Jamming-Resistant MAC Protocols for Wireless Networks

Andrea W. Richa
Arizona State University

Motivation

Channel availability hard to model:

- Background noise
- Temporary Obstacles
- Mobility
- Co-existing networks
- Jammer

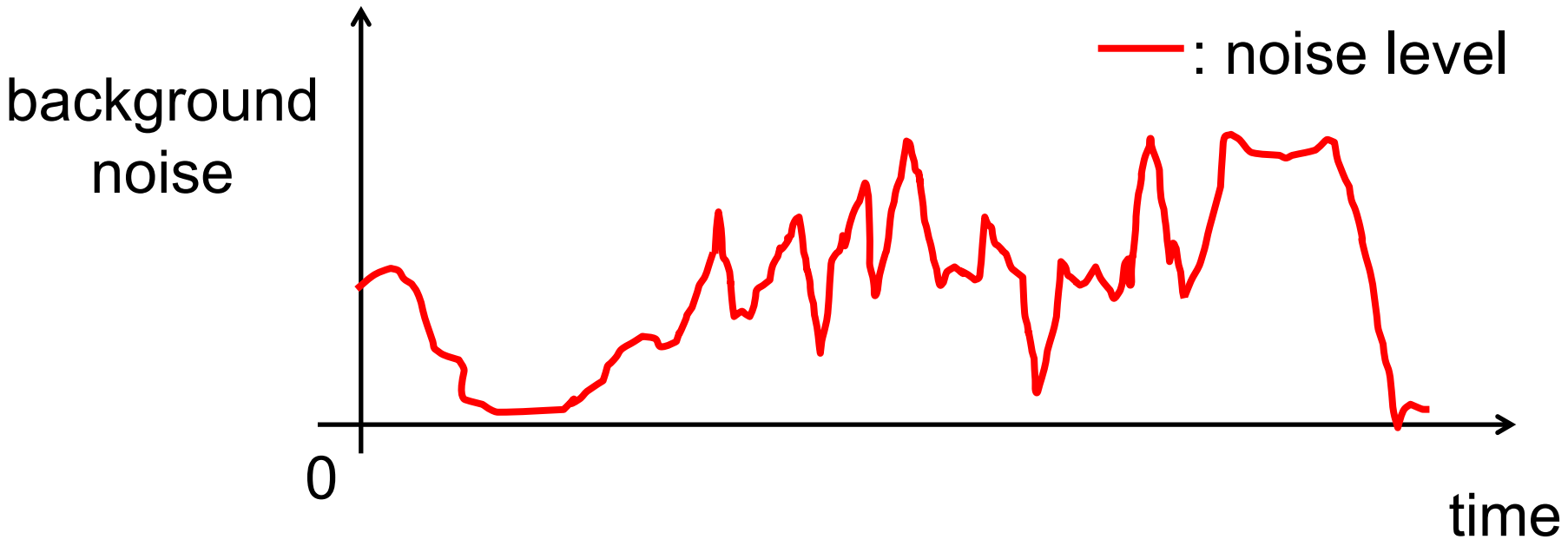
Motivation

Ideal world:

Usual approach adopted in theory.

Motivation

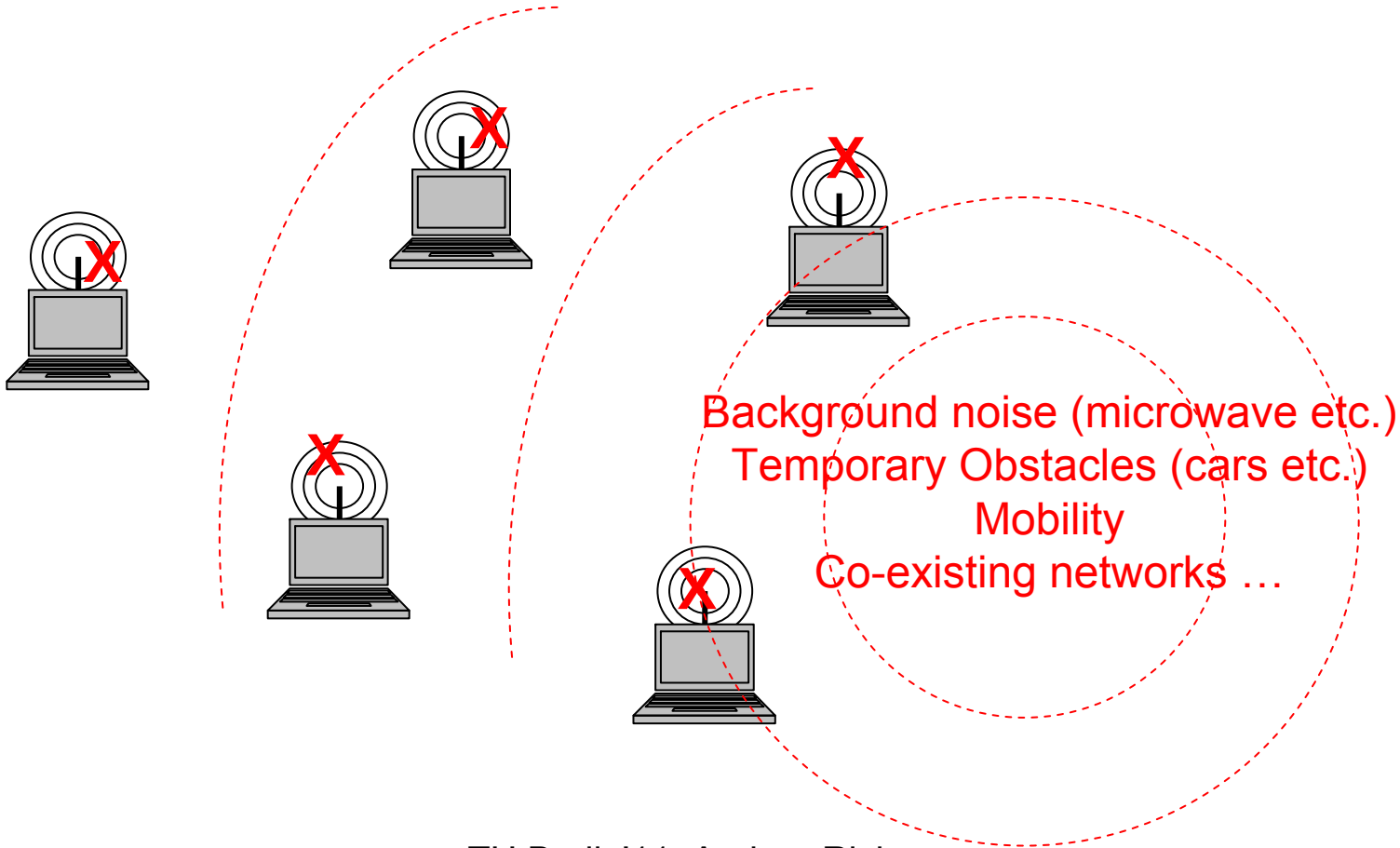
Real world:



How to model this???

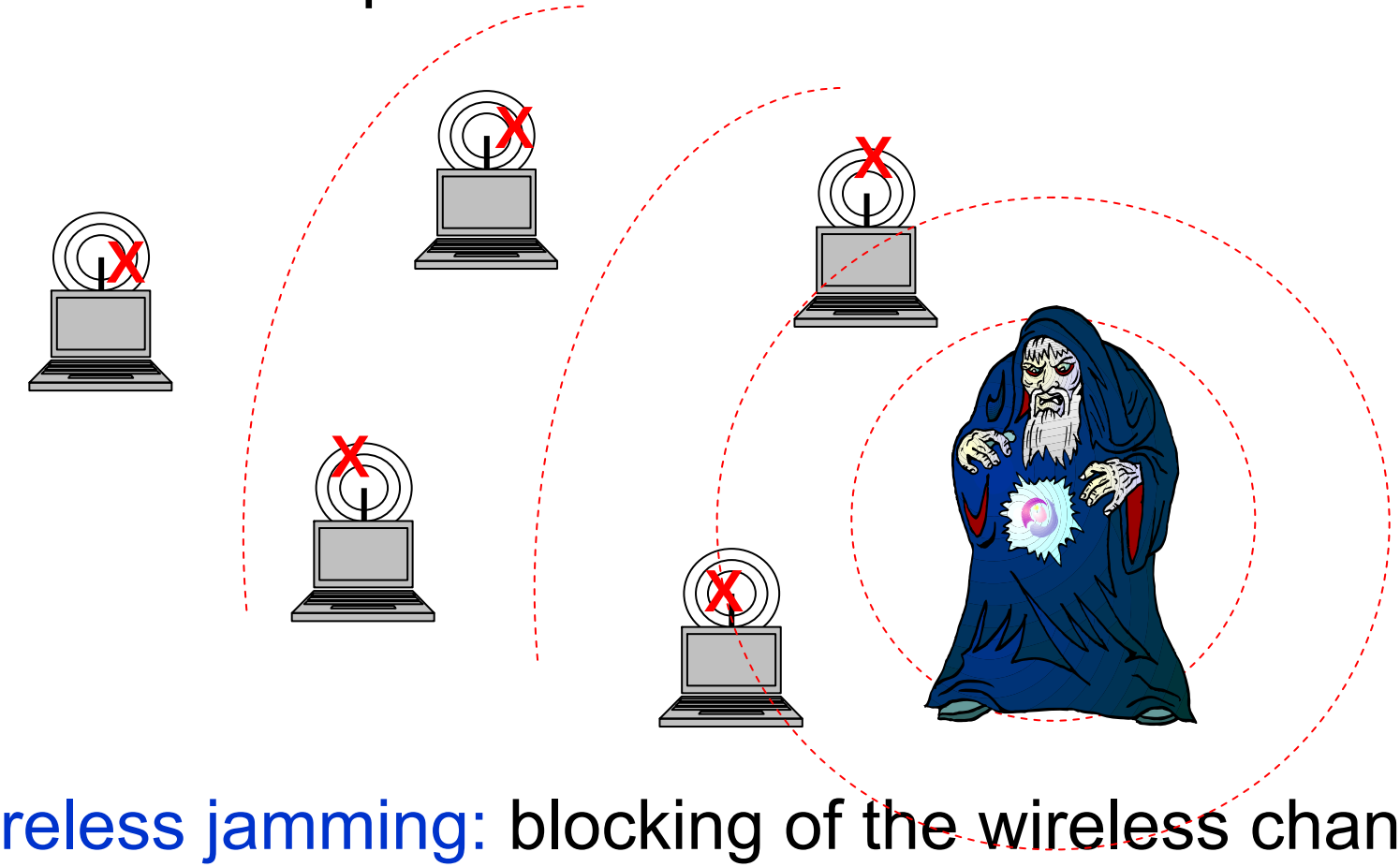
Our Approach: Adversarial Jamming

- Idea: model unpredictable behaviors via adversary!



