ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

IKEA
Departement Informationstechnologie
und Elektrotechnik

On Flow Concurrency in the Internet and its Implications for Capacity Sharing

Brian Trammell and Dominik Schatzmann
Communications Systems Group, ETH Zürich

What's a measurement guy doing at a capacity-sharing workshop?

- Concern about flow state requirements in audit
  - Capacity sharing verification at network interconnection points necessary for algorithms requiring trust between policing and audit
    - expensive → large links, higher concurrency
  - Current capacity-sharing approaches don't need this property
    - independent edge-policing architecture
    - and as long as they don't, remain scalable → small links, lower concurrency
  - But it's always nice to have data to back up these assumptions
- *however…*
- increasing use of flow-state-keeping devices in the network
  - Proliferation of protocols requiring middleboxes. (CGN scares me.)
- Is flow state a new resource requiring fair allocation?

Flow concurrency distribution characteristics

Total Flow Concurrency

network	median	95th	peak
all (/11)	148k	322k	436k
university (5x /18)	3.2k	4.3k	22.9k

- Flow concurrency highly dependent on network type
  - In general, less variable at higher levels of aggregation
- Rule of thumb: ~20k peak per /16
  - adjusted for host type / popularity

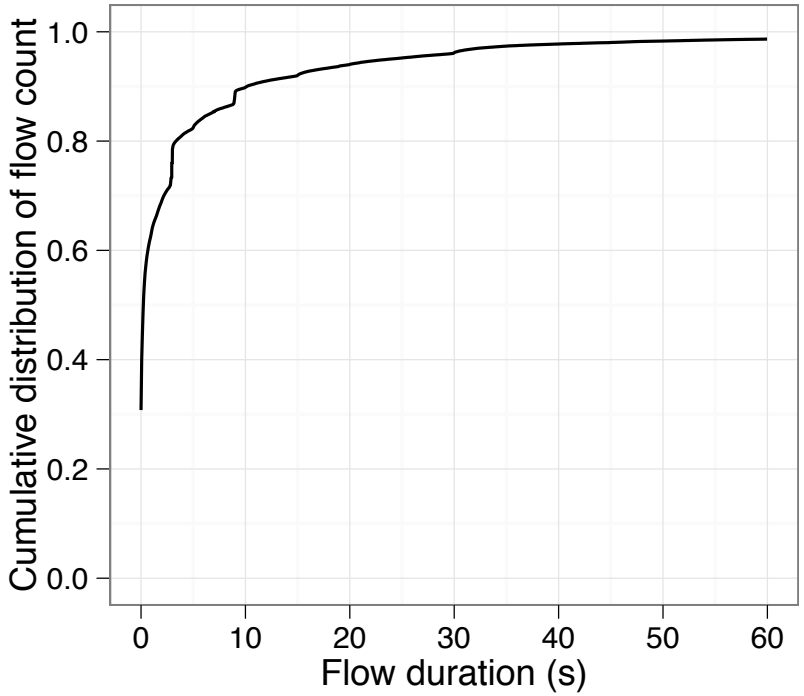
Flow Concurrency per Active Host

Host type	median	95th	peak
clients (13.7k)	5.8	11.8	53.8
servers (13.4k)	10.3	13.4	23.0
CDN (1.25k)	16.5	43.3	49.8
all (2.4M)	5.2	7.7	9.7

- Flow concurrency per active host much more stable per host type
  - (with some noise: 53.8 peak → outbound scan activity)
- 5th percentile client flow concurrency is 3.8 → web client behavior
- Client concurrency a function of behavior
- Server concurrency a function of popularity
- Large-scale rule of thumb: 10 peak per active host.

