Models and Algorithms for Robust Medium Access

Stefan Schmid

T-Labs & TU Berlin

A joint research project with

Baruch Awerbuch

Adrian Ogierman

Andrea Richa

Christian Scheideler

Jin Zhang

Distributed Wireless and Sensor Networks

Ad-hoc wireless communication:

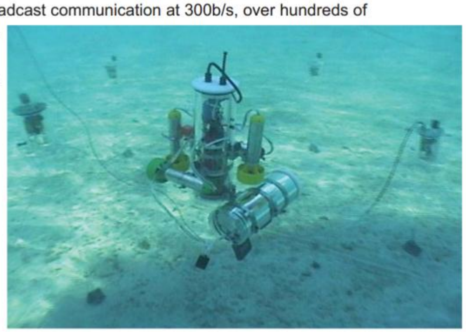
- no centralized control
- nodes must coordinate medium access in a distributed fashion!

Today: e.g., farming

Multi-hop sensor networks

Future: underwater robots?

- Static sensor nodes plus mobile robots
- Dually networked
  - optical point-to-point transmission at 300kb/s
  - acoustical broadcast communication at 300b/s, over hundreds of meters range.
- Project AMOUR [MIT, CSIRO]
- Experiments
  - ocean
  - rivers
  - lakes



Future: self-managed cow herds?

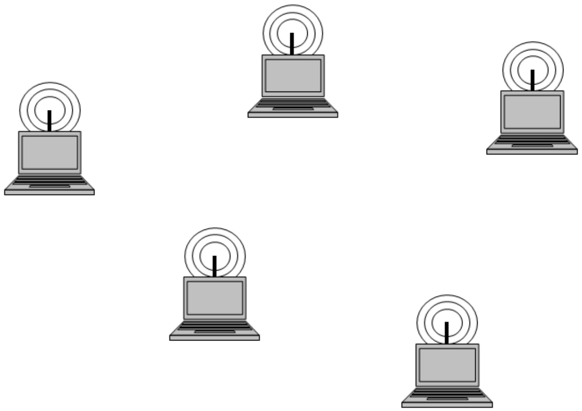
Virtual Fence (CSIRO Australia)

- Download the fence to the cows. Today stay here, tomorrow go somewhere else.
- When a cow strays towards the co-ordinates, software running on the collar triggers a stimulus chosen to scare the cow away, a sound followed by an electric shock; this is the "virtual" fence. The software also "herds" the cows when the position of the virtual fence is moved.
- If you just want to make sure that cows stay together, GPS is not really needed…

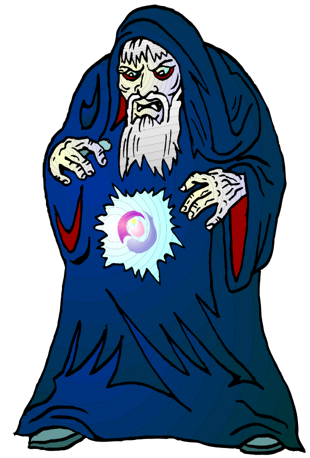
Cows learn and need not to be shocked later… Moo!

The long journey to resilient MAC protocols!

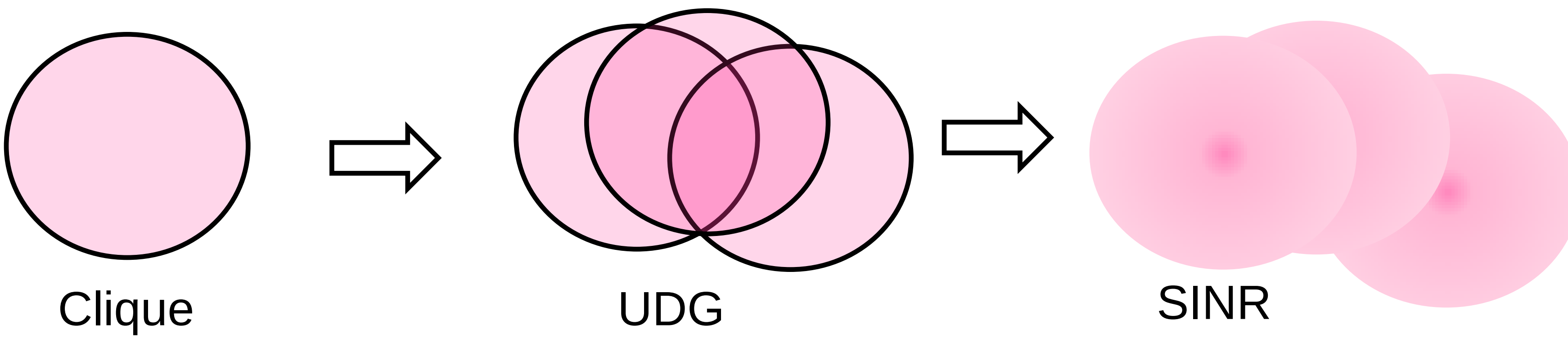
Goal of our robust MAC project: competitive throughput despite jammer!



vs

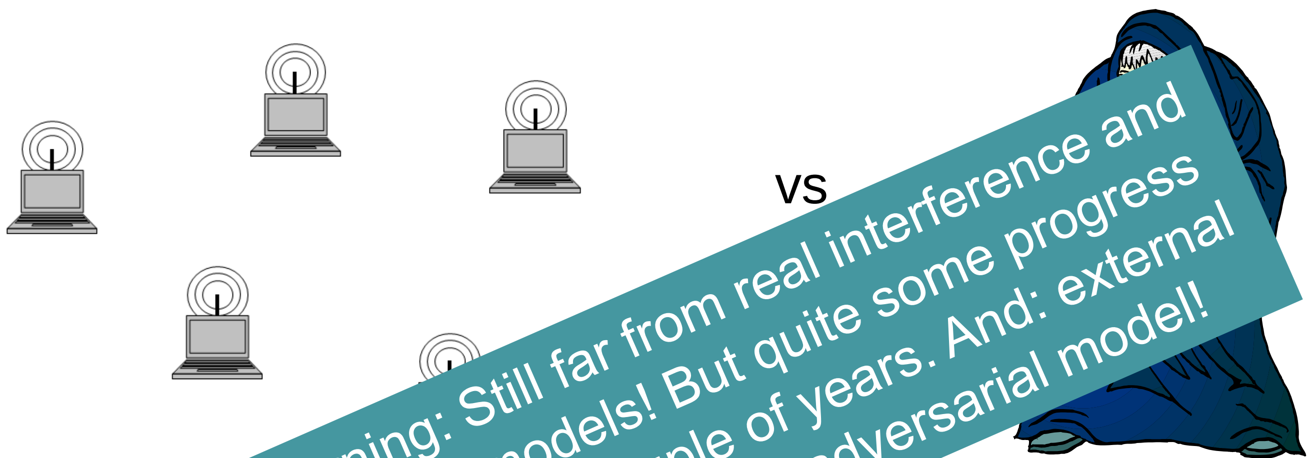


The journey towards more and more realistic node interference models:



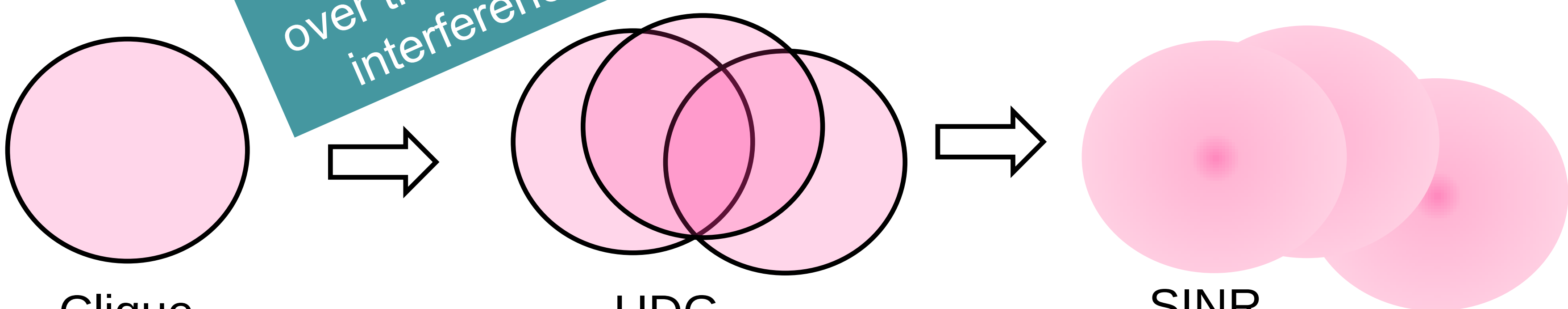
The long journey to resilient MAC protocols!

Goal of our robust MAC project: competitive throughput despite jammer!



Warning: Still far from real interference and antenna models! But quite some progress over the last couple of years. And: external interference strong adversarial model!

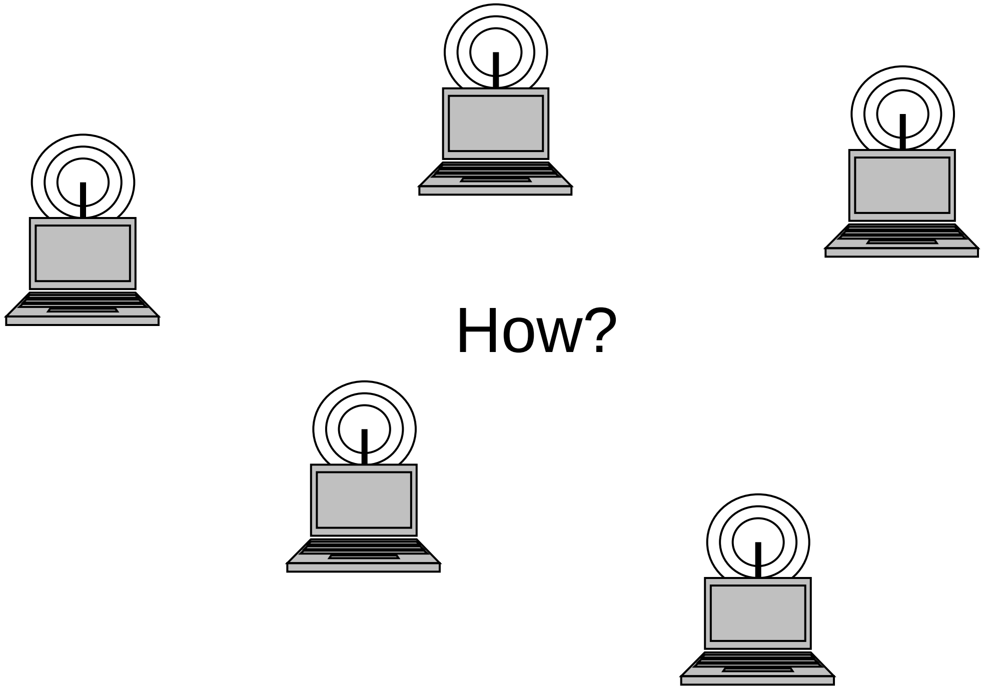
The journey t... ore realistic node interference models:



The MAC Problem

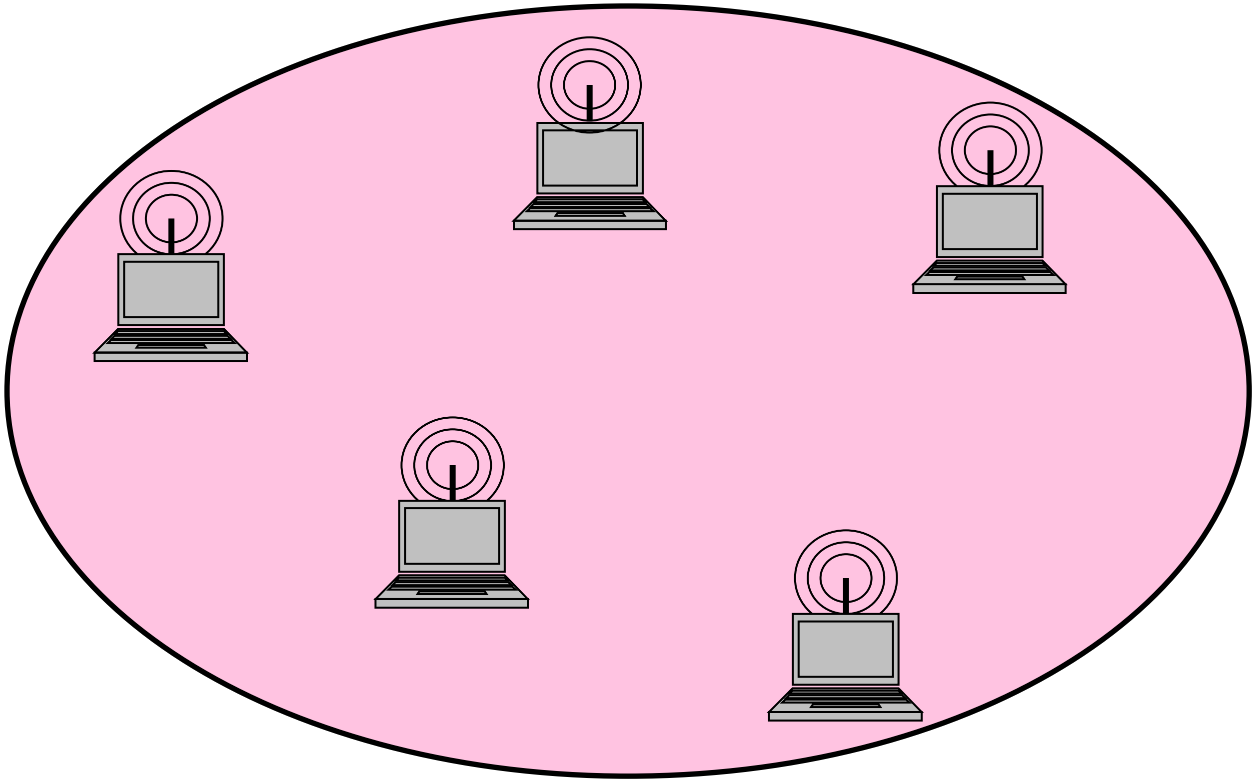
Given: a set of wireless nodes distributed in space

Goal: efficient medium access over a single channel?



Single-Hop Network

All nodes are within transmission / interference range of each other.

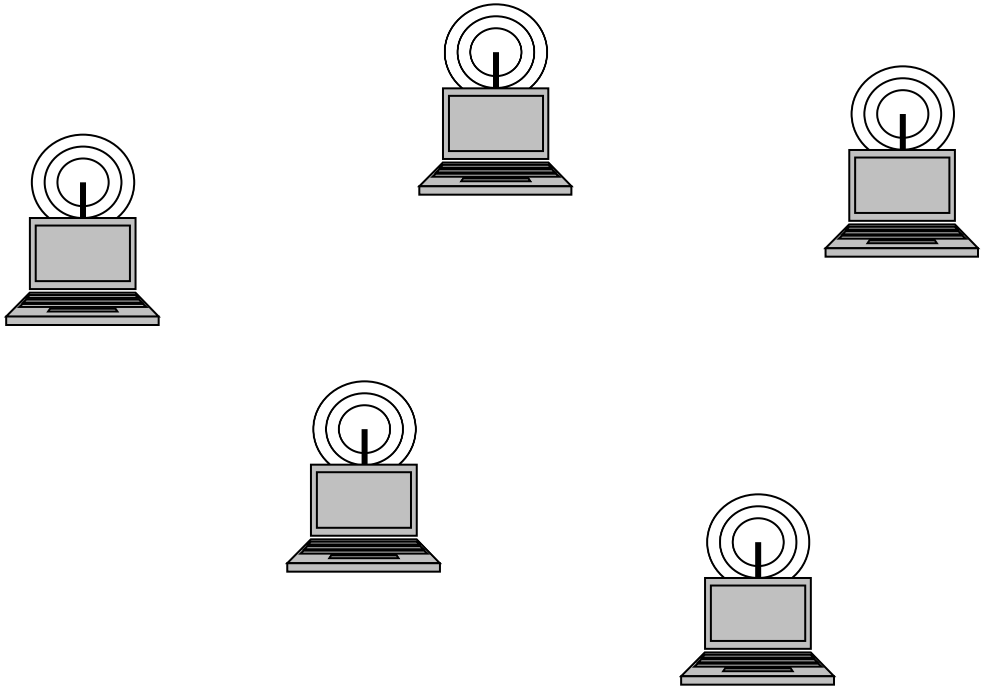


The MAC Problem

Solution: just let one node transmit after the other (round-robin)!

☺: efficient, fair, predictable, ...

☹: organize such a schedule in a distributed system? joins/leaves?



Well-known solutions: ALOHA, Wifi, ...

ALOHA: invented in Hawai!

(Simplified) ALOHA

Send with probability 1/n, where n = # nodes.

Distributed and good throughput (20-40%) but what if n changes over time?

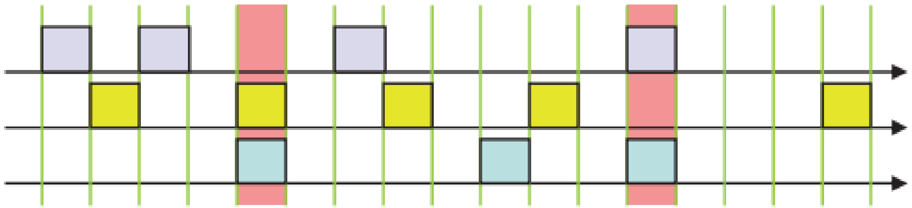
(Simplified) Wifi

Send with probability 1, if collision with probability 1/2, then 1/4, then 1/8, etc.: random backoff

Good solution! Resolves conflicts quickly!

Proof for Slotted ALOHA

- We assume that the stations are perfectly synchronous
- In each time slot each station transmits with probability *p*.



$$P_1 = \Pr[\text{Station 1 succeeds}] = p(1-p)^{n-1}$$

$$P = \Pr[\text{any Station succeeds}] = nP_1$$

$$\text{maximize } P: \frac{dP}{dp} = n(1-p)^{n-2}(1-pn) \overset{!}{=} 0 \quad \Rightarrow \quad pn = 1$$

$$\text{then, } P = (1-\frac{1}{n})^{n-1} \geq \frac{1}{e}$$

- In Slotted Aloha, a station can transmit successfully with probability at least $1/e$, or about 36% of the time.

