A brief introduction to Flow-Aware Networking

Jordan Augé <jordan.auge@cl.cam.ac.uk>

NETOS talklets - April 22nd, 2008

UNIVERSITY OF CAMBRIDGE

Introduction

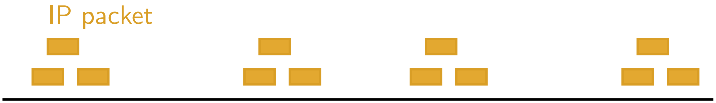
- Overview related to work I have done during my PhD
- Flow-Aware Networking is a concept proposed by James Roberts, my advisor
- **Motivation**: inefficiency and complexity of standardized QoS architectures...
- ... because traffic is hard to characterize (e.g. failure of token bucket)
- **Objective**: define a simple and robust architecture to provide QoS (initially in the backbone)

Introduction

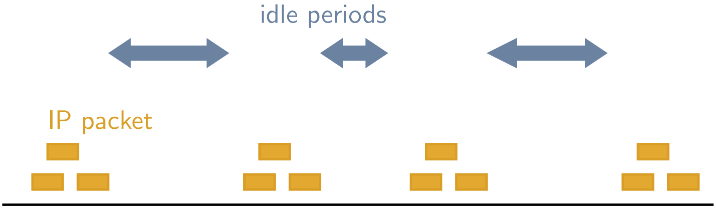
- Overview related to work I have done during my PhD
- Flow-Aware Networking is a concept proposed by James Roberts, my advisor
- **Motivation**: inefficiency and complexity of standardized QoS architectures...
- ... because traffic is hard to characterize (e.g. failure of token bucket)
- **Objective**: define a simple and robust architecture to provide QoS (initially in the backbone)

Observing traffic at different scales

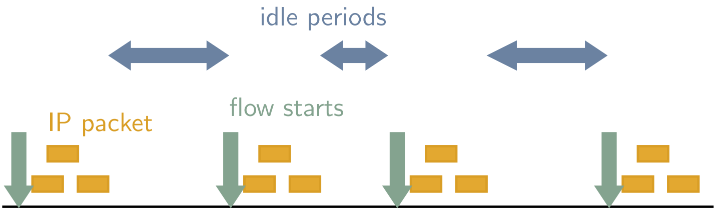
Observing traffic at different scales



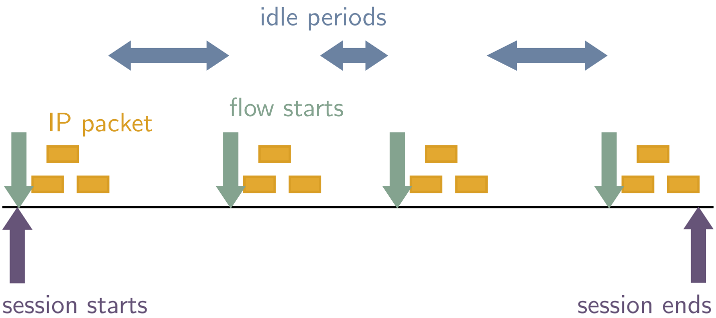
Observing traffic at different scales



Observing traffic at different scales



Observing traffic at different scales



Flow structure of traffic

- Example of flow in IPv4:
  (src/dst IPs, src/dst ports, protocol) + timeout
- Finite size flow arrivals, according to a stochastic process,
- #flows varies: significant characteristic = average link load
- The flow peak rate defines :
  - **bottlenecked flows**: peak rate limited flows: the link only "sees" a packet from time to time
  - **non-bottlenecked flows**: usually high exogeneous rate, share bandwidth thanks to protocols like TCP
- The vast majority of flows are *bottlenecked*...

Flow structure of traffic

- Example of flow in IPv4:
  (src/dst IPs, src/dst ports, protocol) + timeout
- Finite size flow arrivals, according to a stochastic process,
- #flows varies: significant characteristic = average link load
- The flow peak rate defines :
  - **bottlenecked flows**: peak rate limited flows: the link only "sees" a packet from time to time
  - **non-bottlenecked flows**: usually high exogeneous rate, share bandwidth thanks to protocols like TCP
- The vast majority of flows are *bottlenecked*...

