Dynamic communication in a single-hop radio network

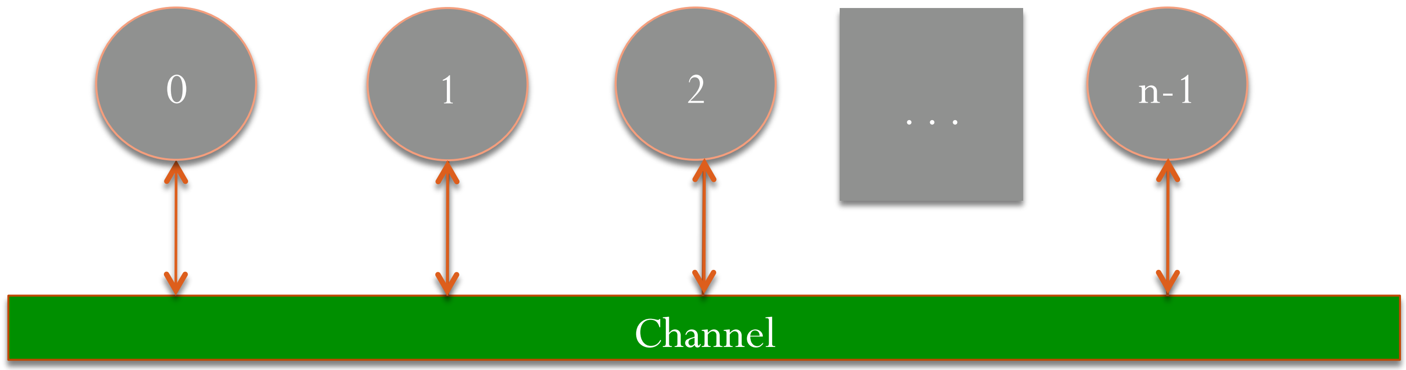
Darek Kowalski

Collaborators:
L. Anantharamu, B.S. Chlebus, M.A. Rokicki

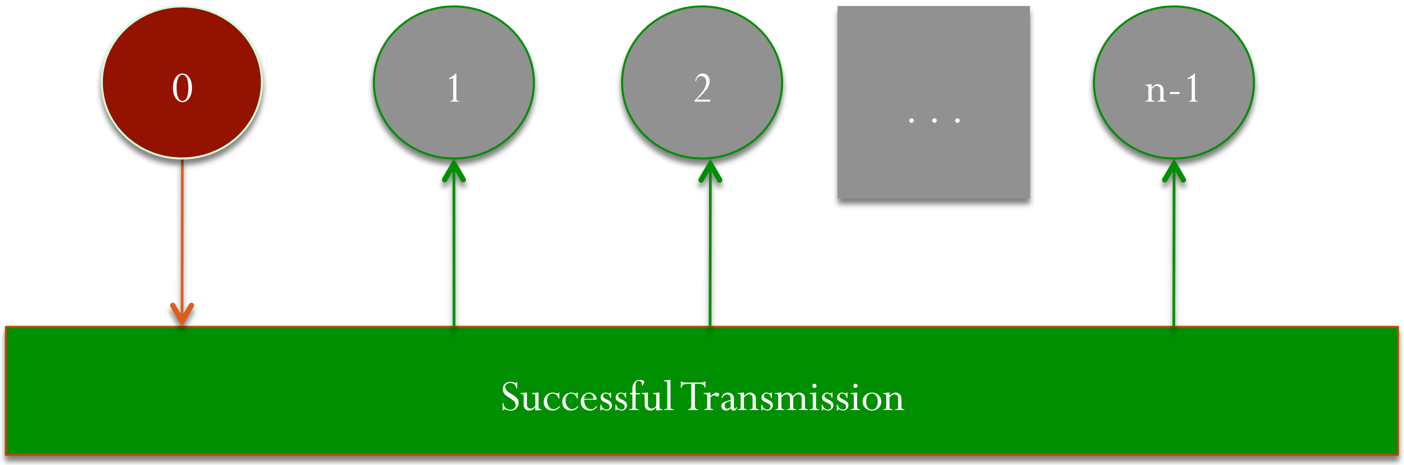
Founding: NSF (0310503), EP/H018816/1, EP/G023018/1

Single-hop radio network

- Single-hop radio network consists of ***n*** stations
- Each station has unique ***id*** from {*0,…,n-1*}
- Stations transmit packets in discrete rounds (slots)
- Transmission reaches all the station in the same round