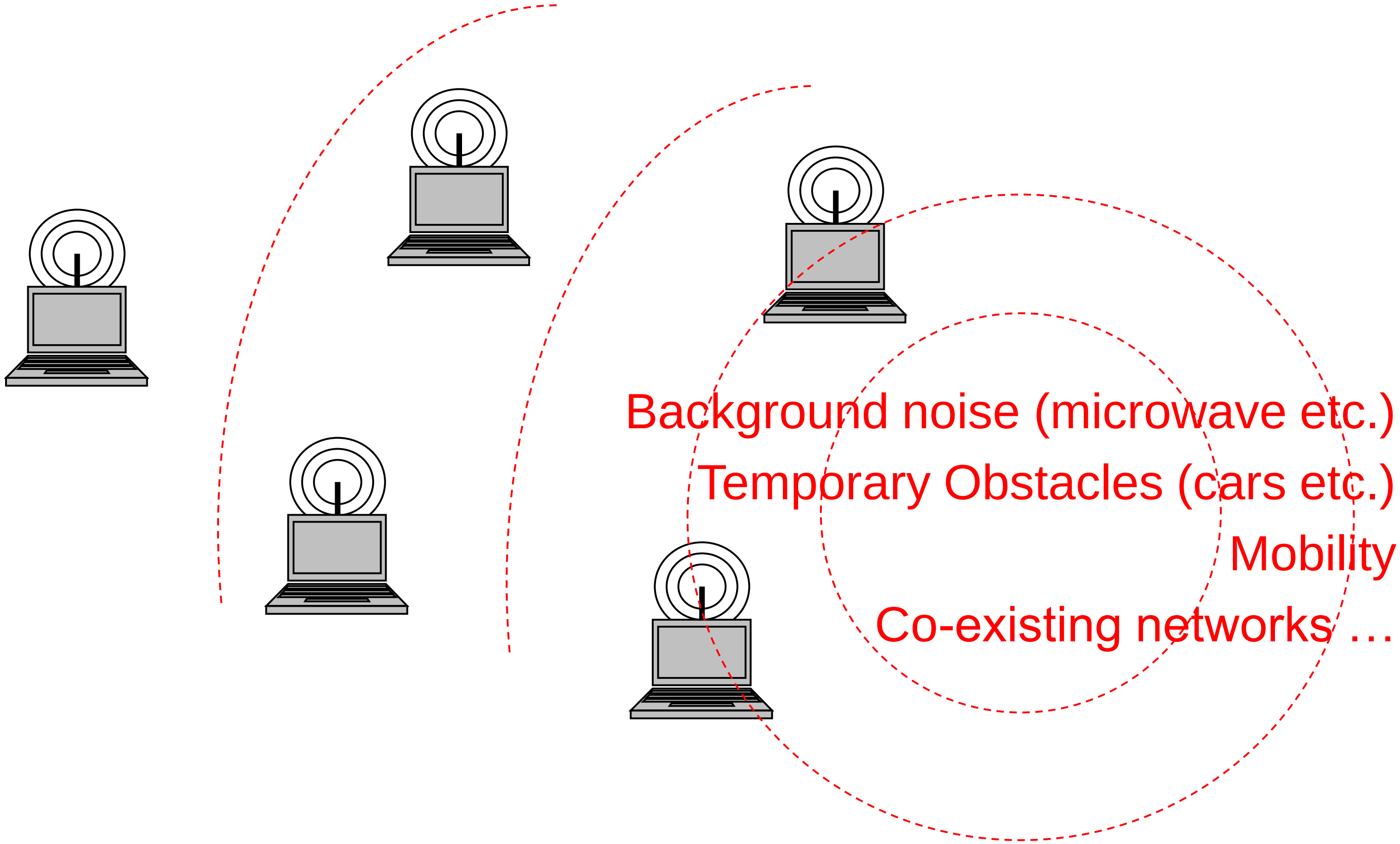
Theorem: ALOHA, Wifi, ... are competitive!

Competitive Throughput

On average, every O(1)-th time slot is a successful transmission.
This is asymptically optimal!

In other words: the percentage of successful transmissions over time does not depend on n, the number of nodes in the system.

But what if there is external interference....?



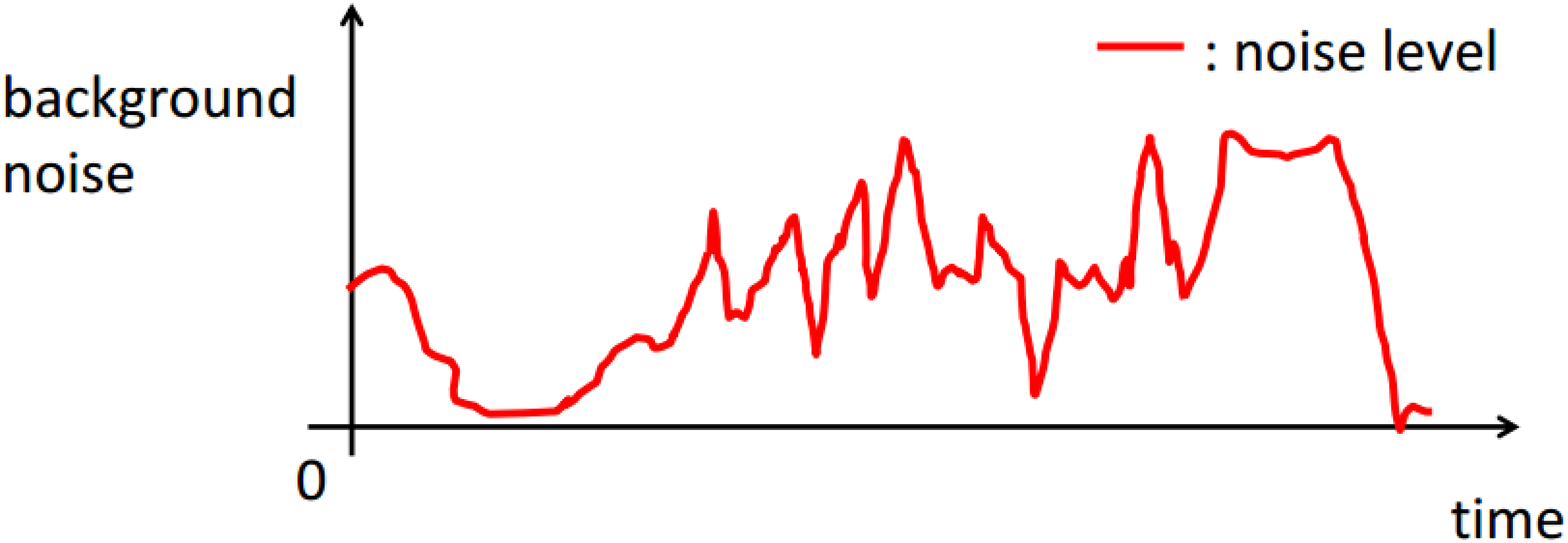
How to model interference?

Ideal world:

Usual approach adopted in theory.

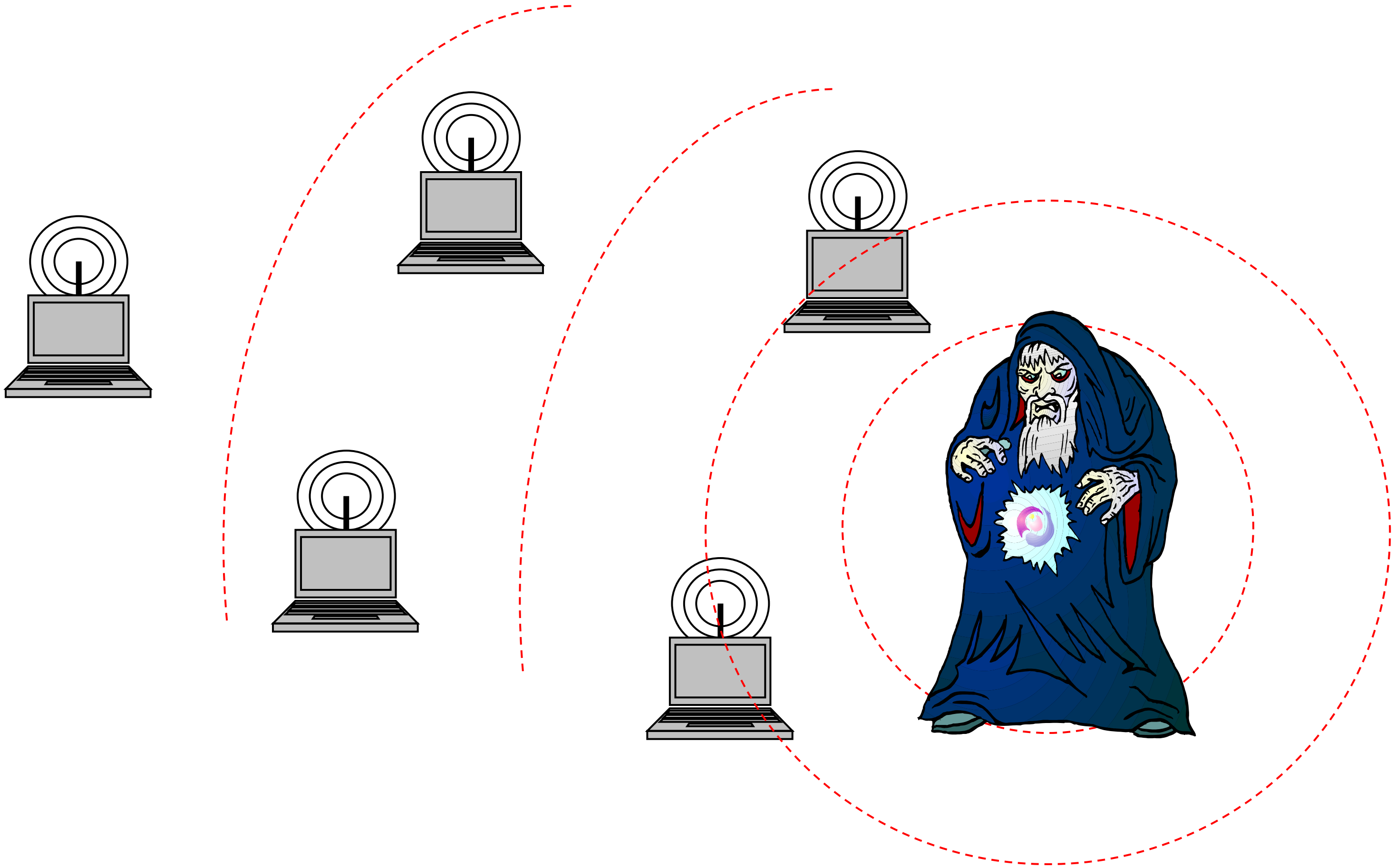
How to model interference?

Real world:



How to model that???

Our Approach: An Adversary / Jammer (Strong Model!)



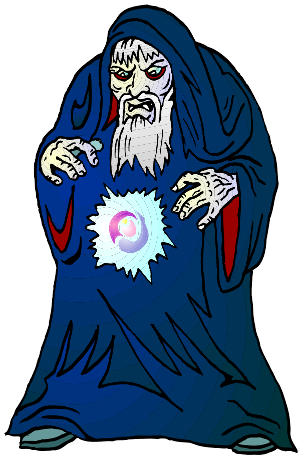
A Strong Adversary Model

Our adversary model captures all sorts of external interference!
And even malicious behavior. That's why we call it jammer/adversary!

The Adversary

In any time time period of duration **T**, the adversary can jam a time period of length **(1-ε)T**!

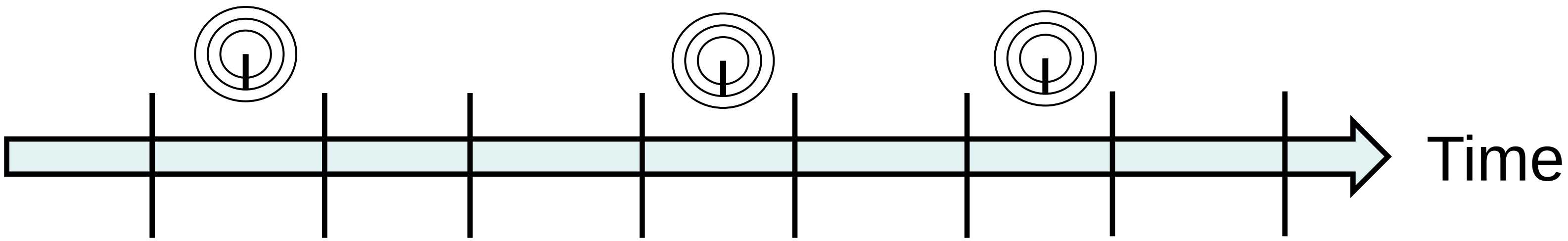
Only an ε-fraction of the time the medium is not blocked! Let us assume that $\varepsilon>0$ is an arbitrary constant.



More Formal Model

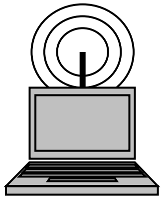
We consider a model with synchronous time!

Time is divded into time slots / rounds.



In each round, a node:

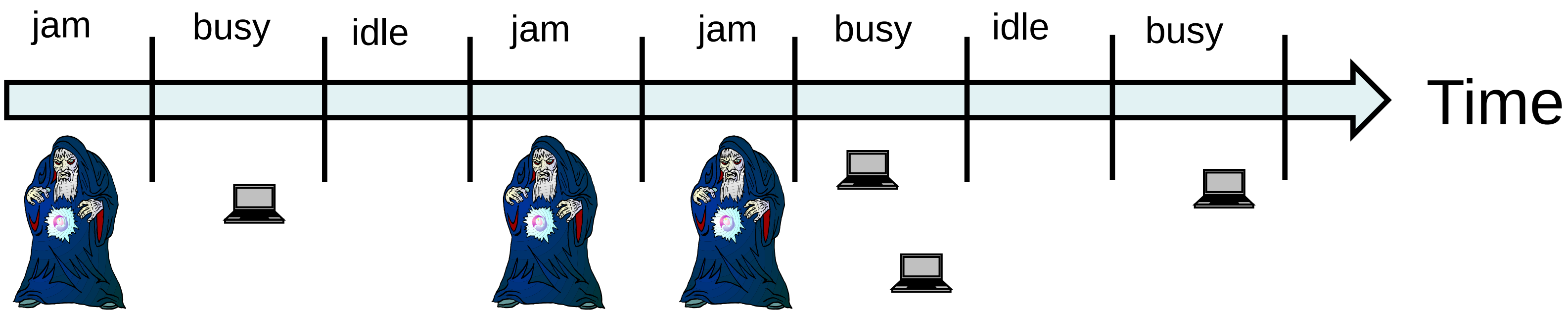
1. Can send a message
2. Or sense the channel
3. Not both (one antenna)



More Formal Model

We consider a model with synchronous time!

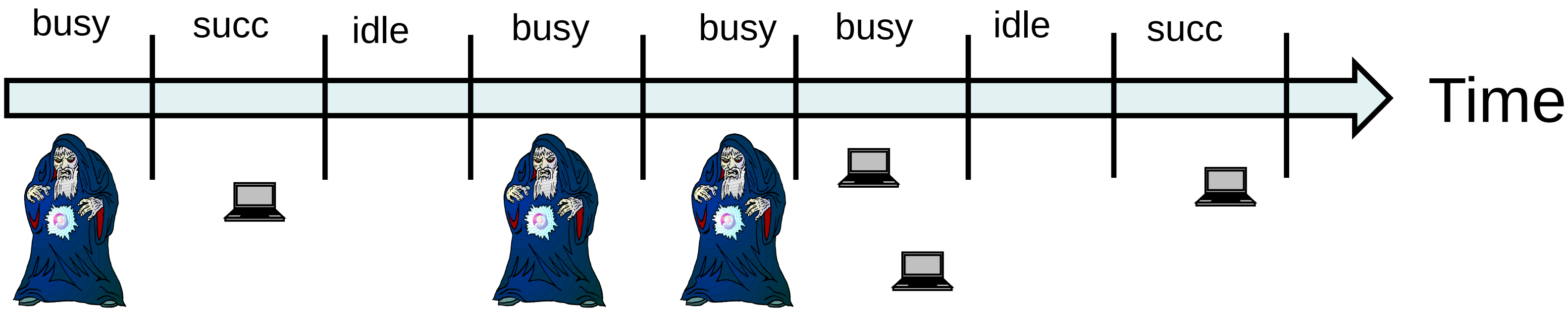
Time is divded into time slots / rounds.



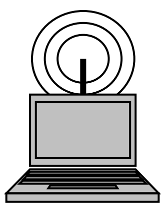
In a round, the channel can be:

1. idle
2. busy (at least one transmission)
3. jammed

More Formal Model



When a node does not send a message, it:

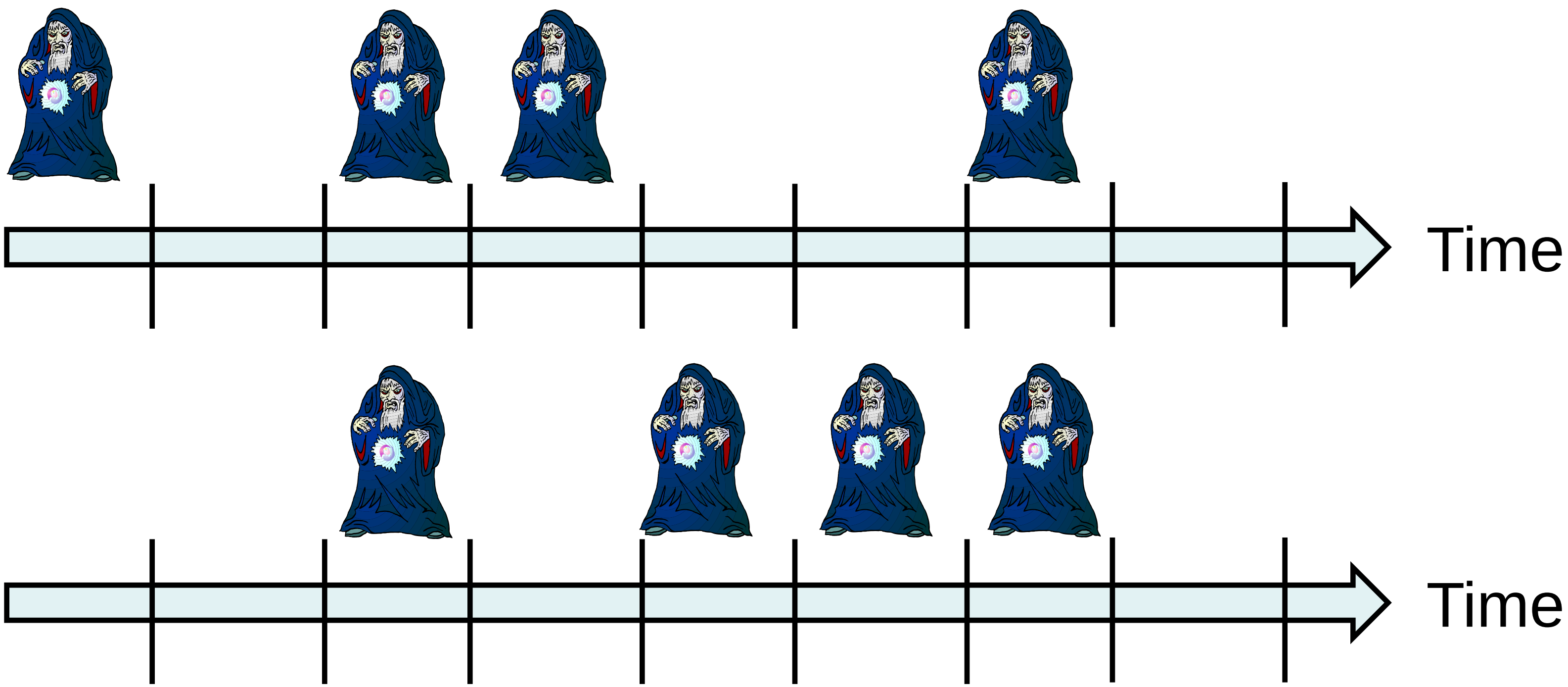


1. Can successfully receive a message
2. Sense a busy channel
3. Sense an idle channel

Note: 1. A node cannot distinguish between collisions or jamming!
2. A node that successfully sends does not know it was successful (only one antenna)!

