# QoS Control and Internet Traffic

QoS (quality of service) control

- feedback traffic control
  - → closed-loop/adaptive control
  - → small time scale: msec
  - → end-to-end and/or with router support
- resource provisioning
  - → open-loop control: allocate/reserve resources in advance
  - → large time scale: seconds, minutes, and longer
  - → router-based control with admission control at edge

Packet scheduling:

→ at routers/switches

→ default: FIFO with drop-tail packet drop

Granularity

- per-flow scheduling

  → e.g., single TCP session, UDP streaming session

- aggregate-flow scheduling

  → classes of traffic: e.g., platinum, gold, silver, bronze

Scheduling algorithms:

- priority scheduling

  → each flow or class assigned priority

  → simple but can lead to starvation

- weighted fair queueing

  → assign each flow/class a weight

  → generalizes round-robin

  → prevents starvation

  → requires traffic rate control to maintain service differentiation

- normalized weighted fair queueing

  → imposes strict ordering of service differentiation

- real-time scheduling

  → e.g., EDF (earliest deadline first)

QoS control:

- per-user (or flow) reservation for super-quality service

  $\rightarrow$ guaranteed service

- shared service classes (platinum, gold, silver, bronze) for good service

  $\rightarrow$ differentiated service

Internet standards:

- IETF IntServ

  $\rightarrow$ RSVP protocol

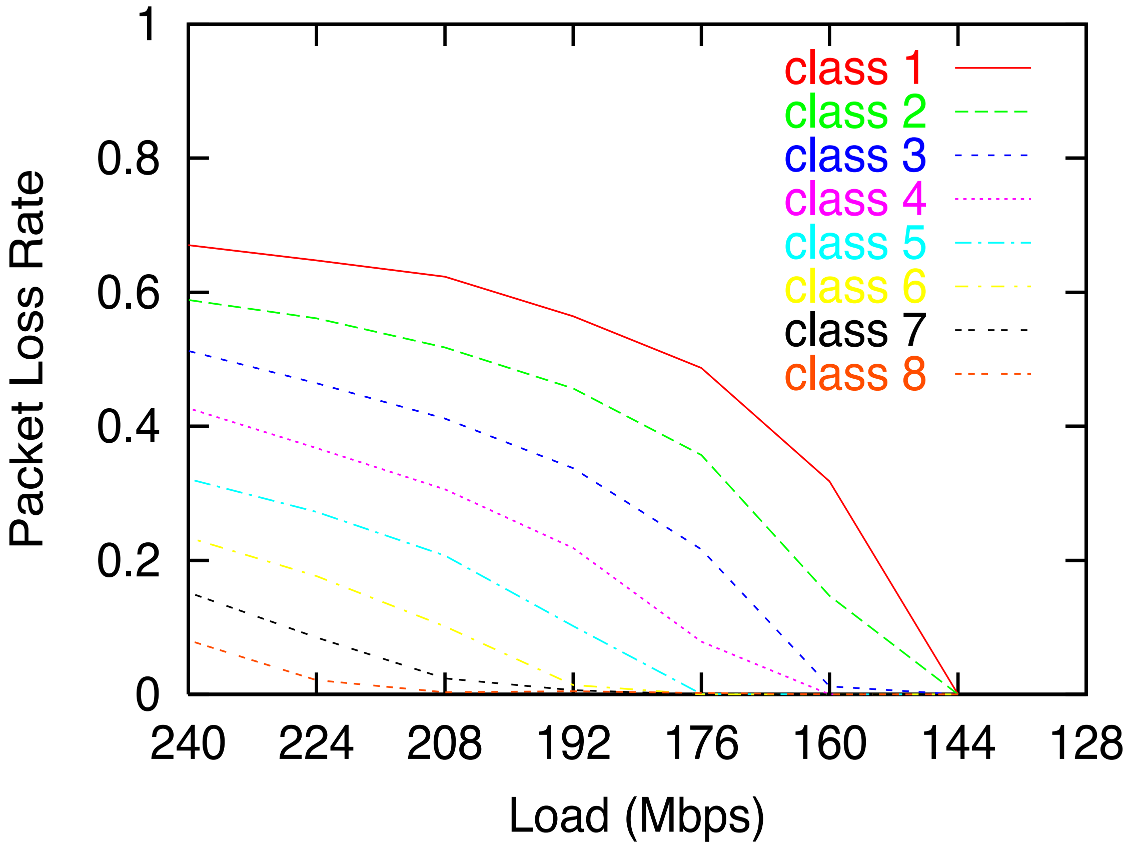
  $\rightarrow$ analogous to leasing a line

