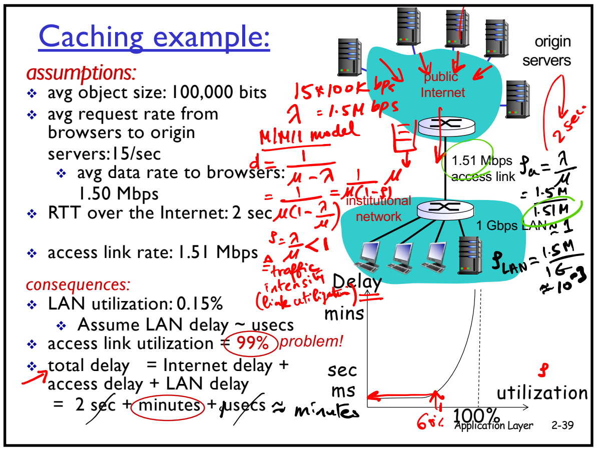


Caching example:

assumptions:

- avg object size: 100,000 bits
- avg request rate from browsers to origin servers:15/sec
  - avg data rate to browsers: 1.50 Mbps
- RTT over the Internet: 2 sec
- access link rate: 1.51 Mbps

consequences:

- LAN utilization: 0.15%
  - Assume LAN delay ~ usecs
- access link utilization = 99% *problem!*
- total delay = Internet delay + access delay + LAN delay
  = 2 ~~sec~~ + minutes + ~~usecs~~ ≈ minutes

$15 \times 100K$ bps
$\lambda = 1.5M$ bps

M/M/1 model

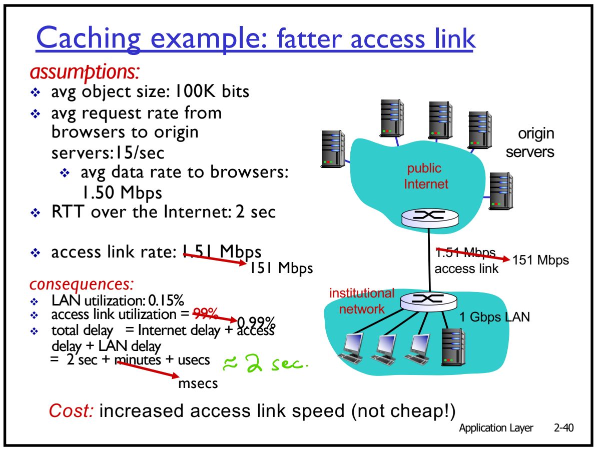
$d = \frac{1}{\mu - \lambda} = \frac{1}{\mu(1-\frac{\lambda}{\mu})} = \frac{1}{\mu(1-\rho)}$

$\rho = \frac{\lambda}{\mu} < 1$
$\triangleq$ traffic intensity (link utilization)

$\rho_a = \frac{\lambda}{\mu} = \frac{1.5M}{1.51M} \approx 1$

$\rho_{LAN} = \frac{1.5M}{1G} \approx 10^{-3}$

39



Caching example: fatter access link

assumptions:

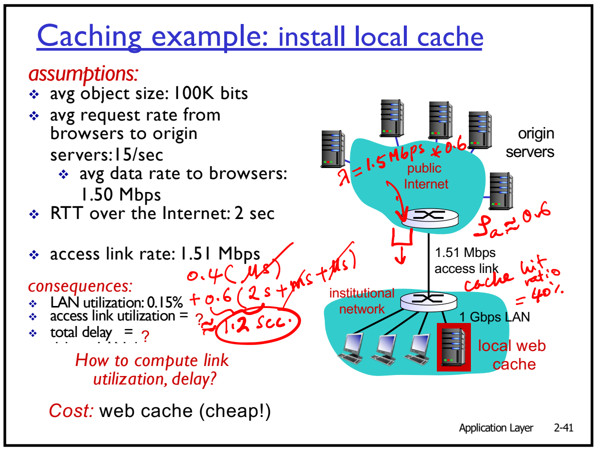
- avg object size: 100K bits
- avg request rate from browsers to origin servers:15/sec
  - avg data rate to browsers: 1.50 Mbps
- RTT over the Internet: 2 sec
- access link rate: ~~1.51 Mbps~~ → 151 Mbps

consequences:

- LAN utilization: 0.15%
- access link utilization = ~~99%~~ → 0.99%
- total delay = Internet delay + access delay + LAN delay
  = 2 sec + ~~minutes~~ + usecs ≈ 2 sec.
  → msecs

Cost: increased access link speed (not cheap!)