The Adversary

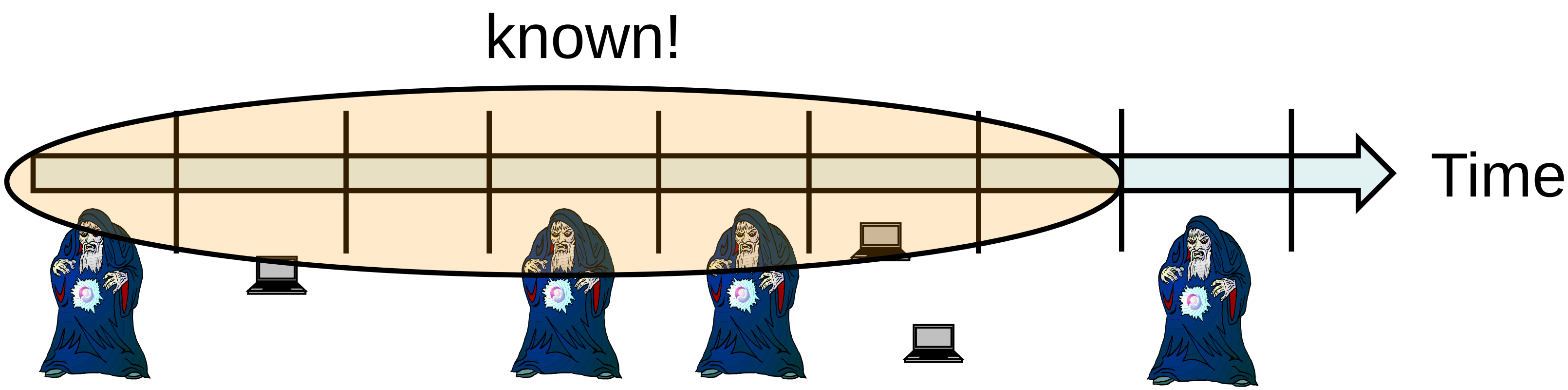
The adversary can block an arbitrary subset of rounds!



How can nodes exploit the remaining ε rounds?!

Don’t know n, don’t know ε, adversary can jam arbitrarily / deterministically!

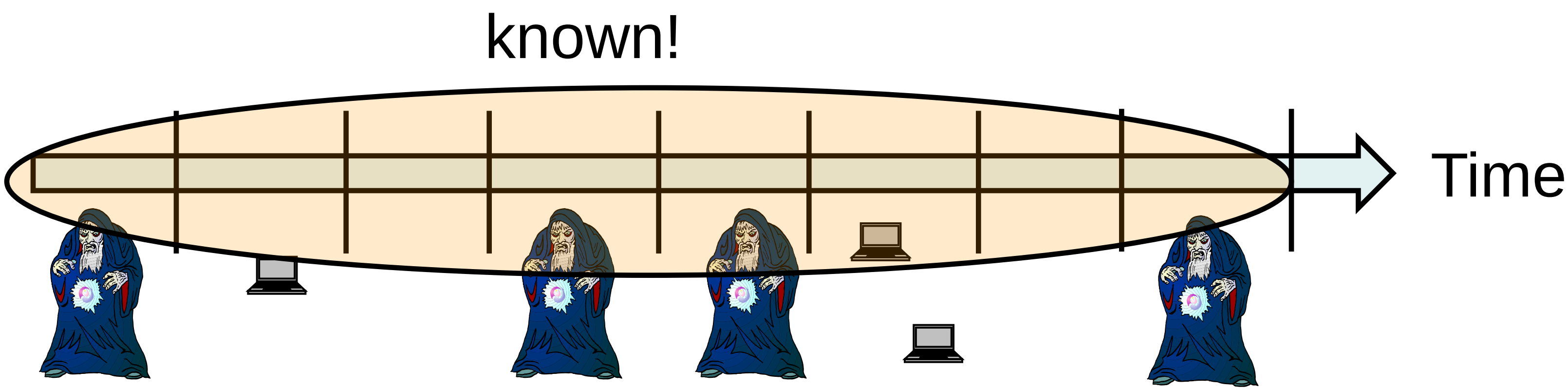
The Adversary Can Even Be Adaptive!



The Adaptive Adversary

In any time time period of duration **T** rounds, the adversary can jam **(1-Ɛ)T** rounds! These jamming decisions can depend on the entire history of the protocol execution!

The Adversary Can Even Be Reactive!



The Reactive Adversary

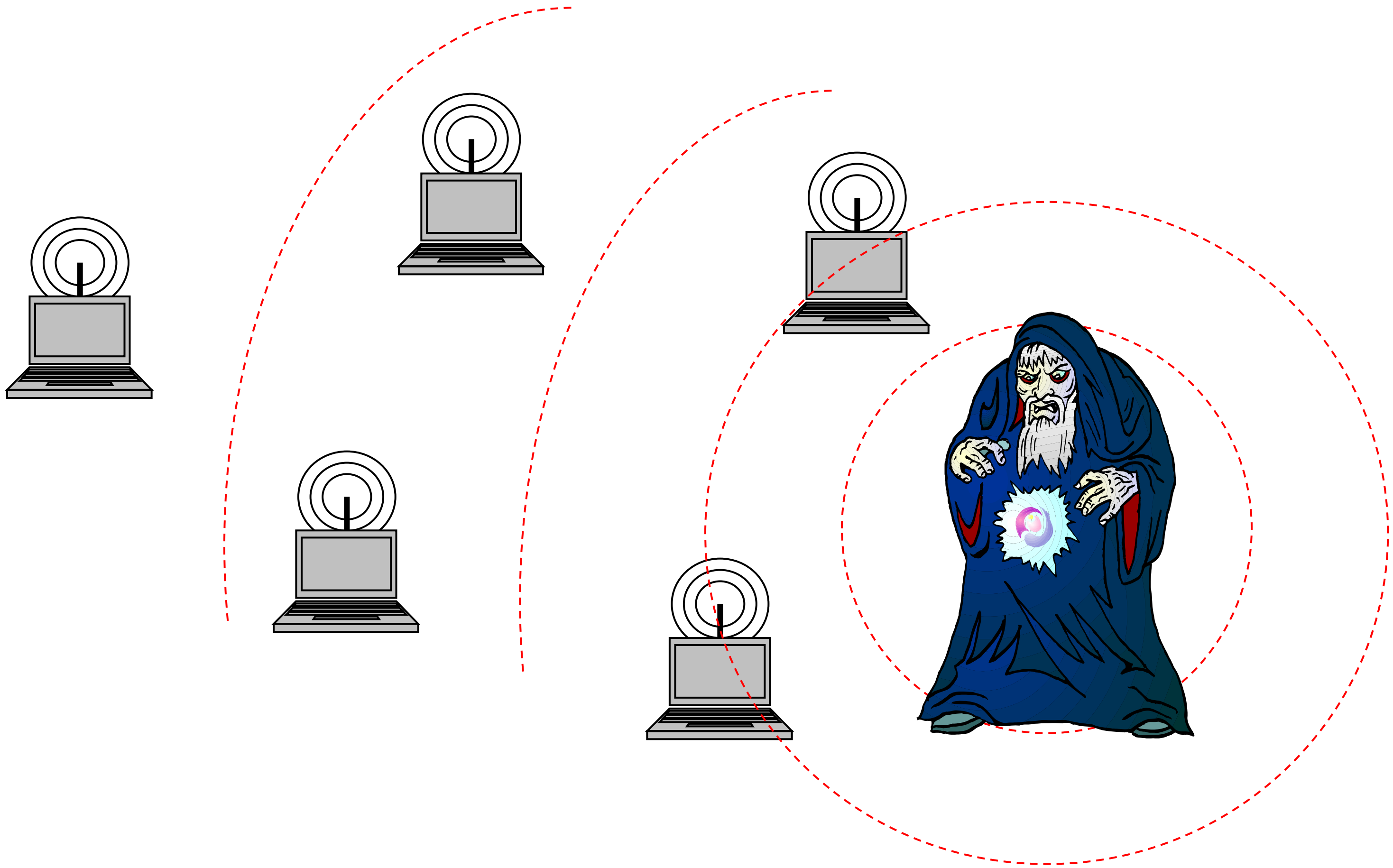
Sometimes, we can even let the adversary be reactive!

That is, he even knows what the node will do in this round!

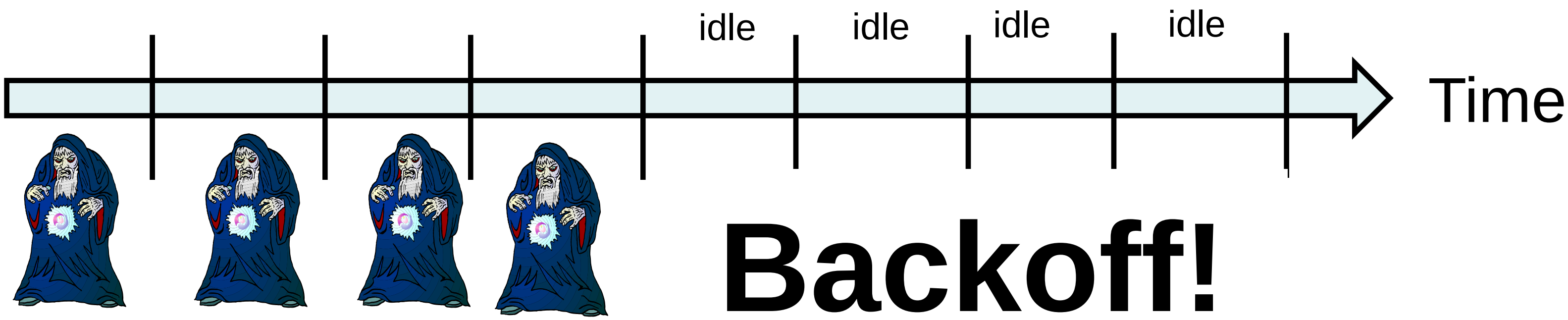
The Problem With Exponential Backoff?

(Simplified) Wifi

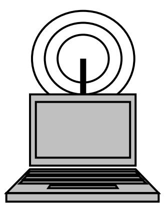
Send with probability 1, if collision with probability 1/2, then 1/4, then 1/8, etc.: random backoff



Bad Example for Exponential Backoff



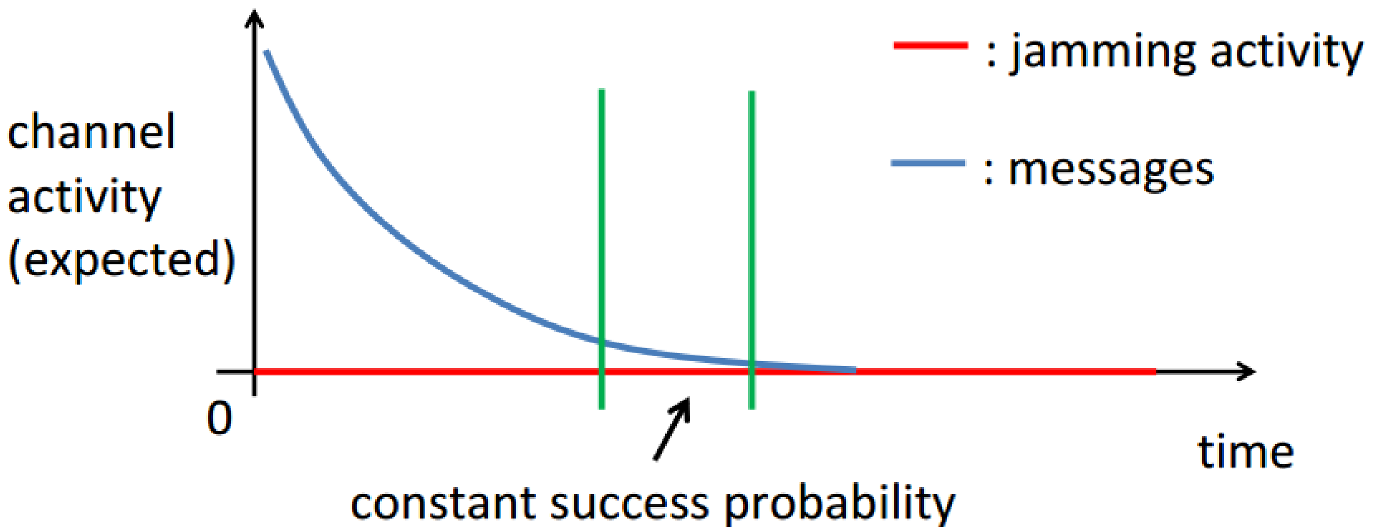
Adversary may jam a lot in the beginning:

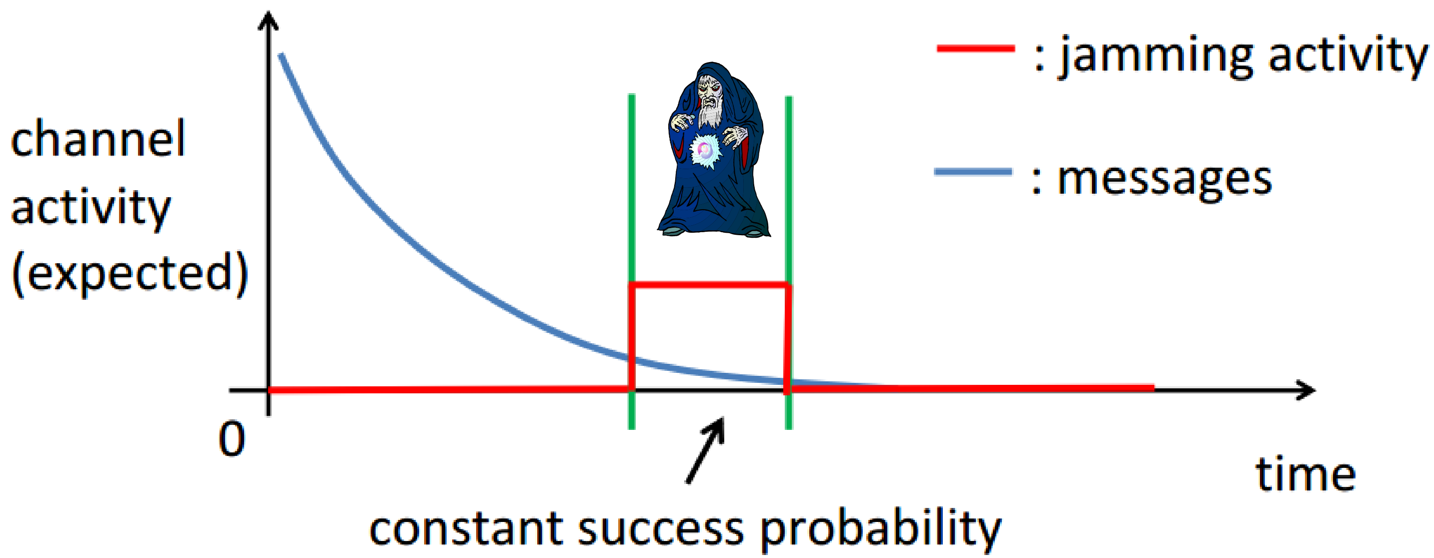


1. Nodes backoff a lot
2. When the adversary stops, everything is idle for a long time!

That’s bad! ☺

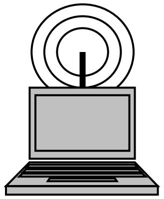
Example for Exponential/Polynomial Backoff



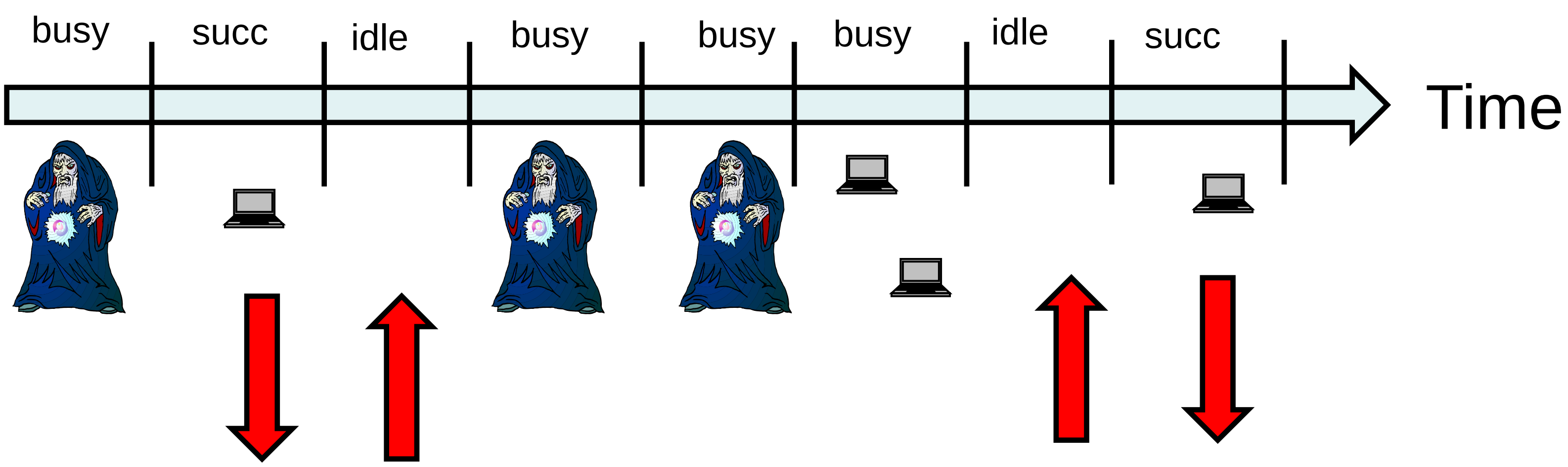


Basic Idea

How to prevent? Idea: do not increase backoff during busy times!



