LAWIN: the Latency-AWare InterNet architecture

Katsushi Kobayashi
ikob@acm.org
University of Tokyo

Copyright (C) 2013, Katsushi Kobayashi. All rights reserved.

Executive Summary
LAWIN: Latency AWare Internet architecture

- Challenges:
  - Support latency sensitive applications for better QoE.
  - Reduce large latency due to large SW/Router buffer
    - Bufferbloat in access NW, buffer buildup and TCP incast in DC
- Components:
  1. Per packet deadline indication in headers
  2. EDF(Earliest Deadline First) packet scheduling
  3. Drop (renege) packets, if elapsed its deadline
- Findings:
  - If deadline is shorter, higher drop probability is imposed.
  - Otherwise, drop probability is similar to FIFO with drop-tail.
    - Familiar to existing transport protocols.

Note: Long term solution, since protocol modification is required

LAWIN : Hawk in Tagalog

LAWIN: Latency AWare InterNet architecture

- Motivation:
  - A lot of latency sensitive applications with different latency requirements.
    - Online Game, Interactive Web
  - Large packet buffer degrade QoE, aka. Bufferbloat.
- Existing approaches:
  - Diffserv : Dedicated queue for specific applications as EF for VoIP.
    ➡Difficult to deploy multi-ISP
  - DOCSIS : recommend small packet buffer
    ➡Large packet buffer is always harmful ?
  - AQM : Drop packets until target queueing delay
    ➡Unable to support different latency requirements
    Though parameter-less AQM, target delay have to be decided.
    - Applications know own latency bound of better QoE,
      not ISP, SDO.

LAWIN and Application

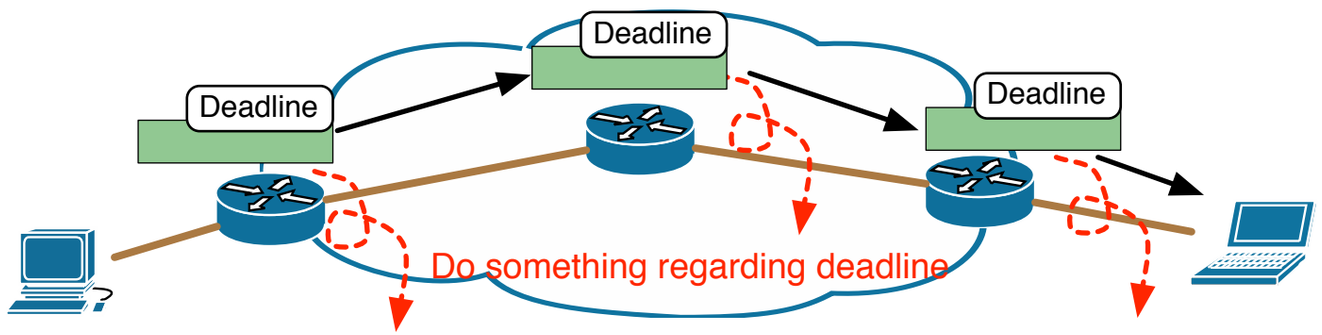
1. Indication: Per packet deadline from application using shim/IP option/unused header.
   - Ends: protocols stack, API, Application supports
2. Scheduling : Earliest Deadline First (EDF)
3. Reneging : Drop deadline elapsed packet

	Loss sensitive	Loss resilience
Latency sensitive	**HFT (1ms) VoIP (150ms)**	**Intra DC (< 10ms) Online Game (100ms) Interactive Web (200ms)**
Not latency sensitive	**File up/download**	**P2P (?)**