- IETF DiffServ

  $\rightarrow$ different types of router behavior

  $\rightarrow$ AF, EF, Cisco's LLQ for VoIP

Cisco 7206VXR router: packet loss rate

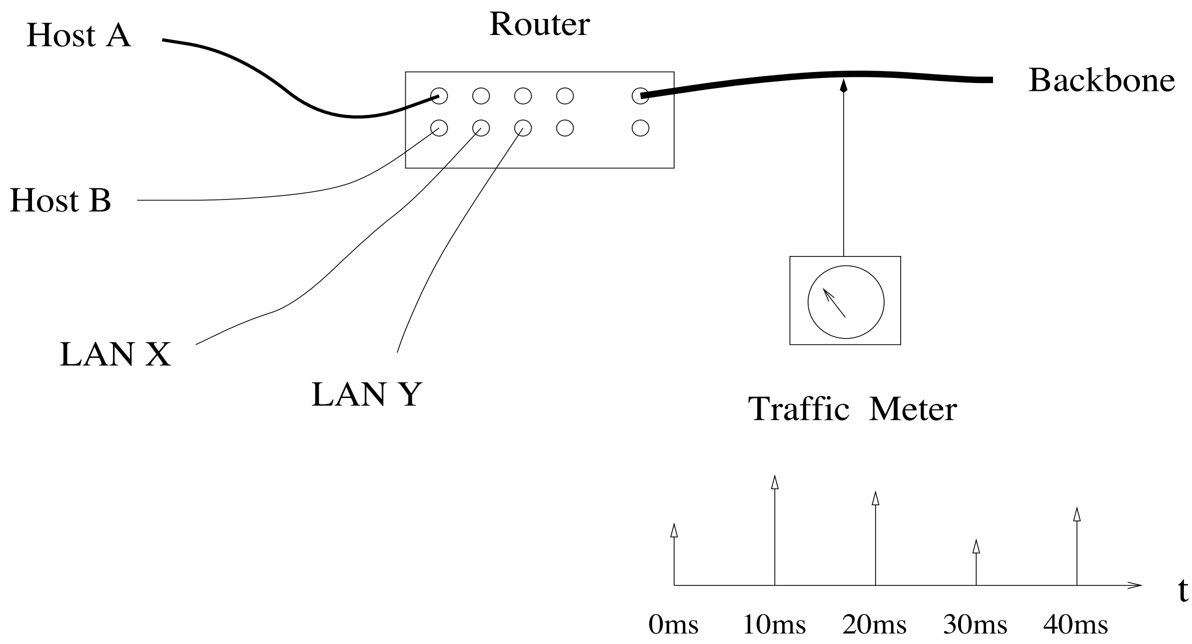



⟶ 8 classes

⟶ OC-3 link

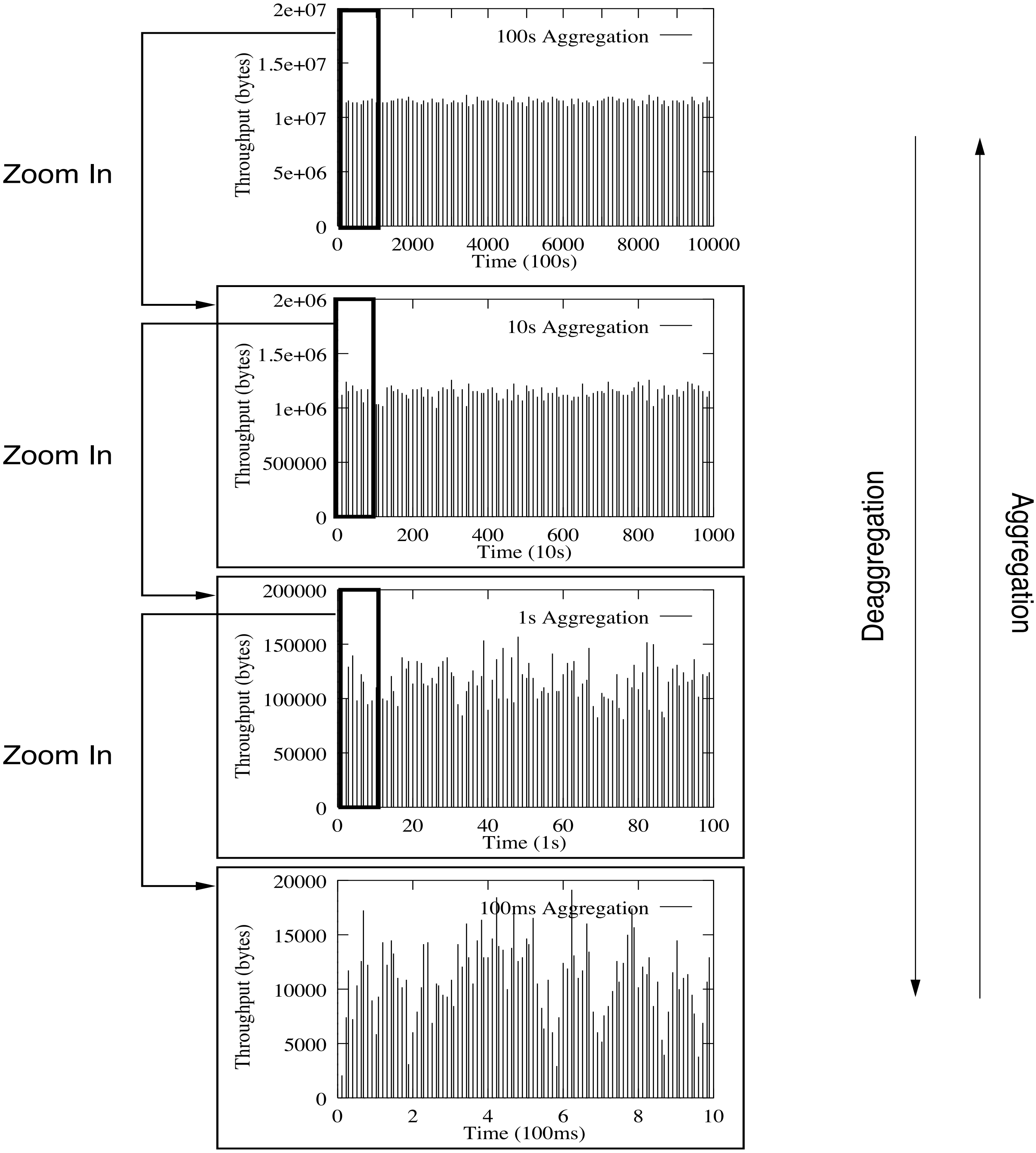
⟶ varying offered load

What does Internet traffic look like?




$\longrightarrow$ logging

$\longrightarrow$ traffic time series (at 10ms granularity)



Zoom In
Zoom In
Zoom In
100s Aggregation
Throughput (bytes)
2e+07
1.5e+07
1e+07
5e+06
0
0
2000
4000
6000
8000
10000
Time (100s)
10s Aggregation
Throughput (bytes)
2e+06
1.5e+06
1e+06
500000
0
0
200
400
600
800
1000
Time (10s)
1s Aggregation
Throughput (bytes)
200000
150000
100000
50000
0
0
20
40
60
80
100
Time (1s)
100ms Aggregation
Throughput (bytes)
20000
15000
10000
5000
0
