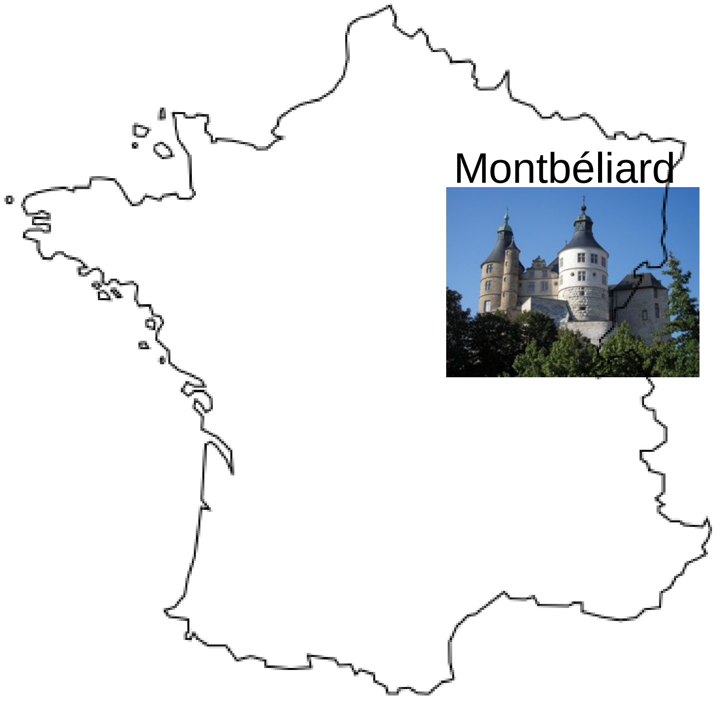
A study on the benefit of TCP packet prioritisation

Eugen Dedu*, Emmanuel Lochin

*Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC)
Montbéliard, France



PDP conference
18-20 February 2009
Weimar, Germany

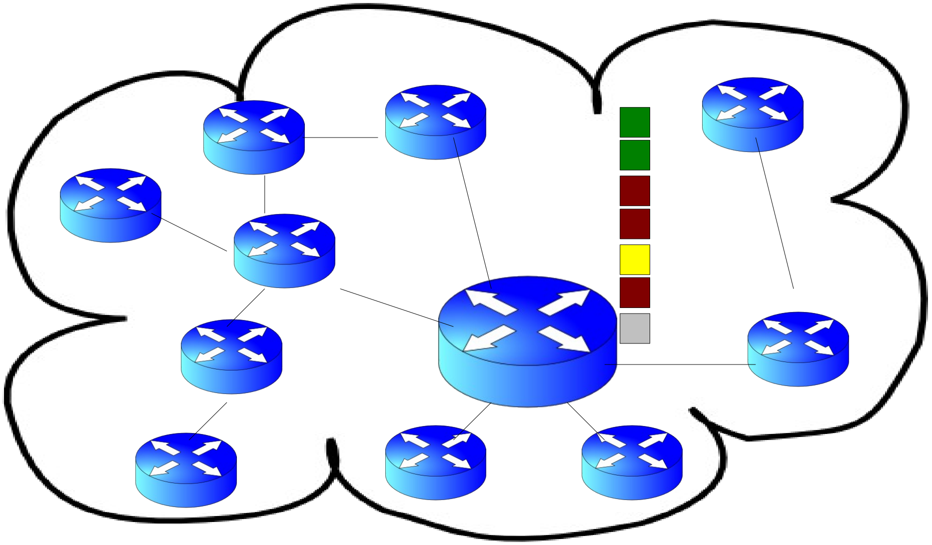
LIFC

Motivation

- User frustration:
  - why downloading a simple Web page takes soo loong?
  - why browsing a Web site is soo loong?
- Network optimisation (“what is the best car?”)
  - money optimisation: replicate inside the ISP (“reduce inter-domain traffic”)
  - traffic optimisation: when a packet is to be dropped, choose a packet whose retransmission uses the least resources
  - user satisfaction optimisation: this paper

Hypothesis

- Packet commutation networks: IP network
- Best-effort
- Router queues =>
  - drop packets
  - increase latency
    - in most networks, packets spend most of the time in router queues
    - **goal** : reduce latency (web traffic) by careful enqueuing