Our Approach: Adversarial Jamming

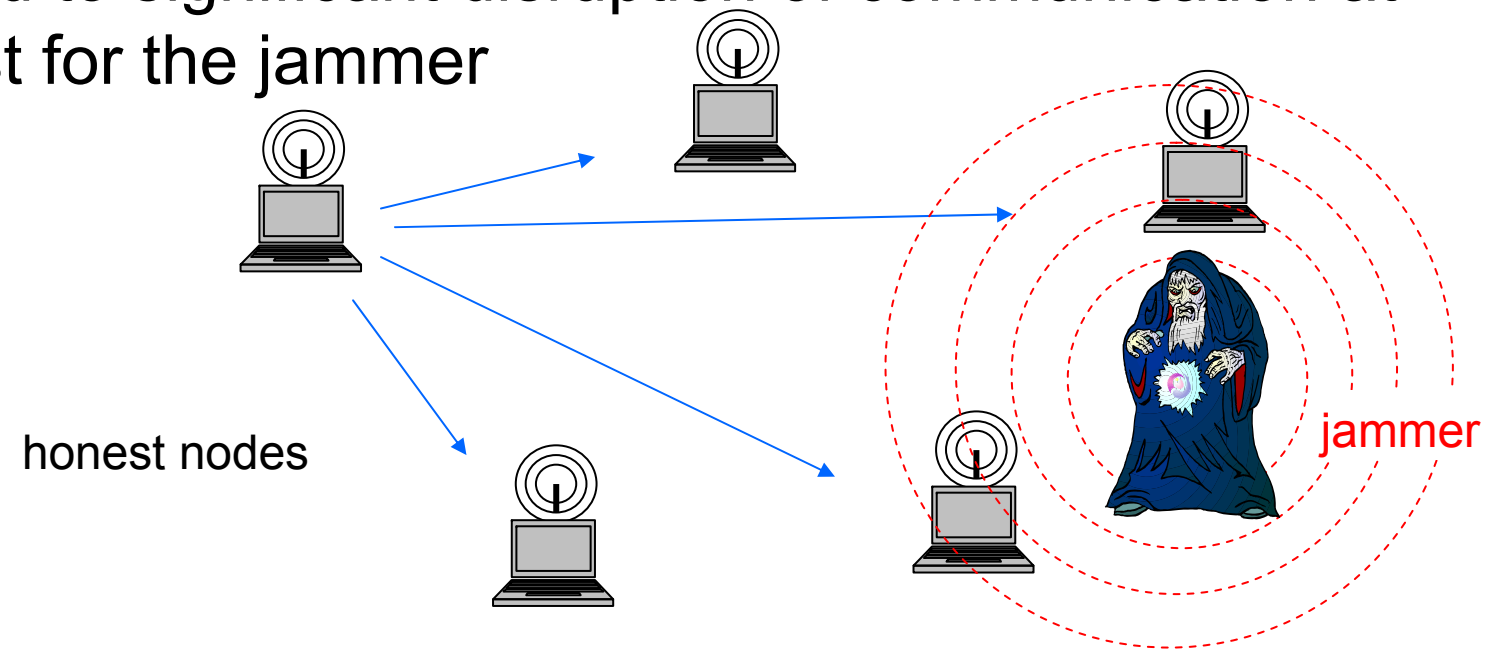
- Idea: model unpredictable behaviors via adversary!



- Wireless jamming: blocking of the wireless channel due to interference, noise or collision at receiver side

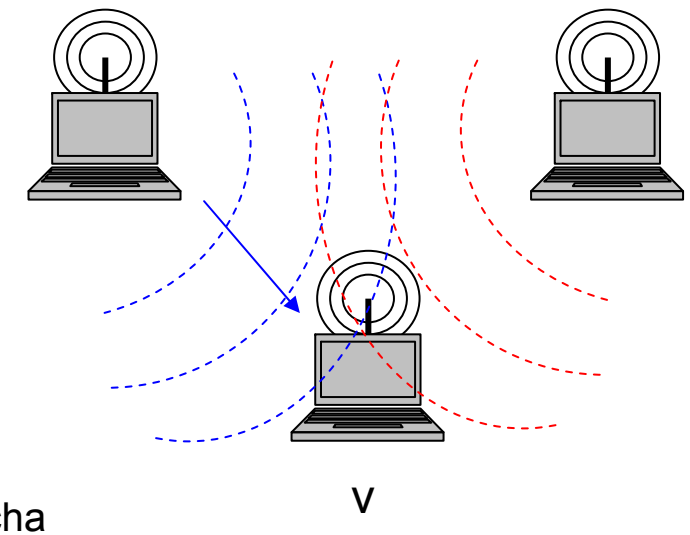
Physical layer jamming

- a jammer (adversary) listens to the open medium and broadcasts in the same frequency band as network
  - no special hardware required
  - can lead to significant disruption of communication at low cost for the jammer



Wireless communication model

- single frequency: e.g., sensor nodes
- at each time step, a node may decide to transmit a packet (nodes continuously contend to send packets)
- a node may transmit or sense the channel at any time step (half-duplex)
- when sensing the channel a node *v* may
  - sense an idle channel
  - receive a packet
  - sense a busy channel (due to collision or jamming)



Physical Layer Traditional Defenses

- spread spectrum & frequency hopping:
  - Many references in the literature (specially more applied work)…
  - rely on broad spectrum (large number of available frequencies). However, sensor nodes or common wireless devices based on 802.11 have narrow spreading factors
  - Our approach is orthogonal to broad spectrum techniques, and can be used in conjunction with those.

MAC Layer Defenses

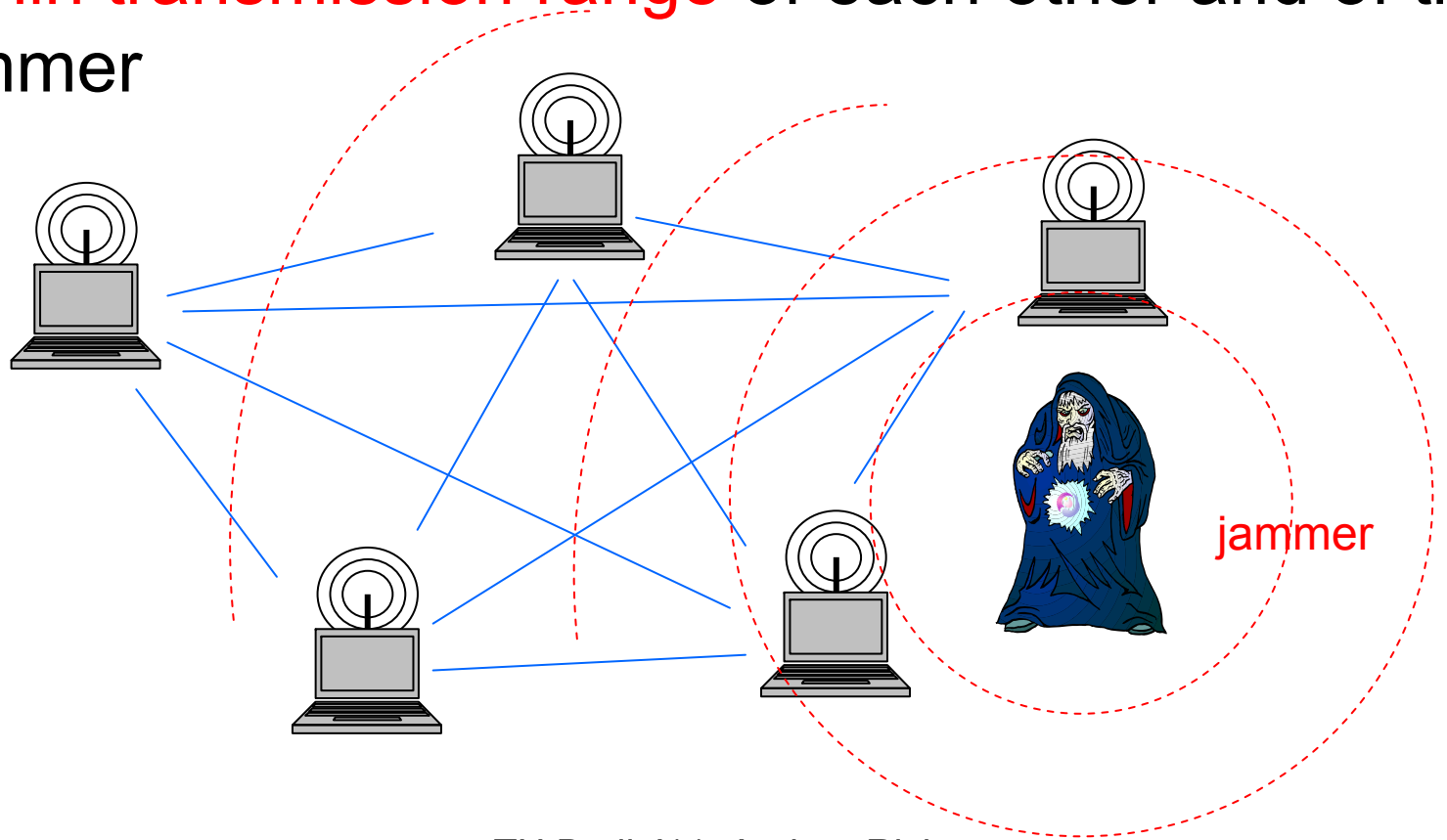
- random backoff:
  - adaptive adversary too powerful for MAC protocols based on random backoff or tournaments (including the standard MAC protocol of 802.11 [BKLNRT'08])
- [GGN'06]: cannot handle adaptive adversaries with high jamming rate
  - more general scenario (adversary can also introduce malicious messages)
  - nodes know *n*
  - not energy efficient
- [DGGN'07,'08,'09]: assume bound on # of channels adversary can jam
- Others (channel surfing, coding strategy,etc.): also cannot handle adaptive adversary

Overview

- MAC protocol : Single-hop
  - Our contributions
  - Simple (yet powerful) idea
  - MAC protocol
  - Fast recovery & Energy Efficiency
- Reactive jammers
  - Fairness
- Multi-hop networks
- Application: Leader Election
- Future work

Single-hop wireless network

- [Awerbuch, R., Scheideler, PODC'08]
- *n* reliable honest nodes and a jammer; all nodes within transmission range of each other and of the jammer



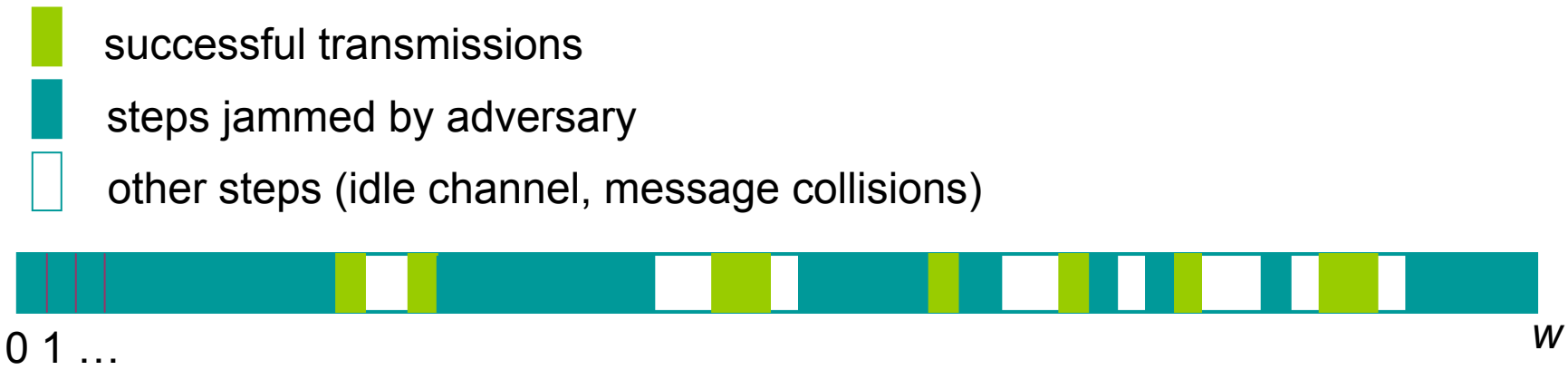
Adaptive adversary

- knows protocol and entire history

- (T,λ)-bounded adversary, $0 < \lambda < 1$: in any time window of size $w \geq T$, the adversary can jam $\leq \lambda w$ time steps

Constant-competitive protocol

- a protocol is called constant-competitive against a (T,λ)-bounded adversary if the nodes manage to perform successful transmissions in at least a constant fraction of the steps (w.h.p. or on expectation), for any sufficiently large number of steps



Our main contribution

- symmetric local-control MAC protocol that is constant-competitive against any $(T,1-\varepsilon)$-bounded adaptive adversary after $\tilde{\Omega}(T/\varepsilon)$ steps w.h.p., for any constant $0<\varepsilon<1$ and any T.