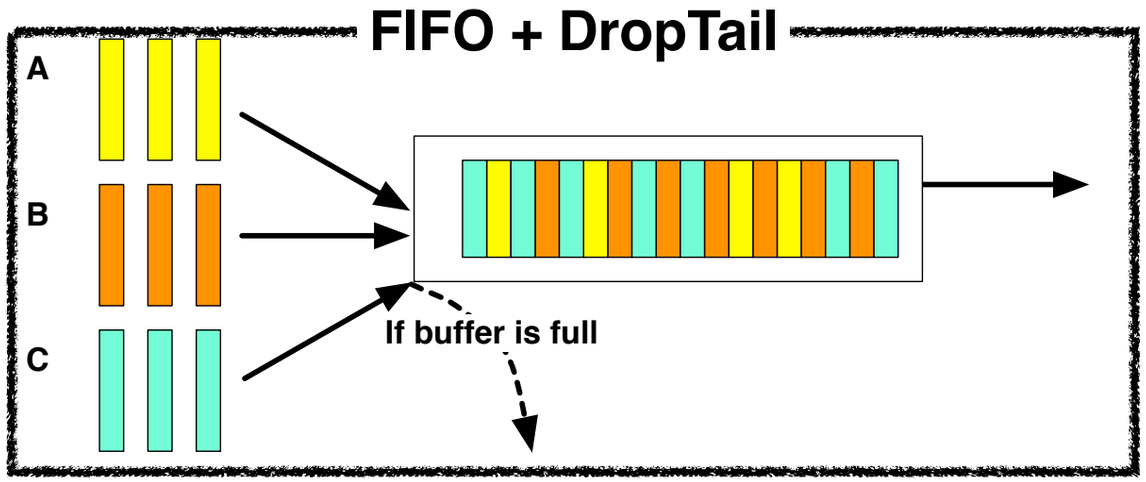
Per-packet deadline indication

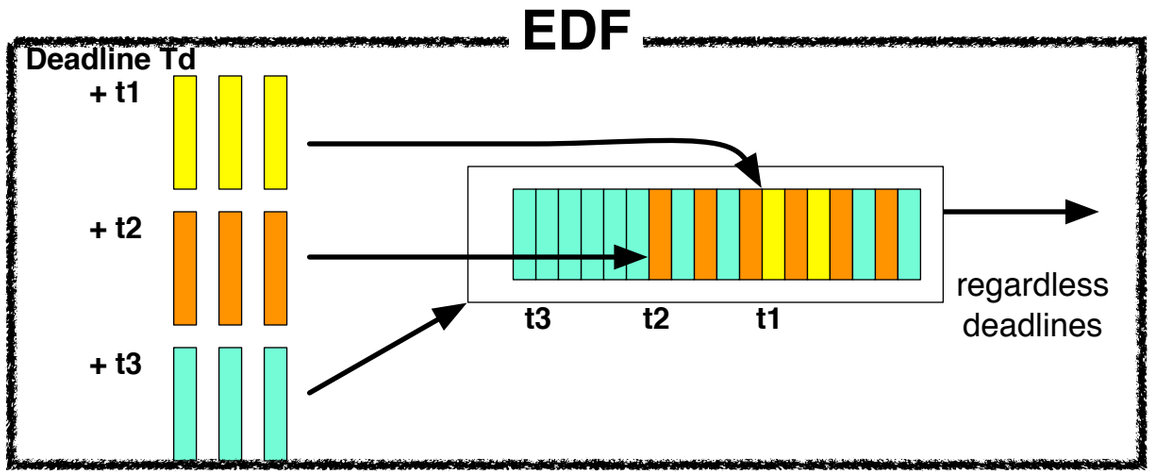
- Specify latency requirement from application.
- Where:
  - Add shim layer, IP option, unused header space.
- How:
  - Relative time format.
    - Absolute format requires global time synchronization.
  - Eliminate propagation delay.
    - Longer distant packets always lose against closer, if incl. propagation.
- Specify max. sojourn time of per hop queue
  - Cumulative (like IPv4 TTL) deadline has similar to propagation delay.

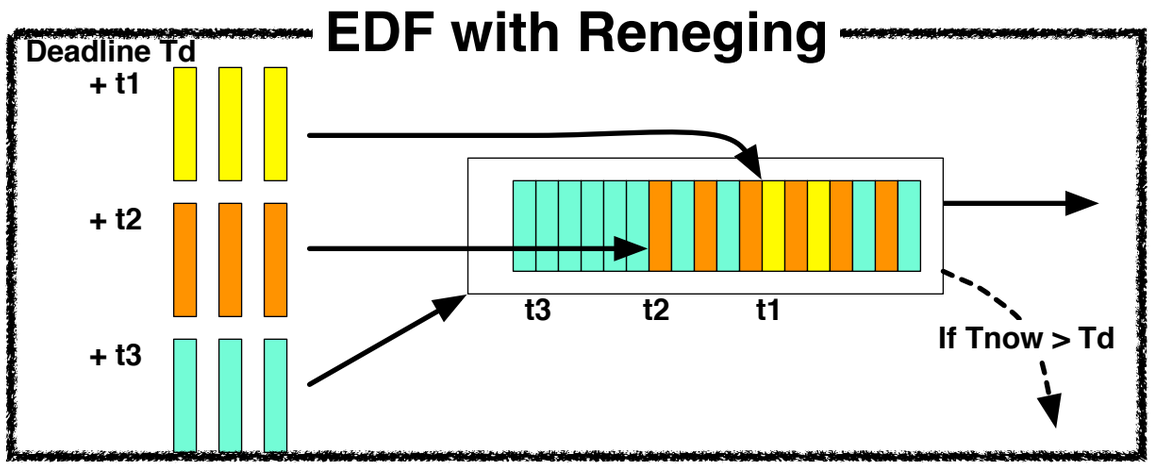
  Note : App. should measure end-to-end latency itself, even telling own requirement.