0
2
4
6
8
10
Time (100ms)
Deaggregation
Aggregation

Aggregated traffic becomes flat

- $\longrightarrow$ "flat is good" rule for QoS provisioning
- $\longrightarrow$ bandwidth dimensioning
- $\longrightarrow$ technically: law of large numbers in action
- $\longrightarrow$ not correlated in time
- $\longrightarrow$ efficient and happy customers

Also aggregation over multiple users

- $\longrightarrow$ called statistical multiplexing
- $\longrightarrow$ assuming independence between different users
- $\longrightarrow$ nice normal distribution shape
- $\longrightarrow$ allows non-peak reservation
- $\longrightarrow$ consider peak-to-mean ratio: less costly

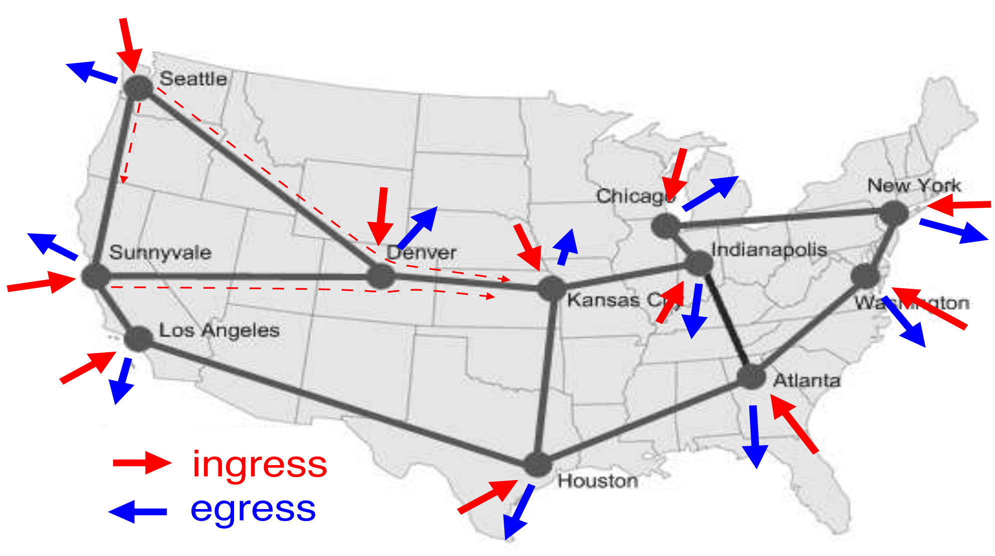
- LLN: principal engineering tool used by large transit providers and large access providers