- energy efficient:
  - converges to bounded amount of energy consumption due to message transmissions by nodes under continuous adversarial jamming ($\varepsilon=0$)
- fast recovery from any state

Pros and Cons

Pros:

- no prior knowledge of global parameters
  - nodes do not know ε
- no IDs needed

Cons:

- nodes know common rough estimate $\gamma = O(1/(\log T + \log\log n))$
  - allow for superpolynomial change in n and polynomial change in T over time
- fair channel use is not guaranteed
  - we will see how to fix that later

Overview

- MAC protocol : Single-hop
  - Our contributions
  - Simple (yet powerful) idea
  - MAC protocol
  - Fast recovery & Energy Efficiency
- Reactive jammers
  - Fairness
- Multi-hop networks
- Application: Leader Election
- Future work

Simple (yet powerful) idea

- each node v sends a message at current time step with probability $p_v \leq p_{max}$, for constant $0 < p_{max} << 1$.
  $p = \sum p_v$ (cumulative probability)
  q_{idle} = probability the channel is idle
  q_{succ} = probability that only one node is transmitting (successful transmission)

- Claim. $q_{idle} \cdot p \leq q_{succ} \leq (q_{idle} \cdot p)/(1-p_{max})$

$\therefore$

if (number of times the channel is idle) $\cong$ (number of successful transmissions) $\rightarrow$ $p = \theta(1) \rightarrow$ $q_{succ} = \theta(1)$! (what we want!)

Basic approach

- a node v adapts p_v based only on steps when an idle channel or a successful message transmission are observed, ignoring all other steps (including all the blocked steps when the adversary transmits!)

Basic approach

- a node v adapts p_v based only on steps when an idle channel or a successful message transmission are observed, ignoring all other steps (including all the blocked steps when the adversary transmits!)!

Naïve protocol

Each time step:

- Node v sends a message with probability p_v . If v does not send a message then
  - if the wireless channel is idle then $p_v = (1+ \gamma)\ p_v$
  - if v received a message then $p_v = p_v/(1+ \gamma)$

  (Recall that $\gamma = O(1/(\log T + \text{loglog } n))$.)

Problems

- Basic problem: Cumulative probability *p* could be too large.
  - all time steps blocked due to message collisions w.h.p.

Problems

- Basic problem: Cumulative probability *p* could be too large.
  - all time steps blocked due to message collisions w.h.p.

Problems

- Basic problem: Cumulative probability *p* could be too large.
  - all time steps blocked due to message collisions w.h.p.
- Idea: If more than *T* consecutive time steps without successful transmissions, then reduce probabilities, which results in fast recovery of *p*.
- Problem: Nodes do not know *T*. How to learn a good time window threshold?
  - It turns out that additive-increase additive-decrease is the right strategy!

MAC protocol

- each node v maintains
  - probability value p_v ,
  - time window threshold T_v , and
  - counter c_v
- Initially, $T_v = c_v = 1$ and $p_v = p_{max}$ (< 1/24).
- synchronized time steps (for ease of explanation)

- Intuition: wait for an entire time window (according to current estimate T_v) until you can increase T_v

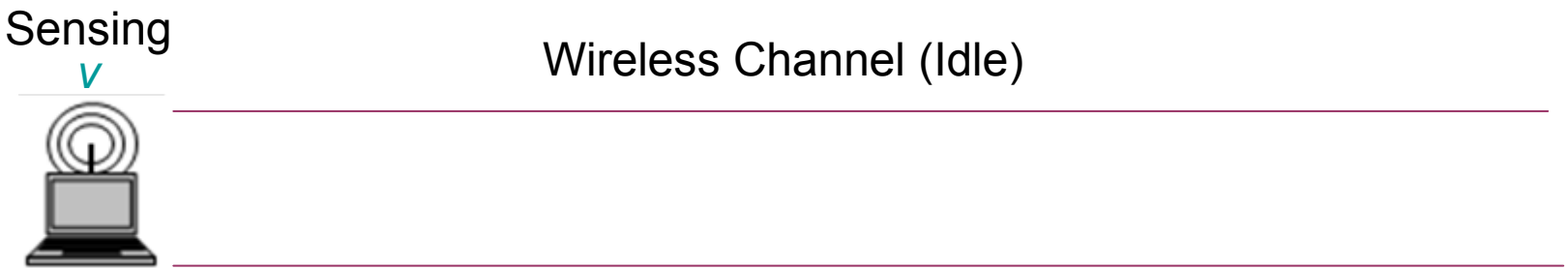
MAC protocol

In each step:

- node v sends a message with probability p_v . If v decides not to send a message then
  - if v senses an idle channel, then $p_v = \min\{(1+\gamma)p_v, p_{max}\}$
  - if v successfully receives a message, then $p_v = p_v/(1+\gamma)$ and $T_v = \max\{T_v - 1, 1\}$
- $c_v = c_v + 1$. If $c_v > T_v$ then
  - $c_v = 1$
  - if v did not receive a message successfully in the last T_v steps then $p_v = p_v/(1+\gamma)$ and $T_v = T_v + 1$

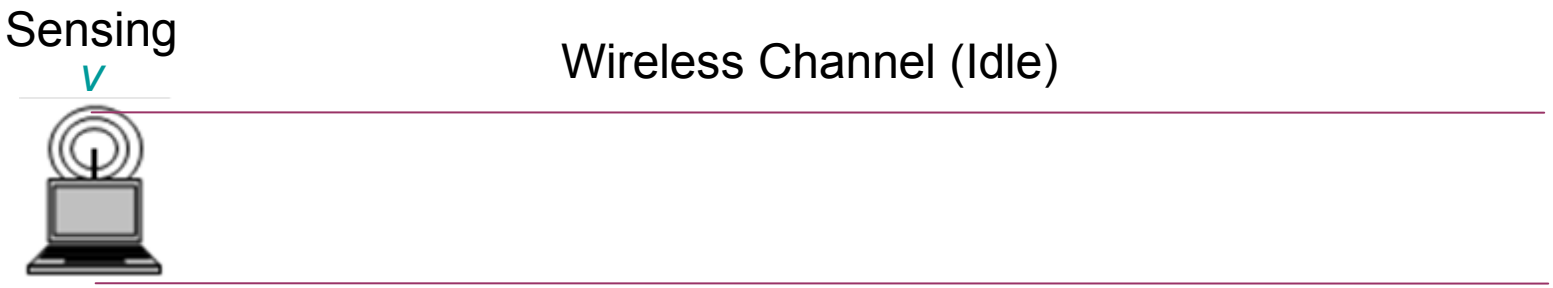
Example: Low value of *p*

- $p_v = 1/n^2$, $T_v = 3$, $c_v = 1$



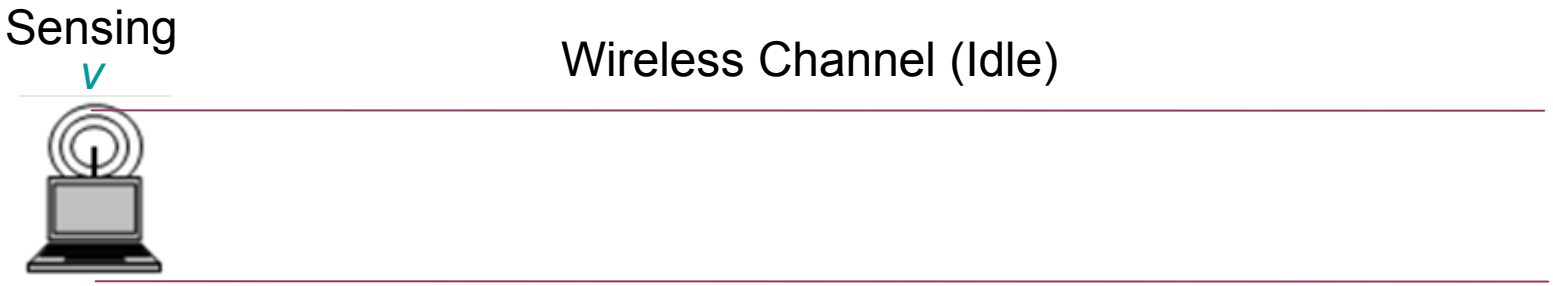
Example: Low value of *p*

- $p_v = (1+\gamma)/n^2$, $T_v = 3$, $c_v = 2$



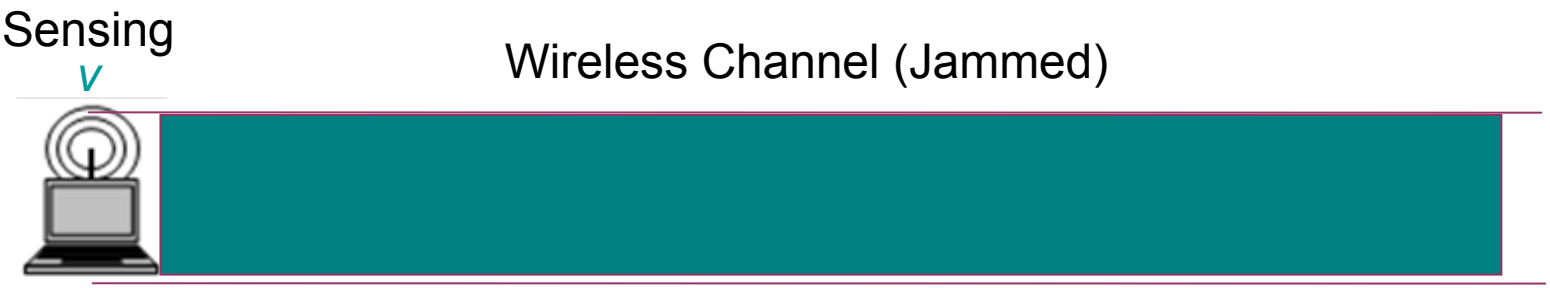
Example: Low value of *p*

- $p_v = (1+\gamma)^2/n^2$, $T_v = 3$, $c_v = 3$



Example: Low value of *p*

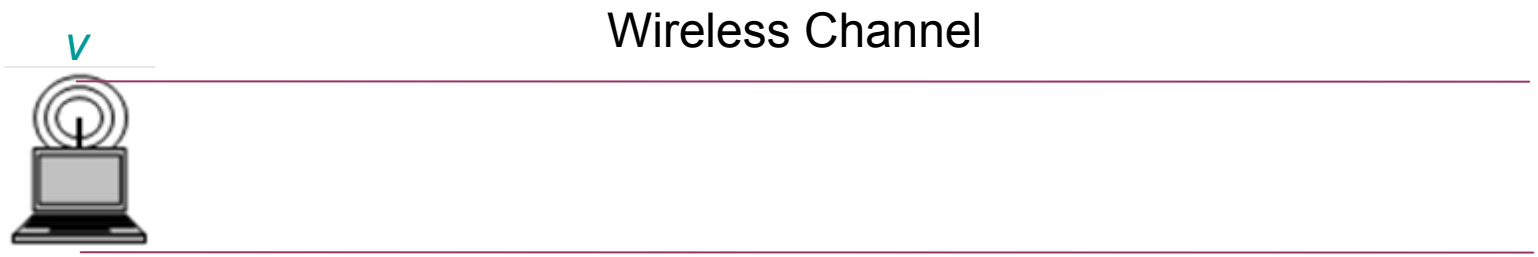
- $p_v = (1+\gamma)^3/n^2$, $T_v = 3$, $c_v = 4$