Flow structure of traffic

- Example of flow in IPv4:
  (src/dst IPs, src/dst ports, protocol) + timeout
- Finite size flow arrivals, according to a stochastic process,
- #flows varies: significant characteristic = average link load
- The flow peak rate defines :
  - **bottlenecked flows**: peak rate limited flows: the link only "sees" a packet from time to time
  - **non-bottlenecked flows**: usually high exogeneous rate, share bandwidth thanks to protocols like TCP
- The vast majority of flows are *bottlenecked*...

Flow structure of traffic

- Example of flow in IPv4:
  (src/dst IPs, src/dst ports, protocol) + timeout
- Finite size flow arrivals, according to a stochastic process,
- #flows varies: significant characteristic = average link load
- The flow peak rate defines :
  - **bottlenecked flows**: peak rate limited flows: the link only "sees" a packet from time to time
  - **non-bottlenecked flows**: usually high exogeneous rate, share bandwidth thanks to protocols like TCP
- The vast majority of flows are *bottlenecked*...

Flow structure of traffic

- Example of flow in IPv4:
  (src/dst IPs, src/dst ports, protocol) + timeout
- Finite size flow arrivals, according to a stochastic process,
- #flows varies: significant characteristic = average link load
- The flow peak rate defines :
  - **bottlenecked flows**: peak rate limited flows: the link only "sees" a packet from time to time
  - **non-bottlenecked flows**: usually high exogeneous rate, share bandwidth thanks to protocols like TCP
- The vast majority of flows are *bottlenecked*...

Flow structure of traffic

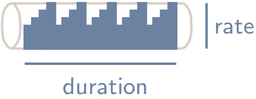
- Example of flow in IPv4:
  (src/dst IPs, src/dst ports, protocol) + timeout
- Finite size flow arrivals, according to a stochastic process,
- #flows varies: significant characteristic = average link load
- The flow peak rate defines :
  - **bottlenecked flows**: peak rate limited flows: the link only "sees" a packet from time to time
  - **non-bottlenecked flows**: usually high exogeneous rate, share bandwidth thanks to protocols like TCP
- The vast majority of flows are *bottlenecked*...

Flow structure of traffic

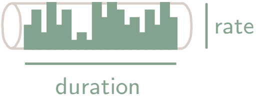
- Example of flow in IPv4:
  (src/dst IPs, src/dst ports, protocol) + timeout
- Finite size flow arrivals, according to a stochastic process,
- #flows varies: significant characteristic = average link load
- The flow peak rate defines :
  - **bottlenecked flows**: peak rate limited flows: the link only "sees" a packet from time to time
  - **non-bottlenecked flows**: usually high exogeneous rate, share bandwidth thanks to protocols like TCP
- The vast majority of flows are *bottlenecked*...