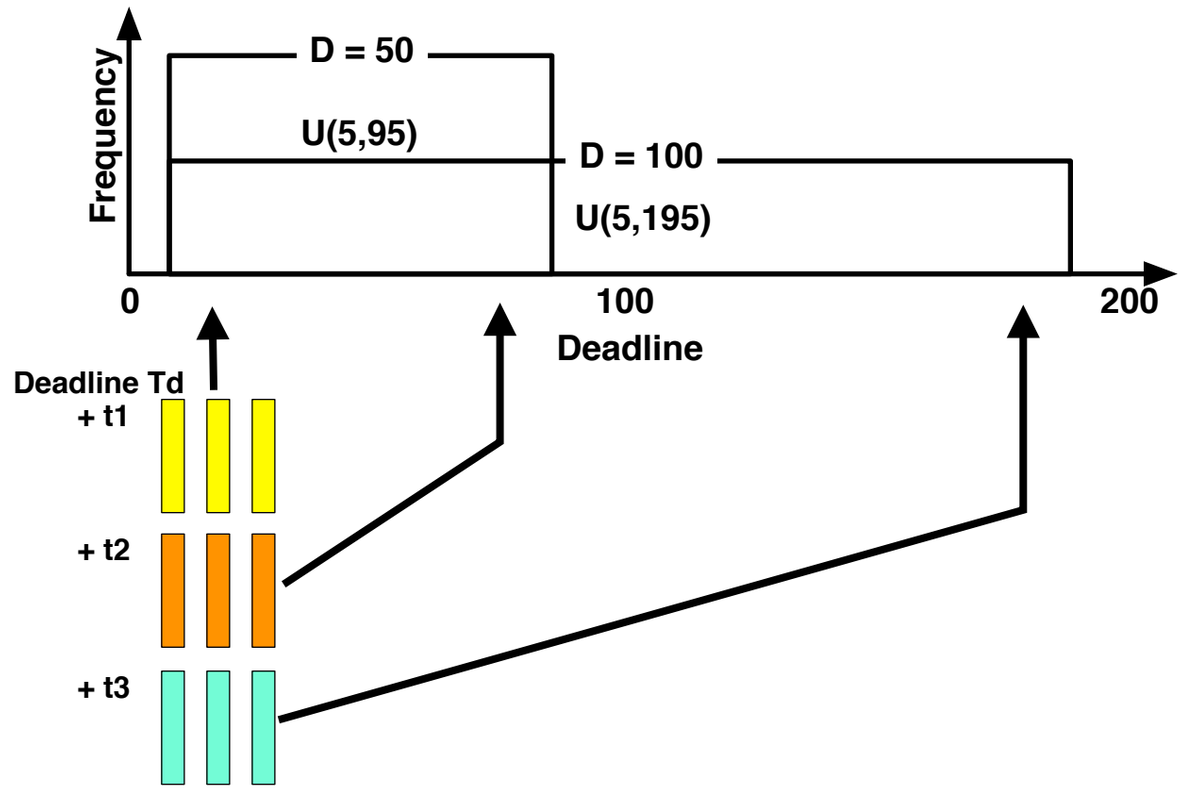


Earliest Deadline First with Reneging

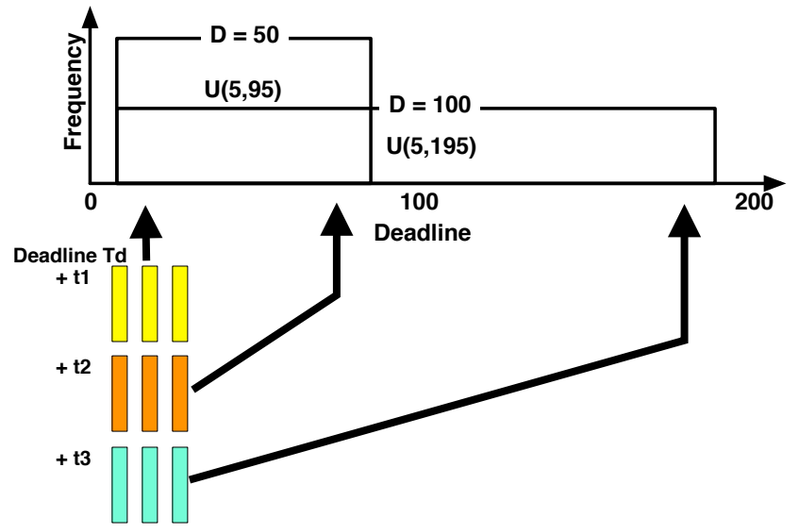








EDF with Reneging (Macro)



$1/\lambda = 2.0$, $1/\mu = 1.96$ ($\rho = 0.98$)
Deadline : U(5, B), $5 < B < 395$