1. Idle round: increase sending probability
2. Successful message: decrease sending probability
3. Busy round: do nothing ☺



Basic Idea

Instead of using a backoff counter, use access probabilities:
each node v has a probability p_v for accessing the channel.

$$\text{If (idle): } p_v := (1+\gamma)\, p_v$$
$$\text{If (success): } p_v := 1/(1+\gamma)\, p_v$$

Here γ is a parameter.

Everything solved?

Motivation

Basic observation: let q_0 be the probability of an idle round, q_1 that exactly one node transmits, let p be the cumulative probability of all nodes, and $\hat{p}$ a cap on p_v.

Claim

$$q_0 * p \leq q_1 \leq p\, q_0/(1-\hat{p})$$

PROOF. It holds that $q_0 = \prod_v (1 - p_v)$ and $q_1 = \sum_v p_v \prod_{w \neq v} (1 - p_w)$. Hence,

$$q_1 \leq \sum_v p_v \frac{1}{1-\hat{p}} \prod_w (1 - p_w) = \frac{q_0 \cdot p}{1-\hat{p}} \quad \text{and}$$

$$q_1 \geq \sum_v p_v \prod_w (1 - p_w) = q_0 \cdot p$$

which implies the claim. □

Claim

$$q_0 * p \leq q_1 \leq p\, q_0/(1-\hat{p})$$

Why is this interesting?

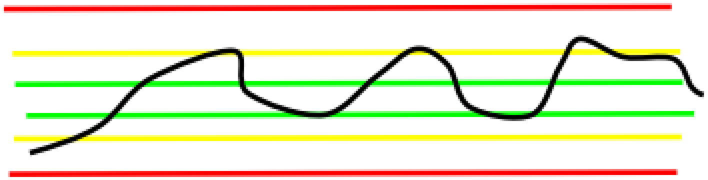
If $q_0 = q_1$, the cumulative probability p must be around a constant!

1. **If p is a constant, we expect a constant throughput in the non-jammed rounds!**
2. **To achieve this, nodes can just seek to balance idle and successful time steps!**

Analysis: Bounds on Cumulative Probability

- Some „ideas" only
- Protocol is interplay of many dependent randomized local algorithms
- Cumulative probability thresholds:

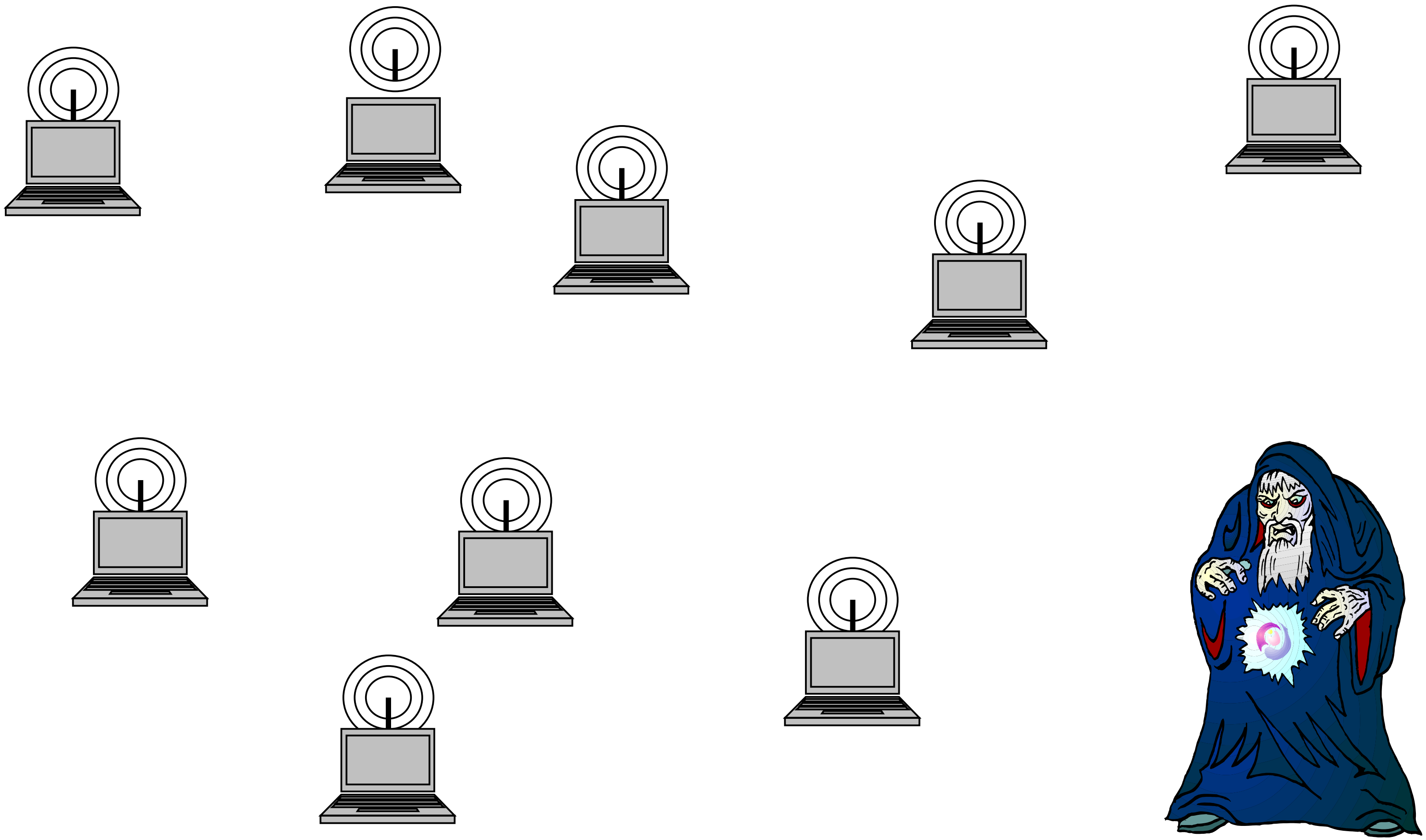
  ρ_{green}, ρ_{yellow}, ρ_{red}



  Show that beyond „good accumulated probabilities", there is a high drift towards „better values"

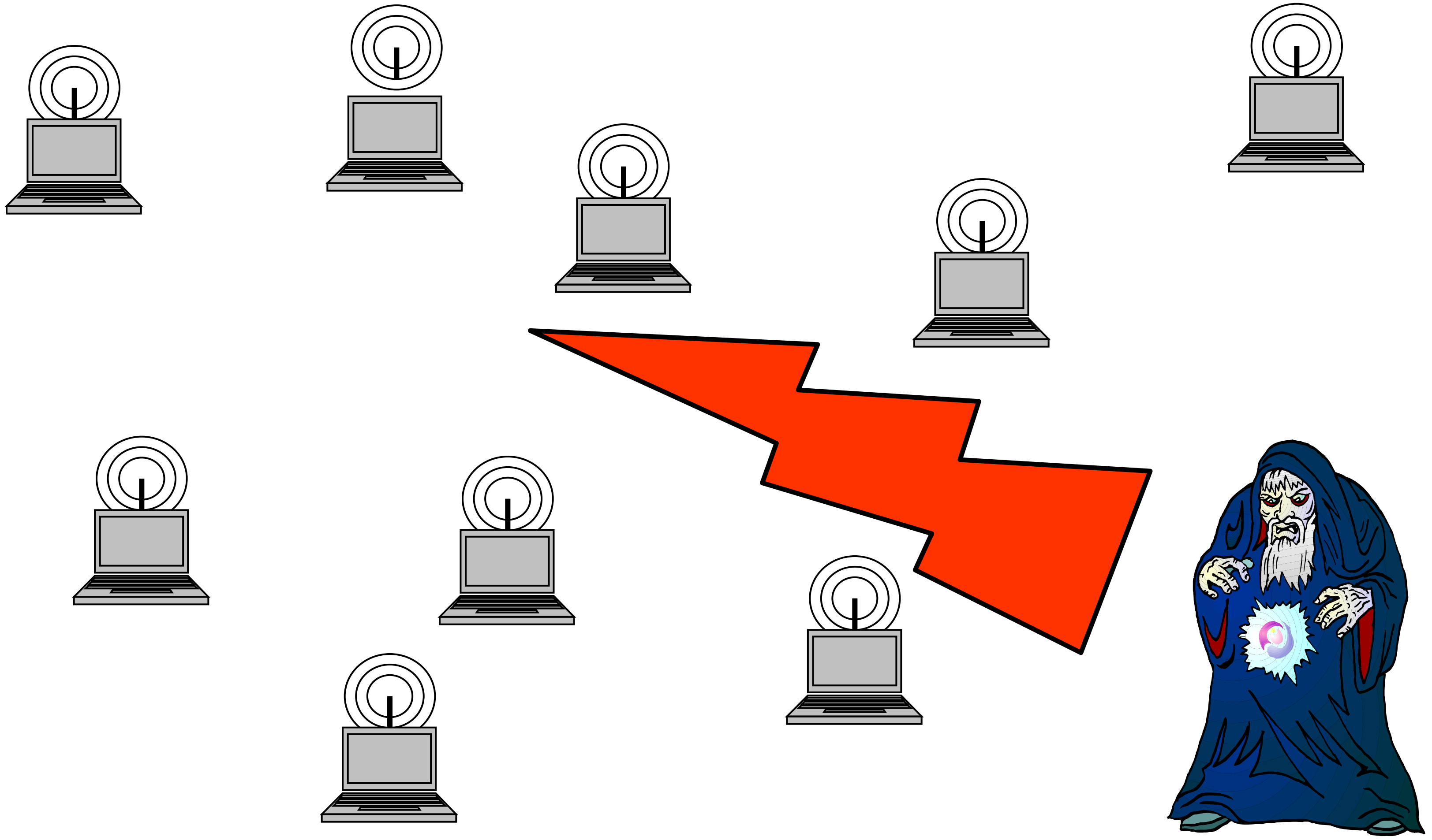
- Techniques: Martingale theory, stochastic dominance, etc.