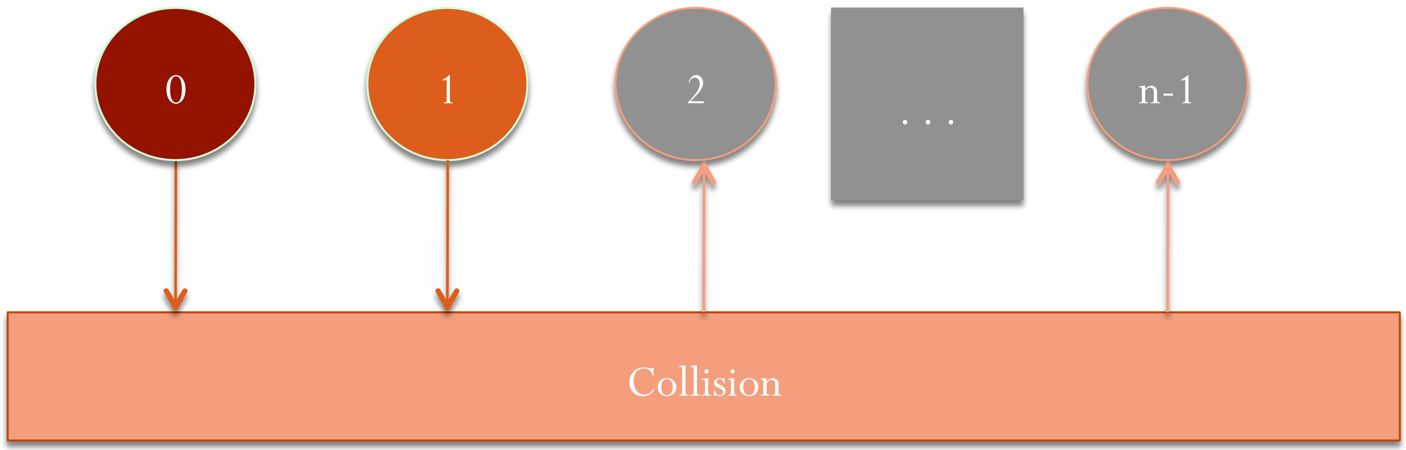


Single-hop radio network communication



Transmission is ***successfully received*** by all the stations in the system *if and only if* one station transmits in a round

Single-hop radio network communication



- If there are at least two stations transmitting in the same round then ***collision*** occurs and no packet is successfully received
- If channel provides ***collision detection*** capability then stations are able to distinguish between silence and collision; otherwise collision is heard as a silence

Dynamic broadcast problem

- Packets are *injected dynamically* into stations
- Injection pattern in modeled by the *worst case adversary*
- Stations store injected packets in their private *queues*
- Each station runs its instance of a protocol, which decides about packet's transmissions

- **Goal: design a protocol that *minimizes queue size and packet latency***

Leaky Bucket Adversary (LBA)

- ***Leaky bucket adversary*** is defined by two parameters:
  - ✓ *injection rate* $\boldsymbol{\rho}$
  - ✓ *burstiness* ***b***
- In each interval of ***t*** rounds adversary can inject at most

$$t\rho + b$$

  packets
- Adversary decides which stations get injected packets

Dynamic protocol

- Input parameters for a protocol contains only:
  - station ***id***
  - ***n*** **-** total number of stations in the system
- Protocol is an automaton. State transition is determined by the following round's events:
  - Feedback from channel:
    - Successful transmission
      Additionally packet may carry extra bits used by a protocol
    - Silence
    - Collision, for the channel with collision detection
  - Number of packets injected into the station

Quality of protocol

- ***Packet latency*** – max time after which a packet is transmitted
  - We say that protocol is (strongly) ***fair*** if packet latency is bounded
- ***Queue size*** – max number of pending packets in the system
  - We say that protocols is ***stable*** if queue size is bounded

Classes of deterministic protocols

- **Full sensing** protocol:

  stations are synchronized by global round number; protocol can transmit only packets (without additional bits)

- **Adaptive** protocol:

  extension of full-sensing protocol in which control bits can be piggybacked on packets e.g., station can indicate that it has transmitted the last packet from its queue

Randomized protocols

- Backoff protocols:
  after i^{th} failure to transmit a packet, wait r rounds to retransmit, where r is chosen uniformly at random from
  $$\{1,\ldots,F(i)\}$$
- Binary Exponential Backoff uses $F(i) = 2^i$;
  this protocol is used in 802.11 standard for wireless network
- Polynomial Backoff uses $F(i) = i^c$, where $c > 1$

Related work: stochastic approach

Packets are injected into stations with some stochastic distribution e.g., Poisson, Bernoulli:

- Analysis of backoff protocols for multiple access channels ***[Hastad, Leighton, Rogoff SICOMP 1996]***
- A bound on the capacity of backoff and acknowledgement-based protocols ***[Goldberg, Jerrum, Kannan, Paterson SICOMP 2004]***
- Contention resolution with constant expected delay ***[Goldberg, MacKenzie, Paterson, Srinivasan JACM 2000]***

Related work: deterministic settings

- Static k -conflict resolution problem:
  given $\boldsymbol{k}$ packets injected into $\boldsymbol{k}$ out of $\boldsymbol{n}$ stations, the goal is to design protocol which transmits all $\boldsymbol{k}$ packets in the shortest possible time

  - ✓ Upper bound: $O\left(k \log \frac{n}{k}\right)$

    [Komlos, Greenberg** IEEE T. Inf. Theory **1985]