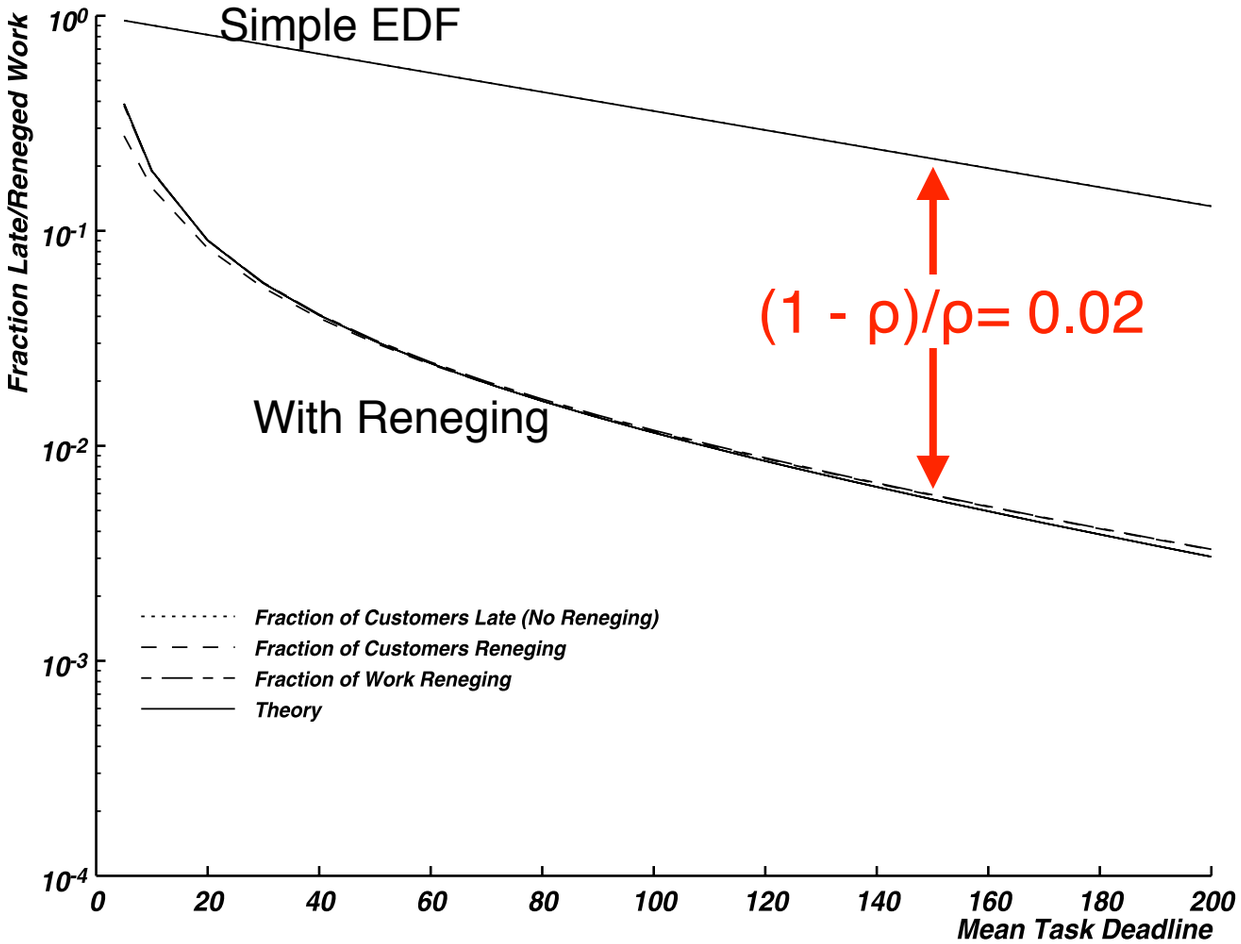


FIG. 1. *M/M/1 queue.*

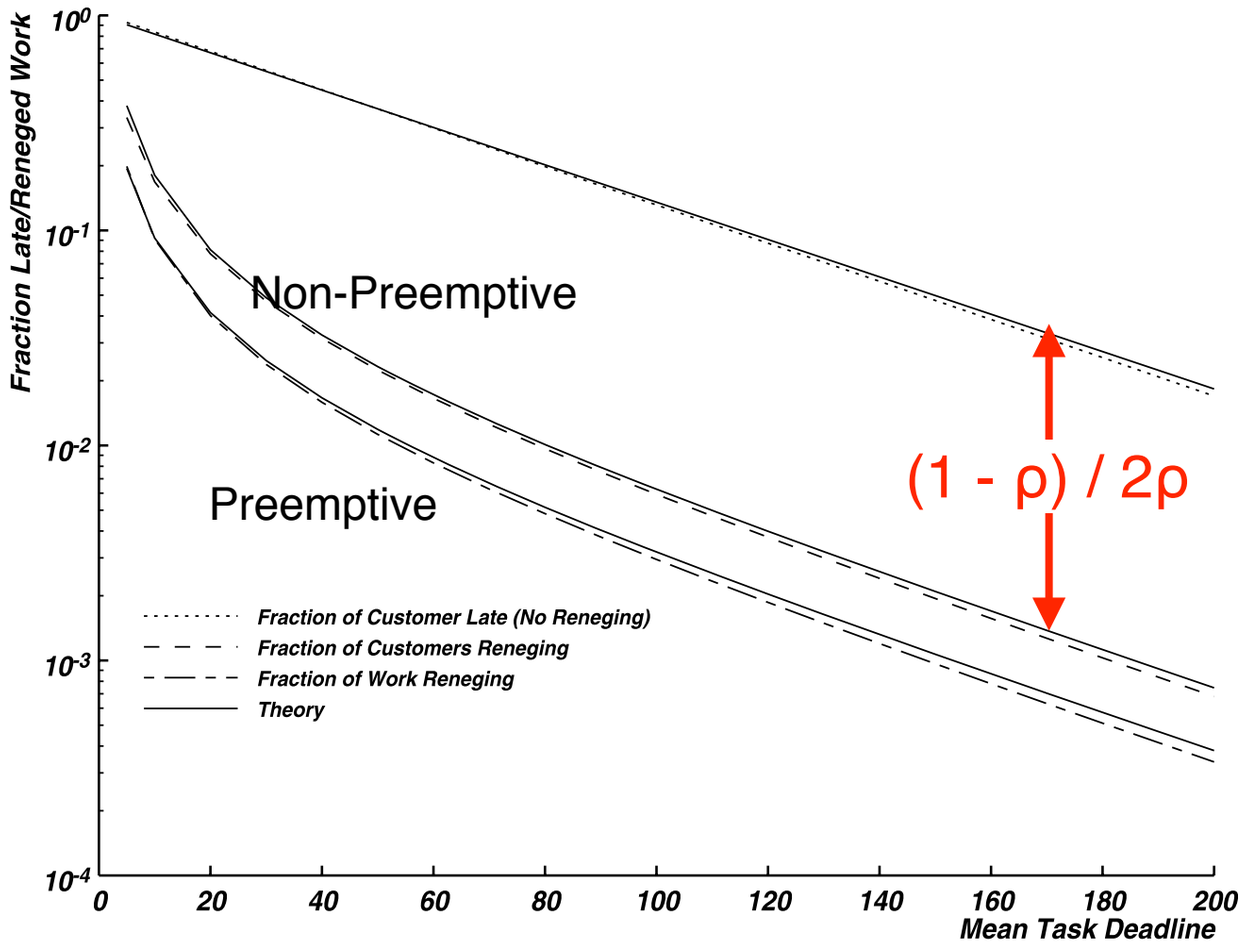
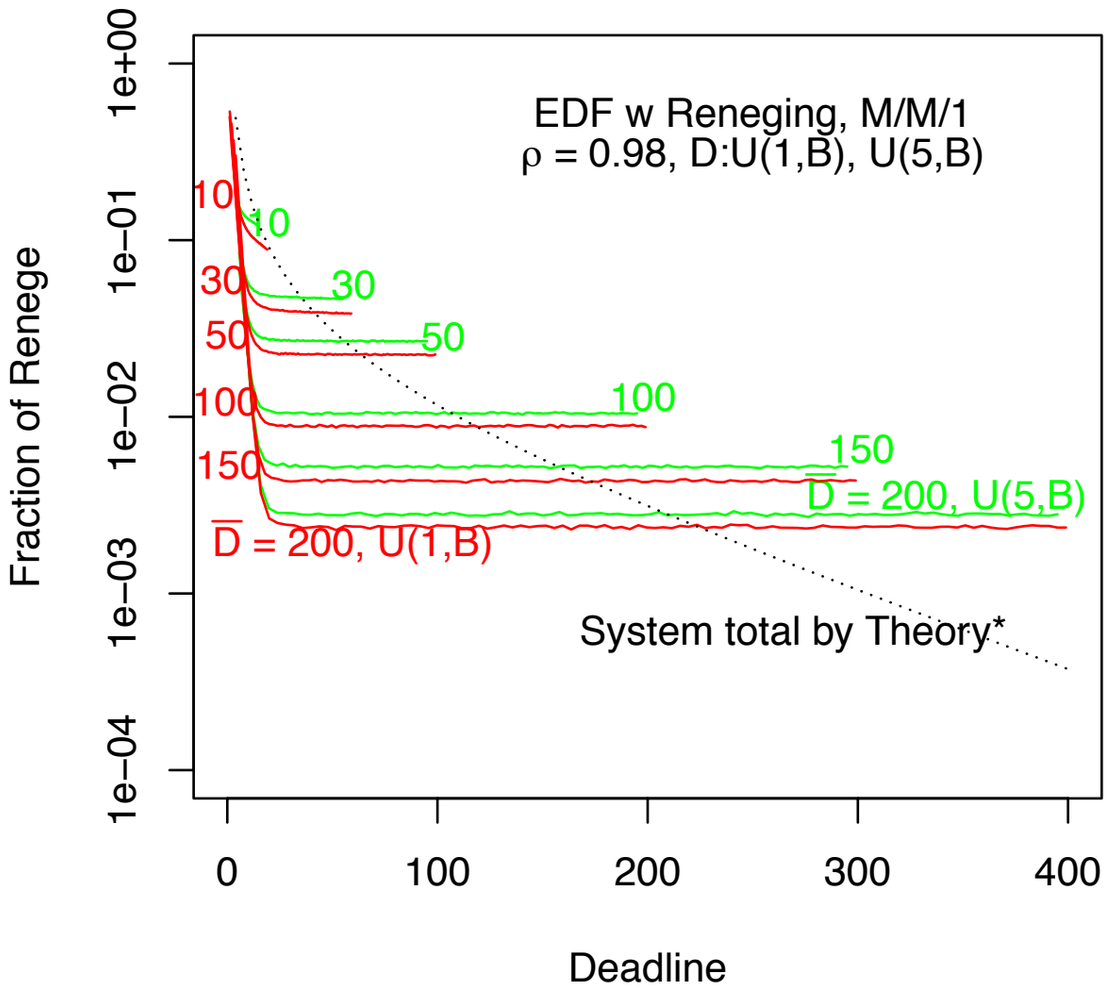