Basic Idea

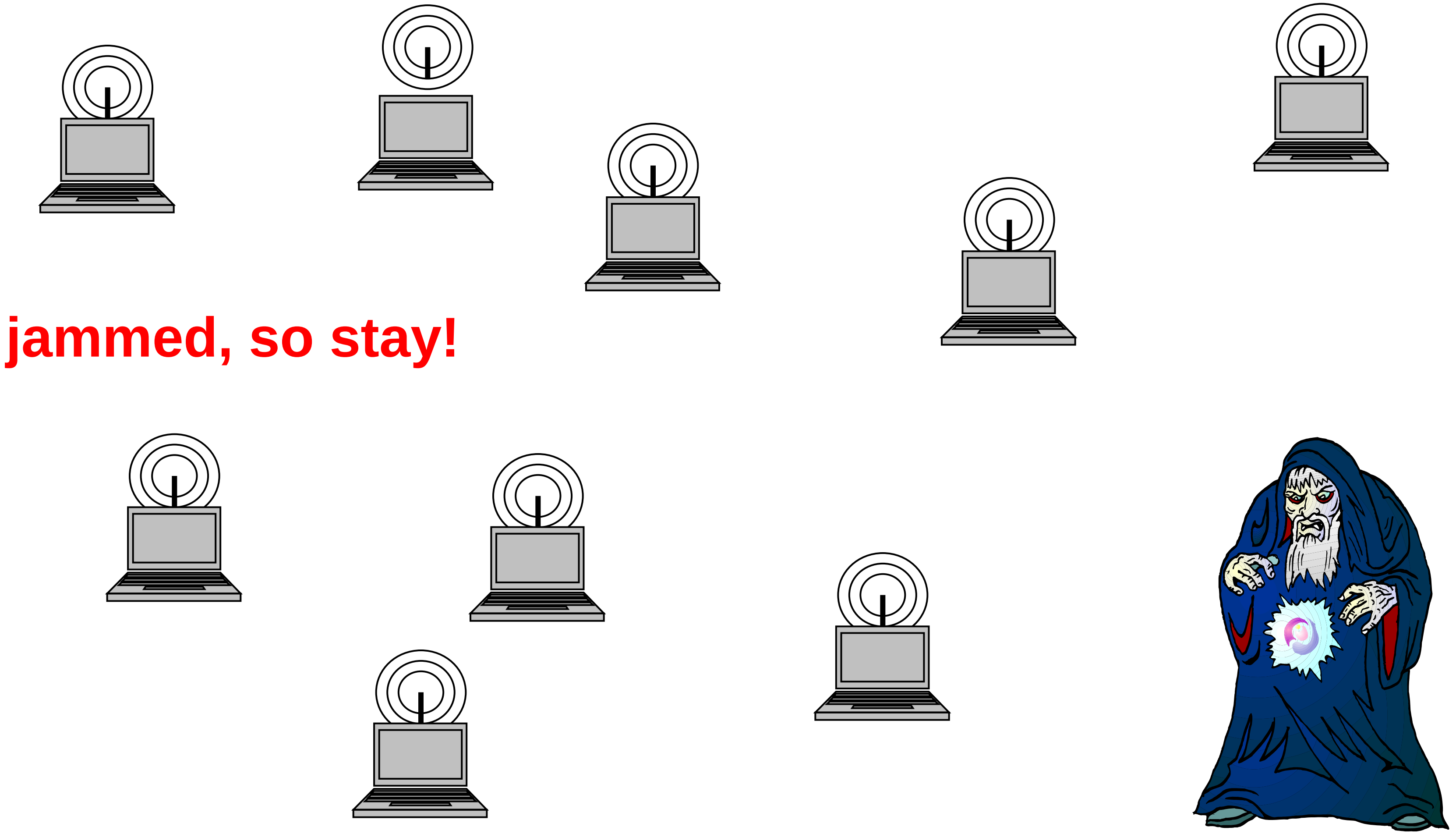


Basic Idea

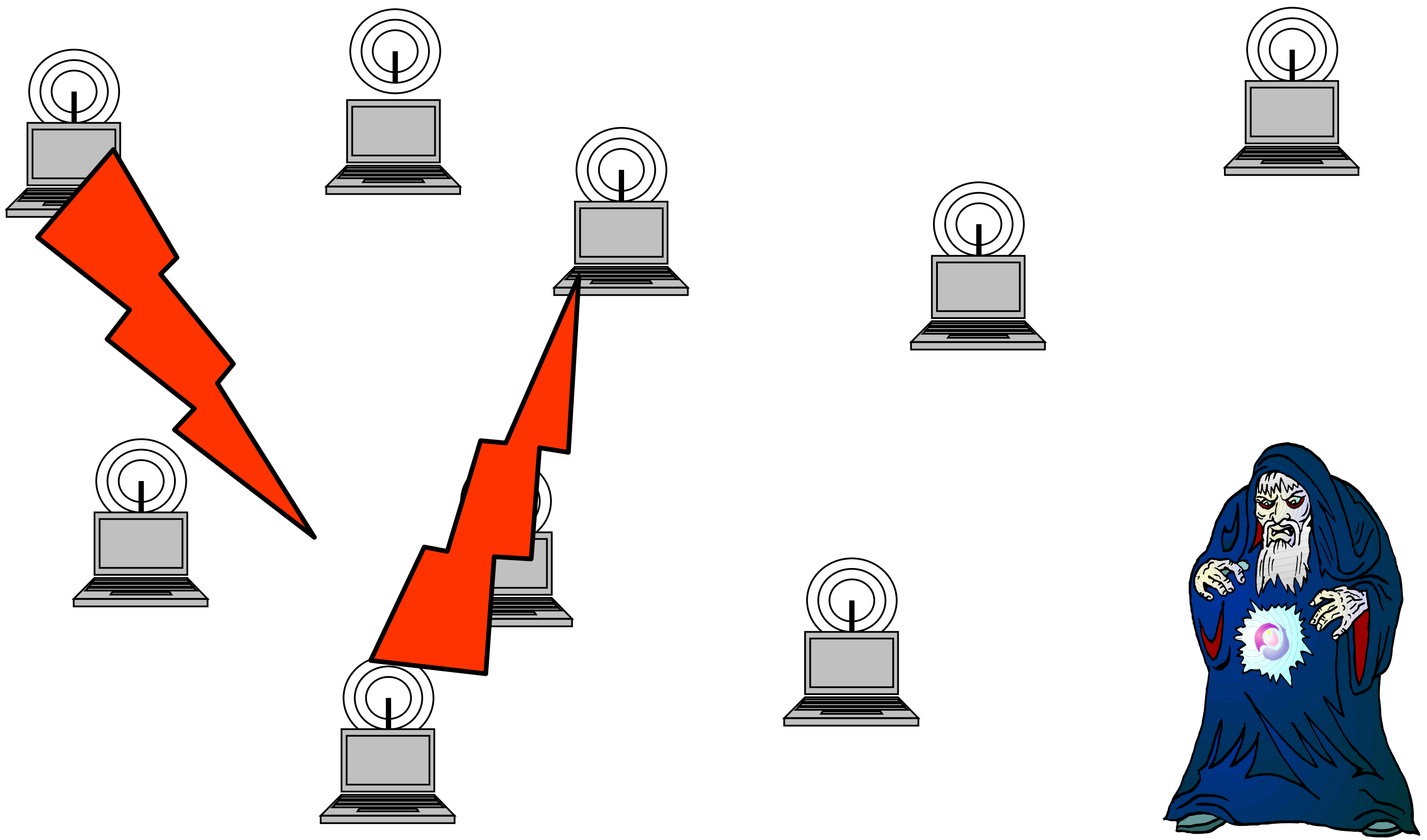
Basic Idea



Basic Idea



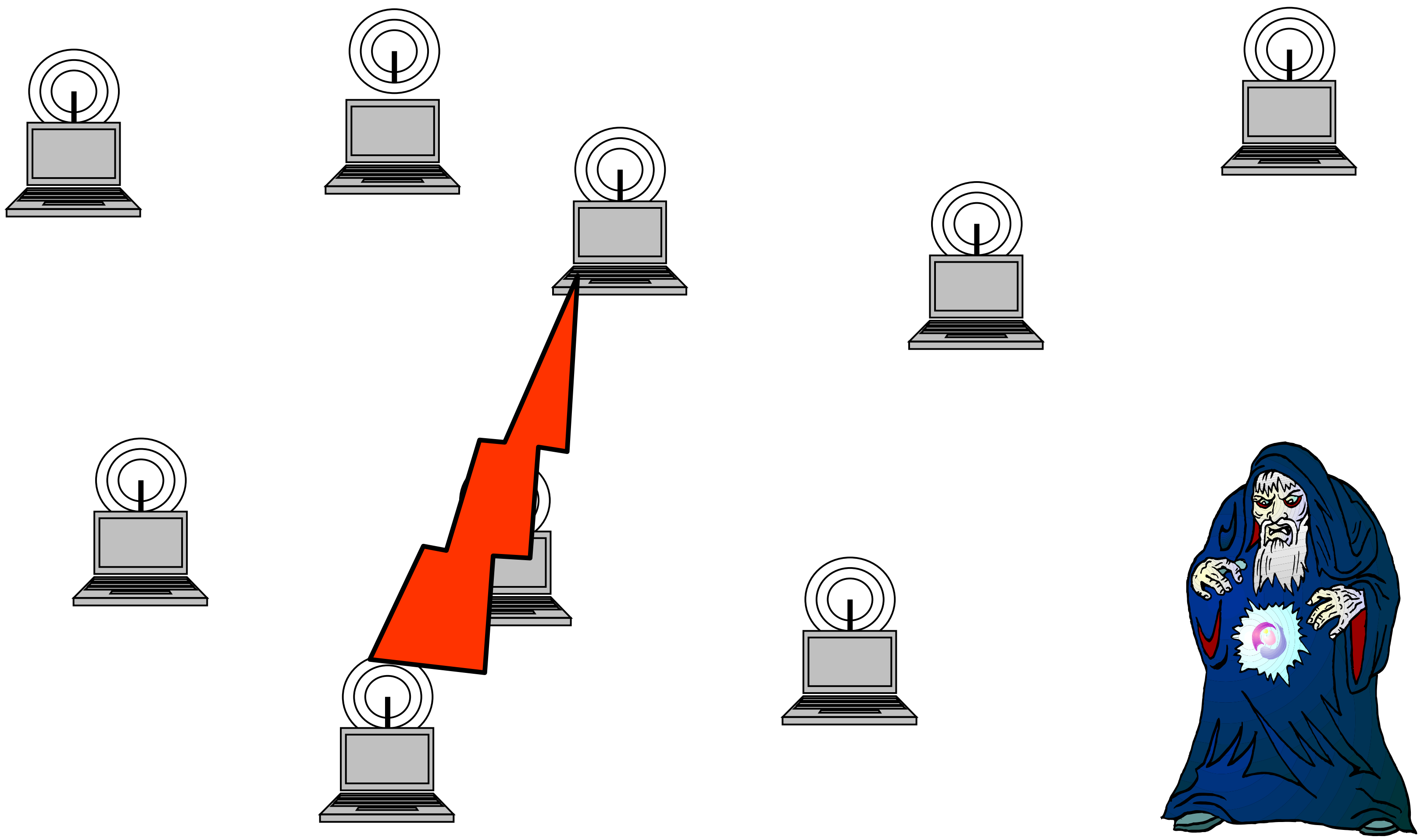
Basic Idea



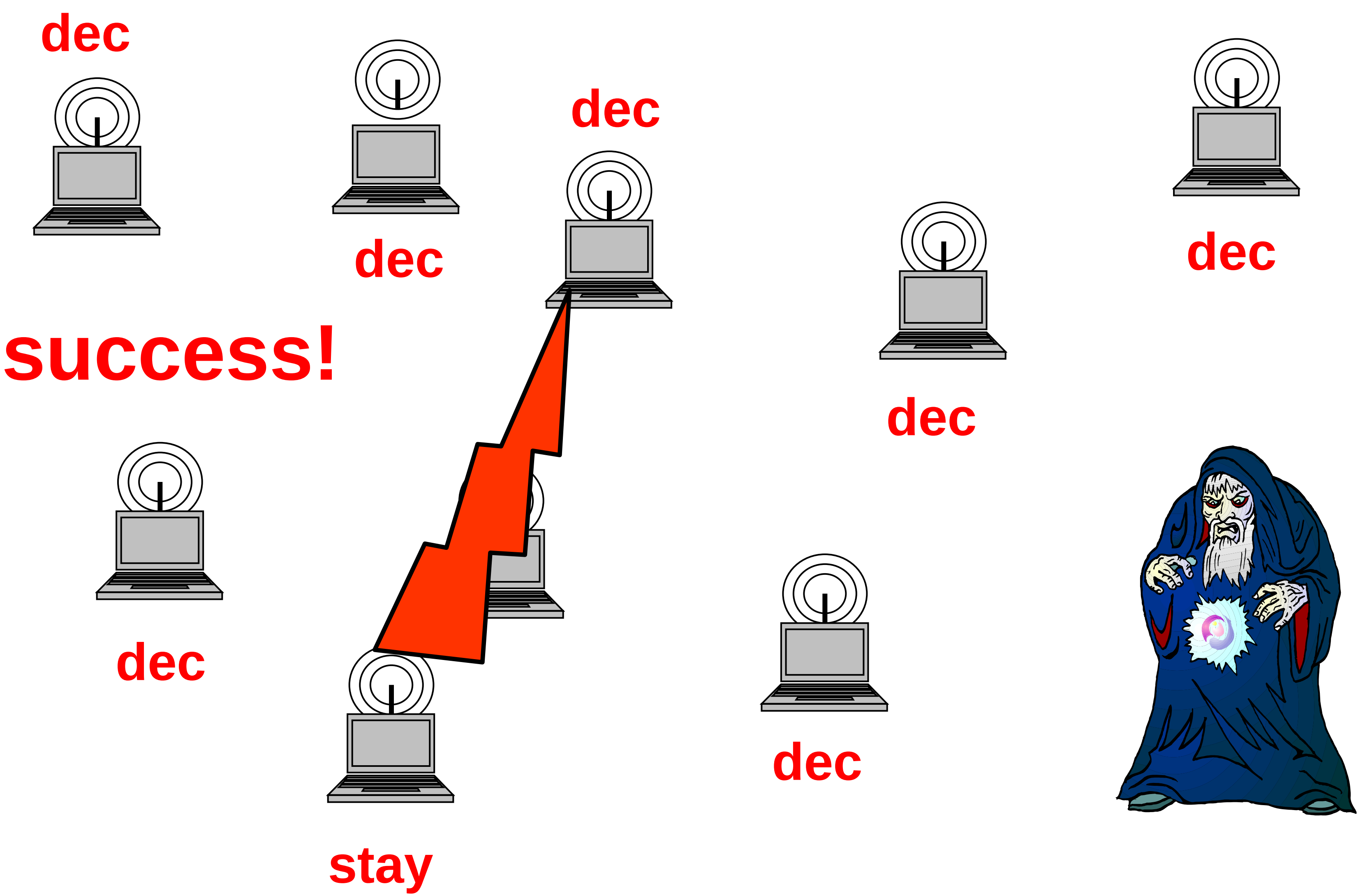
Basic Idea

collisions, so stay!

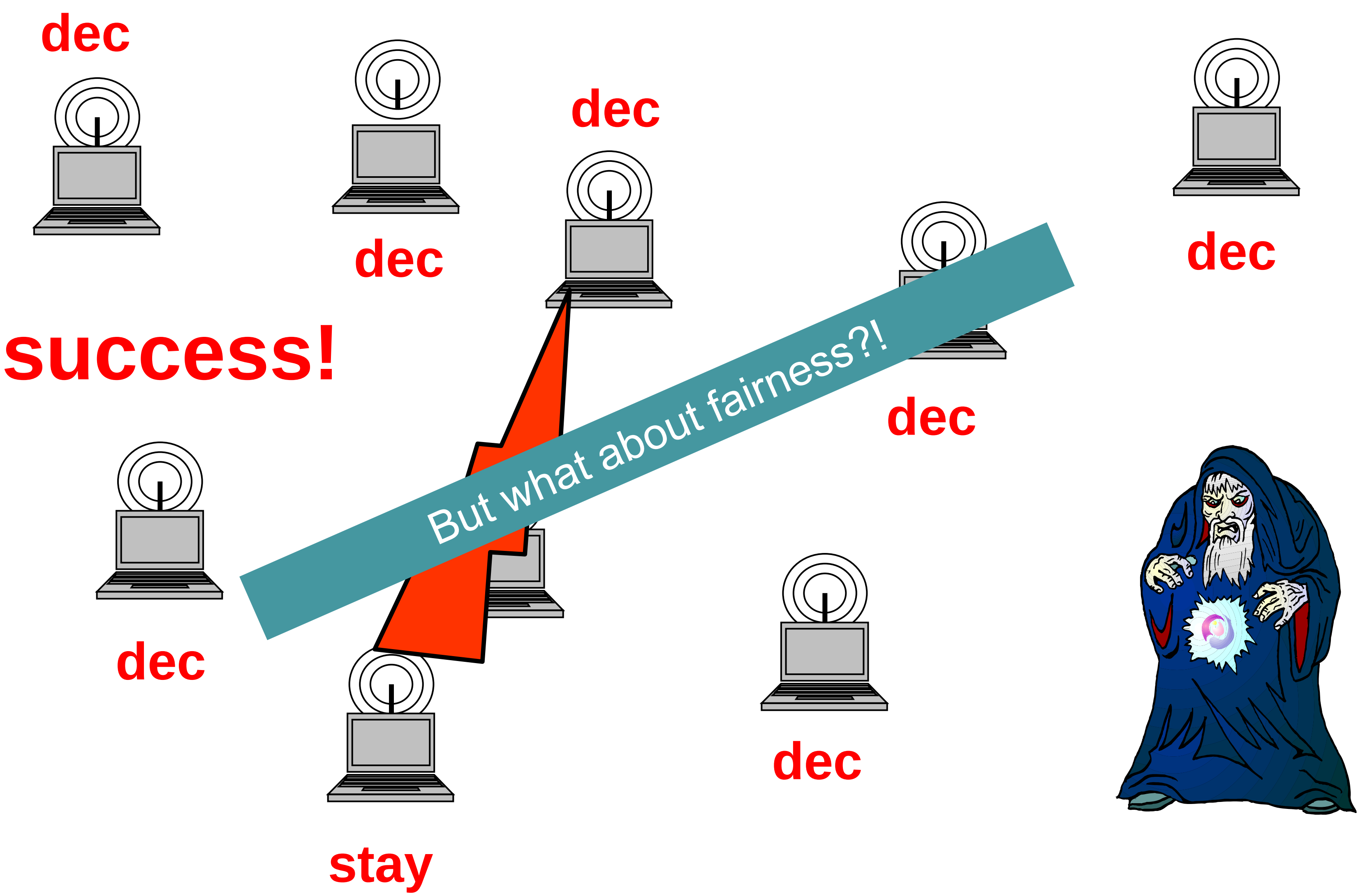
Basic Idea



Basic Idea



Basic Idea


dec
dec
dec
dec
success!
But what about fairness?!
dec
dec
dec
stay
dec

Basic Idea

Problem: if initially all nodes have high probabilities, probabilities stay high!

If (idle): $p_v := (1+\gamma)\ p_v$
If (success): $p_v := 1/(1+\gamma)\ p_v$

We still need a mechanism that reduces the probabilities even during busy times! But make it slowly!

```
T_v=1, c_v=1, p_v = p_max;
In each round:
    decide to send with prob p_v;
    if decide not to send:
        if sense idle channel: p_v = (1+γ) p_v; T_v--;
        if succ reception: p_v = 1/(1+γ) p_v; T_v--;
    c_v++;
    if (c_v>T_v)
        c_v=1;
        if no SUCC in last T_v steps:
            p_v = 1/(1+γ) p_v; T_v=T_v+1;
```

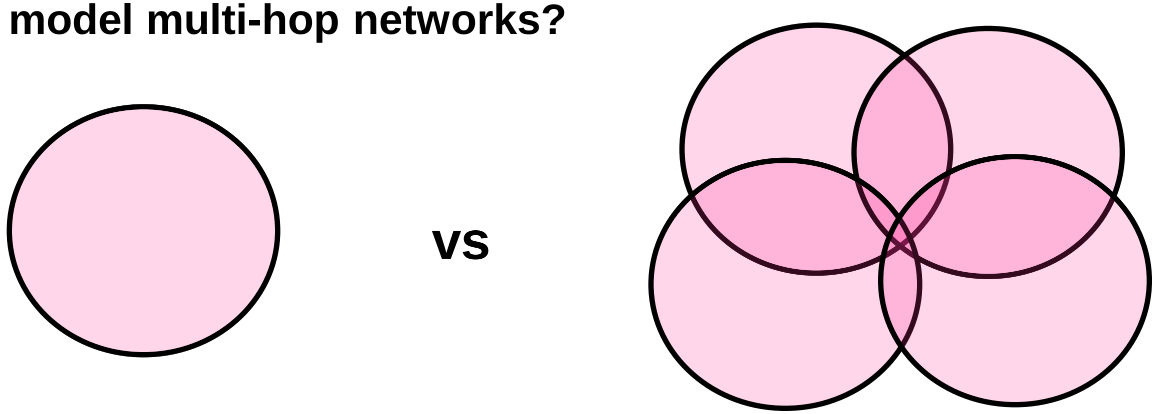
On the Definition of Throughput

Throughput

In a single-hop network easy: fraction of rounds in which a message is successfully sent.

We can prove constant competitive throughput for single-hop networks.

But how to model multi-hop networks?



On the Definition of Throughput

Throughput

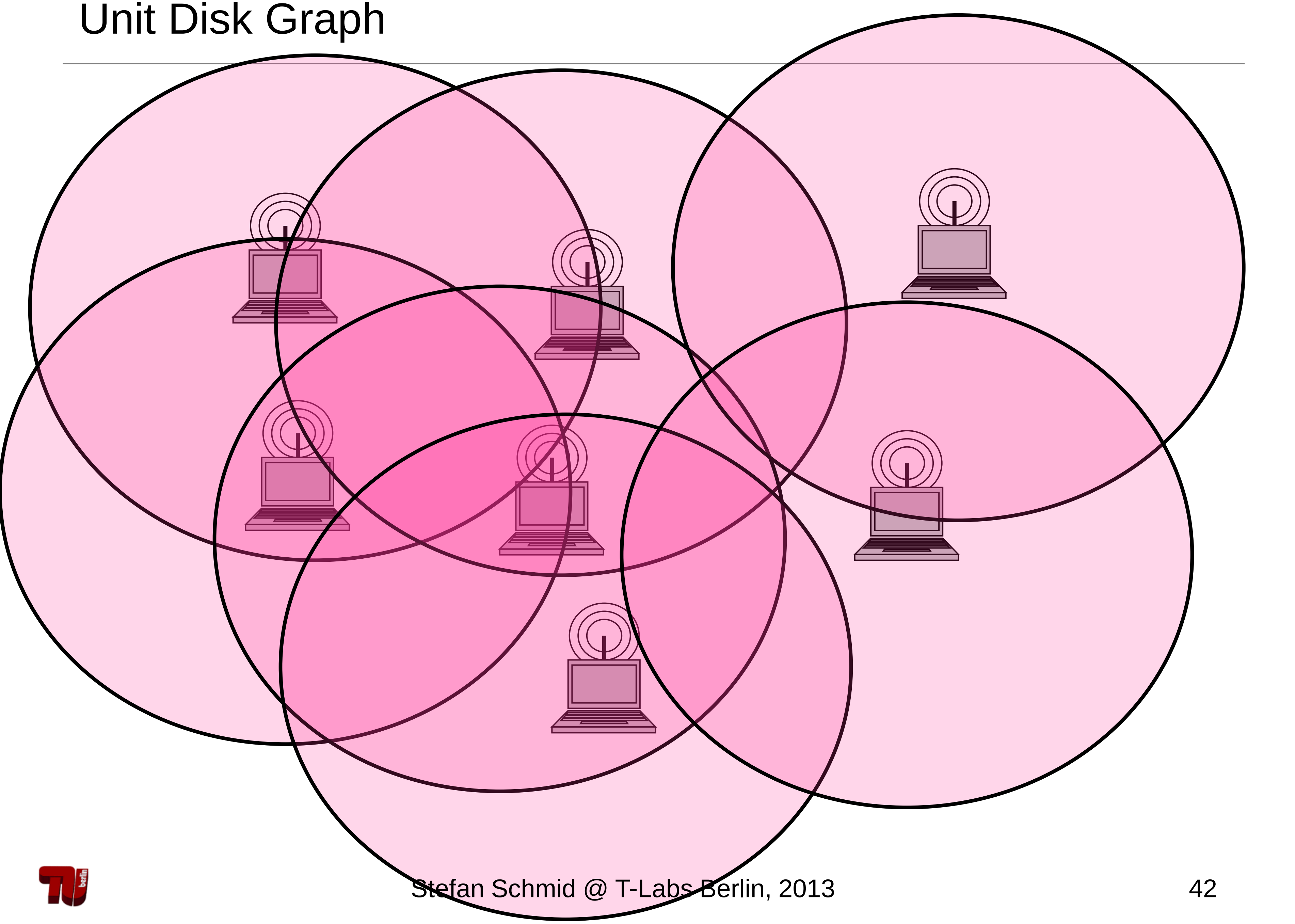
In a single-hop network easy: fraction of rounds in which a message is successfully sent.

What about multi-hop networks?