Characterization of traffic: flows as boxes

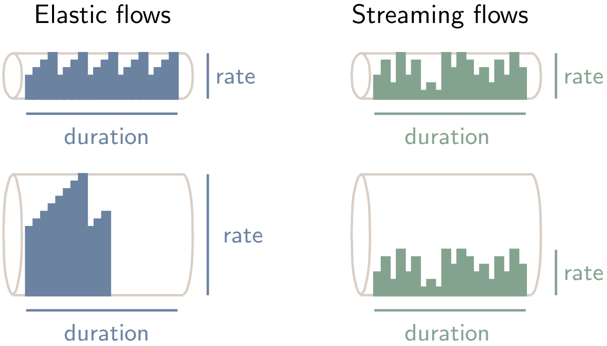
Elastic flows



Streaming flows



Characterization of traffic: flows as boxes



Elastic flows

rate

duration

rate

duration

Streaming flows

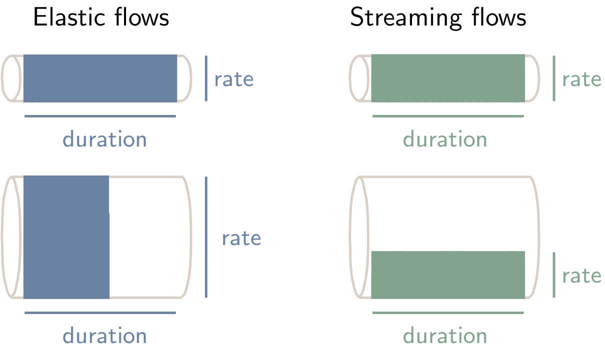
rate

duration

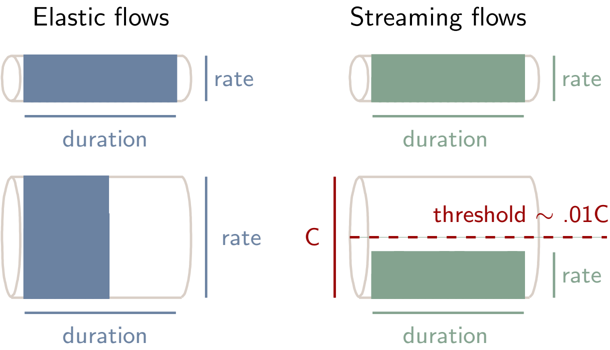
rate

duration

Characterization of traffic: flows as boxes

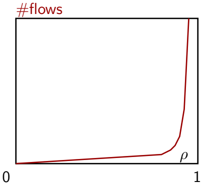


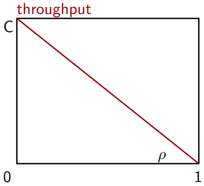
Characterization of traffic: flows as boxes



Elastic (Statistical Bandwidth Sharing)

- Processor Sharing (PS) models : M/M/1/PS queue
- Good approximation of TCP performance
- Insensitivity to flow distributions
  - only depends on the load (ρ = arrival rate x size / C)
  - E[flows in progress] $= \frac{\rho}{1-\rho}$
  - E[throughput] $= C \cdot (1 - \rho)$

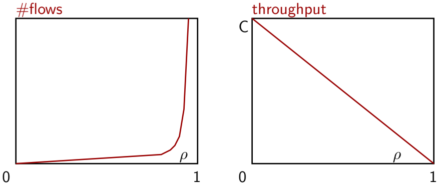




- Small number of flows in normal load but...
- ... in practice $\rho < 0.5$ and E[flows in progress] $= O(10^4)$
- Most of the flows are bottlenecked : limited by their access link for example

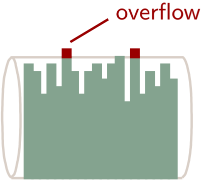
Elastic (Statistical Bandwidth Sharing)

- Processor Sharing (PS) models : M/M/1/PS queue
- Good approximation of TCP performance
- Insensitivity to flow distributions
  - only depends on the load ($\rho =$ arrival rate x size / C)
  - E[flows in progress] $= \frac{\rho}{1-\rho}$
  - E[throughput] $= C \cdot (1 - \rho)$



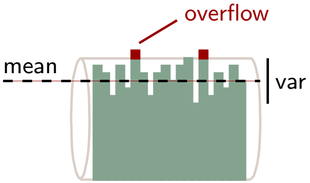
- Small number of flows in normal load but...
- ... in practice $\rho < 0.5$ and E[flows in progress] $= O(10^4)$
- Most of the flows are bottlenecked : limited by their access link for example

Streaming (Bufferless Statistical Multiplexing)