$p_v = (1+\gamma)^2/n^2$, $T_v = 4$, $c_v = 1$

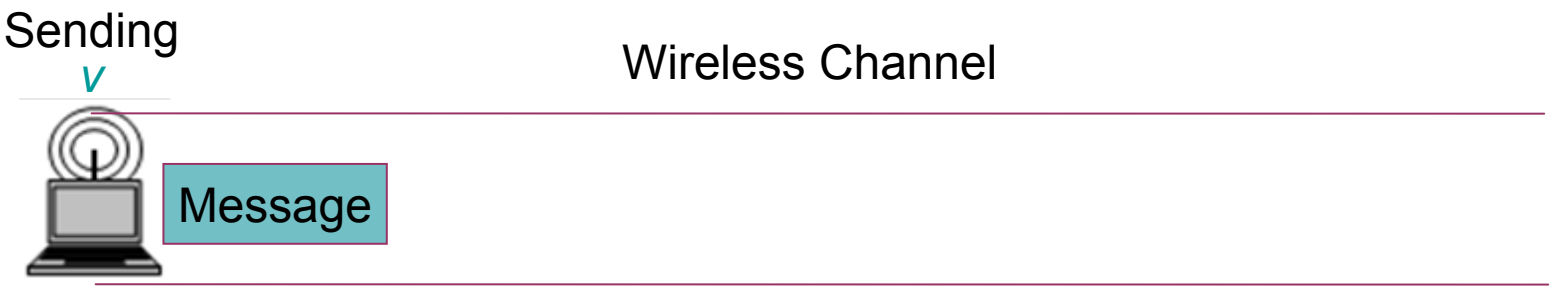
Example: Low value of *p*

- ~ polylog (*n*) idle steps later:
  - $p_v \cong c/n$, $T_v \leq \sqrt{T\ polylog\ (n)}$



Example: Large *p*

- $p_v = 1/c$, $T_v = 2$, $c_v = 1$

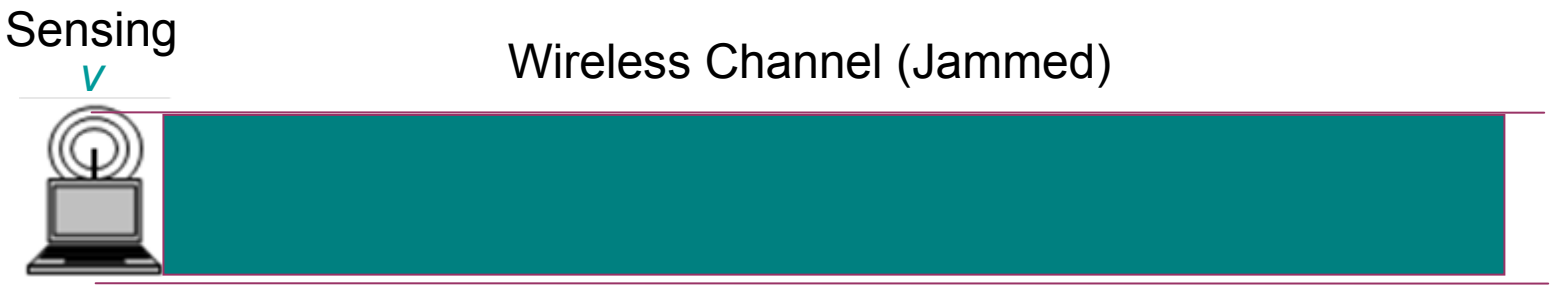


Example: Large *p*

- $p_v = 1/c$, $T_v = 2$, $c_v = 2$

Example: Large *p*

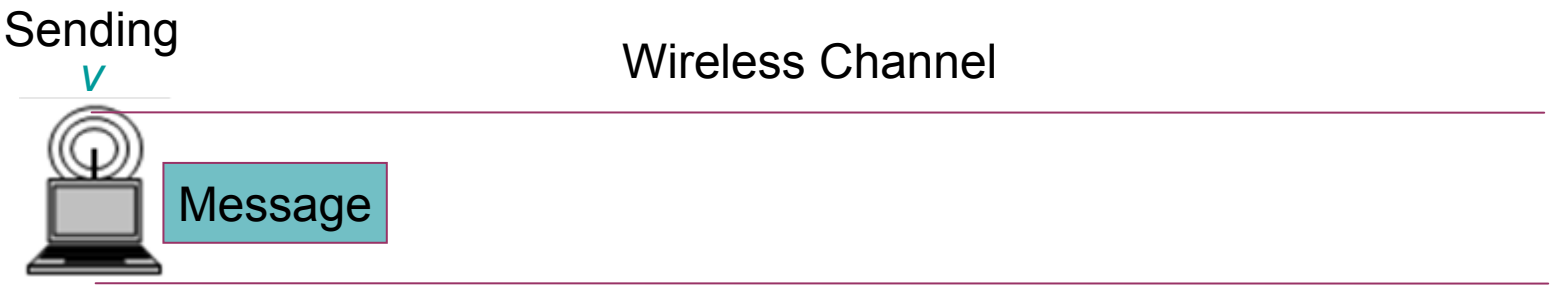
- $p_v = 1/c$, $T_v = 2$, $c_v = 3$



$p_v = 1/[c(1+\gamma)]$, $T_v = 3$, $c_v = 1$

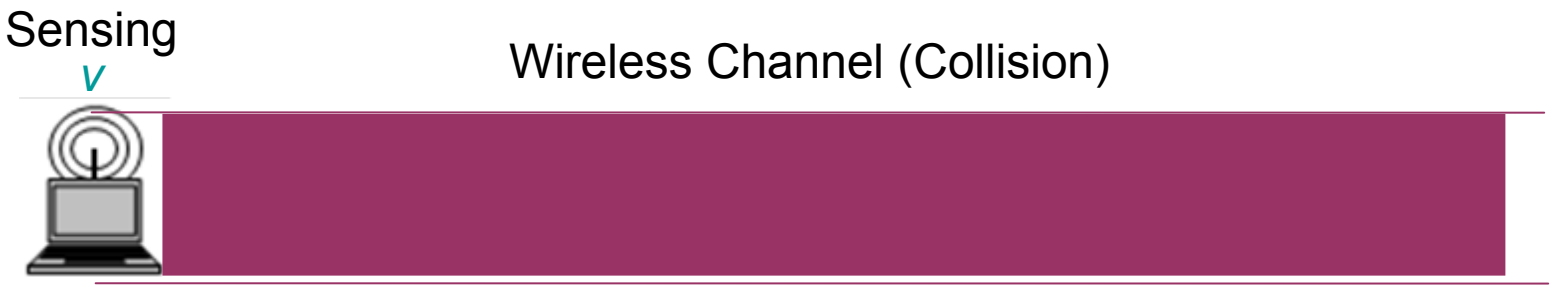
Example: Large *p*

- $p_v = 1/[c(1+\gamma)]$, $T_v = 3$, $c_v = 1$



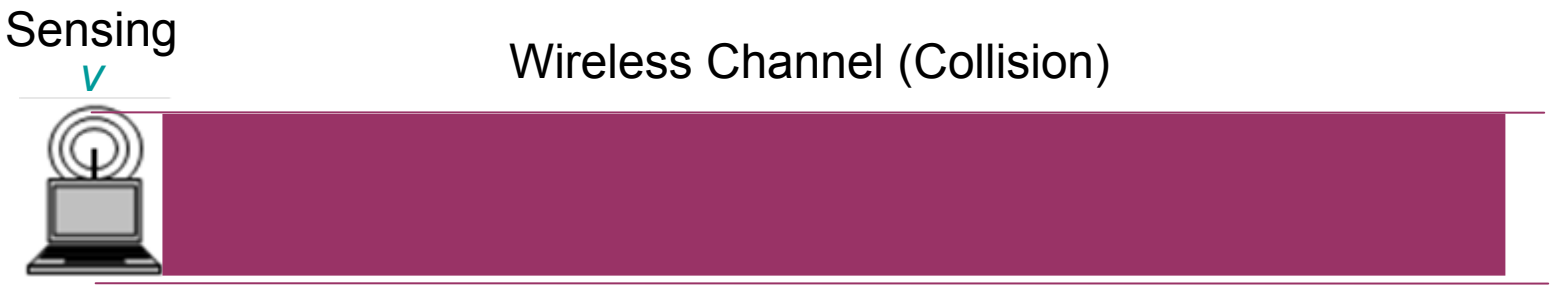
Example: Large *p*

- $p_v = 1/[c(1+\gamma)]$, $T_v = 3$, $c_v = 2$



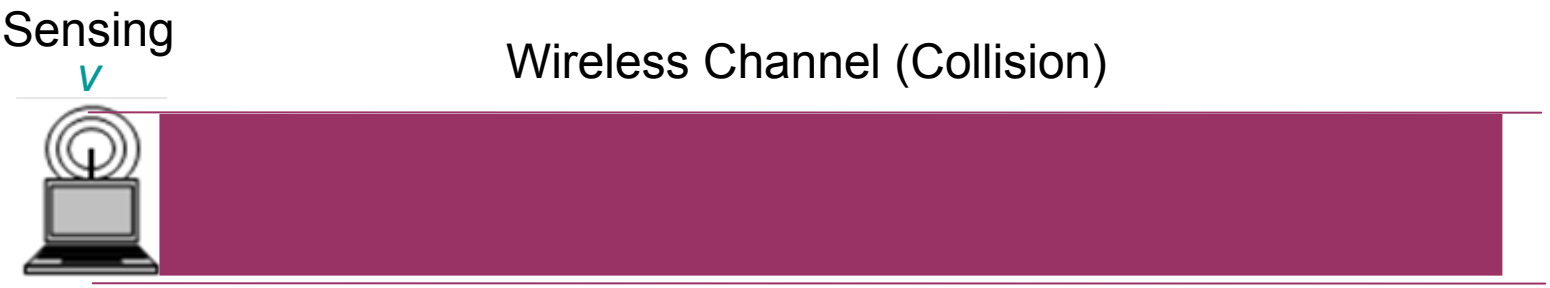
Example: Large *p*

- $p_v = 1/[c(1+\gamma)]$, $T_v = 3$, $c_v = 3$



Example: Large *p*

- $p_v = 1/[c(1+\gamma)]$, $T_v = 3$, $c_v = 4$



$p_v = 1/[c(1+\gamma)^2]$, $T_v = 4$, $c_v = 1$

Our results

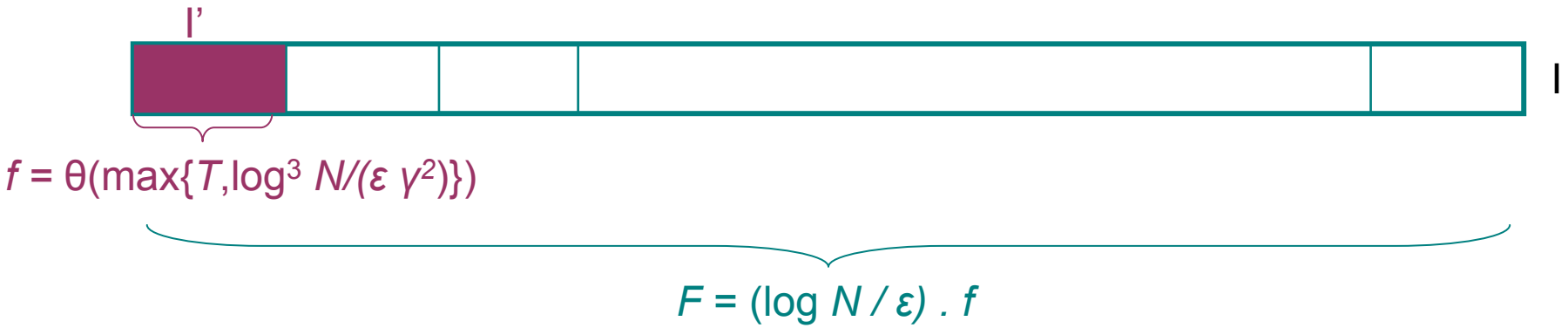
- Let $N = \max\{T, n\}$

- **Theorem.** Our MAC protocol is constant-competitive under any $(T, 1-\varepsilon)$-bounded adversary if the protocol is executed for $\Omega(\log N \cdot \max\{T, \log^3 N/(\varepsilon \gamma^2)\}/\varepsilon)$ steps w.h.p., for any constant $0<\varepsilon<1$ and any T.