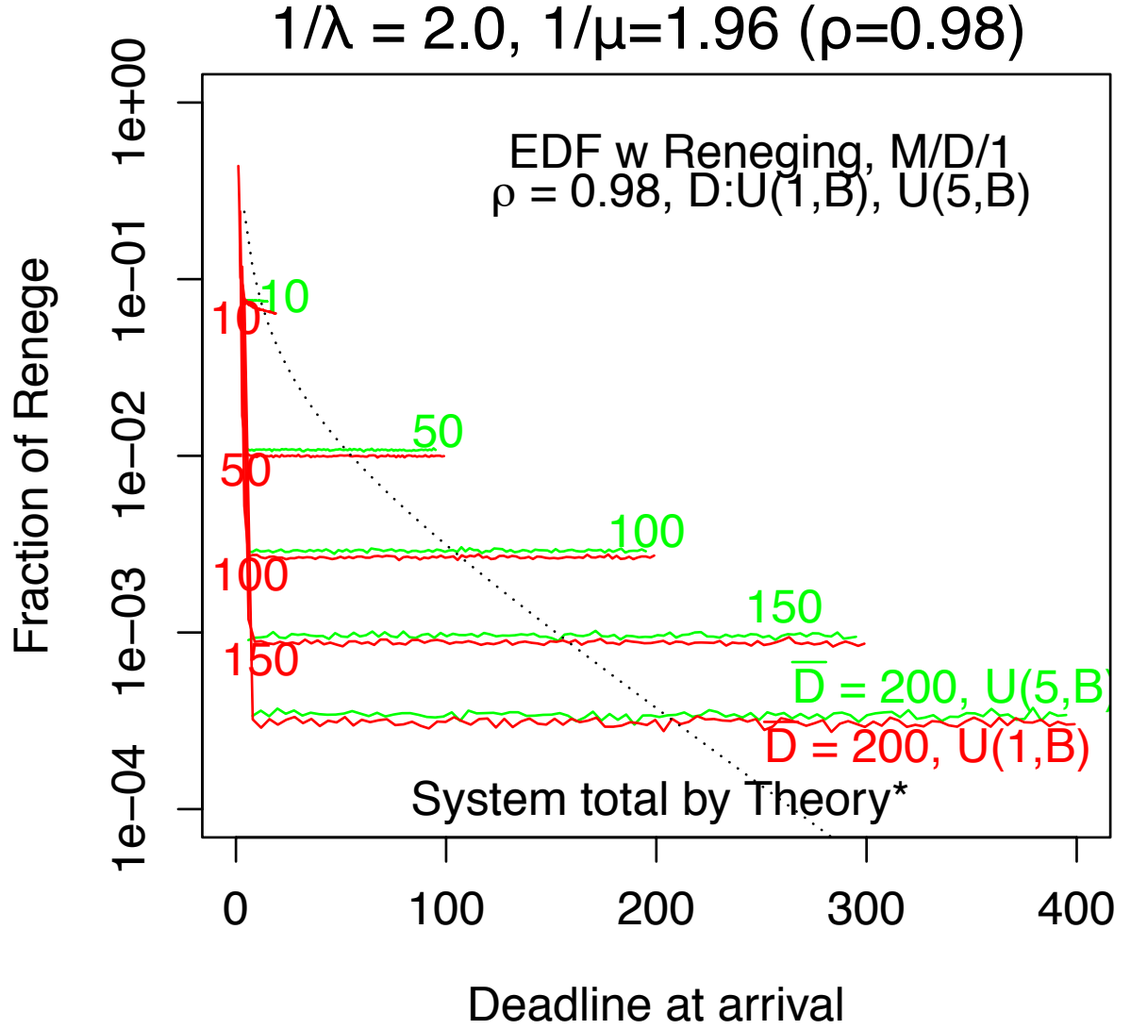
FIG. 2. *M/D/1 queue.*

L. Kruk,J. Lehoczky,K. Ramanan,and S. Shreve. “Heavy traffic analysis for EDF queues with reneging”, The Annals of Applied Probability, 21(2):484–545, 2011