  - ✓ Lower bound: $\Omega\left(\frac{k}{\log k} \log n\right)$

    [Greenberg, Winograd** JACM **1985]

Related work: adversarial queuing

Adversarial queuing was introduces in the context of store and forward networks to study stability of routing protocols:

- Universal-stability results and performance bounds for greedy contention-resolution protocols
  [Andrews, Awerbuch, Fernandez, Leighton, Liu, Kleinberg JACM 2001]
- Adversarial queuing theory
  [Borodin, Kleinberg, Raghavan, Sudan, Williamson JACM 2001]
- Adversarial contention resolution for simple channels, queue-free model, randomized algorithms
  [Bender, Farach-Colton, He, Kuszmaul, Leiserson SPAA 2005]

Injection rate ρ=1

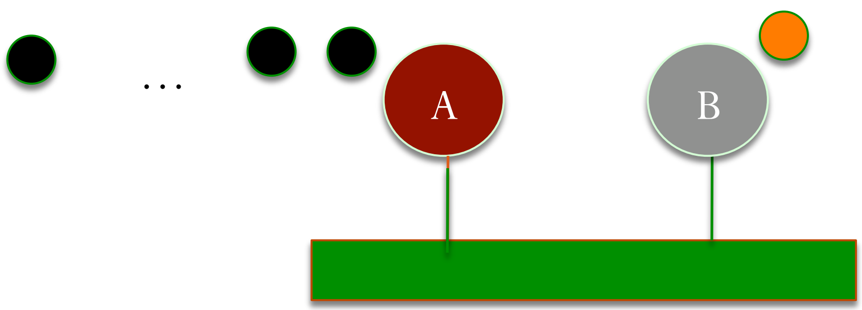
Main results

	$n = 1$	$n \geq 2$
full-sensing	Stable and Fair	No Stable
adaptive	Stable and Fair	No “Stable and Fair” Stable

[Chlebus, Kowalski, Rokicki, DC 2009]

Impossibility result

- No protocol is fair and stable against leaky bucket adversary with $\rho = 1$ and $\mathbf{b} = 1$ in the system of two stations
- Idea of the proof:
  1. Assume that stable and fair protocol exists
  2. Enforce a silent round while maintaining injection rate **1**
  3. Repeatedly enforce a silent round while maintaining injection rate **1**



Impossibility result - proof

Execution 1:

1. Round **t_0** – adversary starts injecting one packet per round into station A; we assume that adversary can use its burstiness **$b = 1$**
2. Round **t_1** – station B becomes empty;
   such round exists since the protocol is fair and the adversary injects packets only into station A
3. Round **$t_1 + 1$** – adversary uses its burstiness **$b = 1$** to inject one packet into station B
4. Round **t_2** – station A pauses its transmission and station B transmits its packet; such round exists due to fairness

Impossibility result - cont.

Execution 2:

1. Round $\mathbf{t_0}$ – adversary starts injecting one packet per round into station A; we assume that the adversary can use its burstiness $\mathbf{b = 1}$
2. Round $\mathbf{t_1}$ – station B becomes empty; such round exists since protocol is fair and the adversary injects packets only into station A
3. *Round $t_1 + 1$ – adversary does not inject into station B*
4. *Round t_2 – silent round; station A pauses its transmission and station B does not transmit*

Introduction to stable protocol

- We say that station is **big** if it queues at least ***n*** packets
- Order of station's transmission is determined by the cycled list with the list beginning pointer (initially set to station 0)
- Each station maintains a copy of the list;
  initially the list is ordered according to stations ***ids***
- Each station maintains a pointer to the **active station**; only one station is active in a round;

  initially station **0** is active

Description of MBTF

- Protocol **Move Big To Front (MBTF)**:

  Repeat:

  - ✓ active station transmits provided it has a packet
  - ✓ if the active station is big then it is moved to the front of the list and it keeps transmitting until its queue contains **n – 1** packets
  - ✓ the next station on the list becomes active

Complexity of MBTF

- Protocol Move Big to Front (MBTF) is stable but not fair against $\mathbf{LB}(\rho = 1, \mathbf{b} \geq 1)$
- MBTF stores at most $\mathbf{O}(\mathbf{n}^2 + \mathbf{b})$ packets in the system
- Each protocol stable against $\mathbf{LB}(\rho = 1, \mathbf{b} \geq 1)$ has to store at least $\Omega(\mathbf{n}^2 + \mathbf{b})$ packets

Injection rate $\rho<1$

Protocols RRW and OFRRW

- **Round Robin Withholding (RRW)**:
  Stations unload their queues in round-robin manner;
  More precisely: let assume that station $\boldsymbol{i}$ is the current station unloading its queue; after the first silent round the next station $\boldsymbol{i+1}$ modulo $\mathbf{n}$ takes over to unload its queue
- ***Phase*** of the protocol is one run over all the stations from station $\mathbf{0}$ to station $\mathbf{n-1}$
- Packet is ***old*** if it was injected in the previous phase
- **Old-First Round Robin Withholding (OFRRW)**:
  In the current phase only *old* packets can be transmitted