Proof sketch

- Show competitiveness for time frames of $F = \theta((\log N \cdot \max\{T, \log^3 N/(\varepsilon \gamma^2)\}/\varepsilon)$ many steps

  If we can show constant competitiveness for any such time frame of size F, the theorem follows
- Use induction over the number of sufficiently large time frames seen so far. We subdivide each frame:



Proof sketch

- $p > 1/(f^2(1+\gamma)^{2\sqrt{f}})$ and $T_v < \sqrt{F}$, in each subframe *I'* w.h.p.
- $p<12$ and $p>1/12$ within subframe *I'* with moderate probability (so that adaptive adversarial jamming not successful)
- Constant throughput in *I'* with moderate probability

- Over a logarithmic number of subframes, constant throughput in frame *I* of size *F* w.h.p.

Overview

- MAC protocol : Single-hop
  - Our contributions
  - Simple (yet powerful) idea
  - MAC protocol
  - Fast recovery & Energy Efficiency
- Reactive jammers
  - Fairness
- Multi-hop networks
- Application: Leader Election
- Future work

Fast recovery

- Our protocol quickly recovers from any (T_v, c_v, p_v)-values.

- Theorem. For any initial $p_0 = \sum p_v$ and $T_0 = max\ T_v$, it takes $O([\log_{(1+\gamma)} (1/p_0)]/\varepsilon + T_0^2)$ w.h.p. until the MAC protocol satisfies again $p \geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$ and $T_v < \sqrt{F}$ for all v.

Proving fast recovery: p

- $p_0 < 1/(f^2(1+\gamma)^{2\sqrt{f}})$
- we show that it takes roughly $[\log_{(1+\gamma)}(1/p_0)]/\varepsilon$ steps to get from p_0 to $(p_0)^{1/2}$; another $[\log_{(1+\gamma)}(1/p_0)]/(2\varepsilon)$ steps to get to $(p_0)^{1/4}$; …

roughly at most $2[\log_{(1+\gamma)}(1/p_0)]/\varepsilon$ steps until cumulative probability $p \geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$

Proving fast recovery: T

- Once cumulative probability $p \geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$, count number of steps until $T_v < \sqrt{F}$ for all v
  - repeated applications of similar inductive argument as MAC protocol's, by repeatedly selecting appropriately geometrically decreasing frame sizes (starting from 4 max T_v).

Energy efficiency

- Corollary. For any time frame of size $F = \Omega((\log N \cdot \max\{T, \log^3 N/(\varepsilon \gamma^2)\}/\varepsilon)$, the total energy spent by all nodes together on sending out messages is bounded by $O(F)$ whp.

- Total amount of energy spent proportional to number of successful transmissions.

Continuous jamming

- Moreover, under a more powerful adversary that can perform continuous jamming (after $\tilde{\Omega}(T)$ steps):

- Lemma. The total energy consumption (sending out messages) during an entire continuous jamming attack is $\tilde{O}(\sqrt{T})$, independent of the length of the attack.

- Exhaust adversary's energy resources

Proving Lemma

- First we show that the total energy consumption is $O(p_0 \cdot T_0/\gamma + \log N)$ whp, where $p_0 = \sum p_v$ and $T_0 = \max T_v$ at the start of the attack
  - compute expected number transmissions out of each node given that p_v decreases by $1/(1+\gamma)$ every T_0+1, then T_0+2 , then T_0+3, … number of steps under continuous jamming
  - sum up expected values over all nodes and use Chernoff bounds
- Note that for an attack starting after $\tilde{\Omega}(T)$ steps, $p_0 = O(1)$ and $T_0 = O(\sqrt{F}) = \tilde{O}(\sqrt{T})$, whp.

Overview

- MAC protocol : Single-hop
  - Our contributions
  - Simple (yet powerful) idea
  - MAC protocol
  - Fast recovery & Energy Efficiency
- Reactive jammers
  - Fairness
- Multi-hop networks
- Application: Leader Election
- Future work

Reactive jammers

- [R.,Scheideler,Schmid,Zhang, ICDCS'11]

- Fully adaptive adversary that in addition can quickly observe the channel at the current time step before deciding to jam
  - i.e., the adversary has some knowledge about the random choices at current time step
  - Distinguishes between idle and non-idle time steps, but cannot distinguish between successful transmissions and collisions (e.g., if packets are encrypted)

AntiJam: Reactive jamming-resistant MAC protocol

- Need to "synchronize" transmission probabilities p_v, as well as counters c_v and T_v
  - Piggyback p_v, c_v, T_v to a message sent by v
  - Better understanding on how cumulative probability changes every time step
  - Achieve fairness for free (basically all nodes have the same transmission probability)!
- Once nodes are synchronized (plus some other smaller changes), one can show that the basic protocol is also robust against reactive jammers

Our results

Theorem. The AntiJam protocol achieves:

- fairness: the channel access probabilities among nodes do not differ by more than a factor of $(1+\gamma)$ after the first message was sent successfully.

- $e^{\theta(1/\varepsilon^2)}$-competitiveness w.h.p., under any $(T,1-\varepsilon)$-bounded reactive adversary if the protocol is executed for $\tilde{\Omega}(T/\varepsilon)$ steps w.h.p., for any constant $0<\varepsilon<1$ and any T

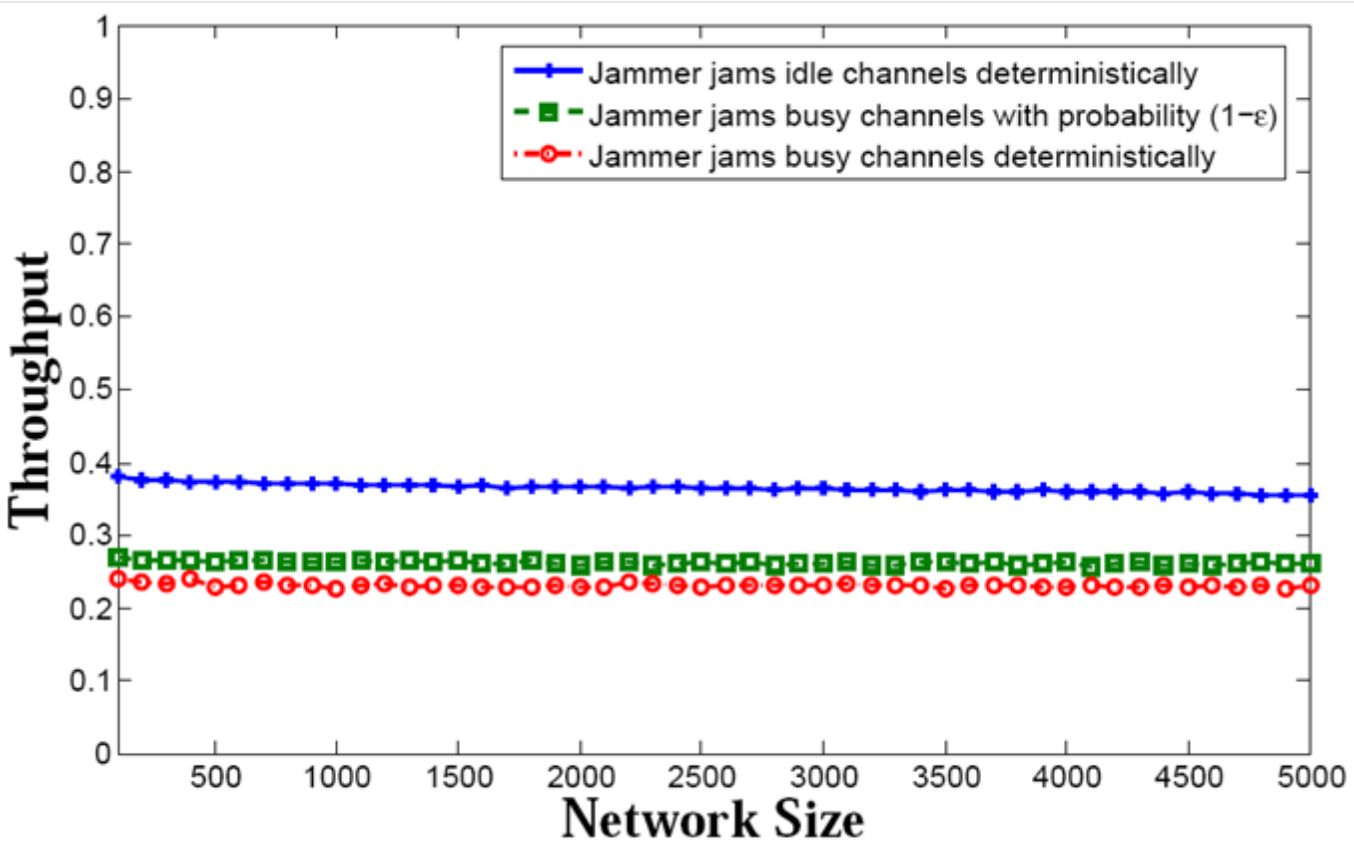
(constant in throughput now depends on ε)