  → "largeness" is key

  → even though components are random, system is well-behaved and predictable

  → apply at ingress/egress and backbone links

  → measurement-based tool: traffic matrix



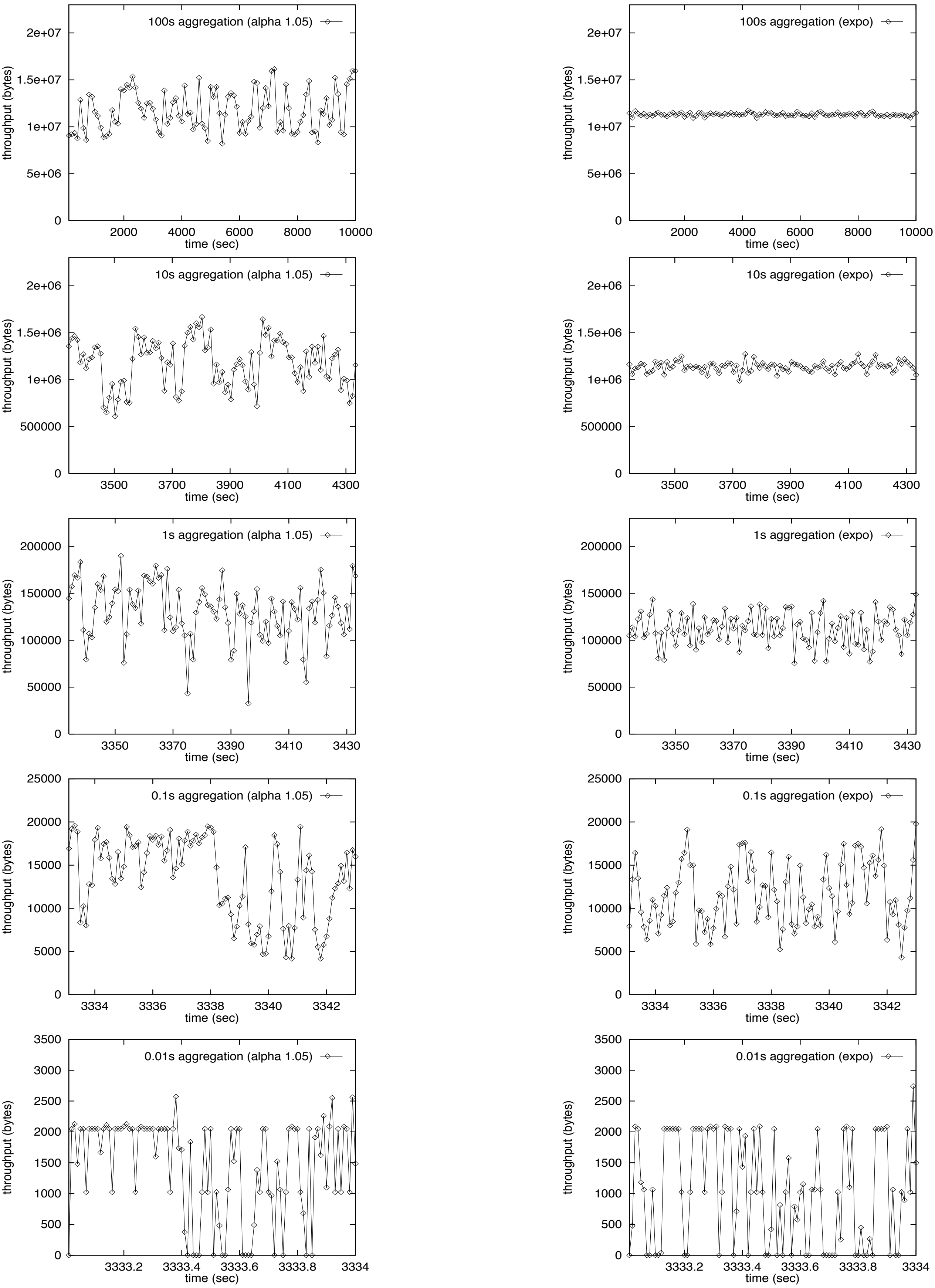
Back to Internet traffic:

$\longrightarrow$ what does it look like?

