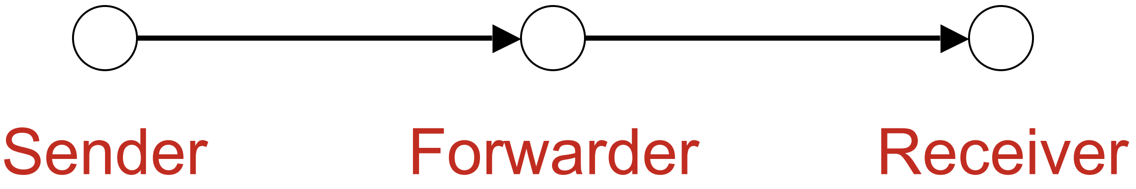
Punishment in Selfish Wireless Networks: A Game Theoretic Analysis

Dave Levin

University of Maryland,
College Park

Wireless Ad Hoc Networks



- What incentive does the forwarder have
  - To use its own battery?
  - To not send its own packets instead?

Solution Flavor #1

- Require external means
  - Trusted component (bank or hardware)

- Pay nodes to forward for you
  - Sprite, Ad Hoc-VCG
- Use tamper-proof hardware
  - [Buttyán & Hubaux, 2000]

- We’ll call these external incentive mechanisms
- Deployment Strategy?

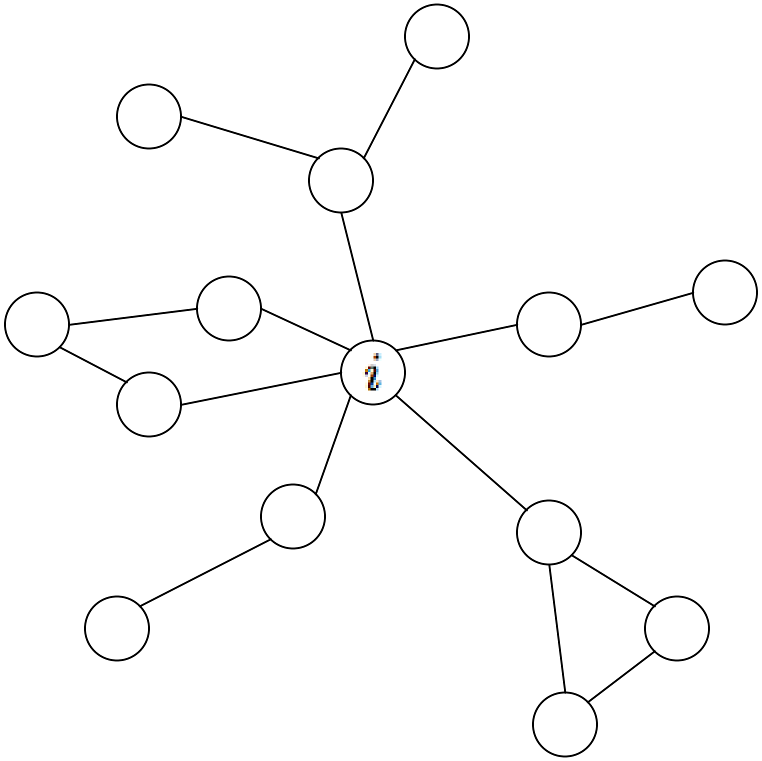
Solution Flavor #2

- Goal: easier deployment
- Approach:

  Internal incentive mechanisms that can be built on top of existing 802.11 primitives

- This work:
  - Model to understand internal mechanisms
  - New mechanism: jamming