Proof sketch: Fairness

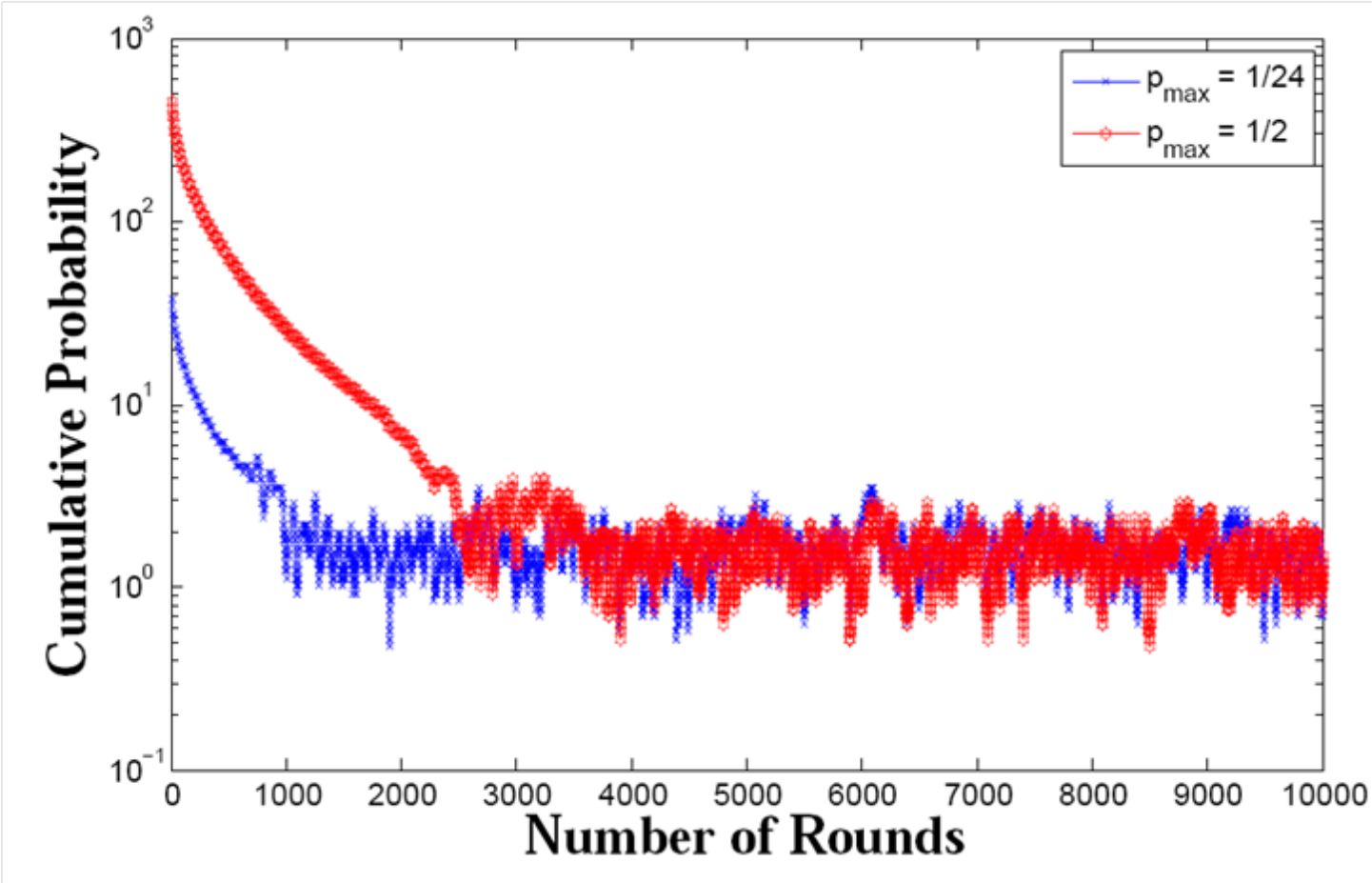
- Fact:
  - Right after u sends a message successfully along with the tuple (p_u, c_u, T_u), $(p_v, c_v, T_v) = (p_u/(1+\gamma), c_u, T_u)$ for all receiving nodes v, while the sending node values stay the same. In particular, for any time step t after a successful transmission by node u, $(c_v, T_v) = (c_w, T_w)$ for all nodes v and $w \in V$
  - This implies that after a successful transmission, the access probabilities of any two nodes in the network will never differ by more than a factor $(1+\gamma)$ in the future.