Protocols SRR and OFSRR

In case collision detection is available:

- **Search Round Robin (SRR)**:
  Similar to RRW, the only difference is that a new station to transmit is found using a binary search on the (cycled) list of subsequent stations; binary search uses collision detection
- ***Phase*** of the protocol is one run over all the stations from station **0** to station **n – 1**
- Packet is ***old*** if it was injected in the previous phase
- **Old-First Search Round Robin (OFSRR)**:
  In the current phase only *old* packets can be transmitted

Parameters and methodology of simulations

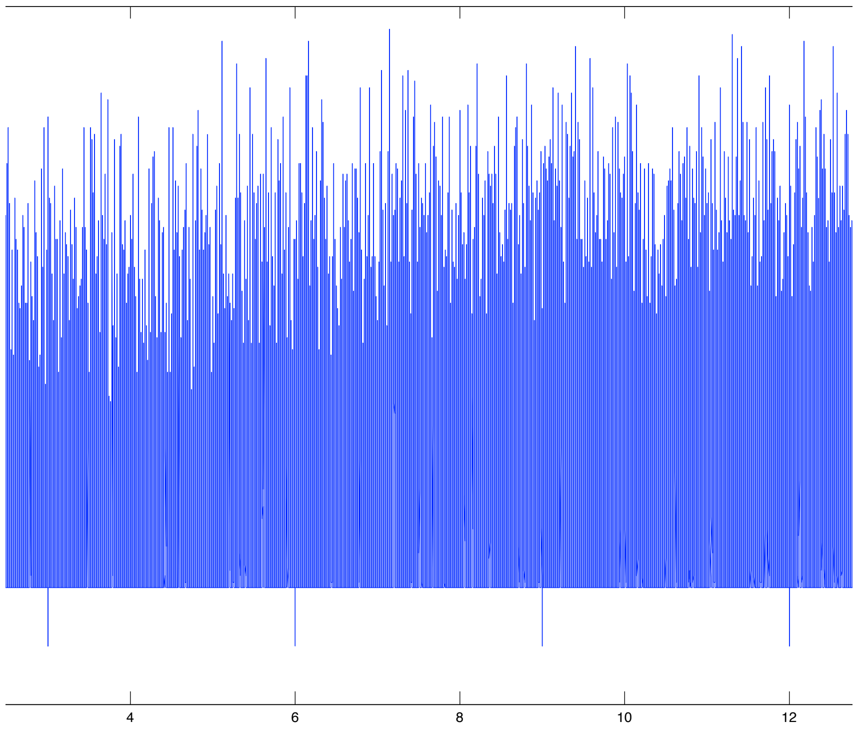
Basic parameters:

- Injection rate
- Burstiness
- Number of stations
- Number of rounds

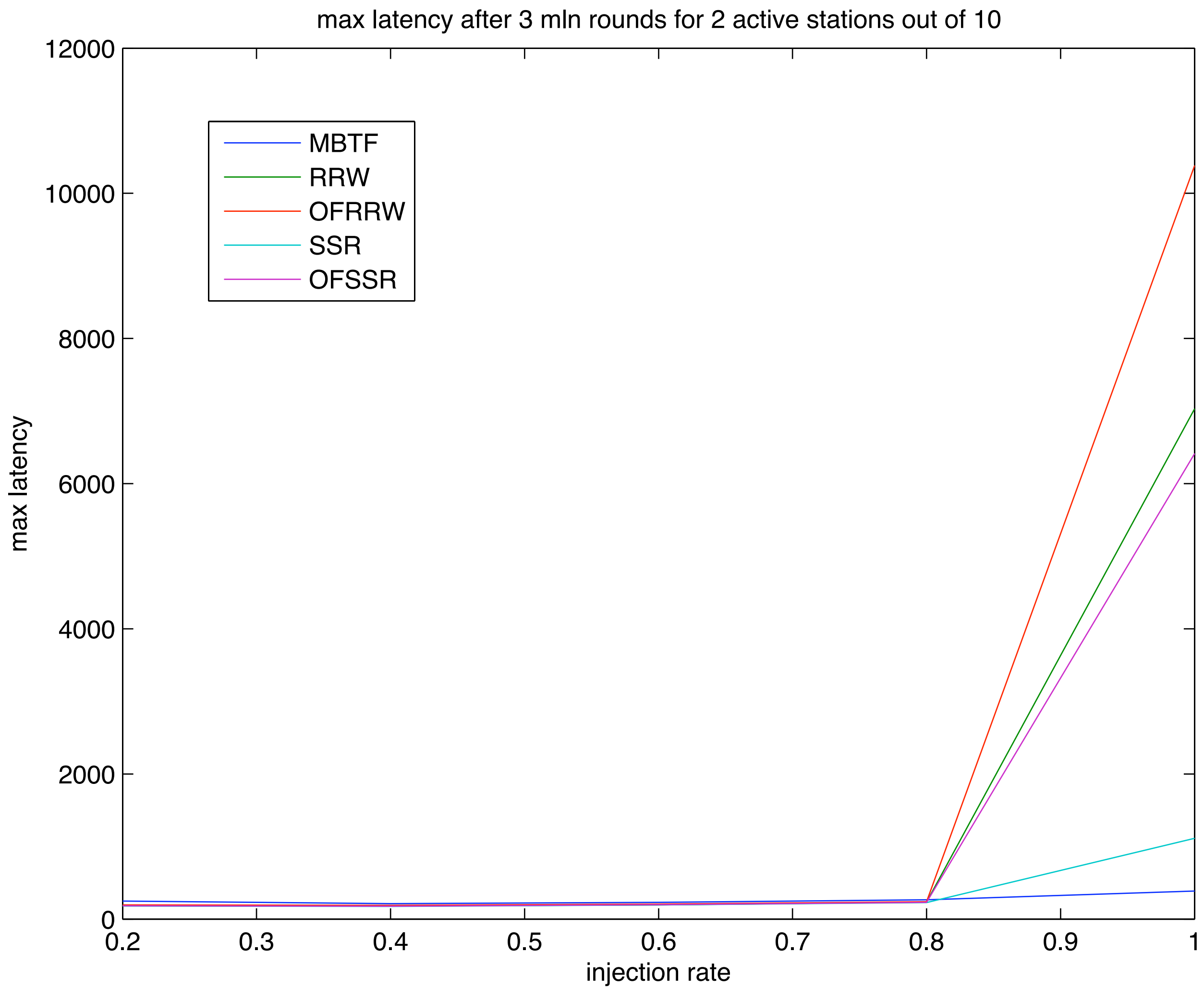
Additional parameters:

- Activity rate (packets injected only to active stations)
- Probability of changing activity status
- Fixed Poisson distribution of the number of injected packets

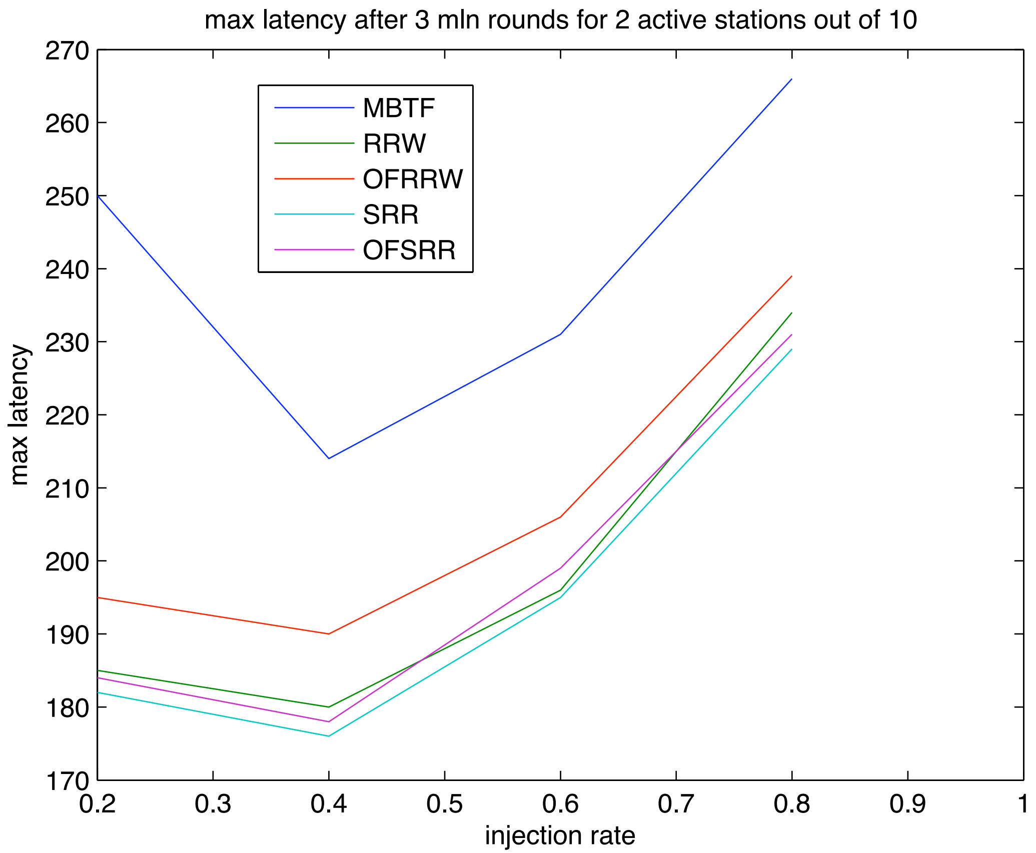