Assumptions: Network



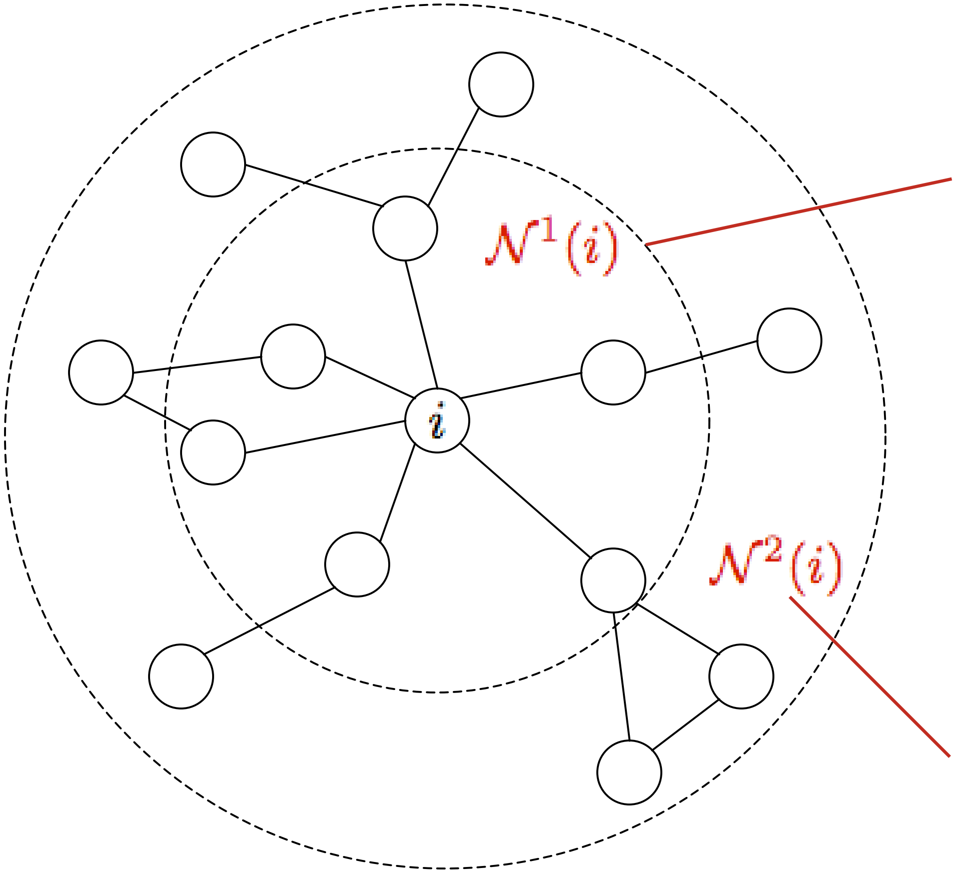
Edges

Can transmit to one another

Bidirectional Links

802.11 requires link-level ACKs

Assumptions: Transmissions



Transmission range

All 1-hop nbrs hear i's transmissions

Carrier Sense Range

No one within 2 hops can send/recv while i is transmitting

Assumptions: Node Behavior

- Nodes are rational but not malicious
  - Maximize their connectivity over time

- Limit of means criterion
  - (u_i^t) preferred to (w_i^t) iff

$$\exists \epsilon > 0 \;\; \text{s.t.} \;\; \frac{1}{T}\sum_{i=1}^{T}(u_i^t - w_i^t) > \epsilon$$