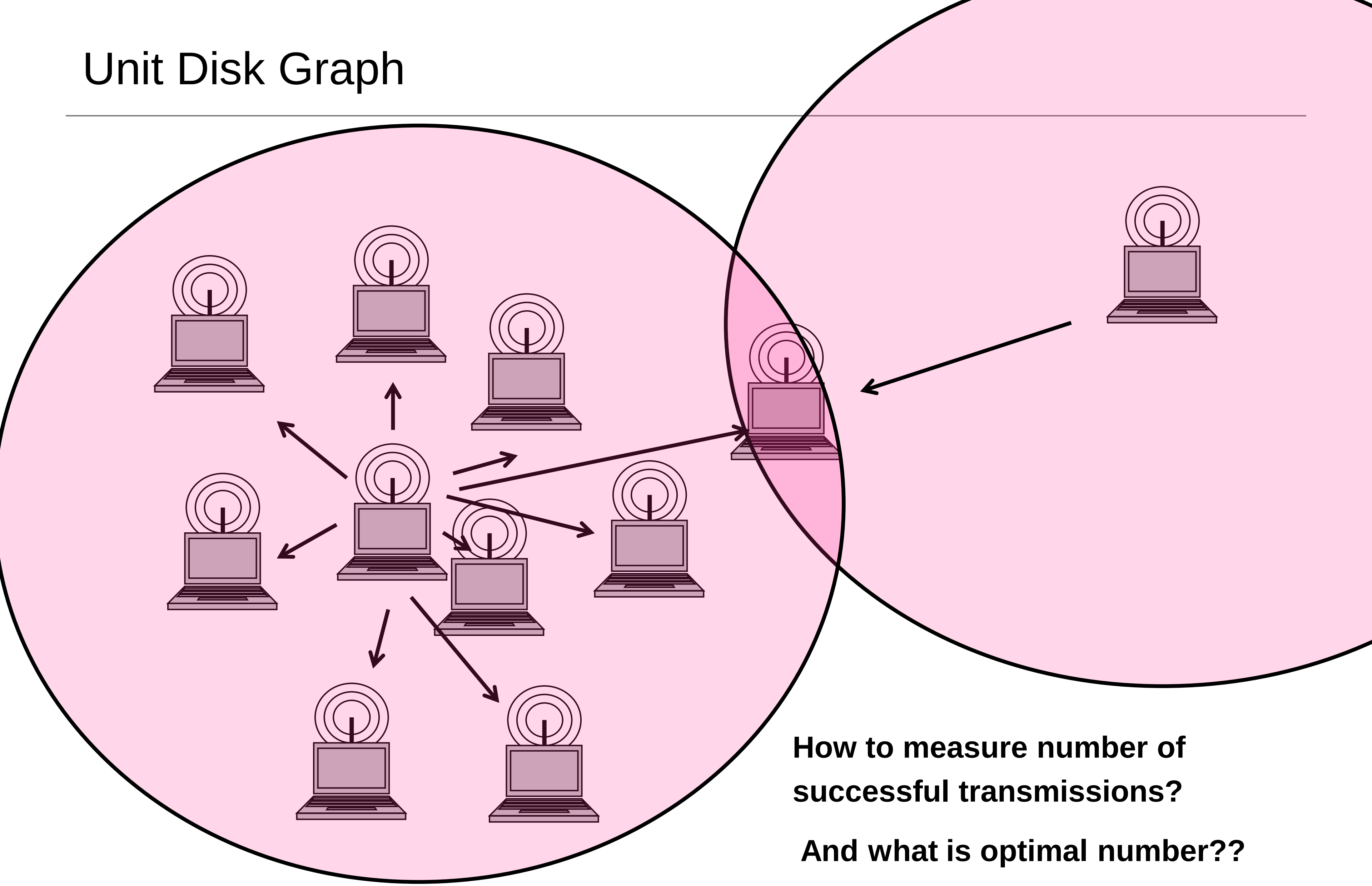
Unit Disk Graph

A most simple multi-hop network: each node has a transmission and interference range of one unit.

Unit Disk Graph



Unit Disk Graph



How to measure number of successful transmissions?

And what is optimal number??

On the Definition of Throughput

Throughput

In a multi-hop network, we define throughput from the perspective of a receiving node v. Given the number of non-jammed time steps **f(v)** at a node v, count the number **s(v)** of successful transmissions at v.

.

Competitive Throughput

A protocol has a competitive throughput if:

$$\sum f(v) \leq c \sum s(v)$$

for some constant c.

Happy with the definition?

On the Definition of Throughput

Actually, it would be even cooler if we could show a competitive throughput as defined as follows!

Strong Competitive Throughput

A protocol has a competitive throughput if:

$$f(v) \leq c\ s(v)$$

for some constant c. That is, it holds for every node v!

How to model the adversary in a distributed setting?

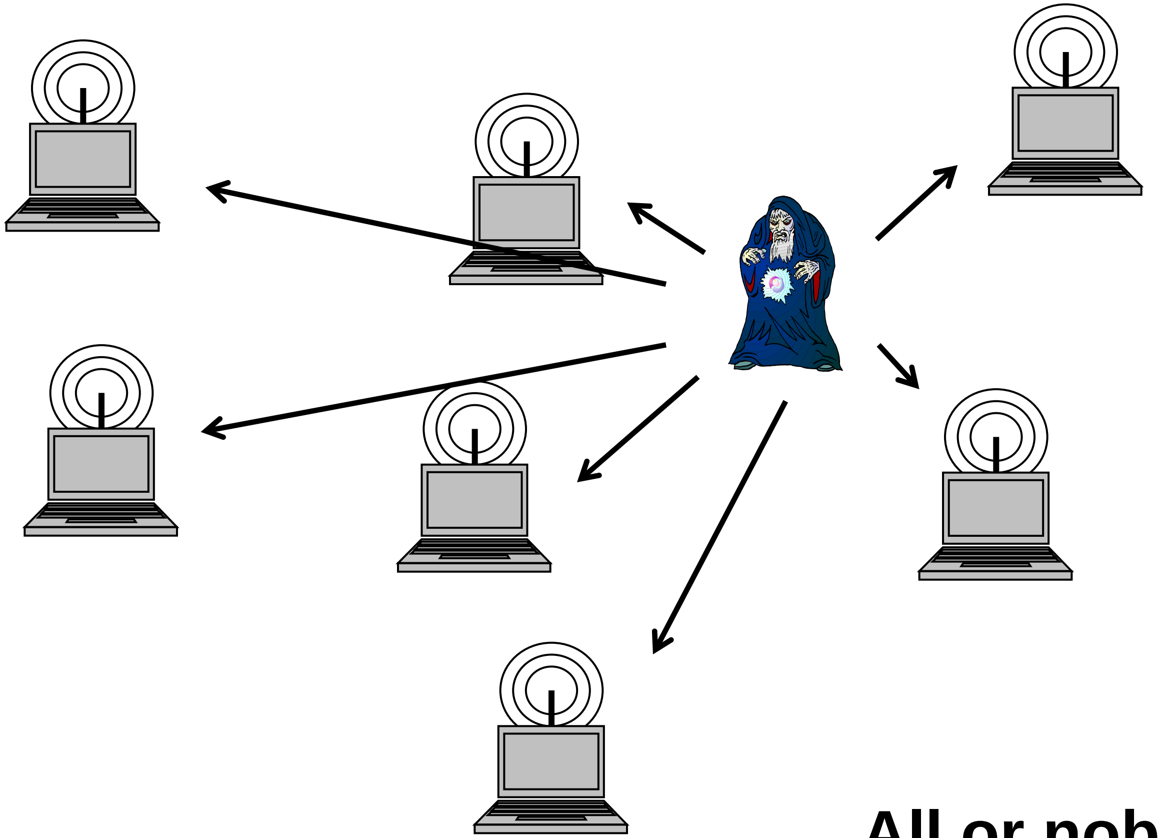
In single-hop network, adversary can jam all nodes or none: it is like a regular node.

In multi-hop network, adversary may even jam at different locations, different nodes!

k-Uniform Adversary

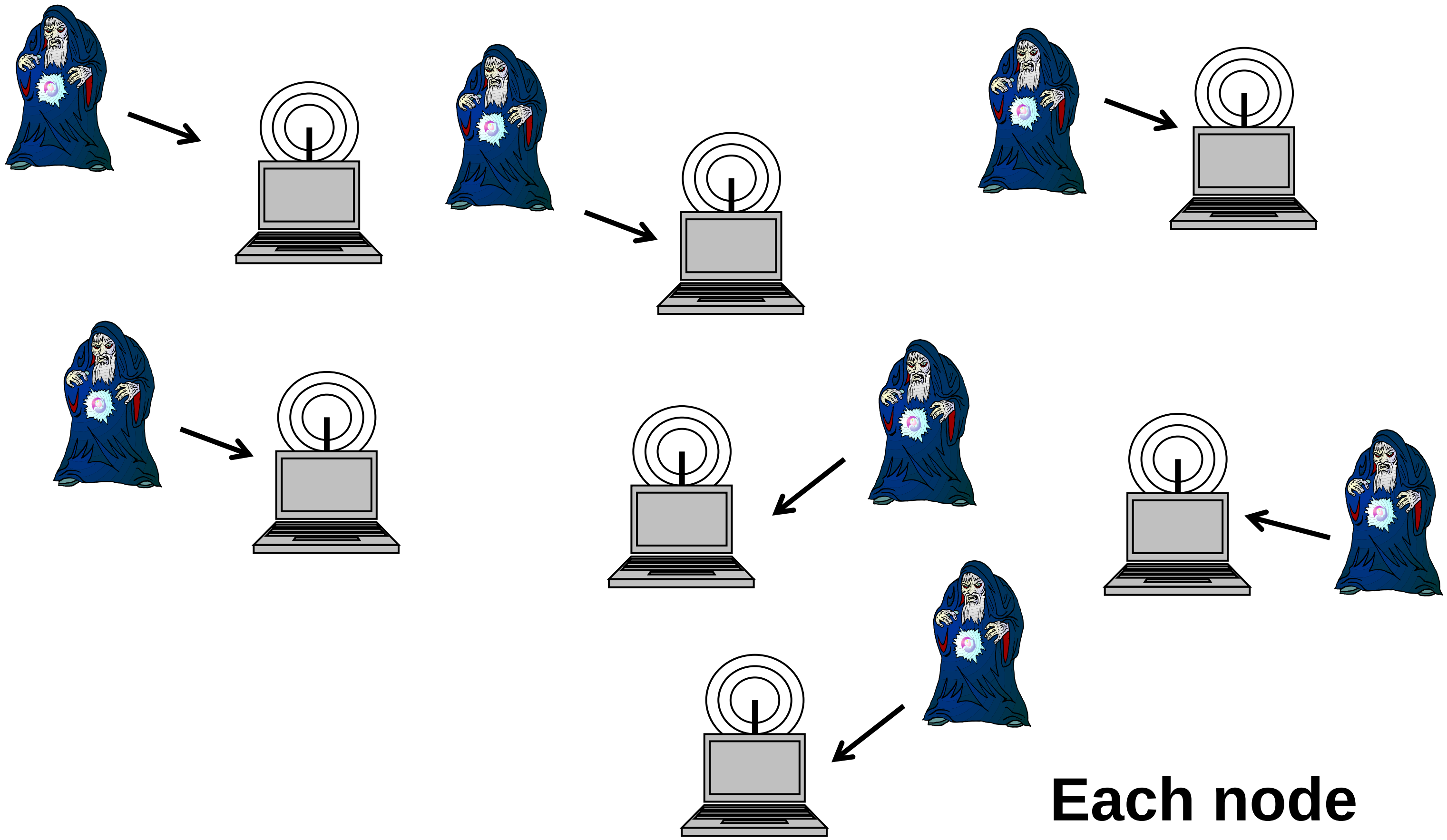
A k-uniform adversary can partition nodes into k groups, and jams each of these groups with the same pattern. (For each group, an **ε-**fraction of steps must be non-jammed.)

1-Uniform Adversary



All or nobody!

n-Uniform Adversary

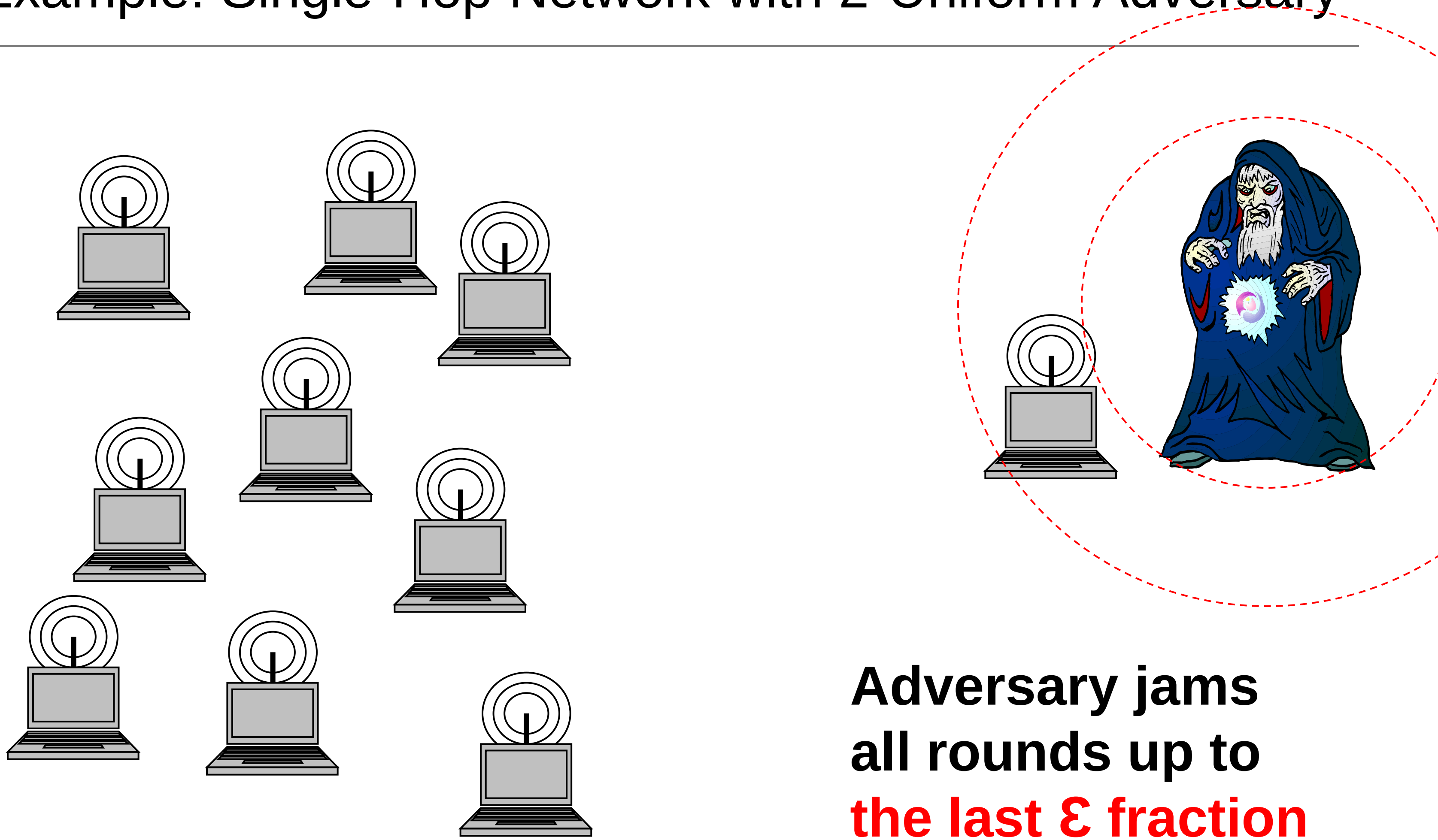


Each node individually!

Theorem

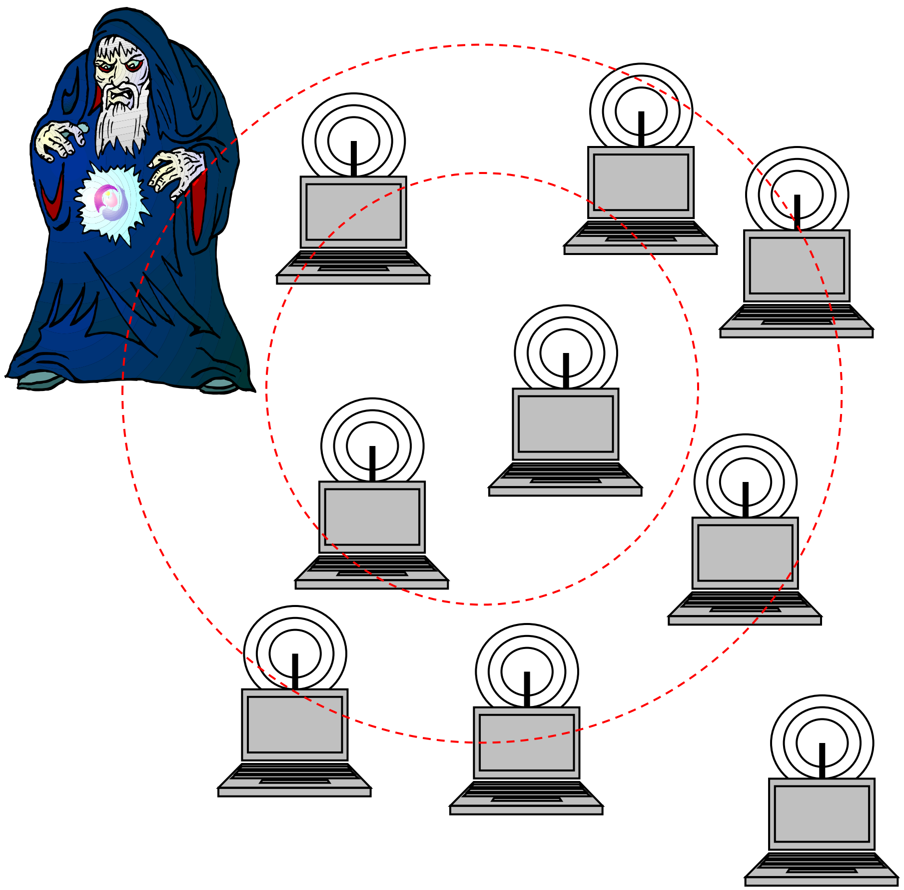
Constant competitive throughput can be achieved!
But not a strongly competitive throughput,
at least with our protocol.

Bad Example: Single-Hop Network with 2-Uniform Adversary



Adversary jams all rounds up to the last ε fraction of node on the right!