EDF with Reneging (Micro)

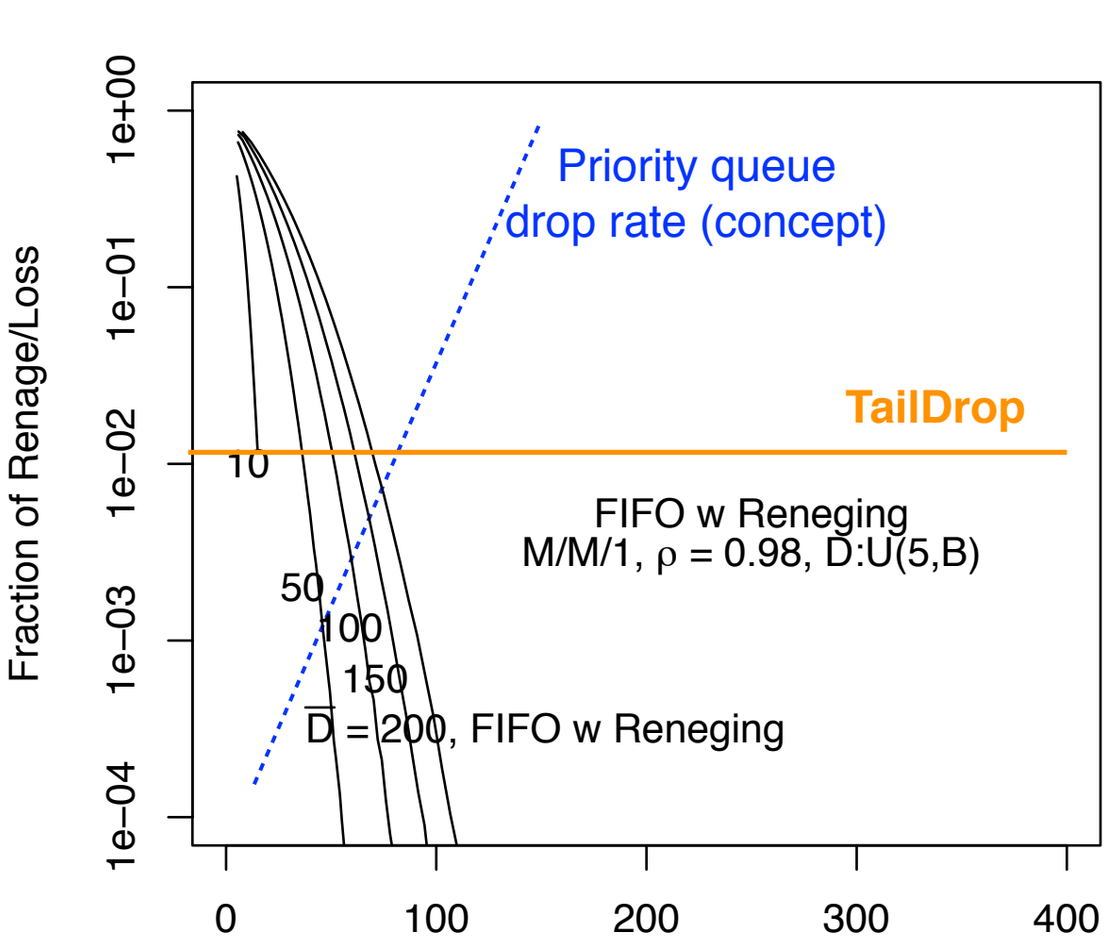
- Key findings
  - Cliffs at smaller deadlines:
    Impose the higher packet loss to the smaller deadlines
  - Flat-bottoms at other area:
    Constant packet loss rate regardless of deadlines
    - Latency : EDF
    - Loss : FIFO with Drop Tail/Head