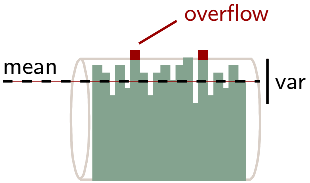
- Controlled performance $\equiv P[\text{input rate} < C] \leq \epsilon$
- Ensure transparent regime for streaming flows
- Performance is insensitive to detailed traffic characteristics
- Locally Poisson arrivals : M/M/1 good approx.
- ex. P[>83 pk] $= 10^{-4}$] for $\rho = 90\%$, that is $\sim$ 1ms with 1Gb/s
- Little scope for differentiation (cf Diffserv)

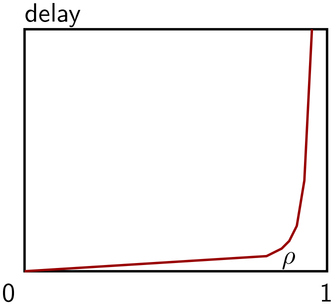
Streaming (Bufferless Statistical Multiplexing)



- Controlled performance $\equiv P[\text{input rate} < C] \leq \epsilon$
- Ensure transparent regime for streaming flows
- Performance is insensitive to detailed traffic characteristics
- Locally Poisson arrivals : M/M/1 good approx.
- ex. P[>83 pk] $= 10^{-4}$] for $\rho = 90\%$, that is $\sim$ 1ms with 1Gb/s
- Little scope for differentiation (cf Diffserv)

Streaming (Bufferless Statistical Multiplexing)

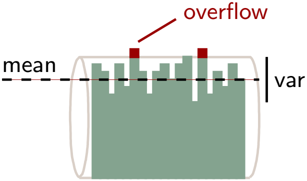


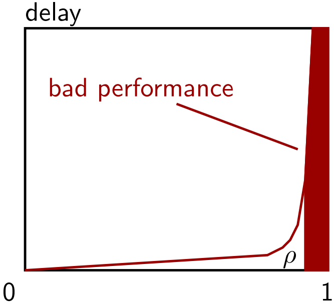


- Controlled performance $\equiv P[\text{input rate} < C] \leq \epsilon$
- Ensure transparent regime for streaming flows
- Performance is insensitive to detailed traffic characteristics

- Locally Poisson arrivals : M/M/1 good approx.
- ex. P[>83 pk] $= 10^{-4}$] for $\rho = 90\%$, that is $\sim$ 1ms with 1Gb/s
- Little scope for differentiation (cf Diffserv)

Streaming (Bufferless Statistical Multiplexing)

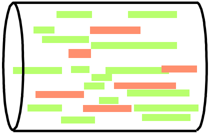




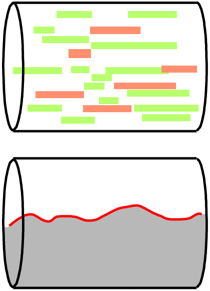
- Controlled performance $\equiv P[\text{input rate} < C] \leq \epsilon$
- Ensure transparent regime for streaming flows
- Performance is insensitive to detailed traffic characteristics

- Locally Poisson arrivals : M/M/1 good approx.
- ex. P[>83 pk] $= 10^{-4}$] for $\rho = 90\%$, that is $\sim$ 1ms with 1Gb/s
- Little scope for differentiation (cf Diffserv)

Link utilization regimes

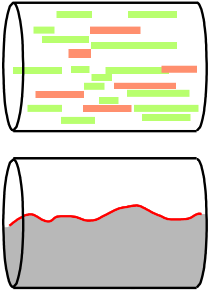


Link utilization regimes



Link utilization regimes

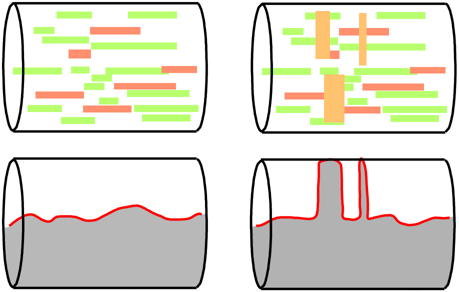
”transparent”



negligible loss
and delay

FIFO sufficient

Link utilization regimes



”transparent”

negligible loss

and delay

FIFO sufficient

”elastic”