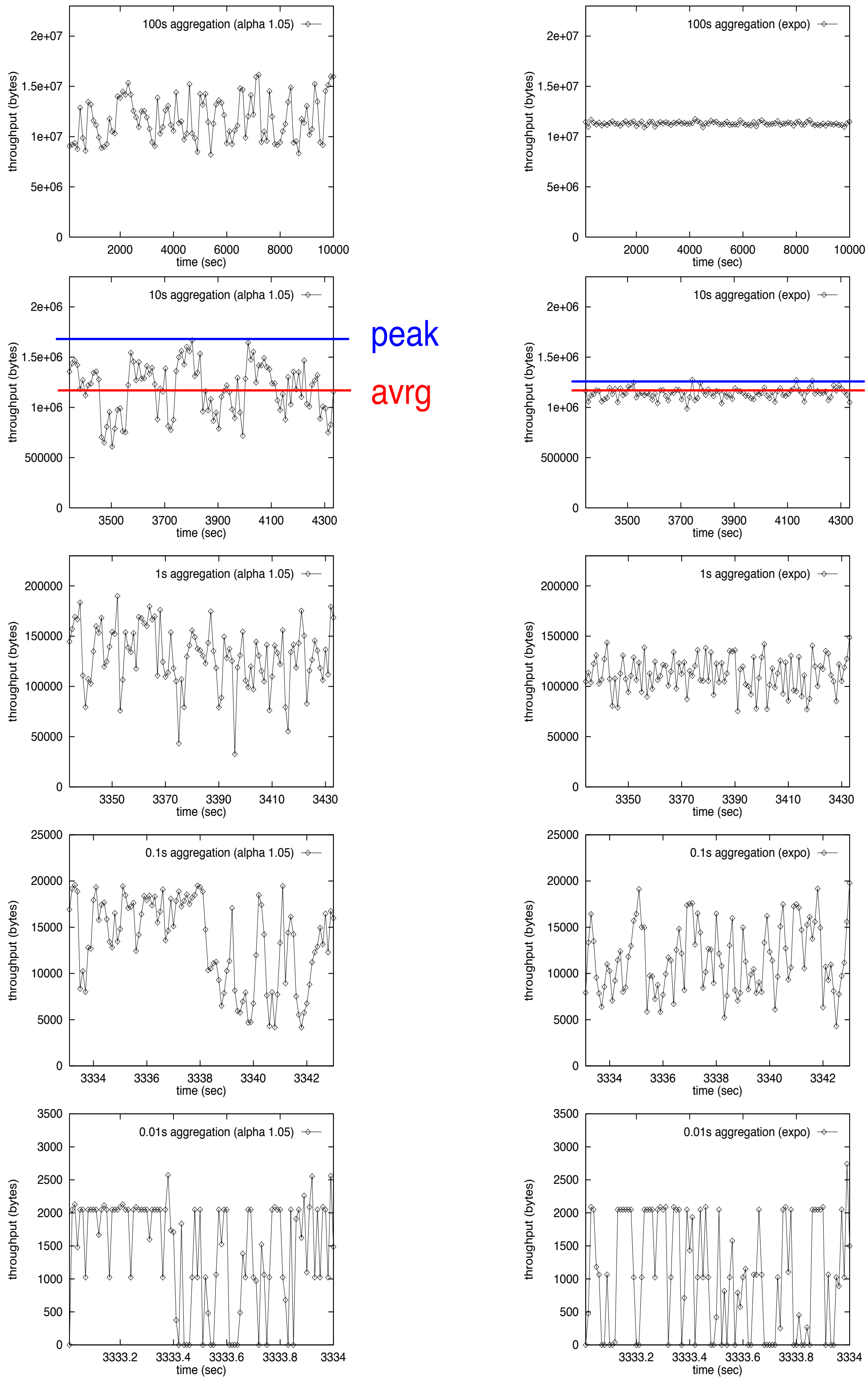
$\longrightarrow$ doesn't become flat with time aggregation

$\longrightarrow$ stays bursty!

$\longrightarrow$ in a peculiar fashion: self-similar or fractal


100s aggregation (alpha 1.05)
100s aggregation (expo)
10s aggregation (alpha 1.05)
10s aggregation (expo)
1s aggregation (alpha 1.05)
1s aggregation (expo)
0.1s aggregation (alpha 1.05)
0.1s aggregation (expo)
0.01s aggregation (alpha 1.05)
0.01s aggregation (expo)
throughput (bytes)
time (sec)

**Internet: self-similar** **Telephony: Poisson-like**



Consequences:

$\longrightarrow$ cannot use "flat is good" method anymore

$\longrightarrow$ intrinsic trade-off between QoS and efficiency

$\longrightarrow$ bad news for QoS provisioning

$\longrightarrow$ traffic must be correlated in time (why?)

We know session arrivals are (approximately) Poisson; what about session lifetimes?

Important fact: TCP session lifetimes are heavy-tailed

$\longrightarrow$ $\Pr\{Z > x\} \approx x^{-\alpha}$

$\longrightarrow$ as opposed to: $\Pr\{Z > x\} \approx e^{-bx}$

$\longrightarrow$ exponent: $1 < \alpha < 2$ (closer to 1)

$\longrightarrow$ note: different from Internet connectivity power-law

$\longrightarrow$ much more likely session will last a long time

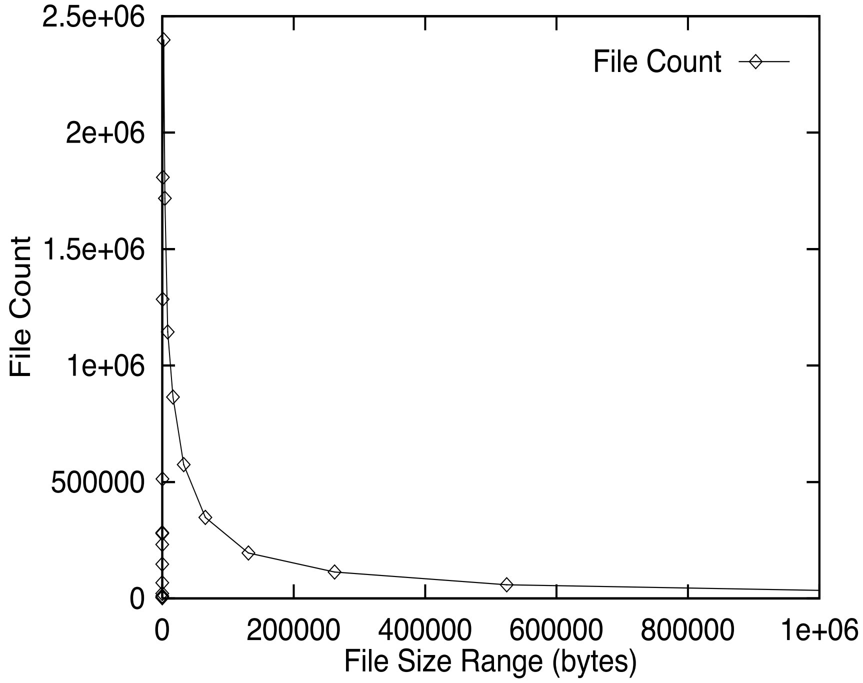
$\longrightarrow$ has finite mean but infinite variance

Why would TCP session lifetimes be heavy-tailed?

