Flow-Aware Networking: an alternative to QoS

Jim Roberts, France Telecom, Division R&D

http://perso.rd.francetelecom.fr/roberts/

From traffic descriptor to SLS?

- principle of QoS architectures
  - based on a traffic descriptor,
  - satisfy the terms of an SLS

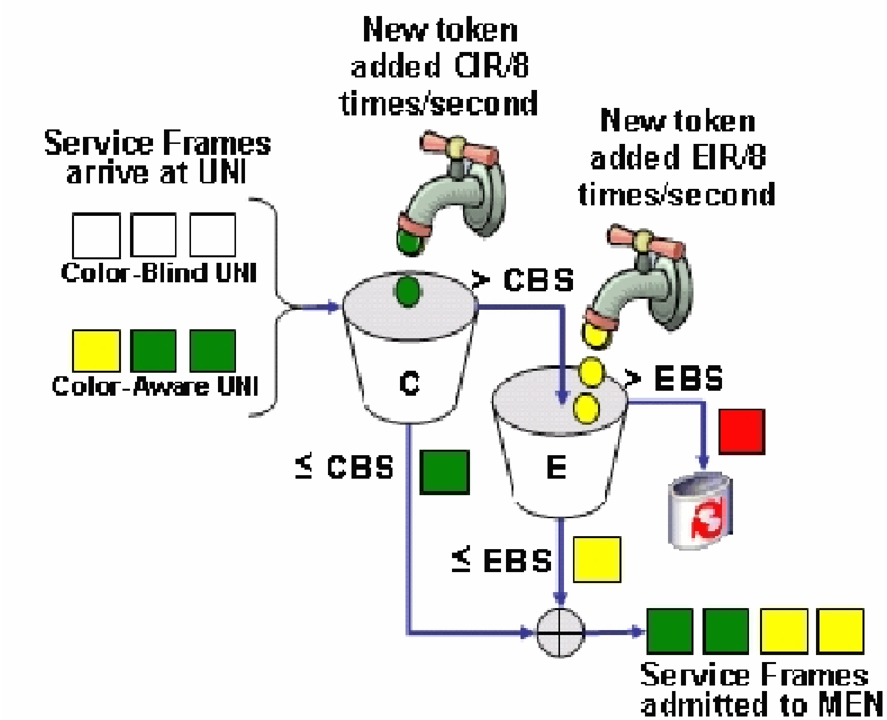
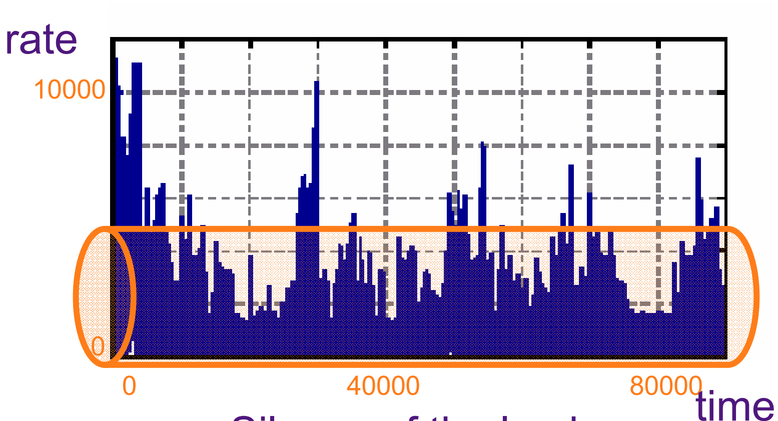