40



Caching example: install local cache

assumptions:

- avg object size: 100K bits
- avg request rate from browsers to origin servers:15/sec
  - avg data rate to browsers: 1.50 Mbps
- RTT over the Internet: 2 sec
- access link rate: 1.51 Mbps

consequences:

- LAN utilization: 0.15%
- access link utilization = ?
- total delay = ?

0.4 (~~μs~~)
+ 0.6 (2 s + ms + ~~μs~~)
≈ 1.2 sec.

How to compute link utilization, delay?

Cost: web cache (cheap!)

$\lambda = 1.5$ Mbps $\times 0.6$

$\rho_a \approx 0.6$

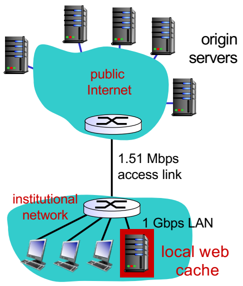
cache hit ratio = 40%

41

Caching example: install local cache

Calculating access link utilization, delay with cache:

- suppose cache hit rate is 0.4
  - 40% requests satisfied at cache, 60% requests satisfied at origin

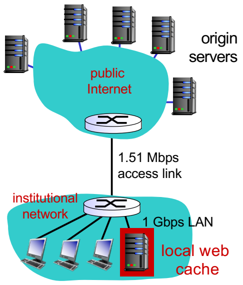


42

Caching example: install local cache

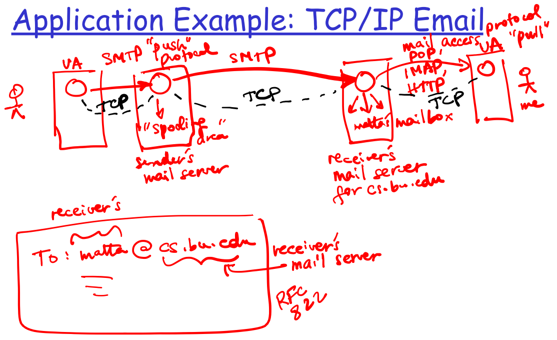
Calculating access link utilization, delay with cache:

- suppose cache hit rate is 0.4
  - 40% requests satisfied at cache, 60% requests satisfied at origin
- access link utilization:
  - 60% of requests use access link
- data rate to browsers over access link = 0.6*1.50 Mbps = .9 Mbps
  - utilization = 0.9/1.51 = .6
  - Assume access delay ~ 700ms
- total delay
  - = 0.6 * (delay from origin servers) +0.4 * (delay when satisfied at cache)
  - = 0.6 (~ 2.7) + 0.4 (~usecs)
  - = ~ 1.6 secs
  - less than with 151 Mbps link (and cheaper too!)



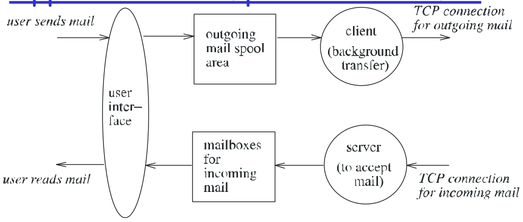
43

Application Example: TCP/IP Email



44

Application Example: TCP/IP Email


user sends mail
user inter– face
outgoing mail spool area
client (background transfer)
TCP connection for outgoing mail
mailboxes for incoming mail
server (to accept mail)
TCP connection for incoming mail
user reads mail

- ❑ Uses client-server communication
- ❑ Not interactive; transfer of e-mail messages takes place in the background (``spooling'')
- ❑ Reliable service; uses TCP (server port 25)

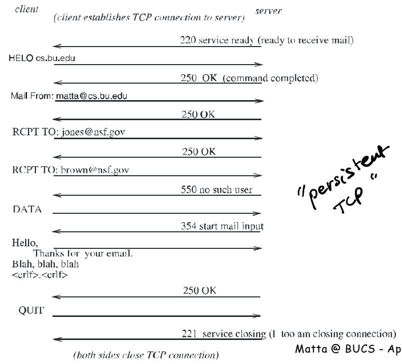
45

TCP/IP Email's Standards

- ❑ RFC 822
  - specifies the format of the e-mail message
  - header and body separated by a blank line
  - header consists of keyword-value pairs: To:, FROM:, ...
- ❑ SMTP (Simple Mail Transfer Protocol)
  - command-response protocol that defines the exchange of mail between client and server
- ❑ MIME (Multipurpose Internet Mail Extensions)
  - Encodes arbitrary data (e.g. binary image) in plain ASCII text
  - SMTP supports only ASCII messages