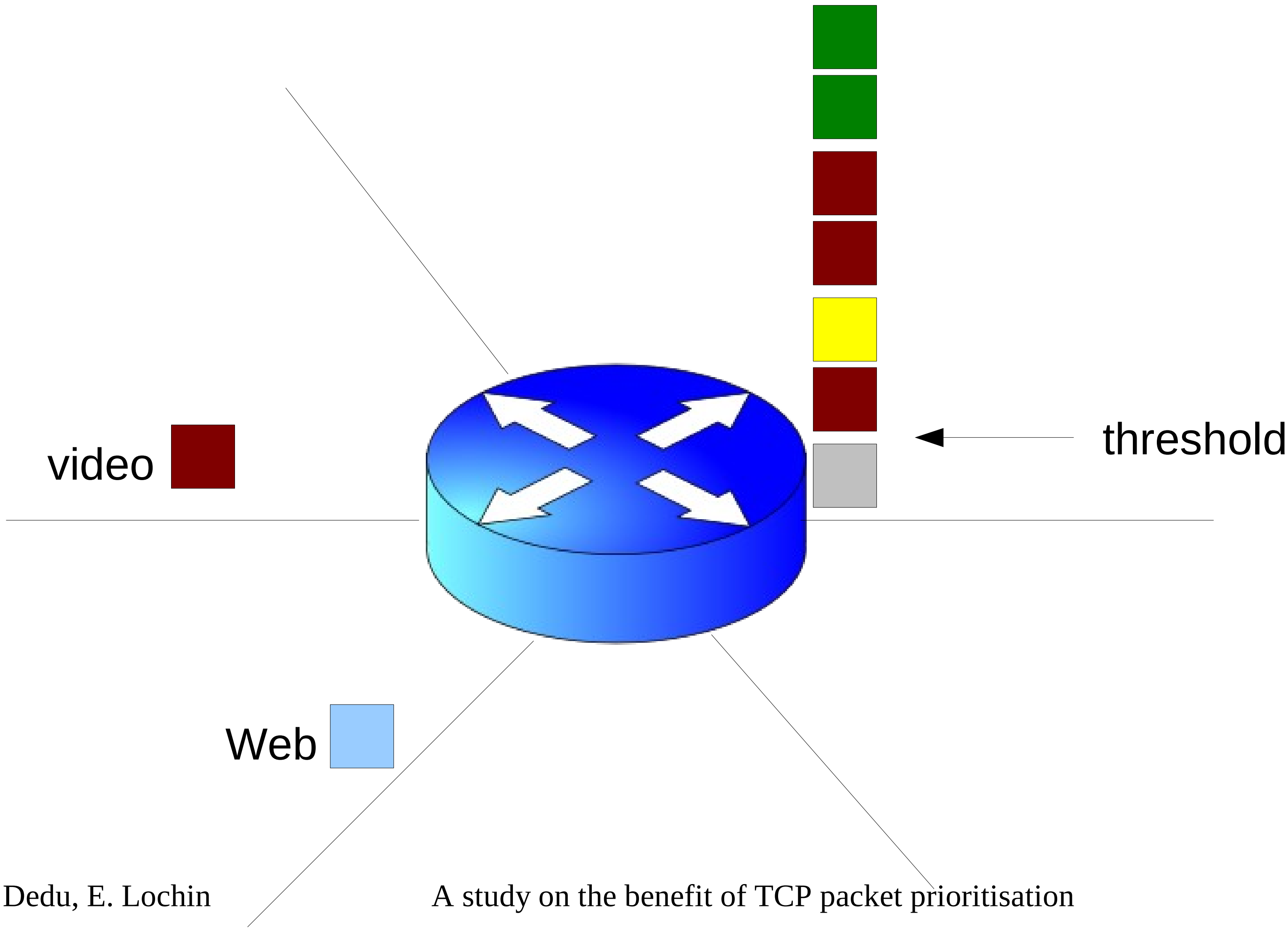
Plan

- Related work
- Our idea, new packet scheduling
- Simulation on simple network: does the idea work as expected?
- Simulation on complex network: is the idea useful?
- Conclusions/perspectives