Figure 6: MEF trTCM algorithm

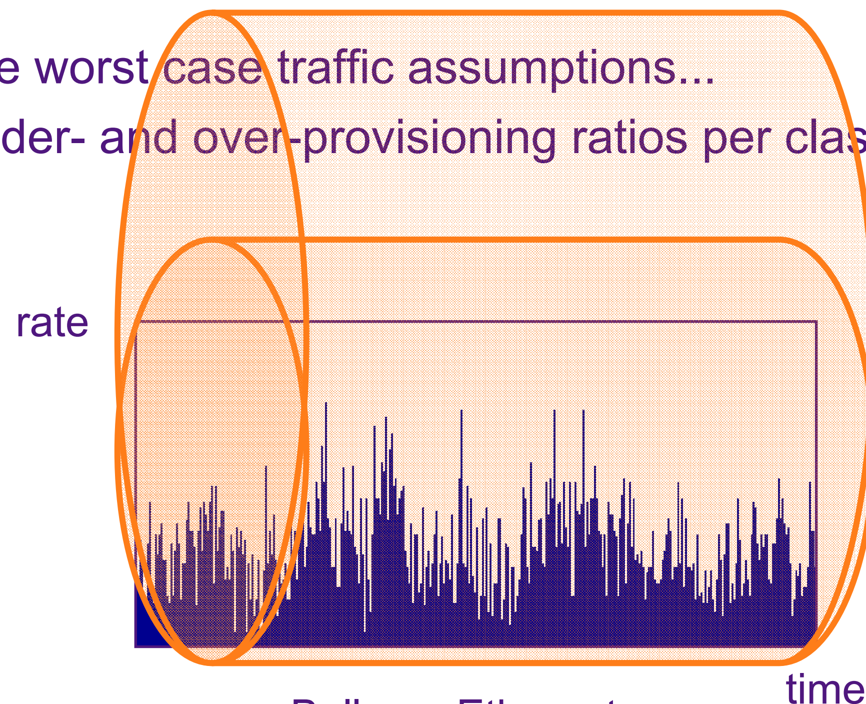
Service Class	Service Characteristics	CoS ID	Bandwidth Profile per EVC per CoS ID	Service Performanc
Premium	Real-time IP telephony or IP video applications	6, 7	CIR > 0 EIR = 0	Delay < 5m Jitter < 1m Loss < 0.001%
Silver	Bursty mission critical data applications requiring low loss and delay (e.g., Storage)	4, 5	CIR > 0 EIR ≤ UNI Speed	Delay < 5m Jitter = N/ Loss < 0.01%
Bronze	Bursty data applications requiring bandwidth assurances	3, 4	CIR > 0 EIR ≤ UNI Speed	Delay < 15ms Jitter = N/ Loss < 0.1
Standard	Best effort service	0, 1, 2	CIR=0 EIR=UNI speed	Delay < 30ms Jitter = N/ Loss < 0.5

From traffic descriptor to SLS?

- principle of QoS architectures
  - based on a traffic descriptor,
  - satisfy the terms of an SLS
- but how ?
  - fit a leaky bucket and make worst case traffic assumptions...
  - or "merely use different under- and over-provisioning ratios per class"