excellent for elastic,

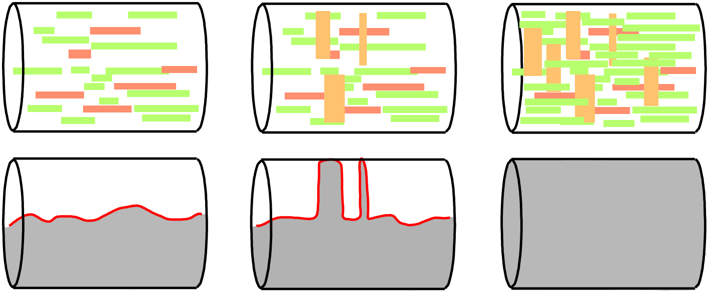
some streaming loss

needs

differenciation

Link utilization regimes

"transparent" "elastic" "congested"



negligible loss and delay

FIFO sufficient

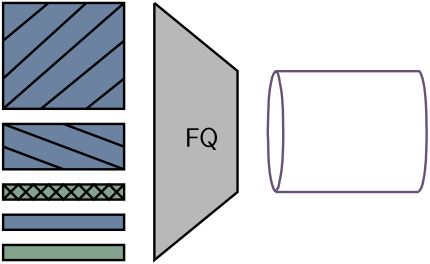
excellent for elastic, some streaming loss