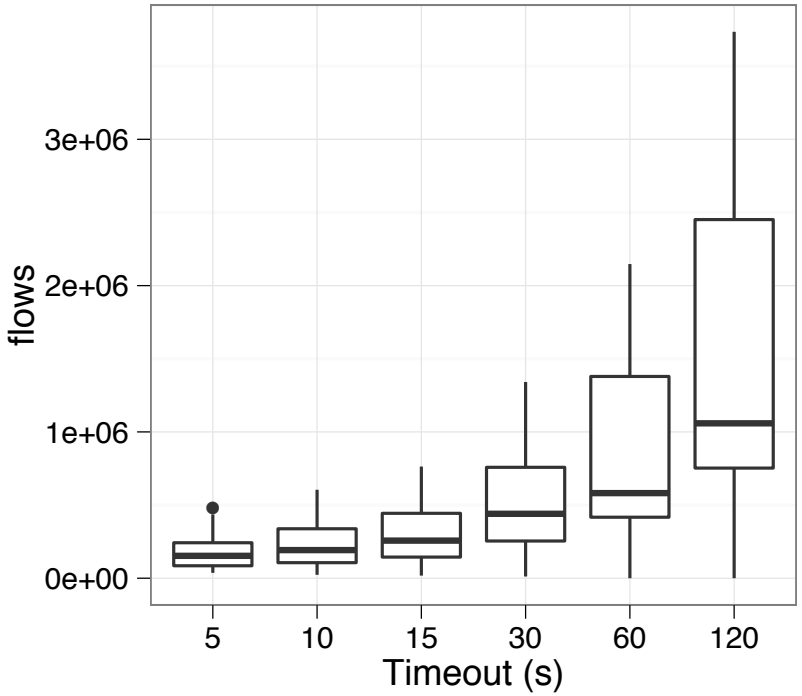
Dominance of short flows

- Flow concurrency is an issue because most flows are short.
- Median flow is ~250ms long
- 23% single-packet flows
- very short flows account for 8% of packets and 5% of bytes…
- …but 52% of flows.



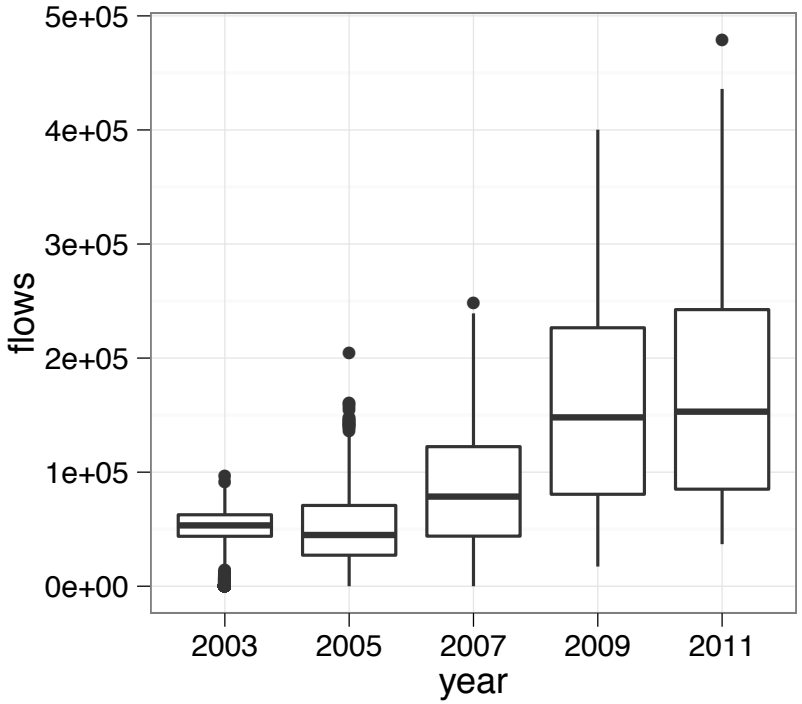
Use aggressive timeouts!

- Dominance of short flows indicates short timeouts can significantly decrease required flow state
- Idle timeouts longer than 15s merely add to state requirements



Development of flow concurrency over time

- Flow concurrency increases with traffic volume.
- Large flows contribute far more to traffic volume than to flow count:
  - Correlation: 0.668



Guidance for edge-policing in capacity sharing

- ~10 peak concurrent flows per active host

- ~12 peak concurrent flows per active *client* (excl. scanning)
- → 200kB per /24 assuming 64B/flow

- Server concurrency depends on popularity
- Even with 100x concurrency, 20MB per /24

- *→ there does not appear to be a problem here*
  - *(but make sure you don't need to police the interconnect)*

Toward flow-state fairness

Decreasing the brittleness of in-network state

- Two solutions to increasing flow concurrency for flow-state keeping devices:
  - Graceful degradation (audit and policing, measurement, etc.)
  - Massive overprovisioning
- Use of devices that must be overprovisioned is increasing in the network
  - Anything with NAT in the name.

- Can capacity-sharing approaches be of use here?

Potential flow-state control schemes

- Temporal offload: delay SYN before forward
  - Most peaks are transient → delay can help ride them out
  - Too much delay leads to retransmit / timeout
  - (Much less delay than this can impact perceived latency)
- Lower-concurrency transport
  - e.g. SPDY: reduce concurrency by opening fewer flows

- Discouragement of flow-state overload
  - Declare flow length in advance, and incentivize longer flows
  - Machinery here looks parallel to conex

Acknowledgements

SWITCH
Serving Swiss Universities

- Thanks to SWITCH for network flow data examined

- mPlane — http://www.ict-mplane.eu/