Silence of the lambs

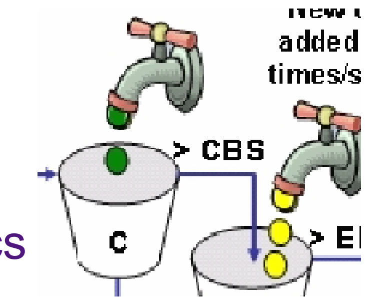


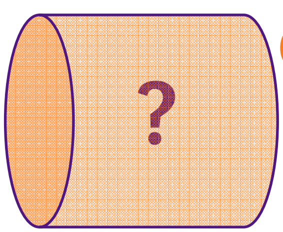
Bellcore Ethernet

Traffic and performance

demand

- volume
- characteristics





capacity

- bandwidth
- how it is shared

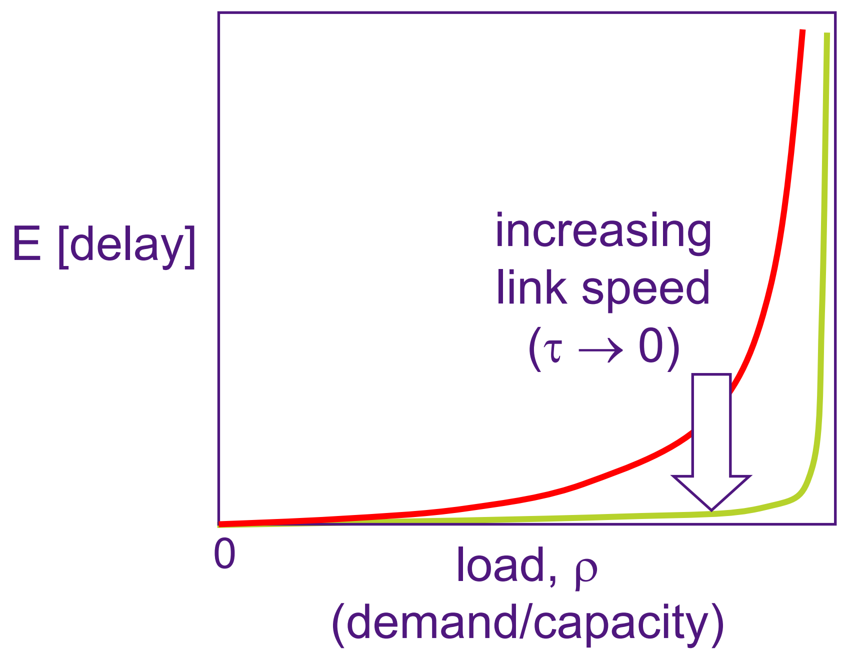
performance

- packet delay
- response time

Delay < 5ms
Jitter < 1ms
Loss < 0.001%

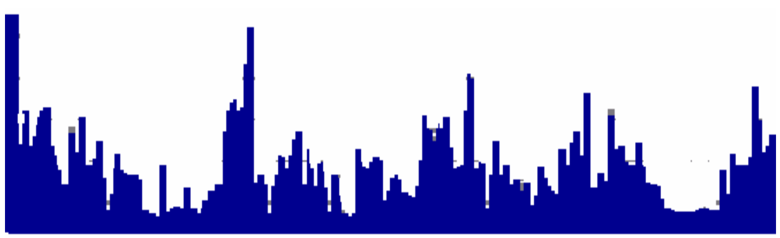
Traffic and performance

- e.g., an M/M/1 queue
  - E [delay] = $\tau \rho / (1 - \rho)$, τ = packet time, ρ = link load
- very little scope for service differentiation
  - quality of service is good or bad
- a need for overload control
  - e.g., admission control



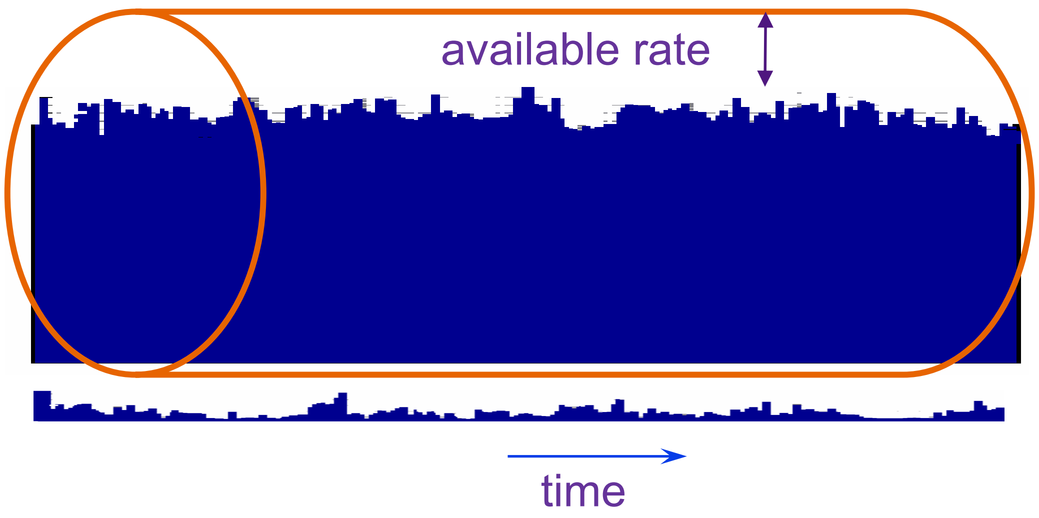
Characterizing Internet traffic

- traffic is composed of flows
  - same identifier, minimum packet spacing
- flows are "streaming" or "elastic"
  - streaming SLS = "conserve the signal"
  - elastic SLS = "transfer as fast as possible"
- the essential characteristic: the flow peak rate
  - streaming peak rate = coding rate
  - elastic peak rate = exogenous rate limit (access line,...)