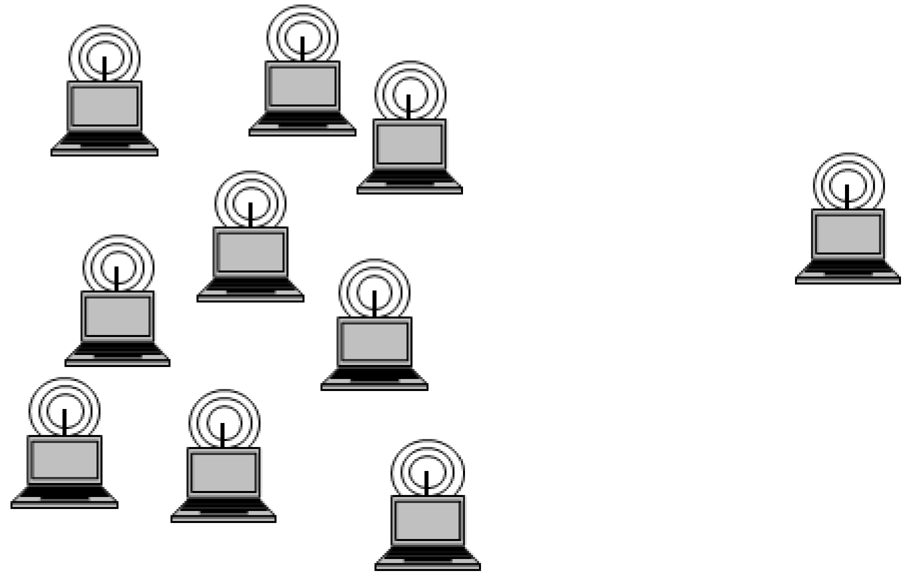
Bad Example: Single-Hop Network with 2-Uniform Adversary

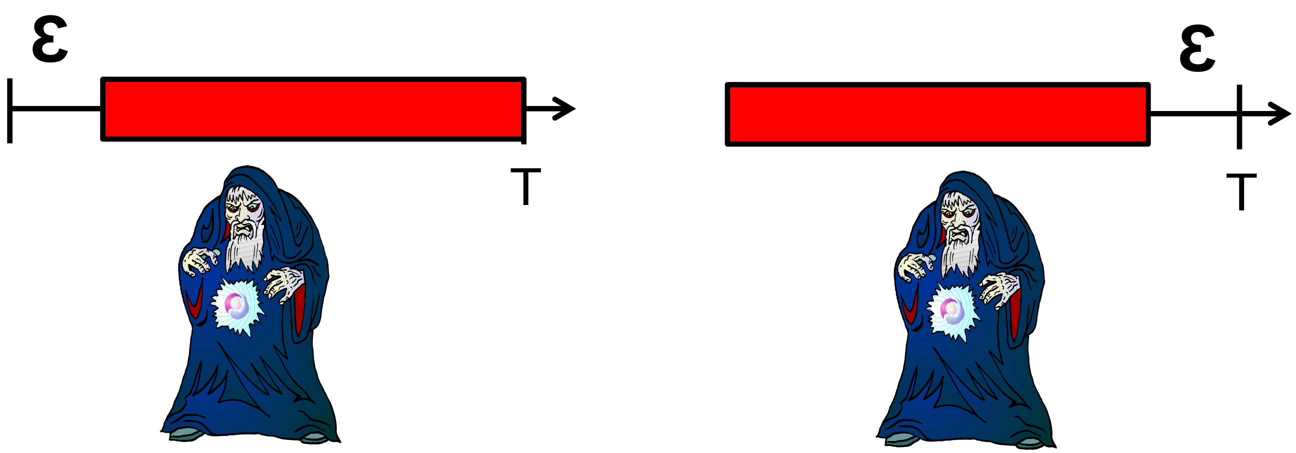


Then adversary jams all rounds up to the first ε fraction of nodes on the left!

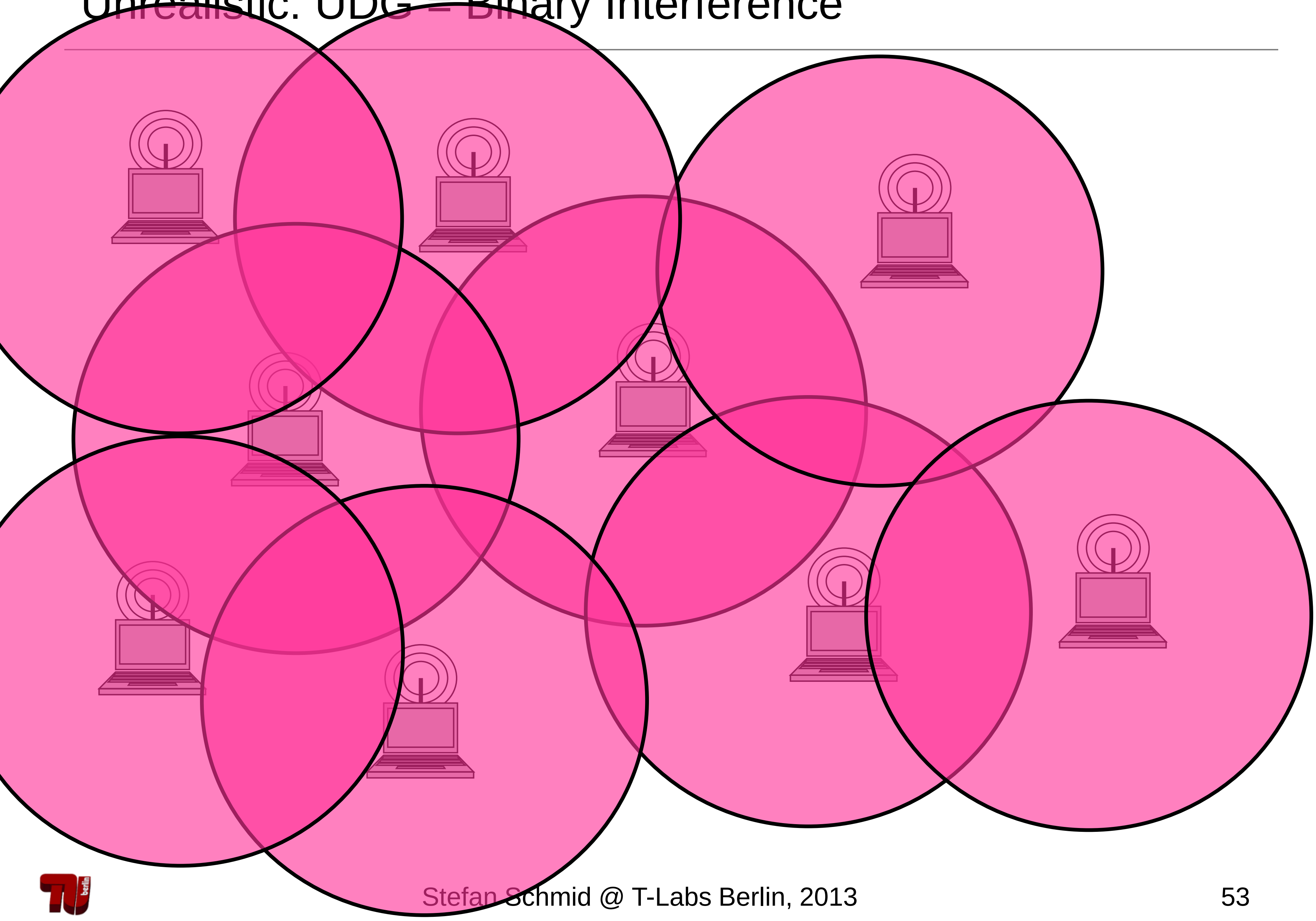
Bad Example: Single-Hop Network with 2-Uniform Adversary

Problem that Tv values are increased and pv values decreased for left nodes during jammed time, and until non-jammed rounds at right node left nodes do not send anything anymore!





Unrealistic: UDG = Binary Interference



SINR = Geometric Power Decrease

From UDG to SINR

Two new challenges:

- Interference range unbounded (but power declines)
- No clear distinction between “idle” and “busy” channel

Our MAC protocol solves these problems as follows:

- Make sure interference from far-away nodes is small
- Define a threshold to distinguish between idle and busy

New adversary model:

- Jammed rounds is no longer bounded
- But adversary has limited energy budget over time

From UDG to SINR

Two new challenges:

- Interference range unbounded (but power declines)
- No clear distinction between “idle” and “busy” channel

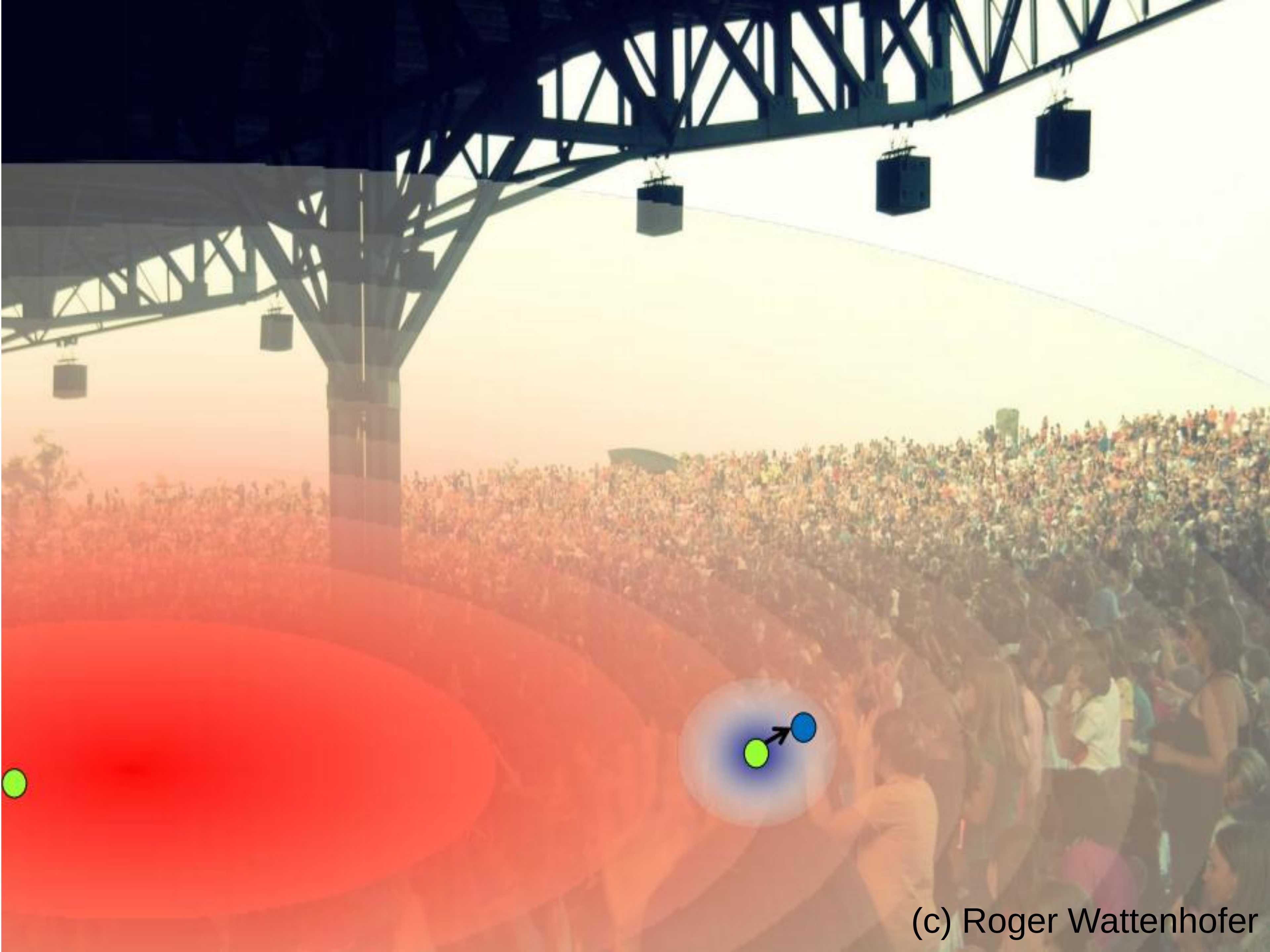
Our MAC protocol solves these problems as follows:

- Make sure interference from far-away nodes is small
- Define a threshold to distinguish between idle and busy

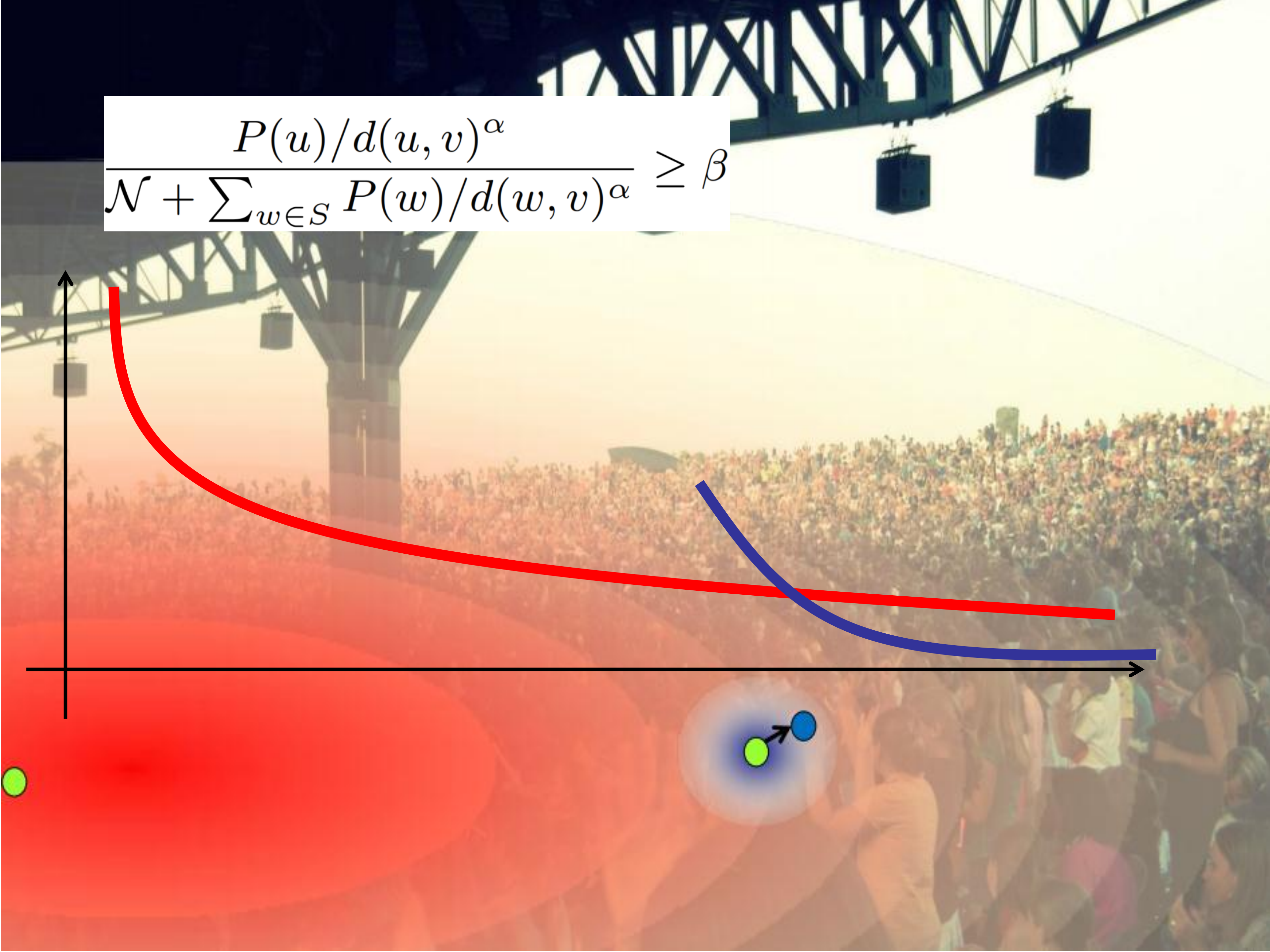
New adversary model:

- Jammed rounds is no longer bounded
- But adversary has limited energy budget over time

First some intuition for SINR...



$$\frac{P(u)/d(u,v)^{\alpha}}{\mathcal{N} + \sum_{w \in S} P(w)/d(w,v)^{\alpha}} \geq \beta$$



From UDG to SINR: what changes?

- New adversary model: energy based

- Adapt protocol: Cannot distinguish idle and busy!

If (idle): $p_v := (1+\gamma)\, p_v$
If (success): $p_v := 1/(1+\gamma)\, p_v$

Robust MAC under SINR: Adversary (1)

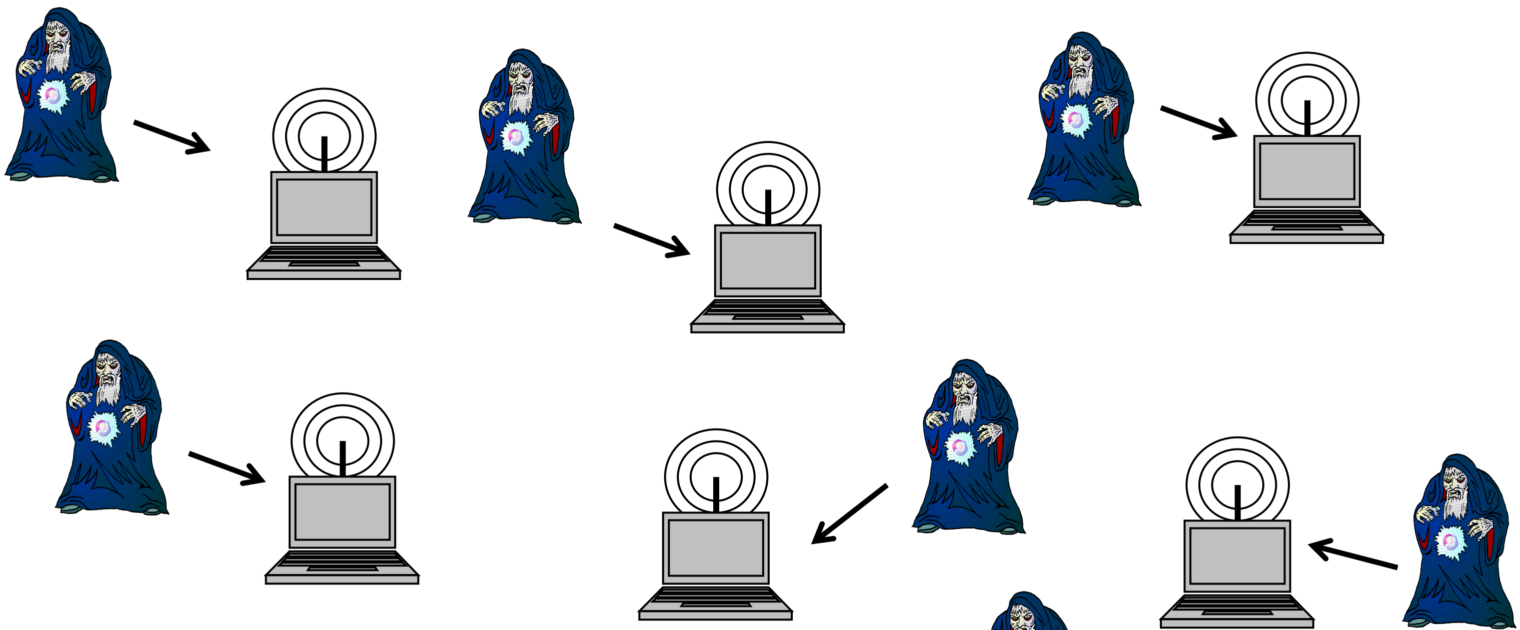
$$\frac{P(u)/d(u,v)^\alpha}{\mathcal{N} + \sum_{w \in S} P(w)/d(w,v)^\alpha} \geq \beta$$

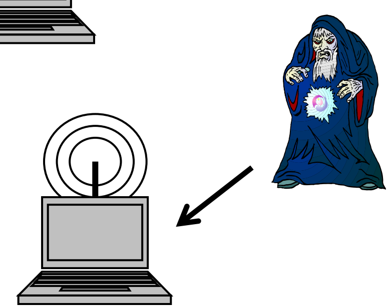
Classic model: receive when close by!