$\longrightarrow$ TCP traffic makes up bulk of Internet traffic

Tale of elephants and mice:

⟶ UNIX and WWW file systems




⟶ many small ones (mice)

⟶ a few very large ones (elephants)

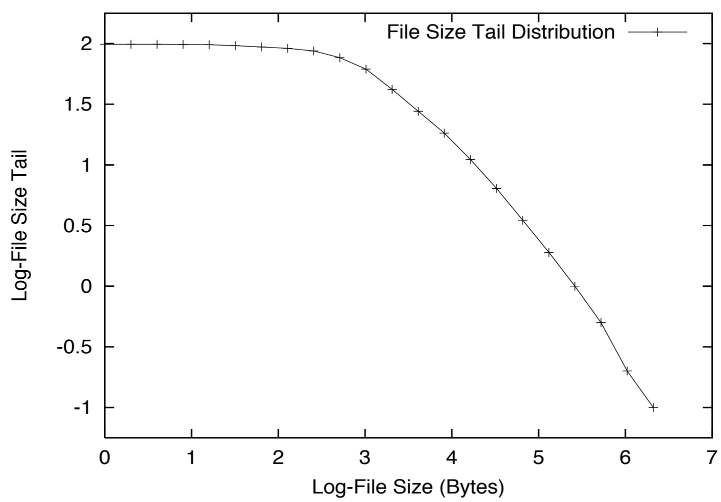
⟶ 10% consumes 90% of bandwidth

How to check if files sizes are heavy-tailed?

Since $\Pr\{Z > x\} \approx x^{-\alpha}$, take logarithm on both sides:

$\longrightarrow$ $\log \Pr\{Z > x\} \approx -\alpha \log x$

$\longrightarrow$ linear function with negative slope $-\alpha$



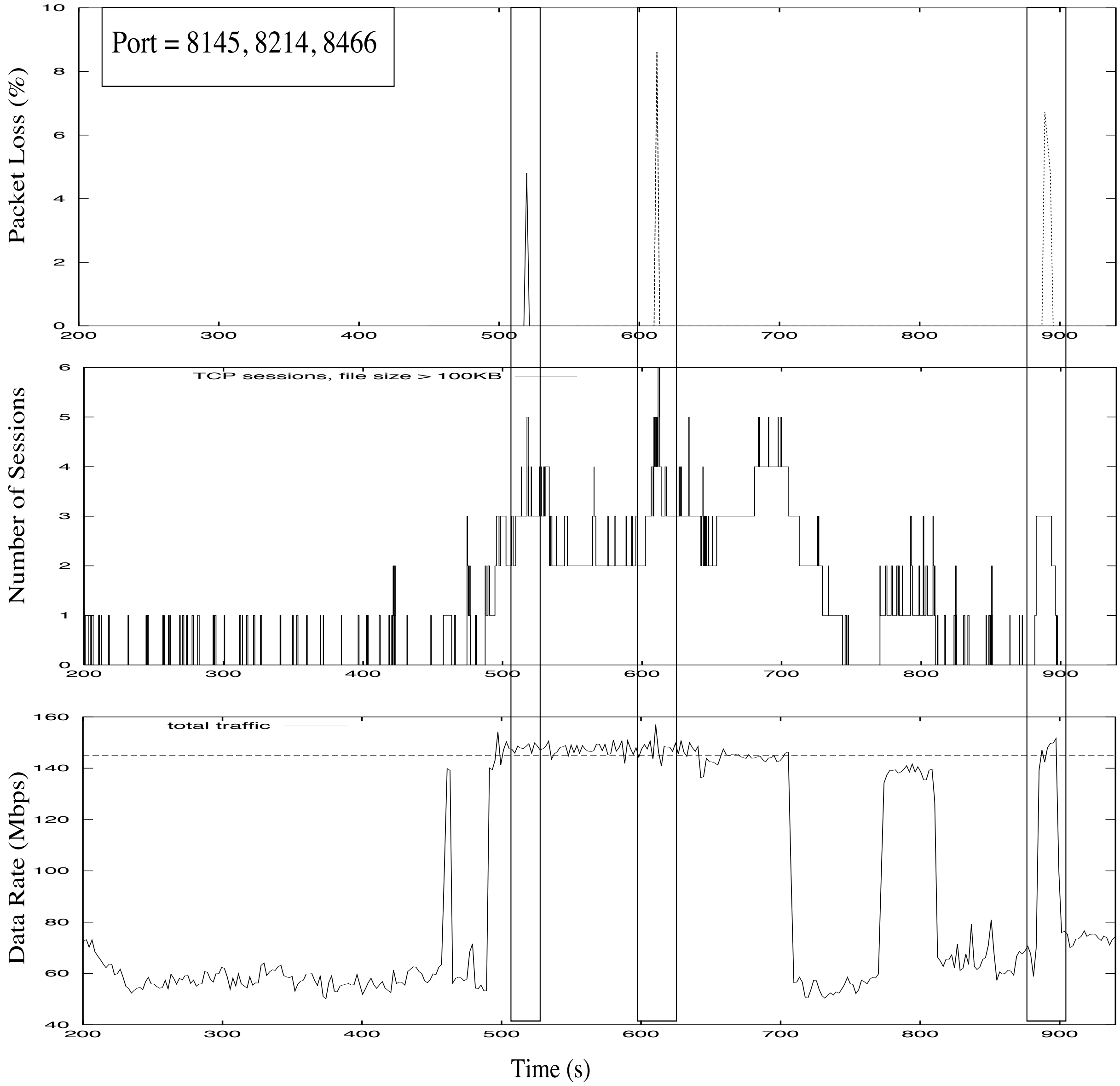

$\longrightarrow$ holds true for large $x$

$\longrightarrow$ what's the slope $\alpha$?

$\longrightarrow$ we don't care about details of small sizes (why?)

Elephants in action:

⟶ at backbone router



Can the problem be solved?

⟶ no: as long as elephants and mice holds

Turns out to be a wide-spread phenomenon is sociology, networks, and elsewhere

⟶ size (population) of cities

⟶ popularity

⟶ frequency of words in books

⟶ etc.

In the real world:

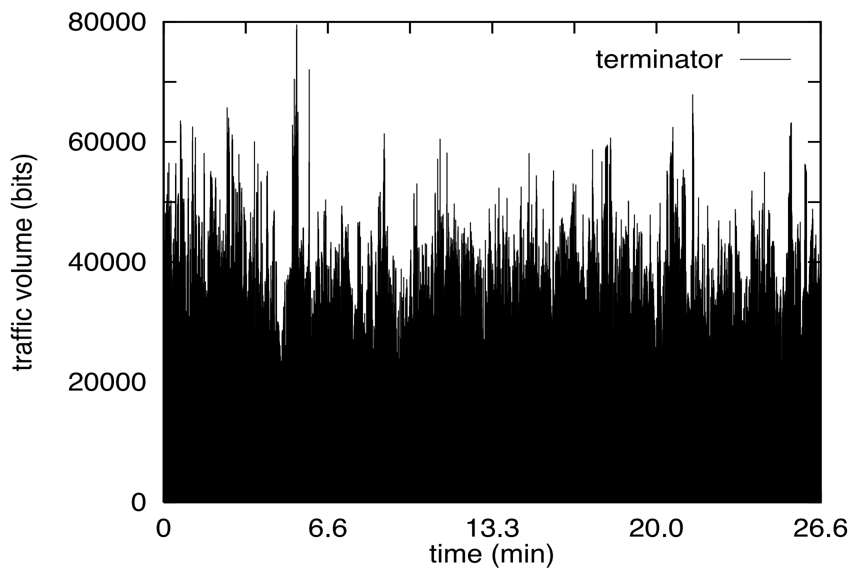
⟶ norm: skewed distribution of sizes

⟶ power-law

Bandwidth allocation rule of thumb: 50% overprovision

## Real-time VBR source profile

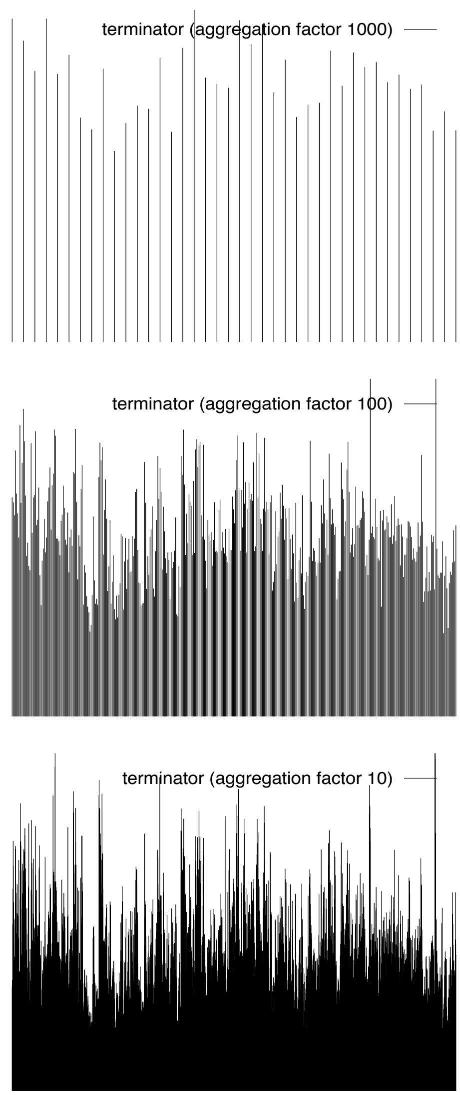
Consider traffic profile of compressed video (e.g., MPEG)