Adversarial injection pattern



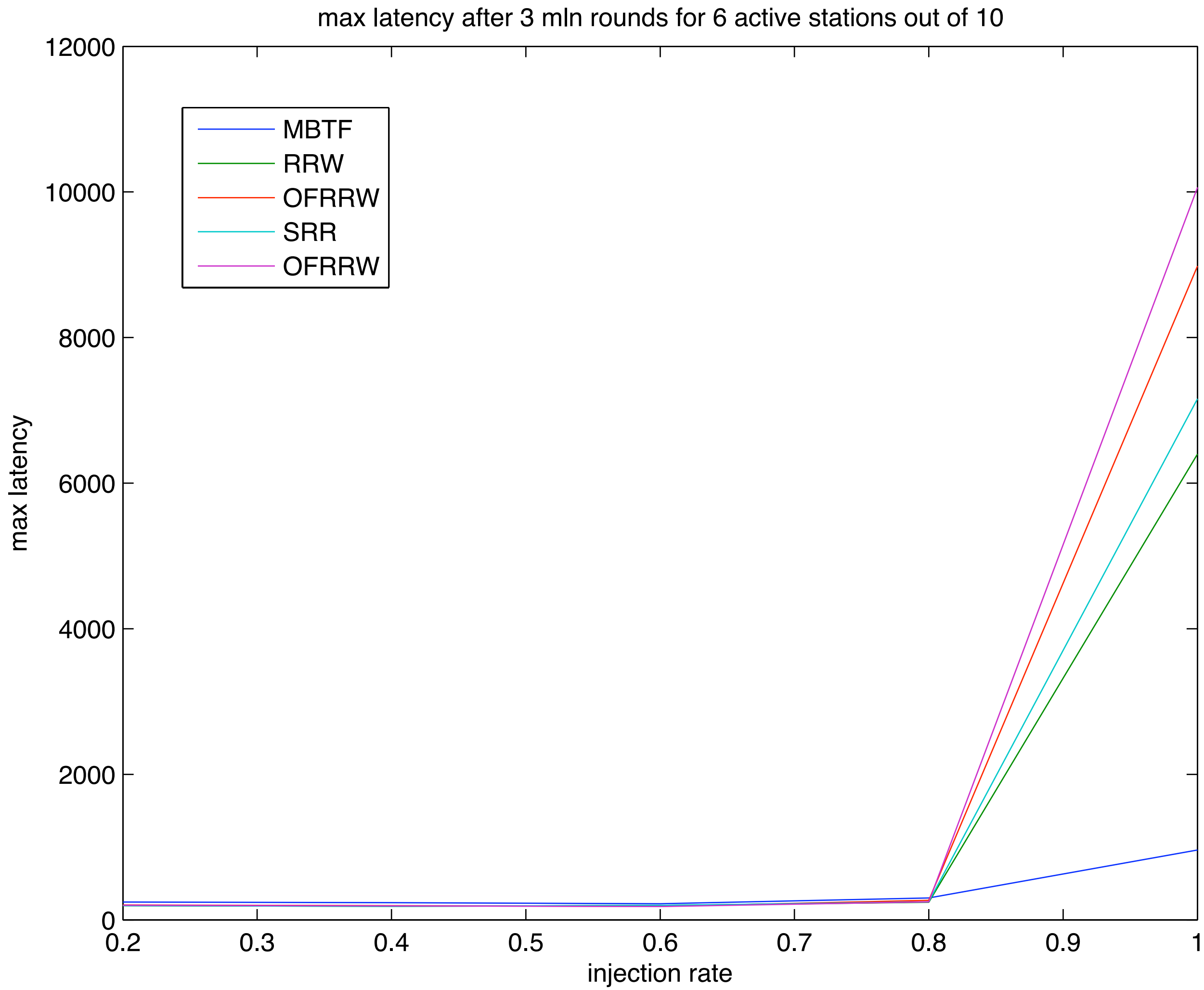
Low activity – deterministic



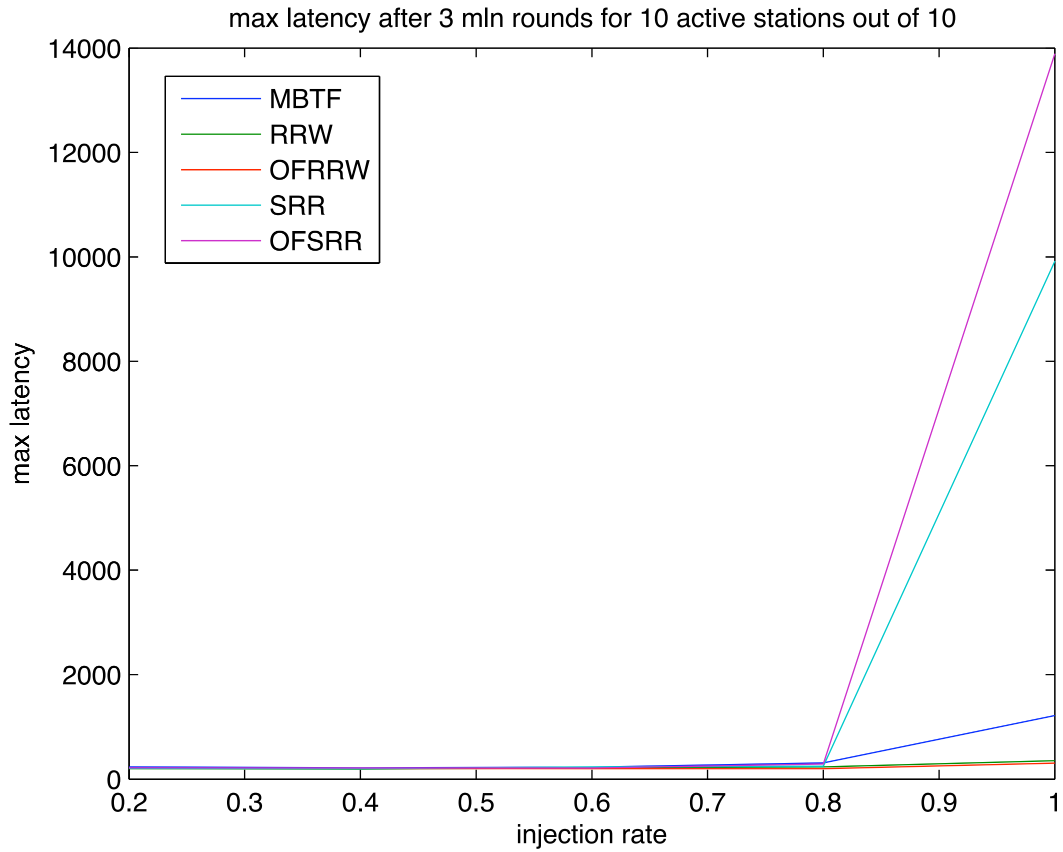
Low activity – deterministic cont.