Our new model: Adversarial SINR!

$$\frac{P/d(u,v)^\alpha}{\mathcal{ADV}(v) + \sum_{w \in S} P/d(w,v)^\alpha} \geq \beta$$

Robust MAC under SINR: Adversary (2)





Energy to jam each node individually!

Much energy needed for close pairs, less energy needed for far pairs!

Robust MAC under SINR: Protocol

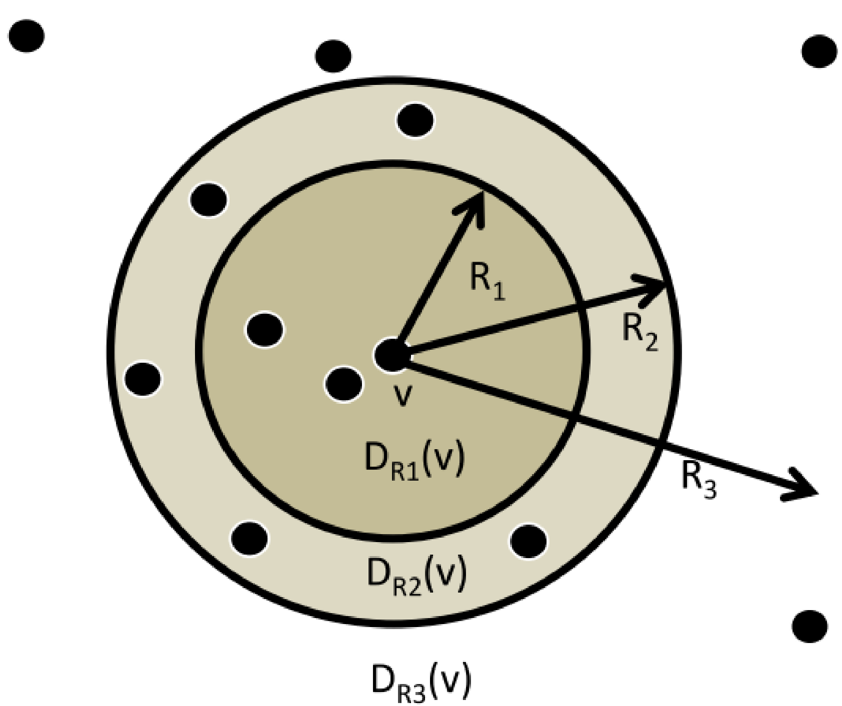
Initially, every node v sets $T_v := 1$, $c_v := 1$, and $p_v := \hat{p}$. In order to distinguish between idle and busy rounds, each node uses a fixed noise threshold of ϑ.

The SADE protocol works in synchronized rounds. In every round, each node v decides with probability p_v to send a message. If it decides not to send a message, it checks the following two conditions:

- If v successfully receives a message, then $p_v := (1+\gamma)^{-1} p_v$.
- If v senses an idle channel (i.e., the total noise created by transmissions of other nodes and the adversary is less than ϑ), then $p_v := \min\{(1+\gamma)p_v, \hat{p}\}, T_v := \max\{1, T_v - 1\}$.

Afterwards, v sets $c_v := c_v + 1$. If $c_v > T_v$ then it does the following: v sets $c_v := 1$, and if there was no idle step among the past T_v rounds, then $p_v := (1+\gamma)^{-1} p_v$ and $T_v := T_v + 2$.

Robust MAC under SINR: Analysis



Many nodes far away, cannot influence center much!

THEOREM 1.1. *When running* SADE *for at least* $\Omega((T \log N)/\epsilon + (\log N)^4/(\gamma\epsilon)^2)$ *time steps,* SADE *has a* $2^{-O\left((1/\epsilon)^{2/(\alpha-2)}\right)}$*-competitive throughput for any* $((1-\epsilon)\vartheta, T)$*-bounded adversary as long as (a) the adversary is uniform and the transmission range of every node contains at least one node, or (b) there are at least* $2/\epsilon$ *nodes within the transmission range of every node.*

How to use the protocol to elect a leader?

How to make the protocol fair?

How to make the protocol fair in the presence of other networks?

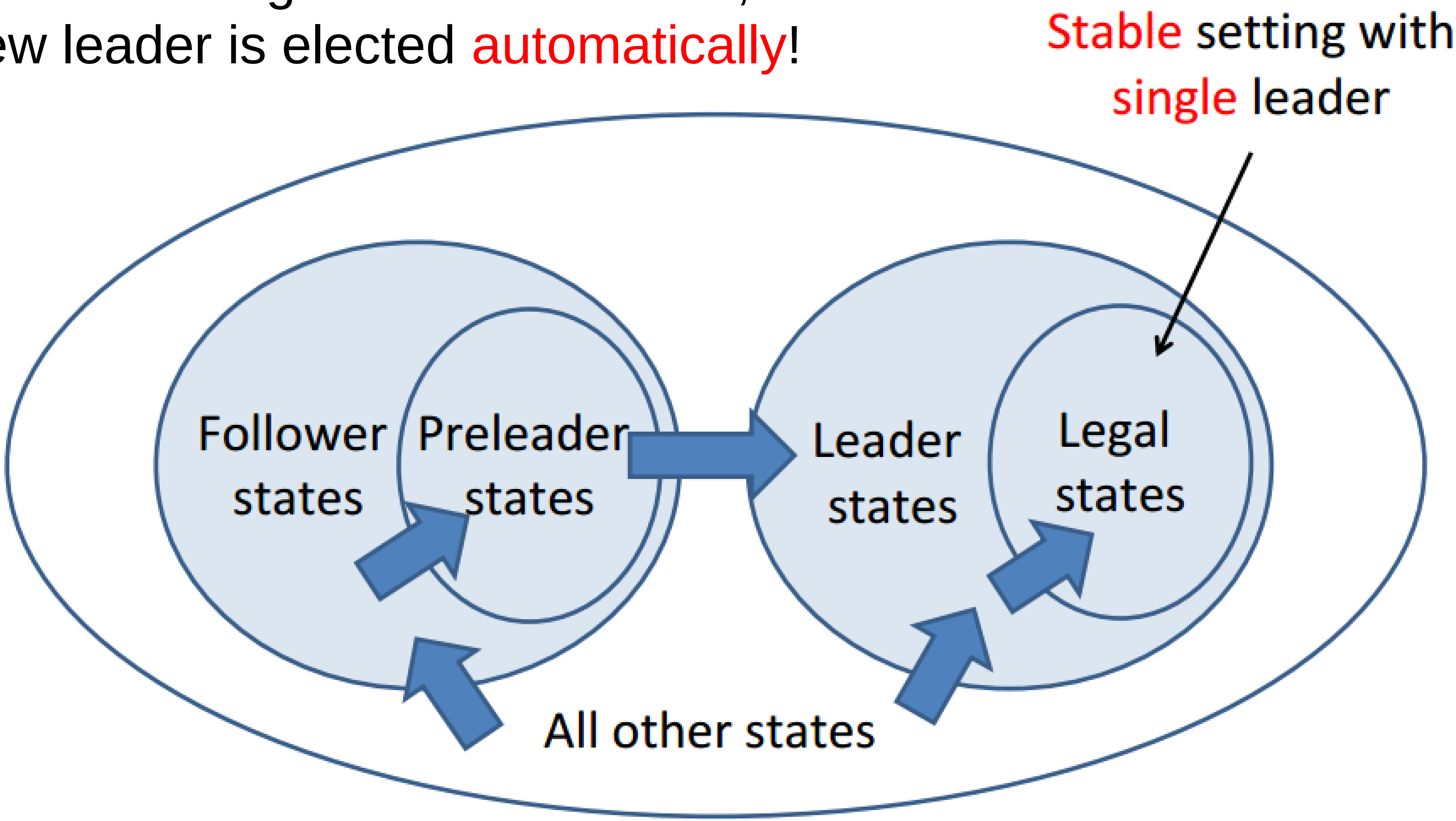
Leader Election

Nodes shall converge to a situation where exactly one node considers itself a leader, and all other nodes followers. (Why good?)

Idea: use MAC protocol we have, but leaders should increase sending probability faster than follower to determine the winner.

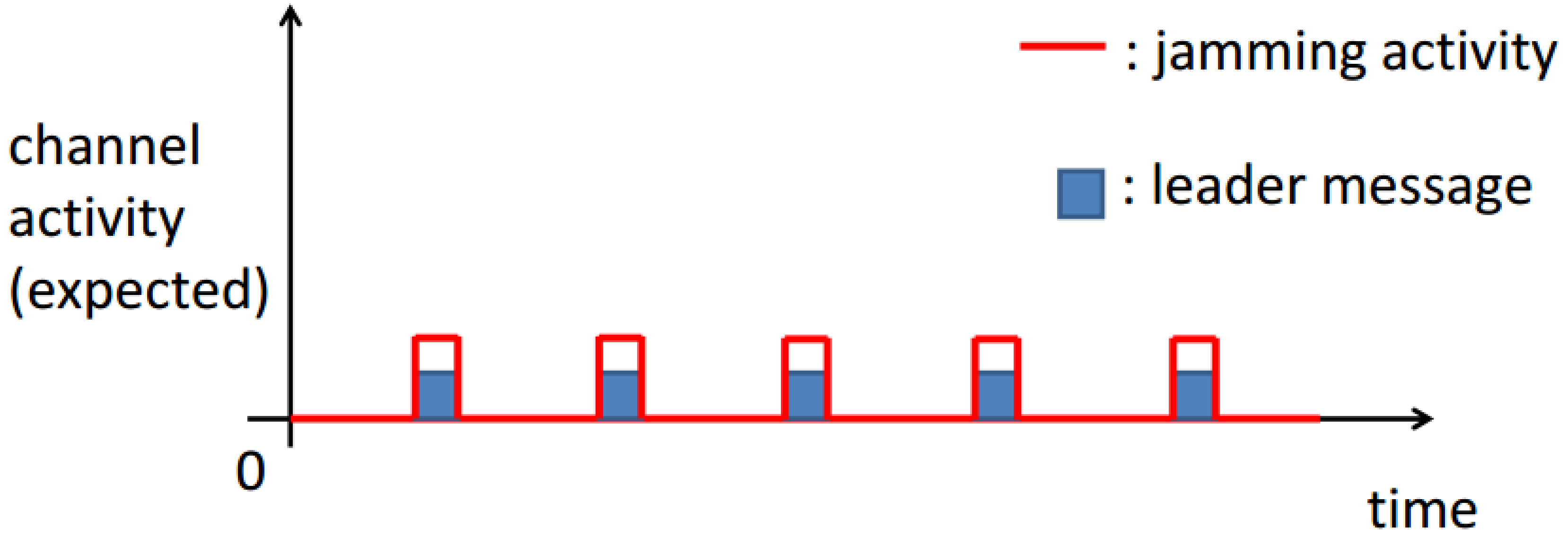
Extension 1: Self-Stabilization

Self-stabilizing: when leader dies,
new leader is elected automatically!



Extension 1: Leader Election

Problem: I cannot rely on leader “keep-alive” messages under jamming! Unless we randomize…!



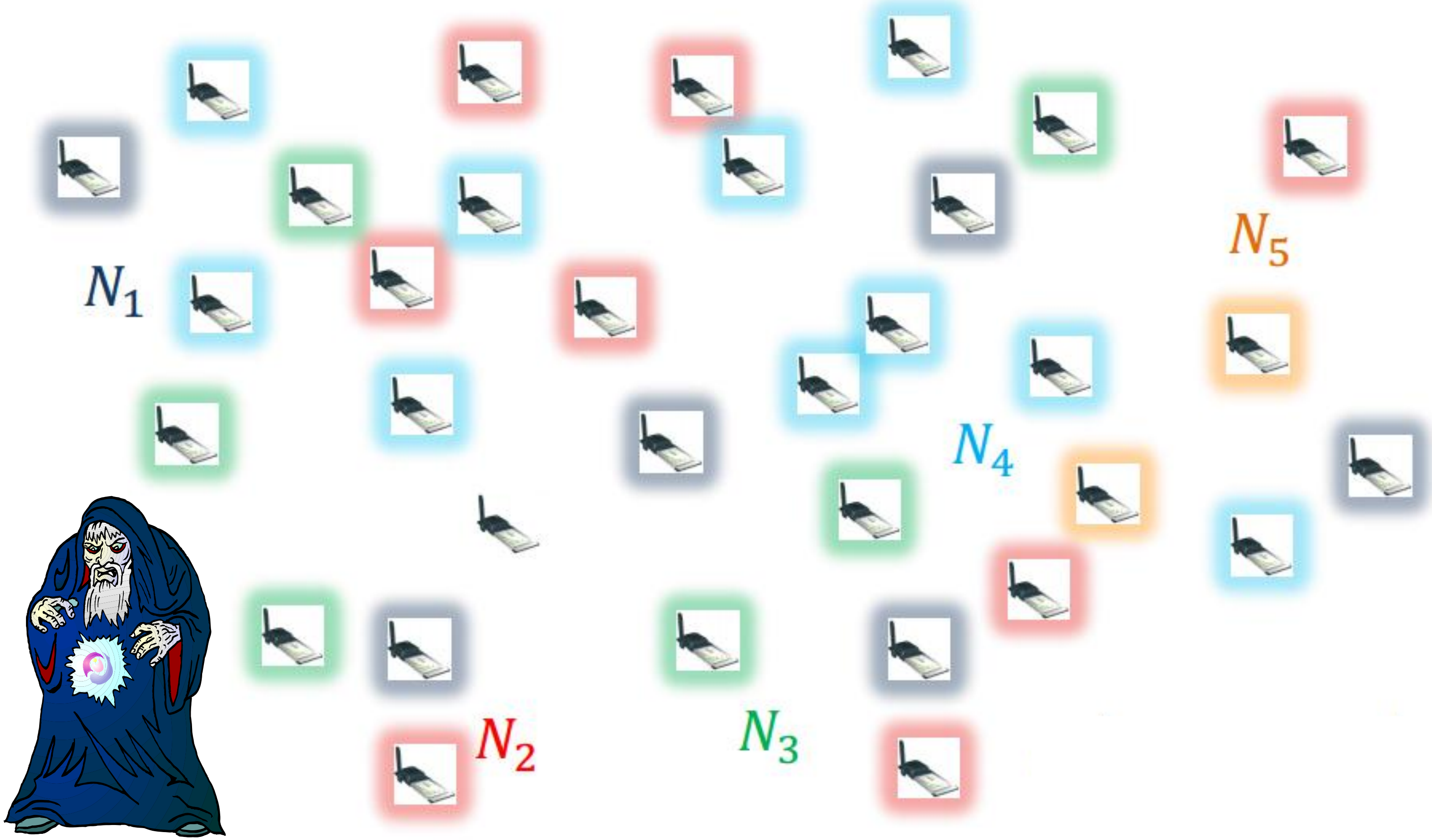
Extension 1: Achieving Fairness

Fairness

Each node should have roughly the same number of successful transmissions.

Idea: nodes synchronize their pv values during transmissions!

Extension 1: Co-Existing Networks



Extension 1: Co-Existing Networks

Security Council, UN

Co-Existing Networks

k networks within transmission range. Should not communicate explicitly. (Different protocols, security levels, …). We want that (1) overall throughput is constant competitive, (2) different networks have same throughput (fairness).

Extension 1: Co-Existing Networks

Main idea:

1. It's not a good idea that each network tries to reach a constant cumulative probability! Because then we have a probability of O(k), which would imply a throughput of exp(-k).
2. Rather, let nodes synchronize implicitly via the idle rounds. increase sending probability slower, and depending on the time period since the last idle time step was observed. (The longer this period, the smaller the increase.)

Research Overview

1. **PODC 2008, Awerbuch et al.: “A jamming-resistant MAC protocol for single-hop wireless networks”**

   Competitive throughput for single-hop network, adaptive adversary

2. **DISC 2010, Richa et al.: “A Jamming-Resistant MAC Protocol for Multi-Hop Wireless Networks” (also in DIST Journal)**

   Competitive throughput for Unit Disk multihop network, adaptive adversary

3. **MOBIHOC 2011, Richa et al.: “Self-Stabilizing Leader Election for Single-Hop Wireless Networks despite Jamming”**

   Robust leader election in single-hop network under reactive adversary

4. **ICDCS 2011, Richa et al.: “Competitive and Fair Medium Access despite Reactive Jamming” (also in journal TON)**

   Competitive throughput in single-hop network under reactive adversary

Research Overview

5. **ACM S3 2011, Richa et al.: "Towards Jamming-Resistant and Competitive Medium Access in the SINR Model"**

   First ideas for SINR network

6. **ACM PODC 2012, Richa et al.: "Towards Jamming-Resistant and Competitive Medium Access in the SINR Model"**

   Competitive throughput for co-existing single-hop networks under adaptive jammer

7. **Under Submission, Ogierman et al.: "Competitive Medium Sharing under Adversarial SINR"**

   Competitive throughput in SINR setting under adaptive jammer

Thank you for your interest!

Dekuji!