Related work on favouring packets

- Kurose book: favour packets with low TTL
- [Rai05]: routers memorise the number of bytes of each flow, and order packets by this number (flow-state on router, heavy computations)
- [Avranchenkov04]: same, but guesses the number of bytes from TCP seqno; two queues (source modification, problems in mixed deployment)
- [Chen03]: edge routers memorise information about flows and set DiffServ bits

Our idea, FavourTail packet scheduling



Our idea, algorithm

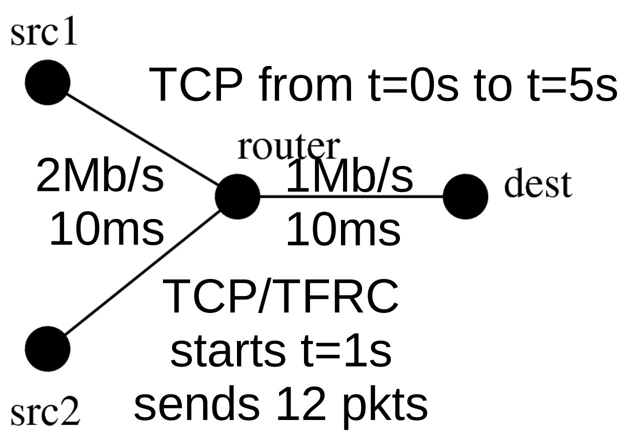
- When a packet arrives, check if the packet is to be dropped
- If not dropped, check if there are packets from the same flow in the queue
- If yes, add the packet at the end, as usually
- If no, prioritise packet by adding it at the end of the priority packets
  - inside the queue, a (changing) threshold pointer delimits priority and normal packets

Our idea, properties

- No reordering inside a flow
- Not only the beginning of a flow is prioritised, but generally any flow during small-cwnd period (few packets in flight)
- The more the routers in the path, the greater the gain in transmission time
- Sources cannot cheat, because they do not guess router load (queue size)
- Starvation may occur in theory (future work)

ns2 simulation on simple network: scenario

- Q: does the idea work as expected?
- Router is: (a) DropTail, (b) FavourTail
- Measure the transmission time for second flow (src2->dest) in both cases



queue size = 50
(default ns2 value)

Simple network: a TCP flow

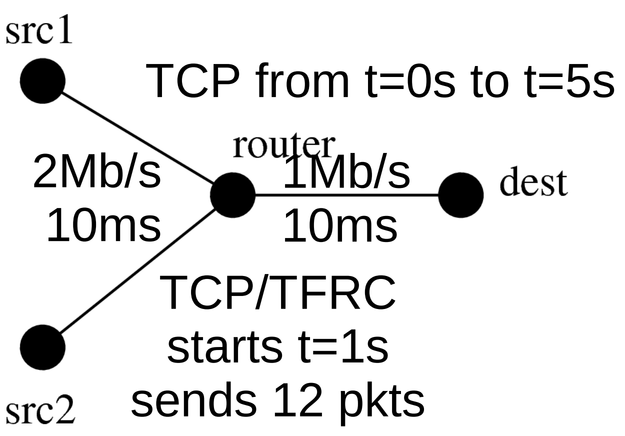
- 1st flow sends 591 packets in both cases
- 2nd flow, trtime = 0.53s for DropTail, 0.43s for FavourTail => 20% gain
- TCP: window-based congestion control, bursty traffic
- Analysis: 1st packet overtakes 13 packets, the 2nd one 14 packets, all the others are not prioritised
  - 0.53-0.43 = 0.10s is the time needed by router to process 13+14 = 27 packets

Simple network: a TFRC flow

- 1st flow sends 591 packets in both cases
- 2nd flow, trtime = 0.54s for DropTail, 0.17s for FavourTail => 70% gain
- TFRC: equation-based congestion control, evenly-spaced packets during one RTT (generally)
- Analysis: 6 packets out of 12 are prioritised, gaining each between 14 and 17 slots