$\longrightarrow$ periodic real-time application

$\longrightarrow$ period $T = 1/f$ ($f$: frame rate)

# Burstiness structure:




⟶ burstiness persists across multiple time scales

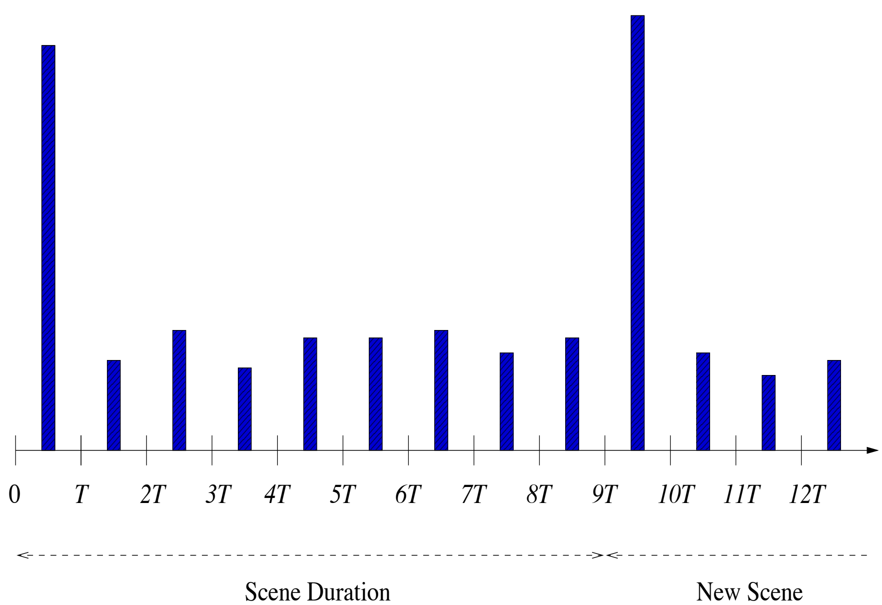
Possible causes:

- Heavy-tailed scene durations

  $\longrightarrow$ facilitates inter-frame compression

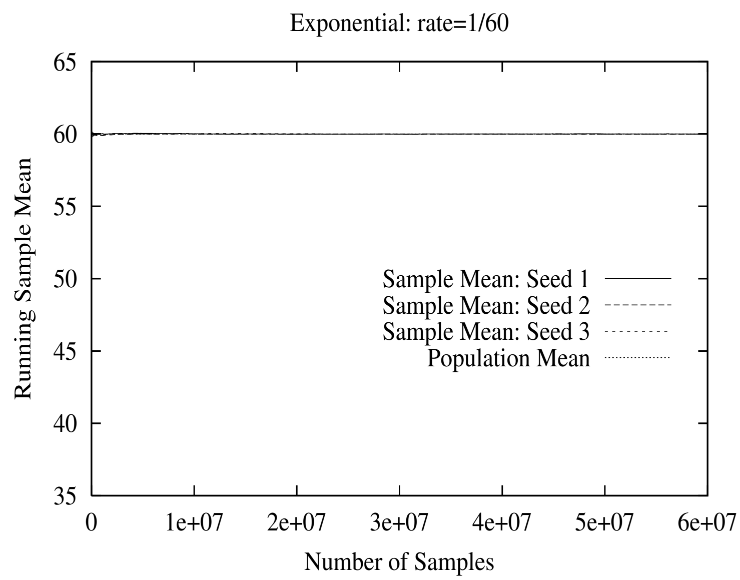
- Repeating GOP pattern

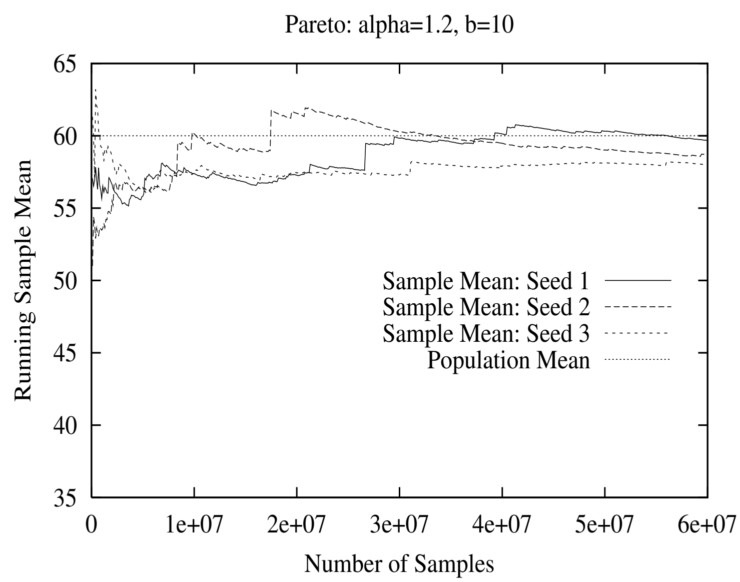
  $\longrightarrow$ e.g., I B B P B B



# Note on sampling and convergence.

## Sample mean convergenge rate: exponential vs. Pareto




extremely slow convergence: concern when generating artificial workload