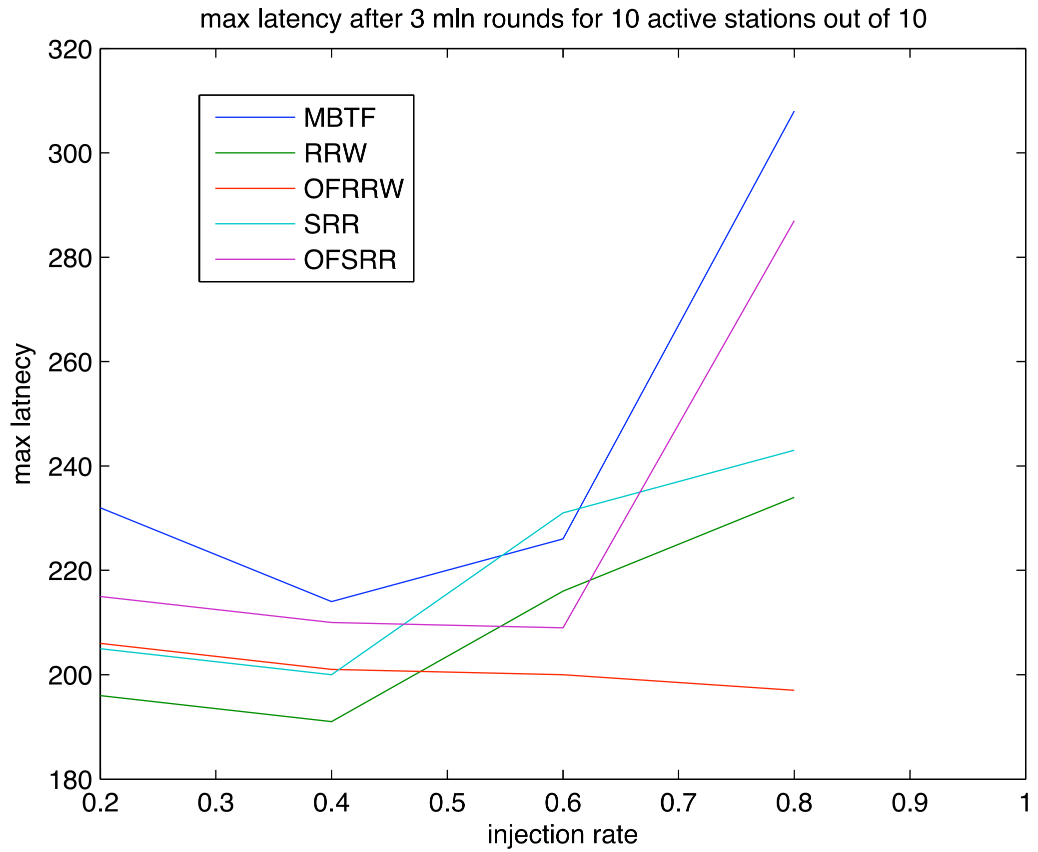
Medium activity – deterministic



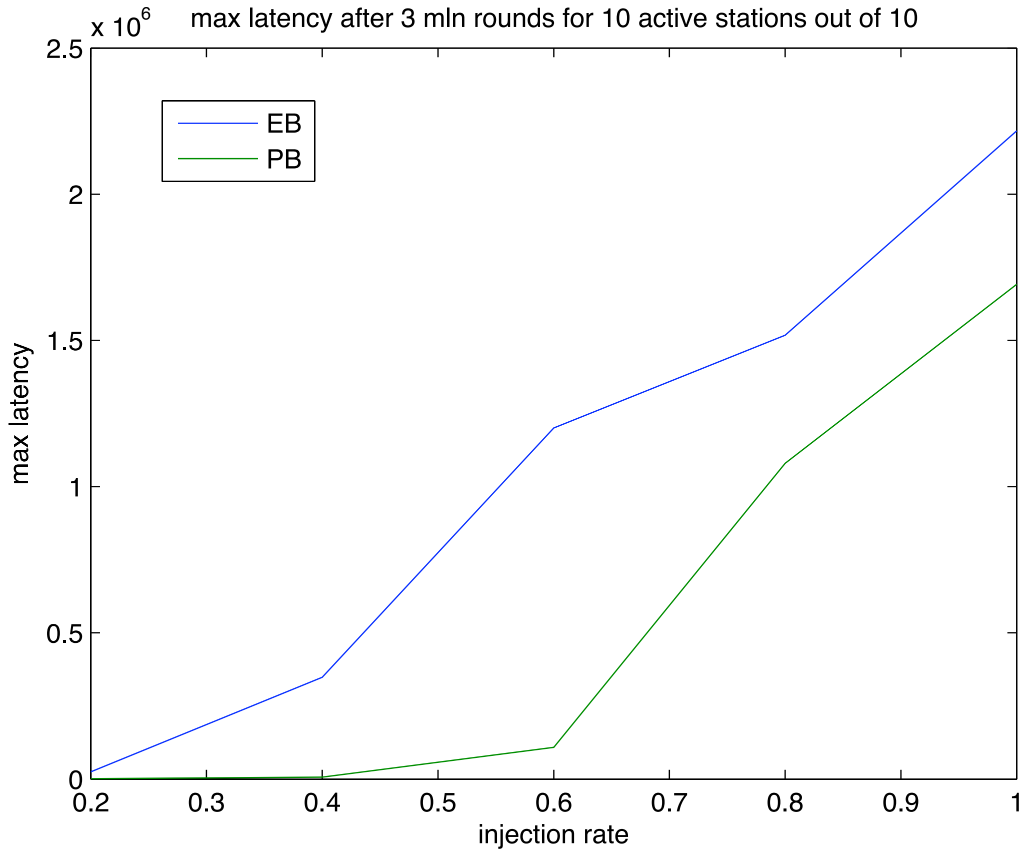
High activity – deterministic



High activity – deterministic cont.



High activity – randomized



Lower bound on queue sizes

- For each protocol ***P*** there exists an execution in which there are at least

$$\Omega\left(\max\left(b, \rho\frac{b}{\log b}\log n\right)\right)$$

pending packets in the system

- For each protocol ***P*** and **LBA(ρ ,b)**, where $\rho \geq \frac{2}{\sqrt{\log n}}$,

there exists an execution in which there are at least

$$\Omega\left(\frac{\rho^2}{4} n^{\rho^2/4} \log n\right)$$

pending packets in the system

Lower bound on packet latency

- Lower bound on queue size imply that latency is at least

$$\Omega\left(\max\left(b, \rho\frac{b}{\log b}\log n, \frac{\rho^2}{4} n^{\rho^2/4}\log n\right)\right)$$

- Each protocol ***P*** stable against **LB(** $\rho \geq$ **½,b** $\geq$**1)** delays some packet by at least