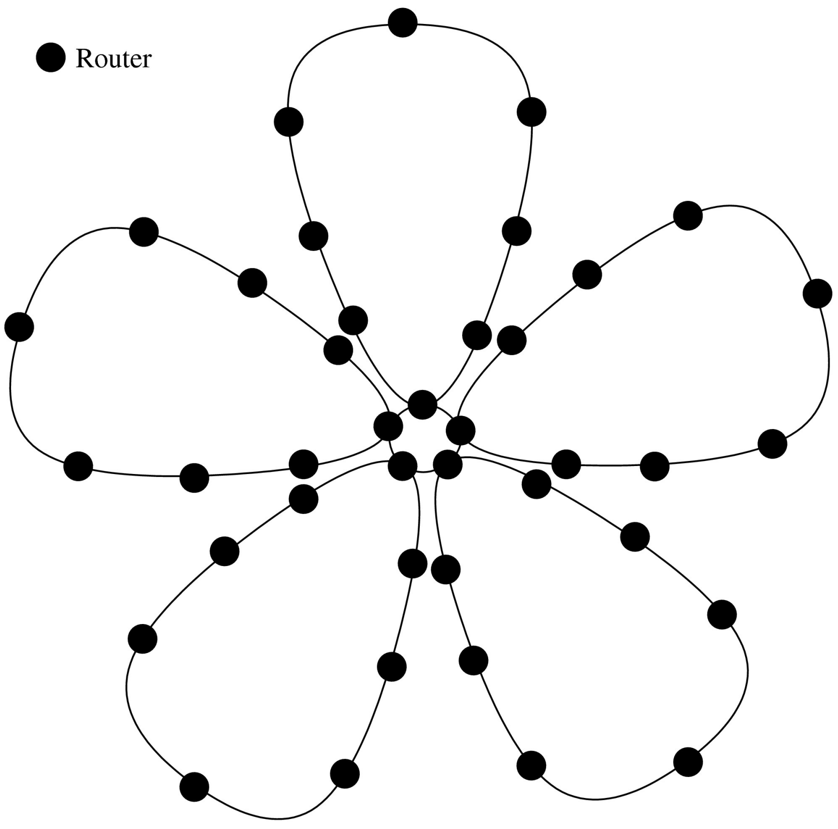
Simple network: conclusions

- TCP:
  - bursts: 1 packet, 2 packets, 4 packets, 8 packets, ...
  - burst of 2 packets: the 2nd packet arrives at R before the 1st leaves the router (2Mb/s vs 1Mb/s)
- TFRC:
  - smooth traffic (generally)
  - the 7th packet and subsequent lose priority, because the throughput becomes a bit higher than 1Mb/s



ns2 simulation on complex network: scenario

- Q: Is the idea useful?
- A regular xDSL backbone
- Compare on several metrics
- Topology:
  - each router (except core ones) has 2 DSLAMs
  - each DSLAM connects 3 hosts
  - 10Mb/s, 10ms
  - all routers DropTail/FavourTail
  - (queue size = 50)
- Flows:
  - 500 FTP TCP/NewReno
  - random src/dest
  - send random 10-600 packets