46

SMTP Client-server Commands



(client establishes TCP connection to server)

220 service ready (ready to receive mail)

HELO cs.bu.edu

250 OK (command completed)

Mail From: matta@cs.bu.edu

250 OK

RCPT TO: jones@nsf.gov

250 OK

RCPT TO: brown@nsf.gov

550 no such user

DATA

354 start mail input

Hello,
Thanks for your email.
Blah, blah, blah
<crlf>.<crlf>

250 OK

QUIT

221 service closing (I too am closing connection)

(both sides close TCP connection)

"persistent TCP"

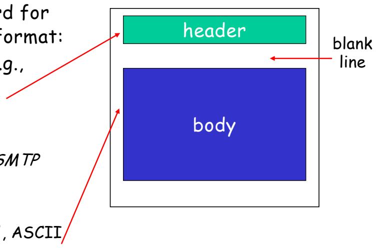
47

Mail message format

RFC 822: standard for text message format:

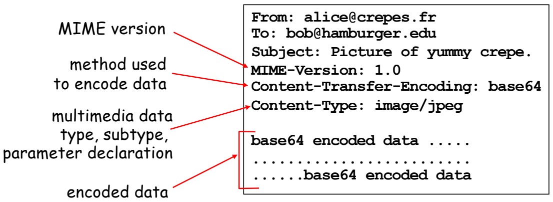
- header lines, e.g.,
  - To:
  - From:
  - Subject:

  different from *SMTP commands*!
- body
  - the “message”, ASCII characters only



Message format: MIME extension

- MIME: Multipurpose Internet Mail Extensions, RFC 2045, 2056
- additional lines in message header declare MIME content type
- A message can have multiple parts, eg text, image, etc.



MIME version

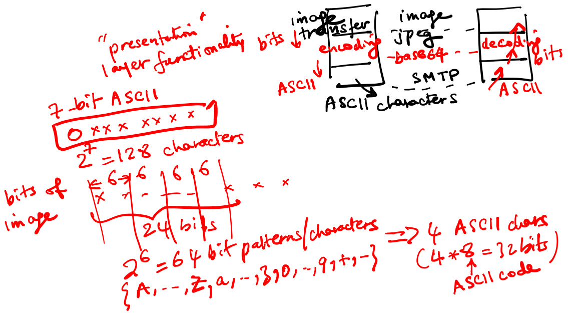
method used to encode data

multimedia data type, subtype, parameter declaration

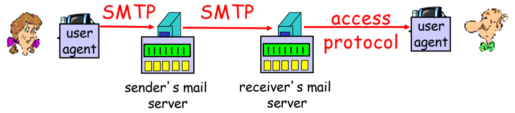
encoded data

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.........................
......base64 encoded data
```



Mail Access Protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]
    - authorization (agent <-->server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored msgs on server
  - HTTP: gmail, Hotmail , Yahoo! Mail, etc.

51

POP3 protocol

authorization phase

- client commands:
  - **user**: declare username
  - **pass**: password
- server responses
  - **+OK**
  - **-ERR**

transaction phase, client:

- **list**: list message numbers
- **retr**: retrieve message by number
- **dele**: delete
- **quit**

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

"download-and-delete"

POP is stateless

52

POP3 (more) and IMAP

More about POP3

- Previous example uses "download and delete" mode
- Bob cannot re-read e-mail if he changes client
- "Download-and-keep": copies of messages on different clients
- POP3 is stateless across sessions

IMAP

POP + "state" @ server

- Keep all messages in one place: the server
- Allows user to organize messages in folders
- IMAP keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name

53

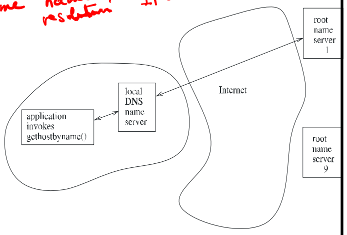
DNS: Internet Domain Name System

$\underbrace{\text{foo.cs.bu.edu}}_{\text{hostname}} \xrightarrow{} \underbrace{128.50.1.3}_{\text{IP address}}$

name resolution

1) how names are structured?

2) how to maintain mapping info & access it efficiently?



Matta @ BUCS - Applications 1-54

54