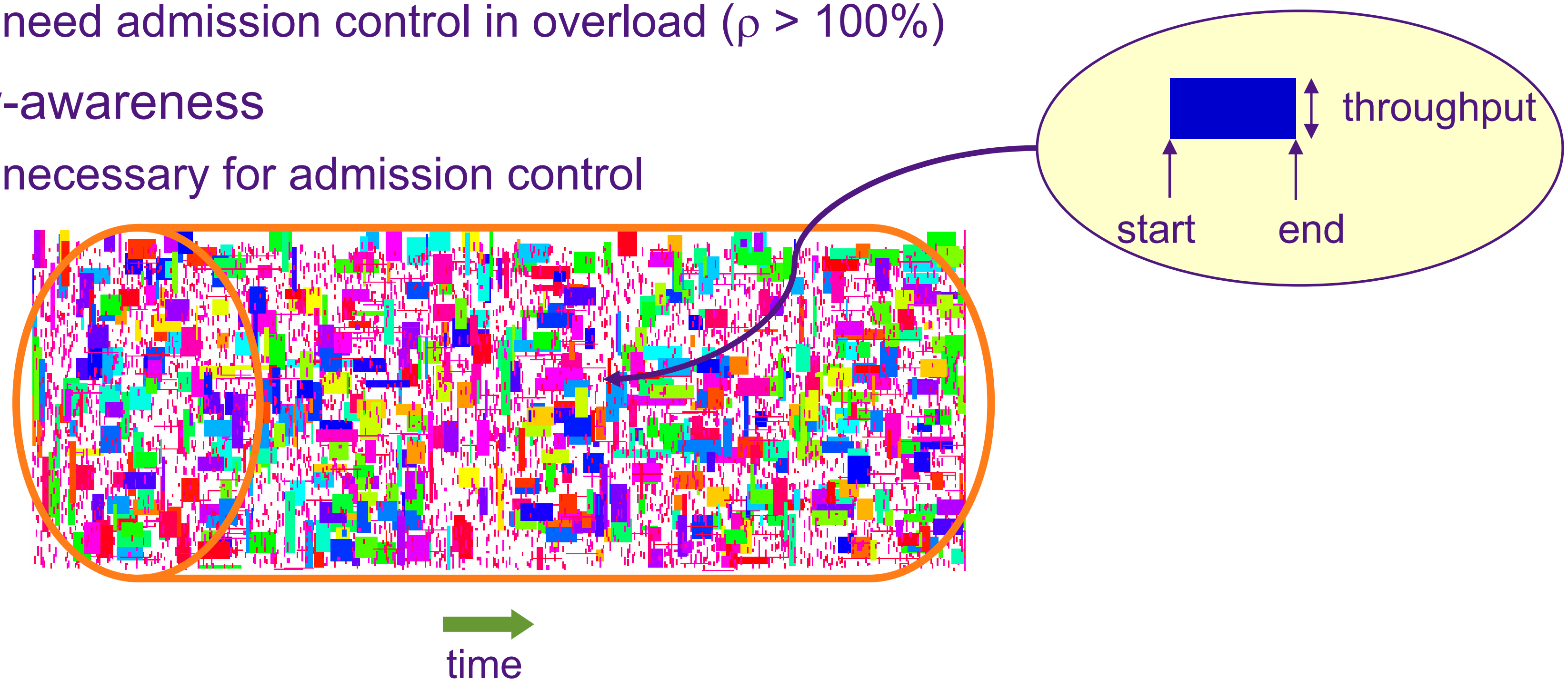
Bufferless multiplexing for streaming flows

- transparency ⇔ Pr [input rate > output rate] < ε
  - efficient when peak rate << link rate
- performance
  - excellent at normal load
  - need admission control in overload
- flow-awareness
  - necessary for admission control