$$\Omega\left(\max\left(\frac{\rho + b}{1-\rho}, \frac{b}{\log b}\log n\right)\right)$$

Protocol queue sizes: summary

	$0 < \rho \leq 1/n$	$1/n < \rho < 1$
OFRRW	$O(b)$	$O\left(\frac{\rho n + b}{1-\rho}\right)$
MBTF	$O(b)$	$O(\rho n^2 + b)$

Packet latency: summary

	$0 < \rho \leq 1/n$	$1/n < \rho \leq 1/(2 \log n)$	$1/(2 \log n) < \rho < 1$
OFSSR	$\min\{n + b, b \log n\}$	$\min\{n + b, b \log n\}$	$\frac{n+b}{1-\rho}$
SSR	$b \log n$	$b \log n$	$\frac{n+b}{(1-\rho)^2}$
OFRRW	$\mathbf{n + b}$	$\frac{n+b}{1-\rho}$	$\frac{n+b}{1-\rho}$
RRW	$\mathbf{n + b}$	$\frac{n+b}{(1-\rho)^2}$	$\frac{n+b}{(1-\rho)^2}$
MBTF	$\frac{n^2+nb}{1-\rho}$	$\frac{n^2+nb}{1-\rho}$	$\frac{n^2+nb}{1-\rho}$

Future work

- Close the gaps between upper and lower bounds on packet latency and queue size
- Improve heuristics for packet injection
- Analyze backoff protocols in adversarial settings
- More practical setting (network, injection patterns, etc.)
- Different classes of protocols e.g. acknowledgement-based
- Individual injection rates