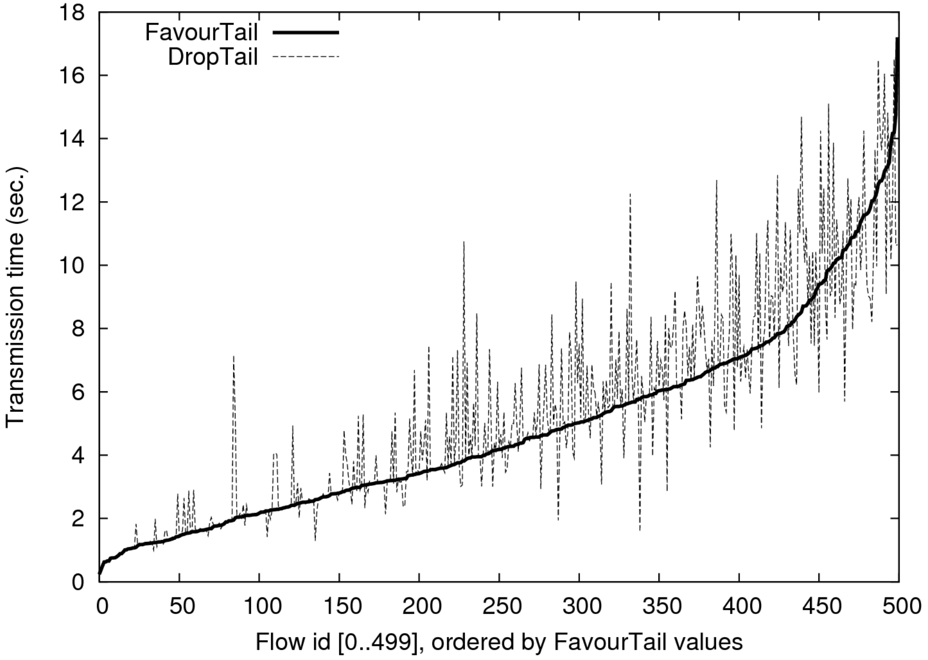
Complex network: global metrics

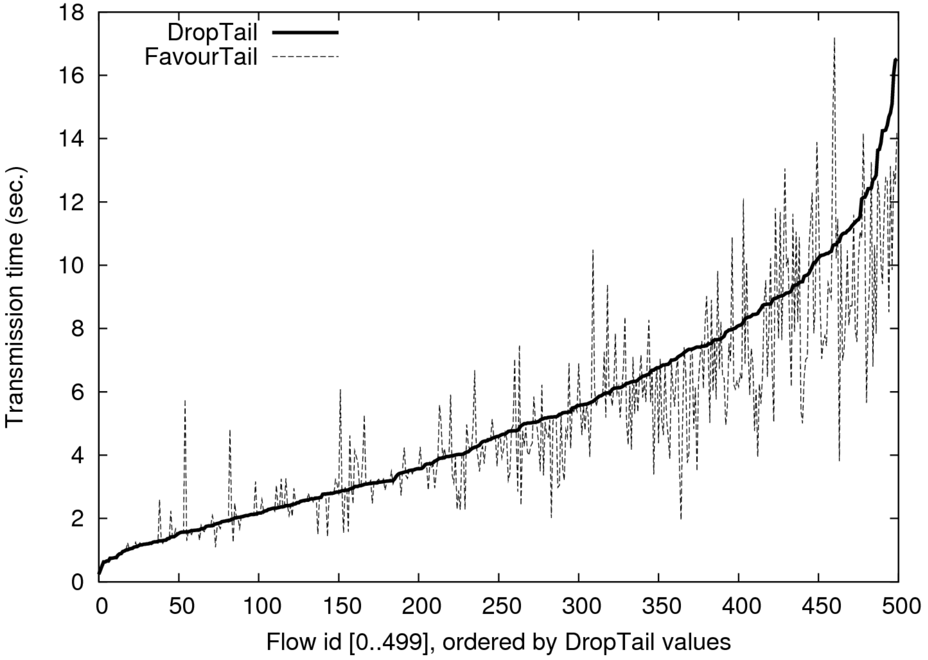
- (Analogy with task scheduler, see also simple network results)

	DropTail	FavourTail
Sum of transmission times	2618	2410
Number of lost packets	2470	1608
Number of lost data packets	913	626

Complex network: short flow metrics (1)

