

CSC104 Winter 2013

Why and how of computing
week 11

Danny Heap

heap@cs.toronto.edu

BA4270 (behind elevators)

<http://www.cdf.toronto.edu/~heap/104/F12/>

416-978-5899

Text: **Picturing Programs**

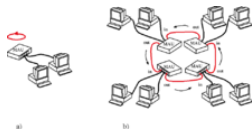
Outline

networks

Notes

roundly connected

ring topology



wait for the token

centrally connected

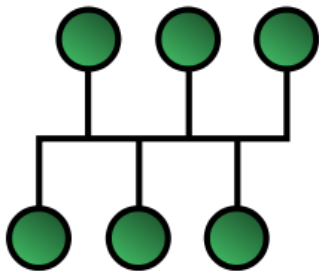
star configuration



server runs things

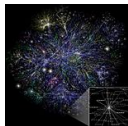
all connected

bus configuration

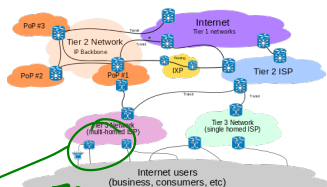


cooperate and back off

network of networks



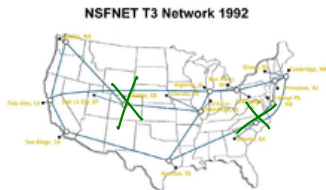
local networks
interconnected by
gateways
(click images)



FG
+ teach
station

gate way
eg ccb-r-fitz

the "medieval internet"



robust:
route-finding
protocol
TCP/IP

- ▶ email — correspond with colleagues.
- ▶ file transfer protocol — move files of info over distance
- ▶ Network File Service
- ▶ Tim Berners-Lee: WWW impossible without open protocols — not proprietary

now the internet \approx WWW

antipodal clicking...

Some "users"
(large media
corps)
more control
than



- ▶ where's the content?
- ▶ how's it move around
- ▶ who's in charge?

→ mostly in Australia,
→ from host → host
hopping along
the internet

each
host & user
has some

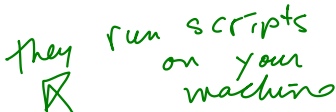
control

no single authority although
ICANN controls some TLDs



can you believe the web?

they run scripts
on your
machine



- ▶ Every browser session requires an **exchange of trust**, from both sides
- ▶ There is a combination of broad-based publishing and powerful media interests
- ▶ The jury is still out on accuracy, for example **Wikipedia versus Britannica**

flatten

```
; flatten : list -> list
(define (flatten L)
  (cond
    [(cons? L) (apply append (map flatten L))]
    [else (list L)]))

; predict what (flatten 3) does

; predict what (flatten (list 3)) does

; predict what (flatten (list 1 2 (list 3))) does
```

depth

```
; depth : list -> number
(define (depth L)
  (cond
    [(cons? L) (+ 1 (apply max (map depth L)))]
    [else 0]))

; predict what (depth 3) does

; predict what (depth (list 3 4)) does

; predict what (depth (list 3 4 (list 5 6))) does
```

Notes