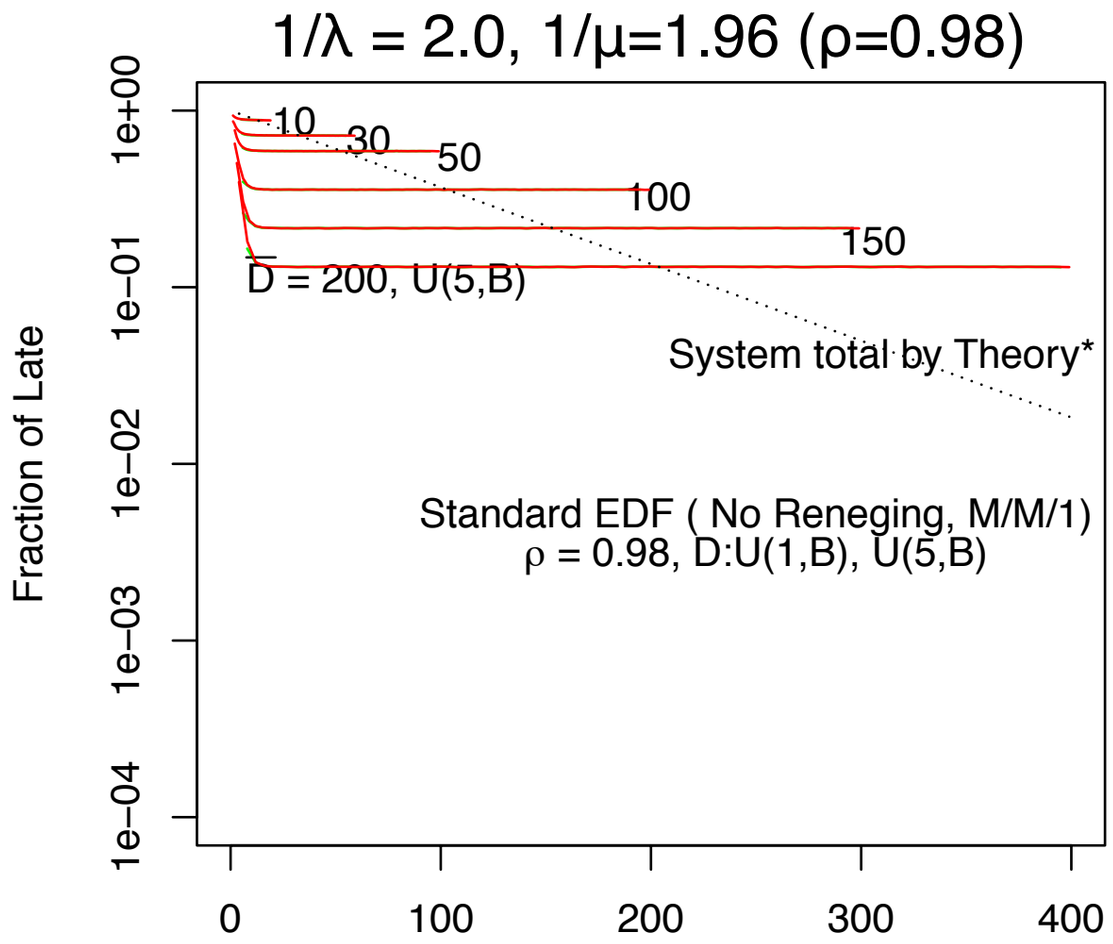




FIFO and Simple EDF (Micro)

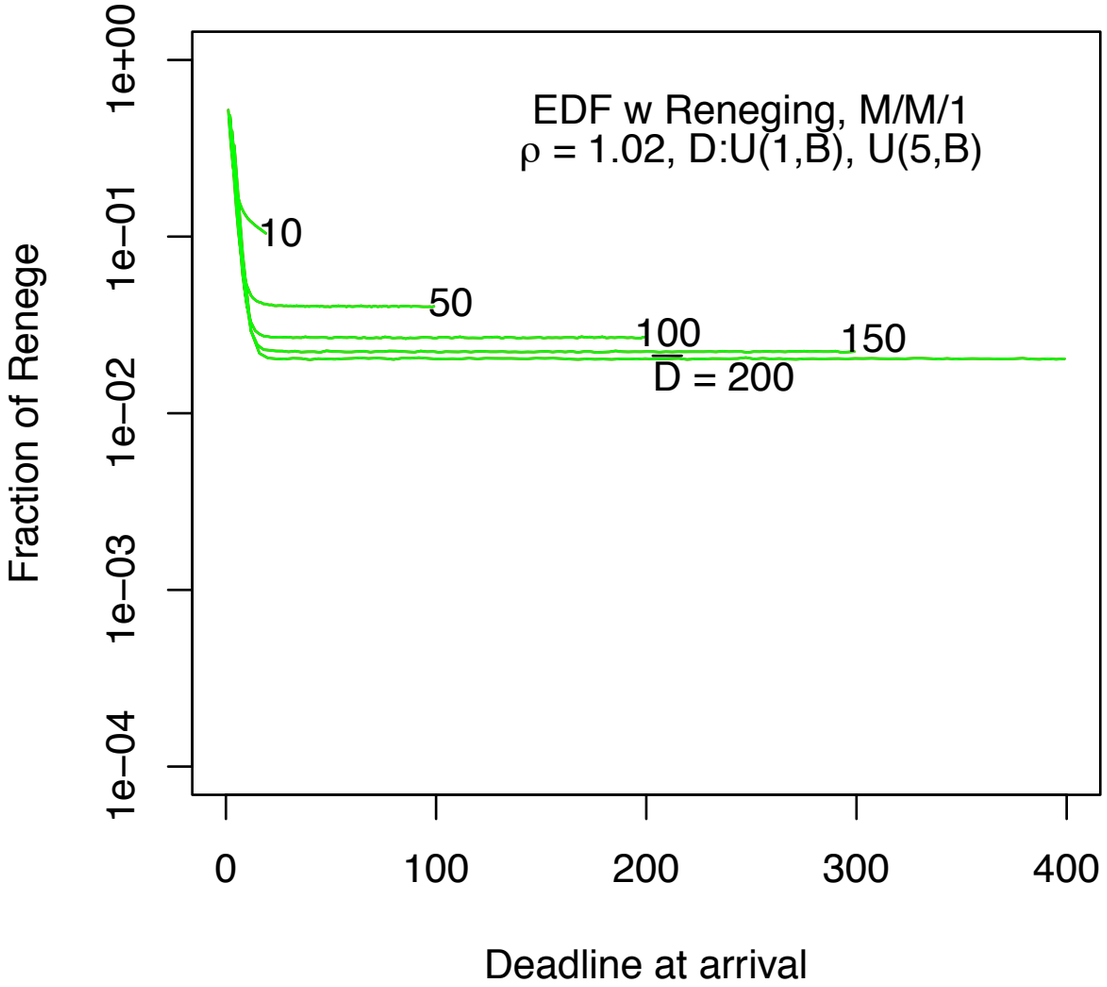
- FIFO with Reneging
  - Too much loss rate, though the higher loss, the shorter deadline.
  - No advantage in shorter deadline.
- Simple EDF
  - Too much late deadline
  - Buffer collapse in overload (ρ > 1)