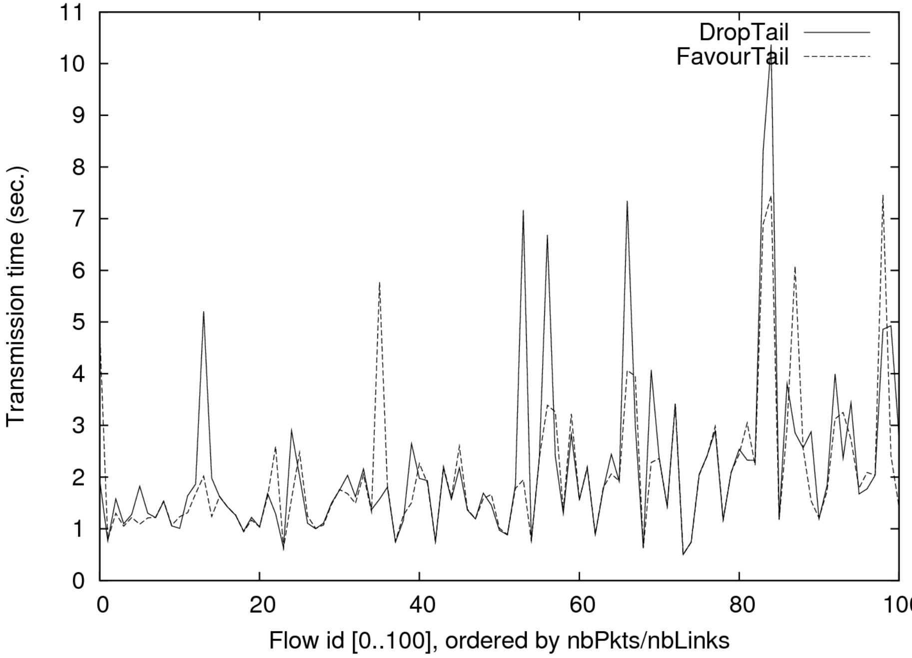
- Are short flows favoured?





Complex network: short flow metrics (2)

- Objective comparison, based on flow “length”:
  - number of packets sent
  - number of packets divided by number of routers <--
  - number of packets divided by the number of concurrent flows
- Are short flows favoured?

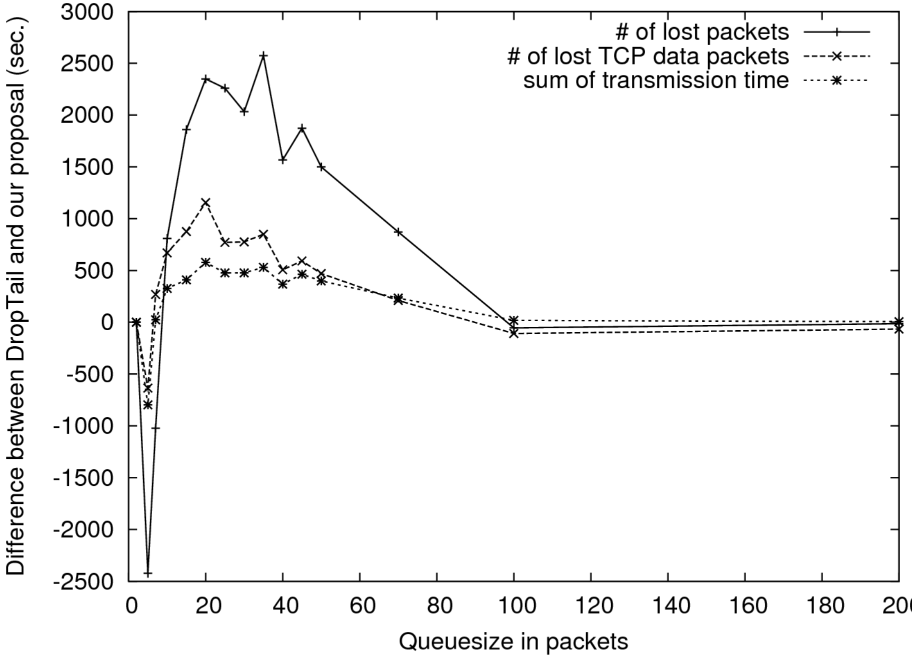


Complex network: queue size influence (1)

- Vary the queue size of each router from 2 to 200 packets
- Same global metrics
- (The first 50 flows send 1-10 packets, all the others send 200-800 packets)

Complex network: queue size influence (2)

- (Positive is better)
- Near 0, same result
- Very high, same result
  - reason: fixed TCP cwnd size cannot make queues overflow
  - => when no congestion, FavourTail similar to DropTail



- For 10-70 (=> when severe to slight congestion), FavourTail better than DropTail
- Also, short flows are not particularly favoured either...

Conclusions/perspectives

- A new packet scheduling for router queues
  - prioritise packets when no other packet from same flow is inside queue
- Intuitively, short flows are favoured
- Surprisingly, all the flows are generally favoured
  - global metrics get better

- Investigate further through a larger measurement campaign
- Study TFRC in complex networks too
- Analyse starvation