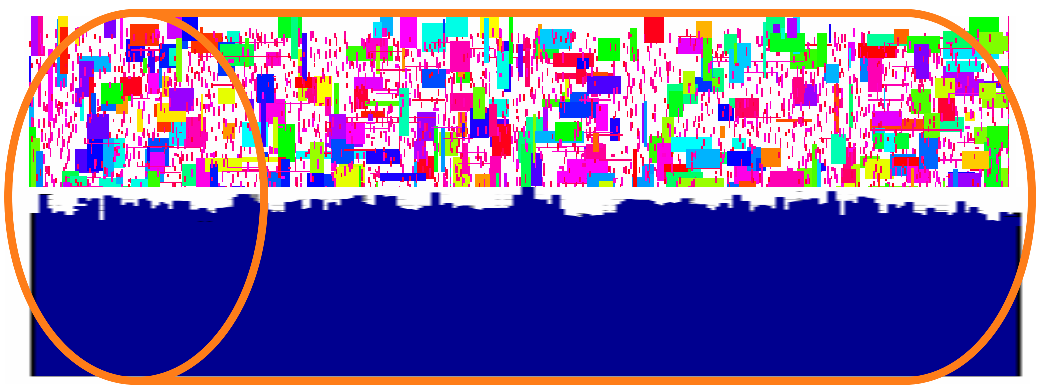
Fair sharing for elastic flows

- peak rate ~ link rate
  - a "processor sharing" queue
- peak rate << link rate
  - bufferless multiplexing, like streaming traffic
- performance
  - excellent at normal load (ρ < 90%)
  - need admission control in overload (ρ > 100%)
- flow-awareness
  - necessary for admission control



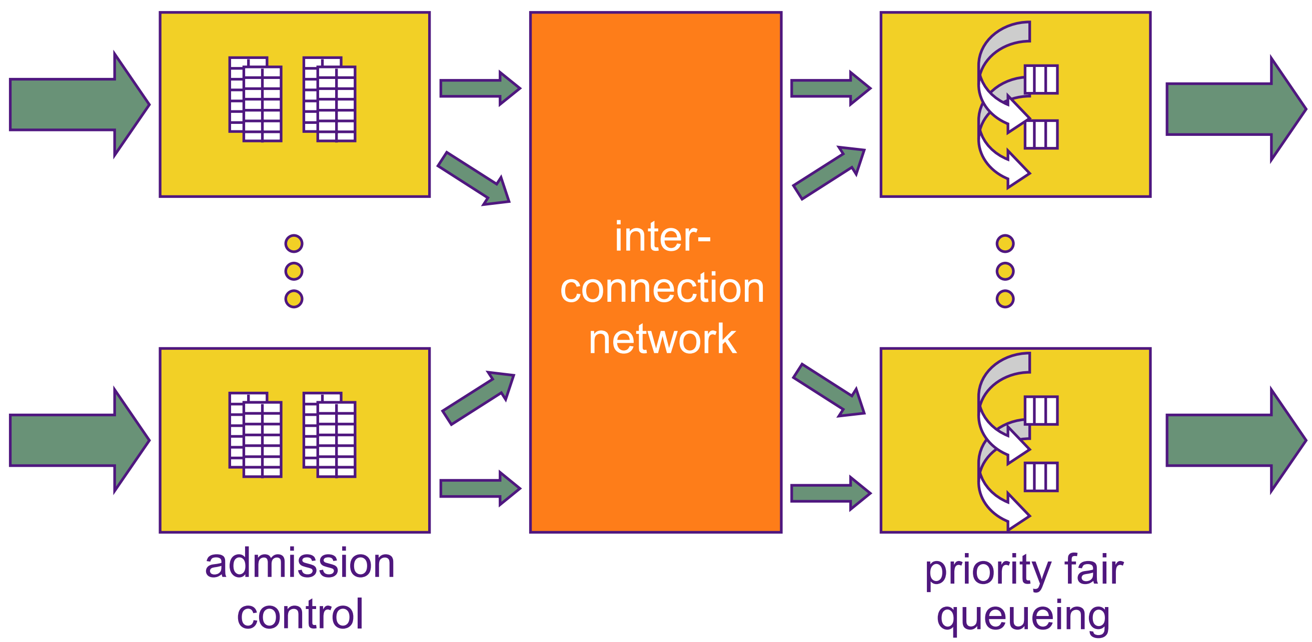
Flow-aware networking with two classes of service

- priority to streaming flows
- fair sharing for elastic flows (end-to-end, by TCP)
- flow-awareness
  - necessary for admission control
- but there are disadvantages
  - marking, policing, fairness