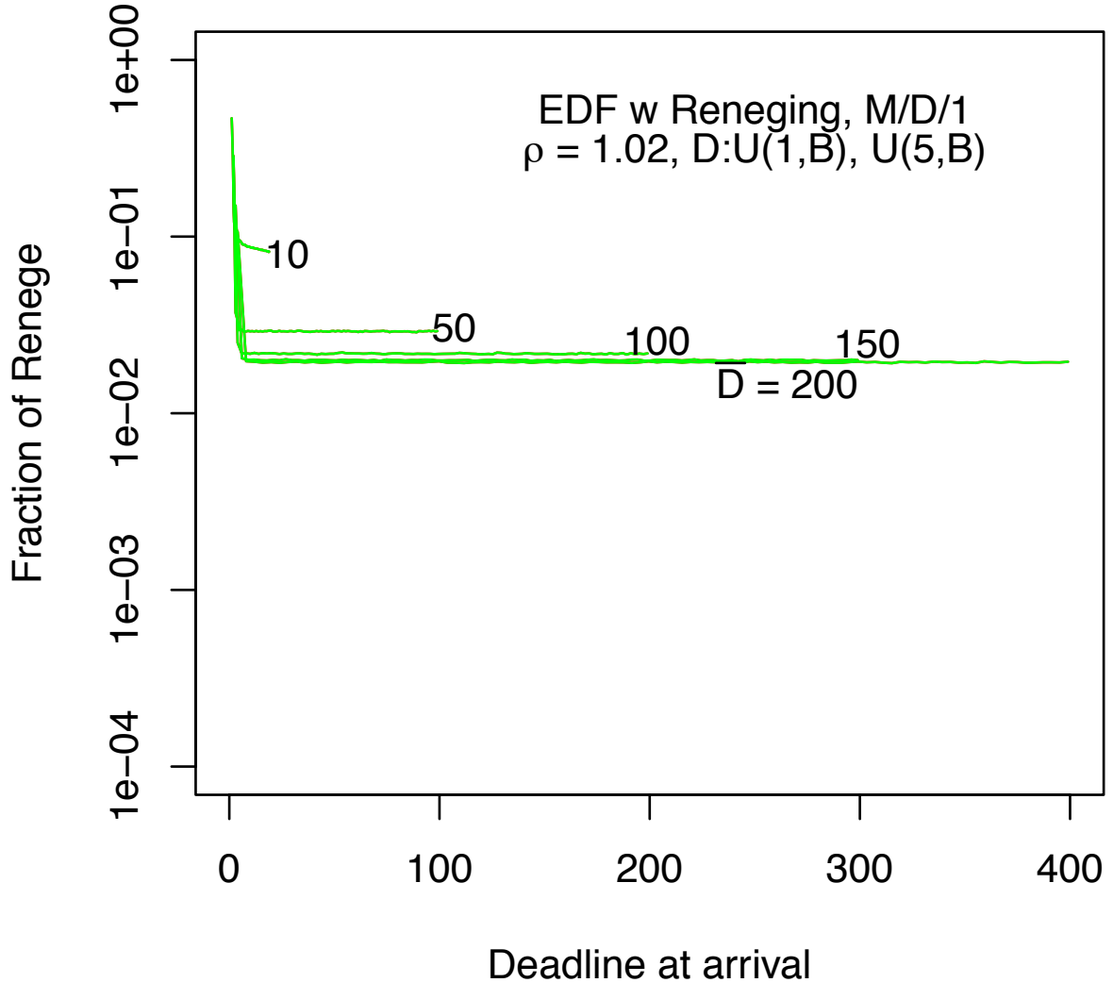




EDF with Reneging in Overload

- Reneging property is similar to underloaded traffic.





EDF with Reneging in Internet

- NS2 simulation with 3 scenarios.
  - Shorter deadline CBR and 4 FTPs
    - VoIP, Online game
  - Shorter deadline Web and 4 FTPs
    - Interactive Web
  - 2 shorter and 2 longer deadline FTPs
    - Impact of scheduling and reneging on existing transport

Related Works

- Idea is SIMPLE, and is NOT something new !
  - Delay bound info. in header - IPv4 TTL (RFC741 ’81)
  - EDF packet scheduler - IEEE TON ’96, ’97, ...
  - Trade delay and loss -
    - Asymmetric Best Effort (IEEE Network ’02)
    - Equivalent Differentiated Services (ISCC ’02)
    - Rate-Delay Service (SIGCOMM ’08)

- Problem is also NOT new !
  - Bufferbloat - Packet Switches With Infinite Storage (RFC 970 ’85)
- Idea is SIMPLE, but theoretical support is recent work*.

*L. Kruk,J. Lehoczky,K. Ramanan,and S. Shreve. “Heavy traffic analysis for EDF queues with reneging”, The Annals of Applied Probability, 21(2):484–545, 2011

Future Works

- Network system:
  - Feasibility of “EDF with reneging” scheduler
    - pFabric*, Calendar queueing
    - How to handle reneged data ?
      - Discard / reschedule with (L)BE
    - Consider serialization delay ?
      - Advantage in smaller packets.
- End system:
  - API design and application

*Alizadeh, Mohammad, et al. "pFabric: Minimal Near-Optimal Datacenter Transport," ACM SIGCOMM ACM, 2013

Conclusion

- Latency AWare InterNet architecture
- EDF with reneging scheduling
  - Simple queueing models
  - Transports and Applications

- This work is partially supported by MEXT, Japan.