AntiJam: Throughput for ε = 0.5

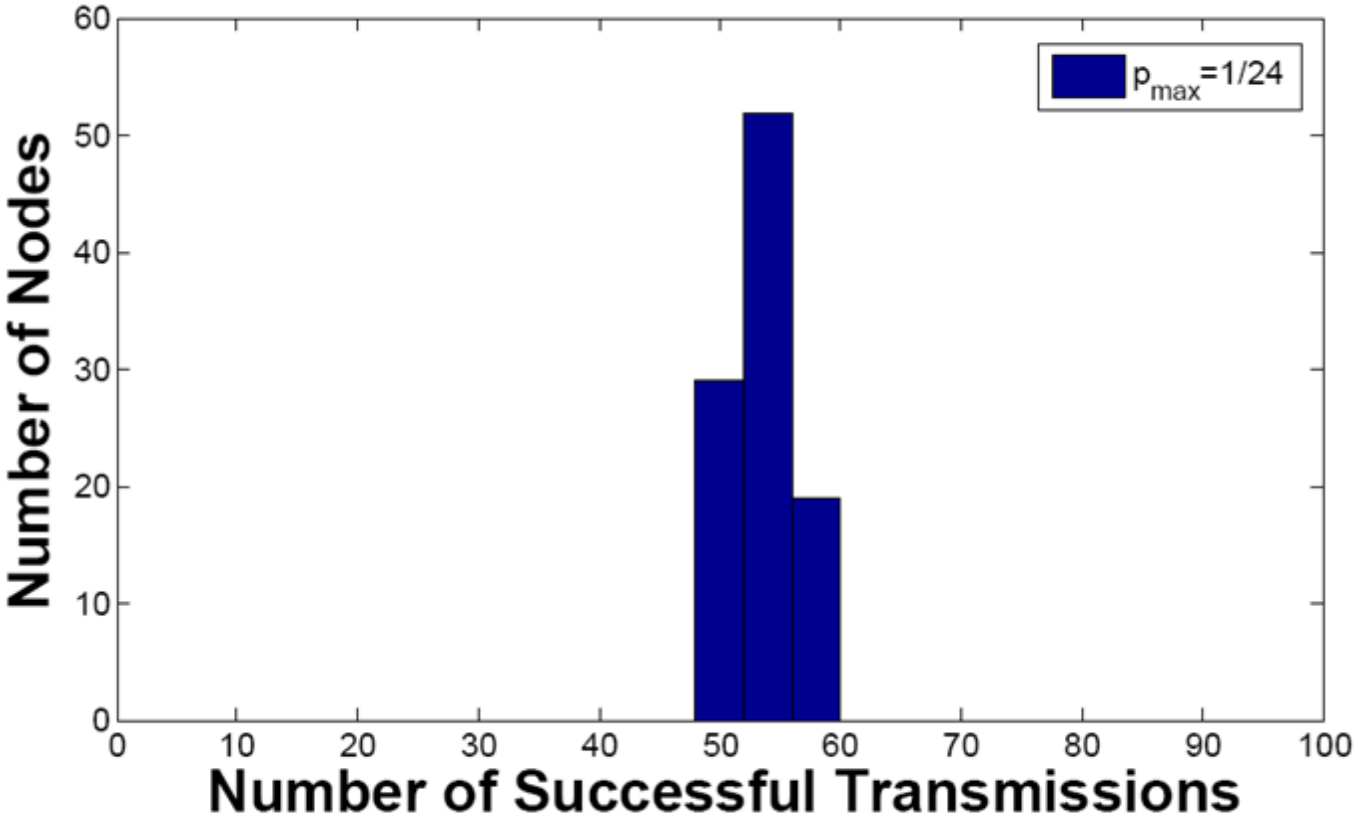
ε = 0.5



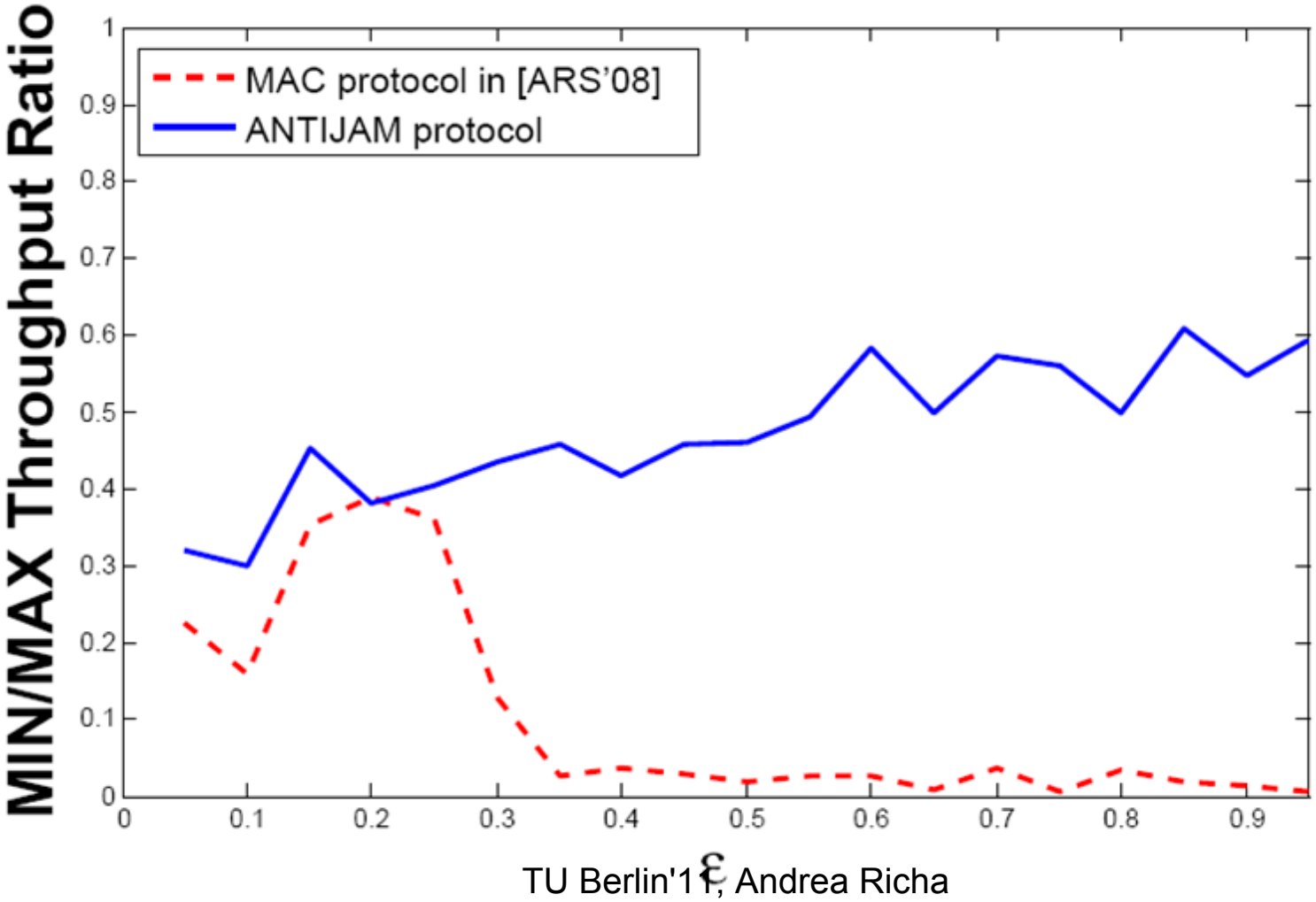
AntiJam: Convergence Time



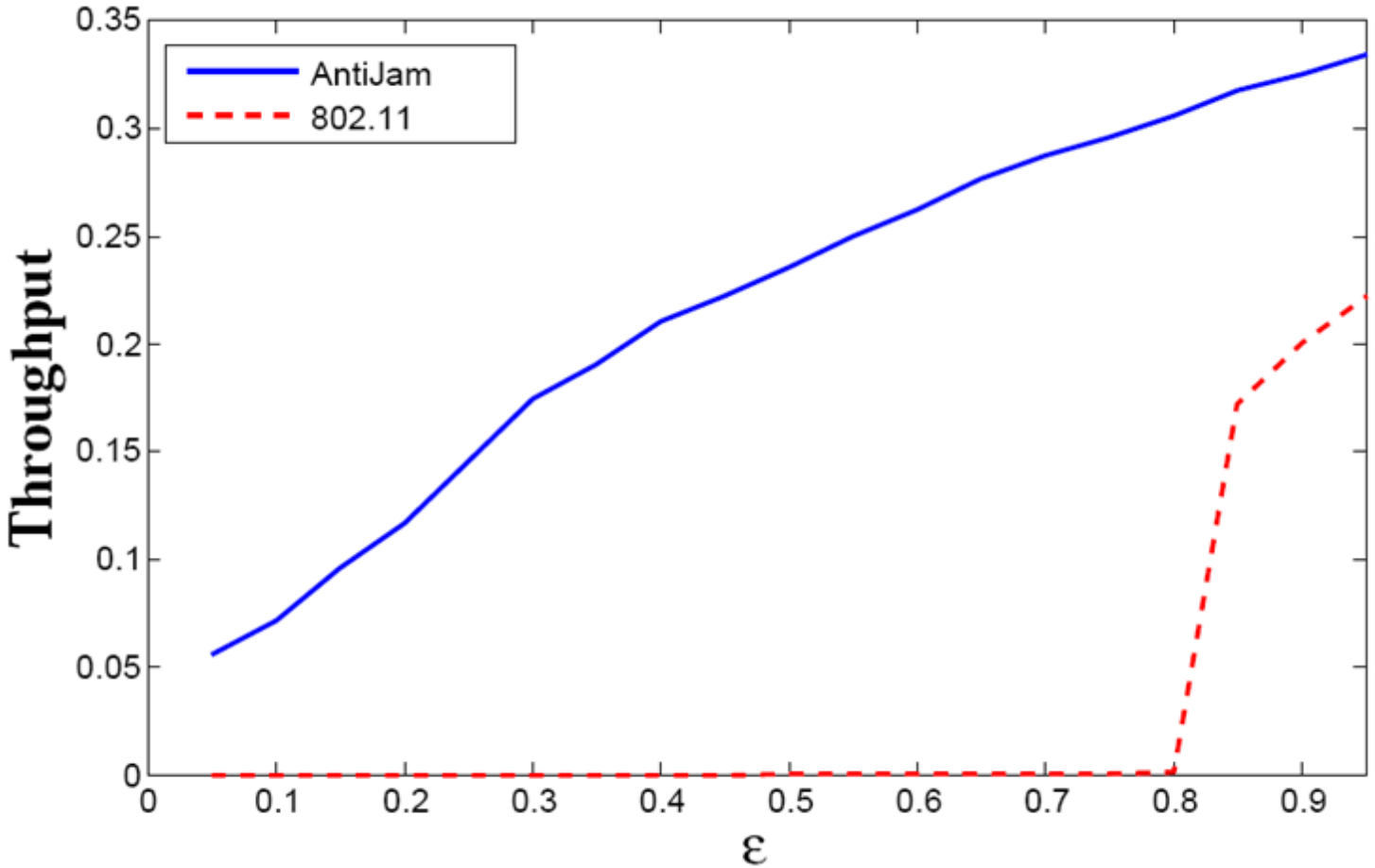
AntiJam: Fairness



AntiJam vs. Non-reactive protocol: Fairness



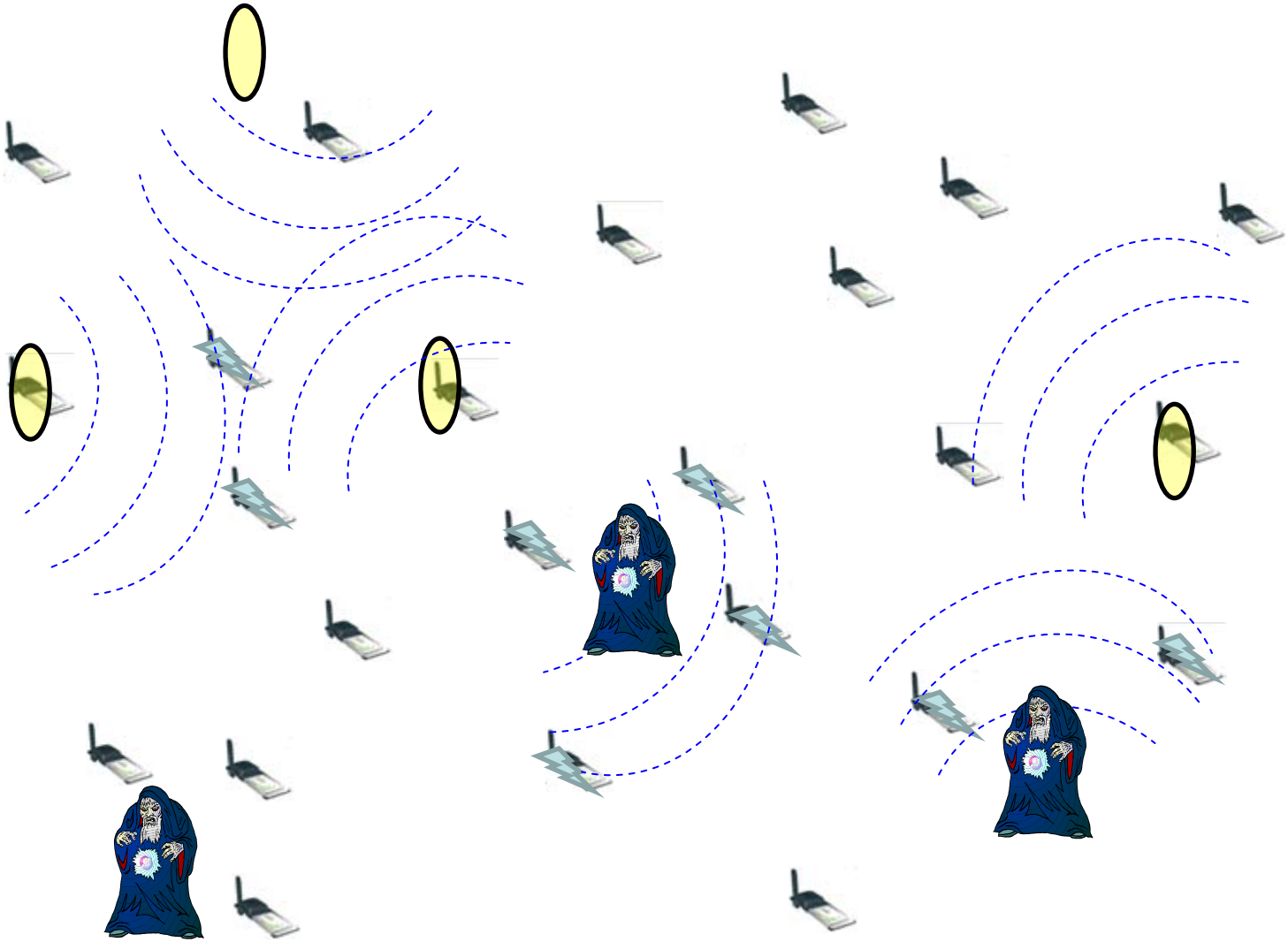
AntiJam vs 802.11



Overview

- MAC protocol : Single-hop
  - Our contributions
  - Simple (yet powerful) idea
  - MAC protocol
  - Fast recovery & Energy Efficiency
- Reactive jammers
  - Fairness
- Multi-hop networks
- Application: Leader Election
- Future work

Multihop wireless networks



Multihop wireless network

- [R., Scheideler, Schmid, Zhang, DISC'10]

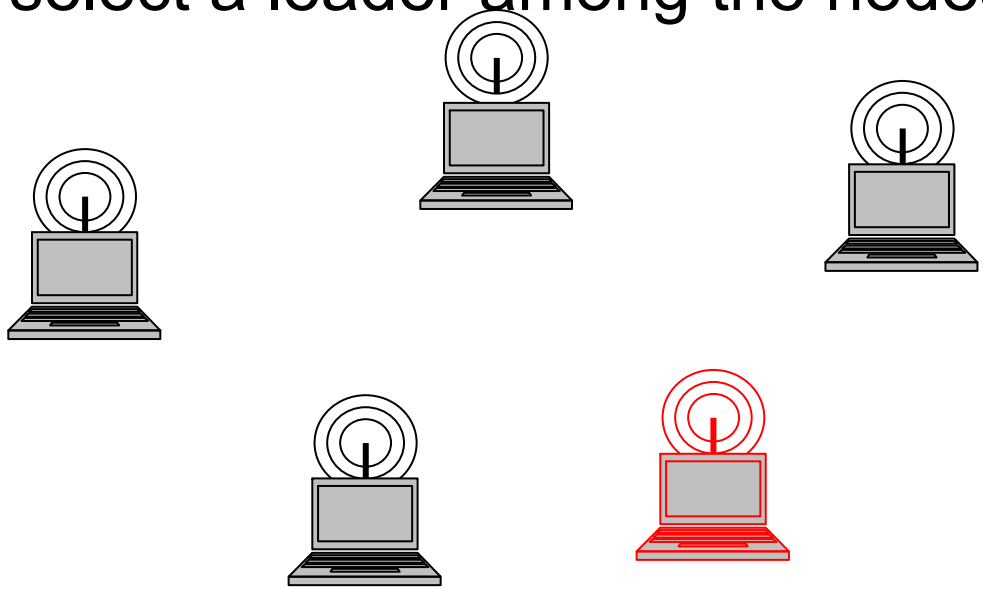
- Unit-disk graph model for both communication and interference
- Also achieve constant-throughput algorithm

$$\text{Throughput} = \frac{\sum_{v} \#\text{successful msgs received by } v}{\sum_{v} \#\text{non-jammed time steps for } v}$$

Application: Leader Election in Single-hop Networks

- [R.,Scheideler,Schmid,Zhang, MobiHoc'11]

- Our goal: select a leader among the nodes



- Challenges: we may start in any state, there is wireless jammer

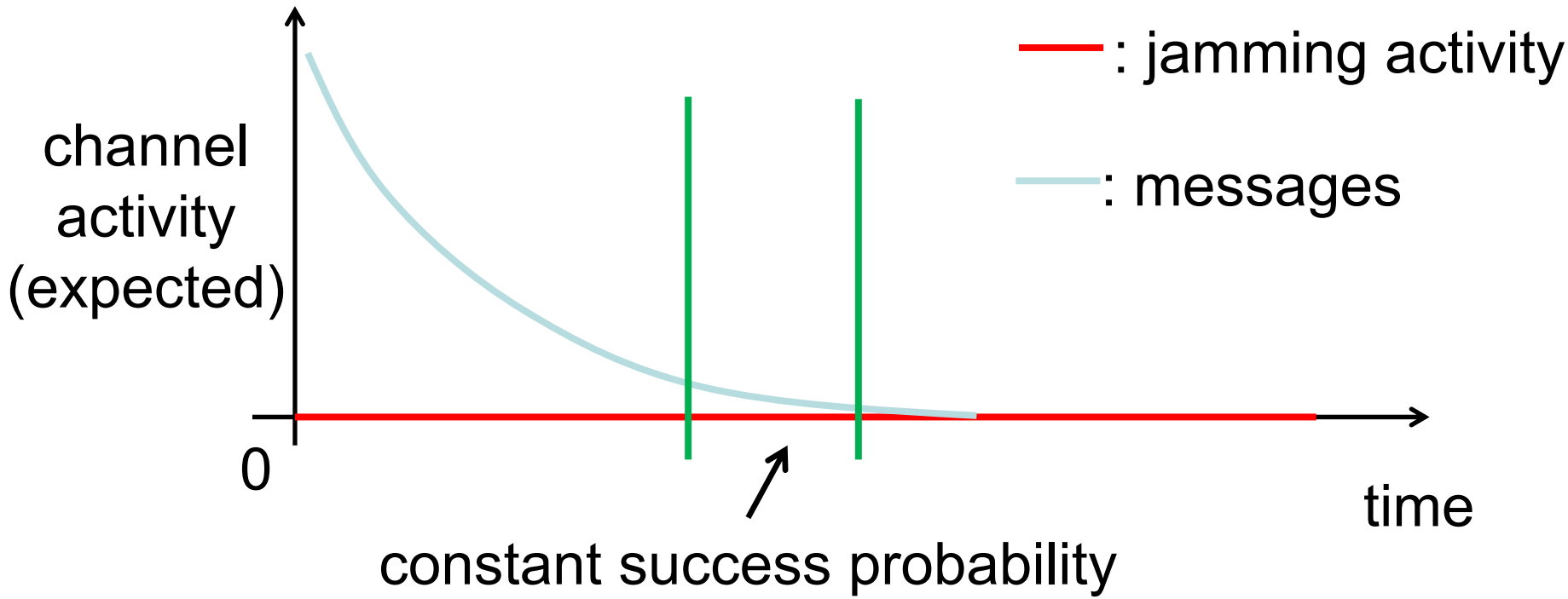
Self-stabilizing Leader Election

- Goal: design a self-stabilizing protocol that elects a single node as the leader, irrespective of the jamming activity
- No leader election protocol proposed so far can achieve that within our model
- Challenges:
  - a leader node should let the others (*followers* or other leaders) know that he is still around
  - the followers should be able to notice when there is no leader in the network

Leader Election

Why is leader election difficult under jamming?