needs differenciation

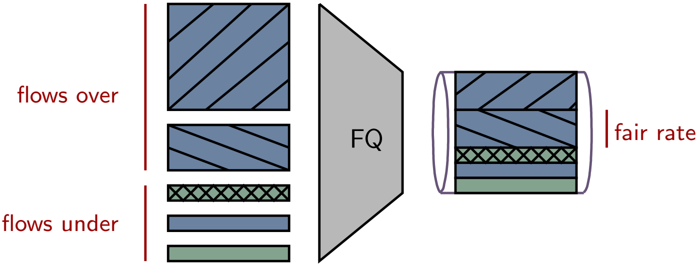
low throughput, significant loss

needs overload control

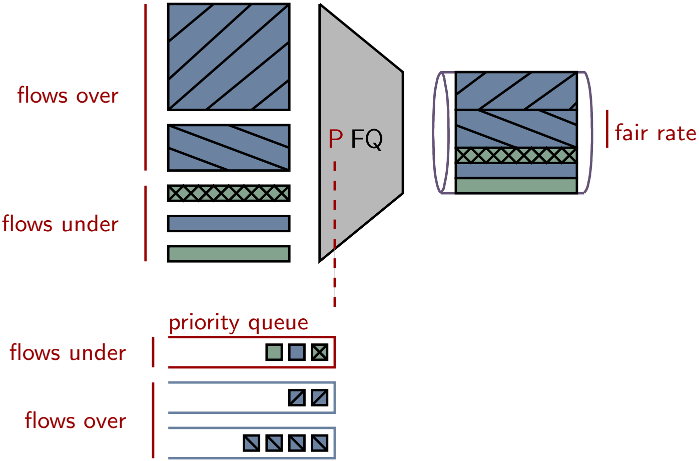
Priority Fair Queueing



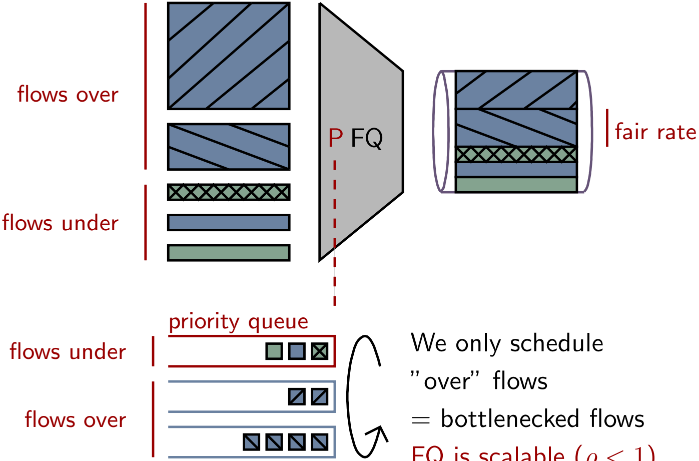
Priority Fair Queueing



Priority Fair Queueing



Priority Fair Queueing

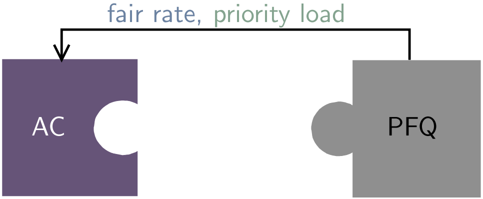


We only schedule "over" flows
= bottlenecked flows

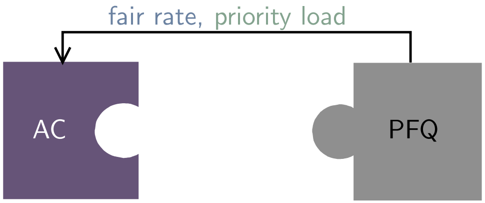
FQ is scalable ($\rho < 1$)

Coupling PFQ with Flow Level Admission Control

Coupling PFQ with Flow Level Admission Control

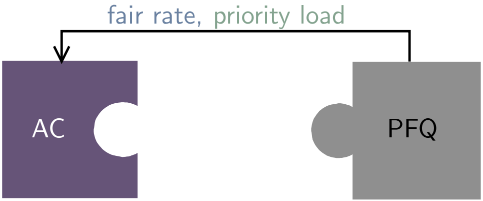


Coupling PFQ with Flow Level Admission Control



🚦 $\text{FR} \geq \theta_{FR}$: QoS elastic (limit nb)

Coupling PFQ with Flow Level Admission Control

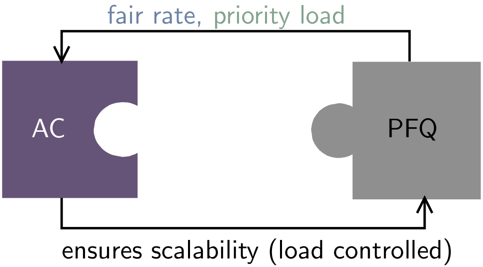


🚦 FR $\geq \theta_{FR}$: QoS elastic (limit nb)

&

🚦 PL $\leq \theta_{PL}$: QoS streaming

Coupling PFQ with Flow Level Admission Control

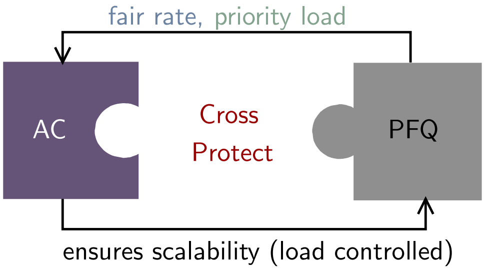


🚦 FR $\geq \theta_{FR}$: QoS elastic (limit nb)

&

🚦 PL $\leq \theta_{PL}$: QoS streaming

Coupling PFQ with Flow Level Admission Control



🚦 $FR \geq \theta_{FR}$: QoS elastic (limit nb)

&

🚦 $PL \leq \theta_{PL}$: QoS streaming

Some work in the FAN context (PhD)

- Simulation (NS-2) and GNU/Linux testbed (XP as a kernel module)
- Some results on a real traffic trace (France Telecom backbone)
- Investigations on the buffer sizing issue for IP routers
- Fair Queueing and TCP performance
- Proposition of a more efficient admission control algorithm for streaming flows
- Thoughts on adapting Cross-Protect for optical networks
- Proposition to introduce Flow-Aware Networking in the access network (Self-Protect) + testbed

Conclusion

- Need to account for the real nature of traffic
- Flow level modelling is efficient
- Important characteristics : load, flow peak rate
- Cross-Protect = Admission Control + Fair Queueing