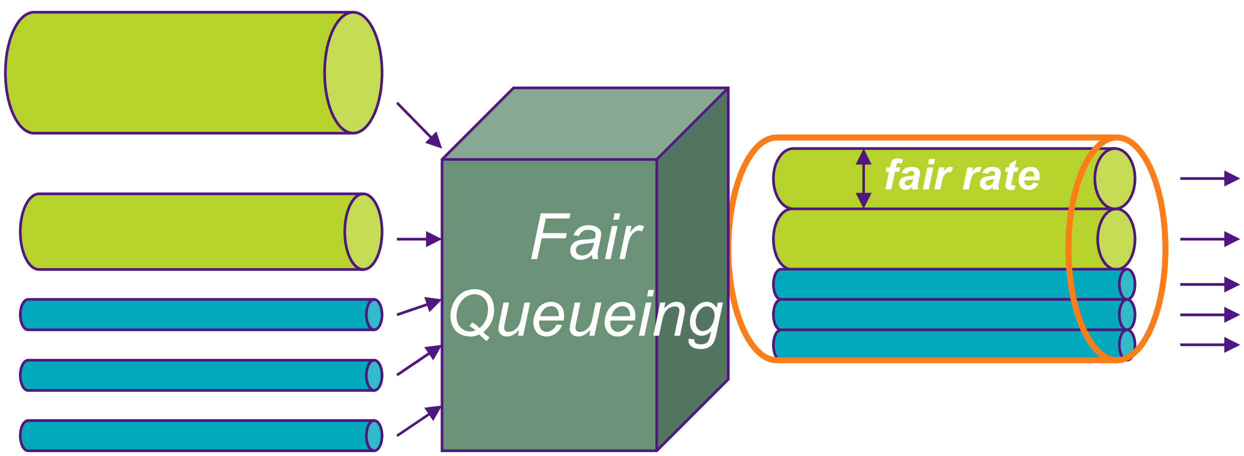
Flow-aware networking without classes of service

- apply per-flow fair queueing in router queues
  - awareness of "active" flows (a small number!)
- per-flow admission control in case of overload
  - awareness of "in-progress" flows (a large number)



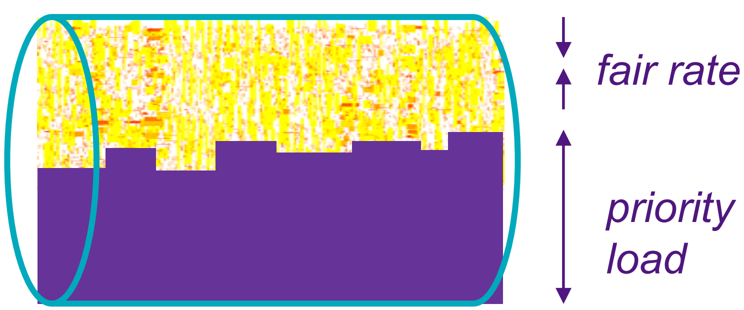
Per-flow fair queueing

- max-min fair sharing by fair sharing
  - e.g., deficit round robin, self-clocked fair queueing,...
  - max active flows ~ 500 (at load $\leq$ 90%), any link rate
- "priority fair queueing"
  - priority to packets of flows of rate < fair rate
- realizes implicit service differentiation
  - when streaming flow rate < fair rate



Measurement-based admission control

- admission control
  - maintain fair rate > $threshold_1$, priority load < $threshold_2$
  - even when offered load > 90%
- maintain a table of flows in progress
  - flow identifier and epoch of last packet
  - time out is no packet in T seconds (e.g., T = 2)
- implicit admission control
  - reject packets of new flows in congestion
  - applications interpret as flow reject



~10000 flows in progress

$flow_n$	$time_n$
$flow_m$	$time_m$

FAN and the "Internet design philosophy"

- respects the end-to-end principle
  - retains the current best effort user-network interface
- retains survivability, reduces vulnerability
  - flow-awareness allows enhanced protection
  - admission control allows adaptive routing
- performance assurance for both types of service
  - through implicit service differentiation
- still based on TCP
  - but fair queueing removes the need for "TCP friendliness"
- enhanced cost-effectiveness, accountability
  - capex & opex reductions, simple billing

Conclusions

- from traffic descriptor to SLS?
  - we need the traffic-performance-capacity relation
- from flow-aware characterization to flow-aware control
  - streaming and elastic traffic
  - bufferless multiplexing and fair sharing
- per-flow fair queueing and admission control
  - scalable and feasible router mechanisms
- flow-aware networking, more than an alternative
  - QoS don't work!
  - FAN respects the "Internet design philosophy"