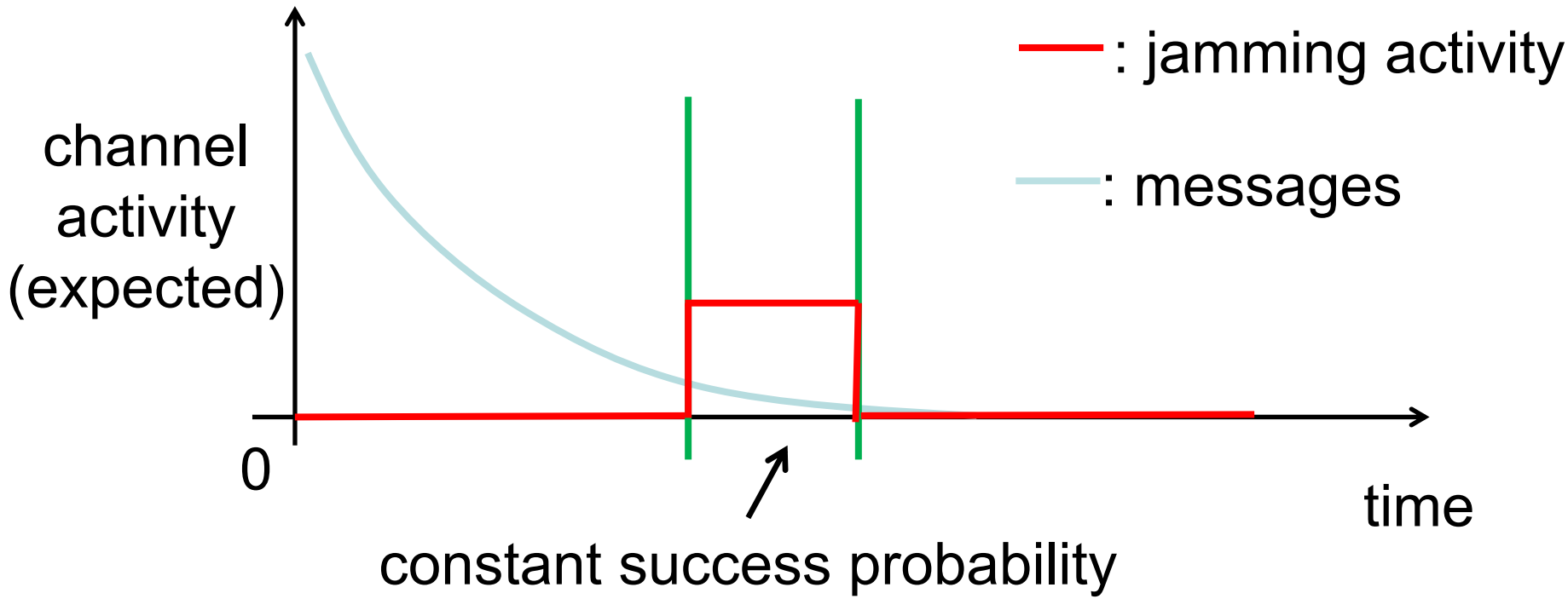
Example: exponential/polynomial backoff



Leader Election

Why is leader election difficult under jamming?

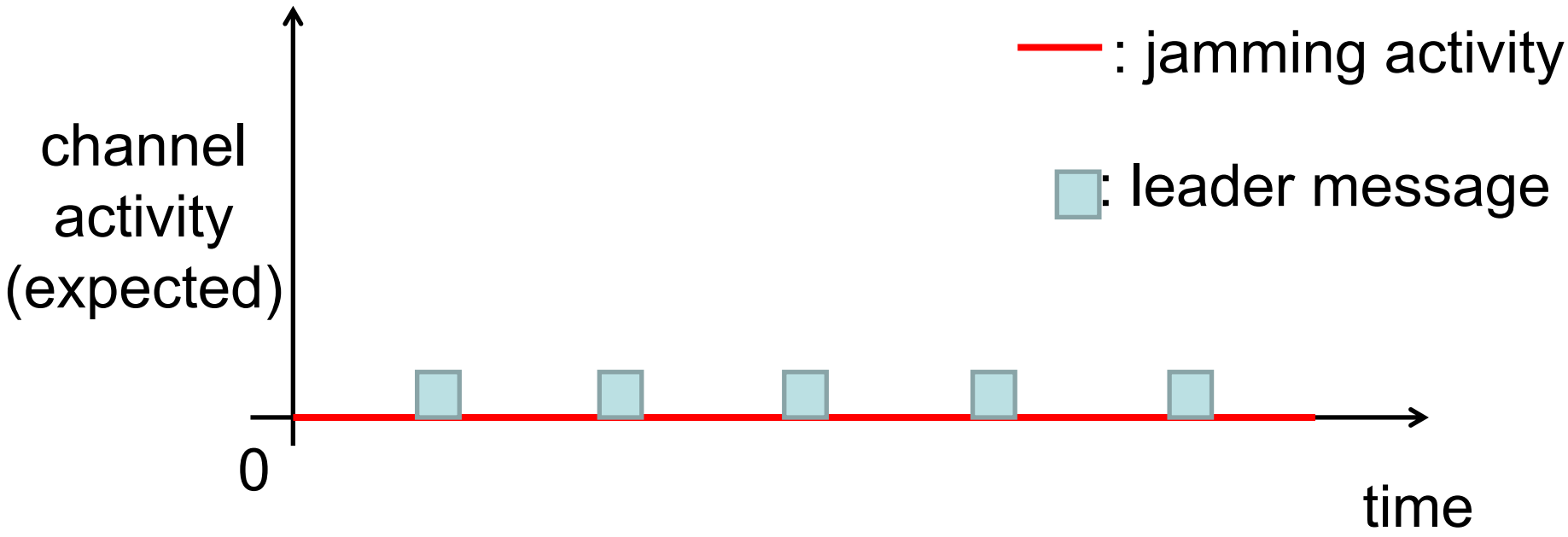
Example: exponential/polynomial backoff



Leader Election

Why is leader election difficult under jamming?

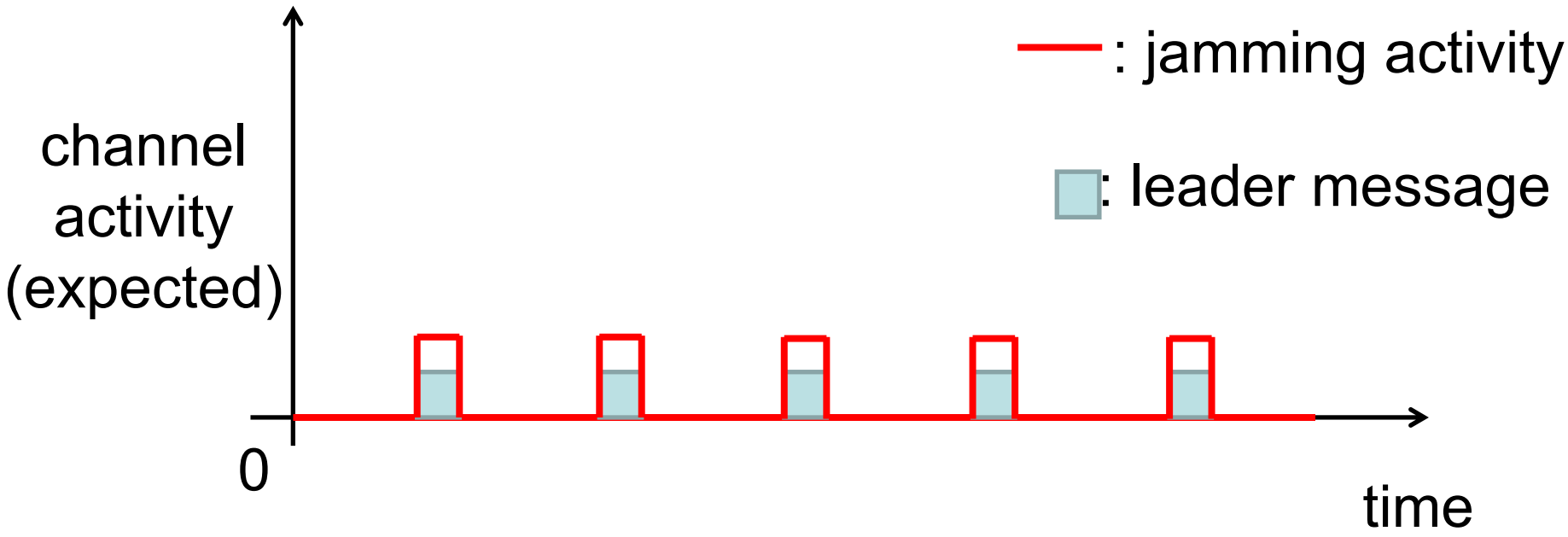
Example: reserved leader slot to notify nodes about leader



Leader Election

Why is leader election difficult under jamming?

Example: reserved leader slot to notify nodes about leader



Future work

- Can the multihop MAC protocol be extended to handle physical interference, e.g., using SINR model?

$$\frac{P_v(u)}{N + \sum_{w \in S} P_v(w)} \geq \beta$$

  (use ideas from our MobiHoc'08 paper?)
- How about reactive jammers in multihop environments? Under SINR?
- Can the protocols be modified so that no rough bound on *n* and *T* are required?
  - stochastic/oblivious jammers: Simpler to handle? E.g., a constant gamma seems to work fine here.
- Other applications of the MAC protocol (e.g., broadcast)?

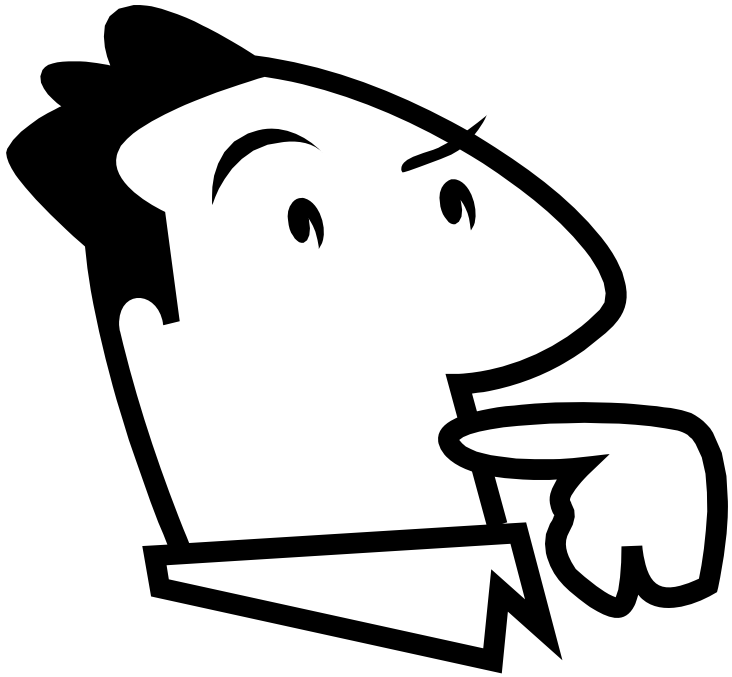
Collaborators

- Baruch Awerbuch (John Hopkins), Stefan Schmid (Tech. U. of Berlin), Christian Scheideler (U. of Paderborn), Jin Zhang (ASU)

- All my papers available from my webpage (for recent submissions, please send me email) at

  www.public.asu.edu/~aricha

  aricha@asu.edu



Thank you!
Questions?