  - i.e., go with the best strategies *on average*

Assumptions: Node Preferences

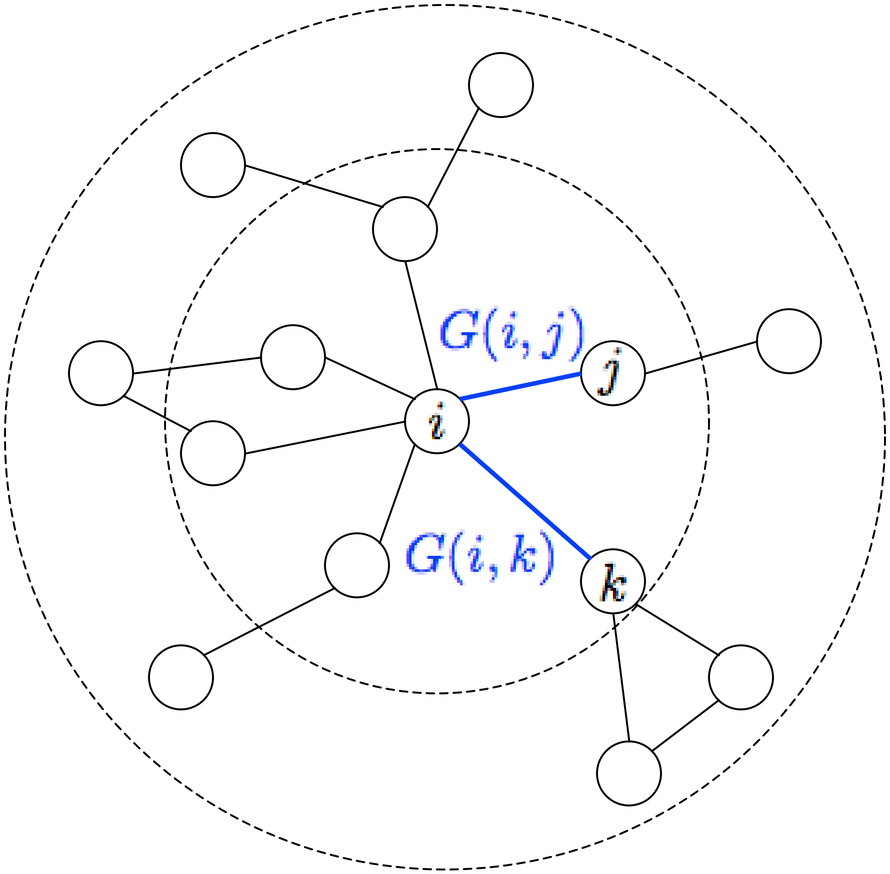
- Connectivity is best
  Utility C
- Disconnected is bad
  $C > D$
- Forwarding costs
  $C > F < 0$
- But connectivity overcomes
  $C + F > D$

Nodes can be disconnected and forward

w.l.o.g.:

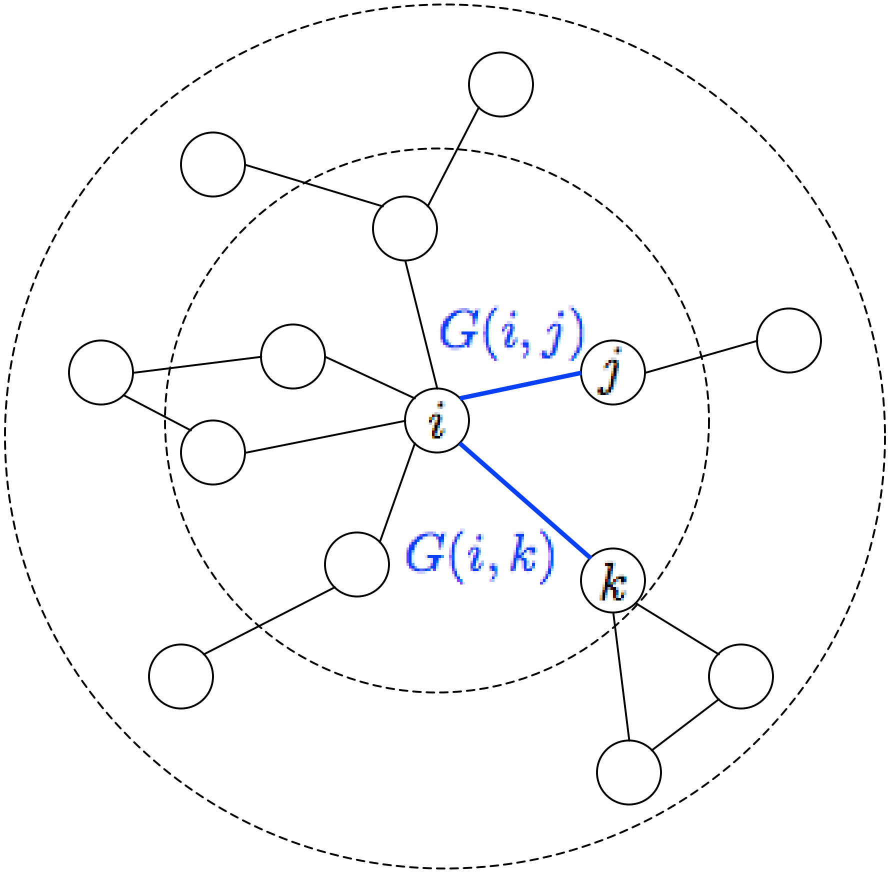
$C = 2$
$D = 0$
$F = -1$

Games with Neighbors



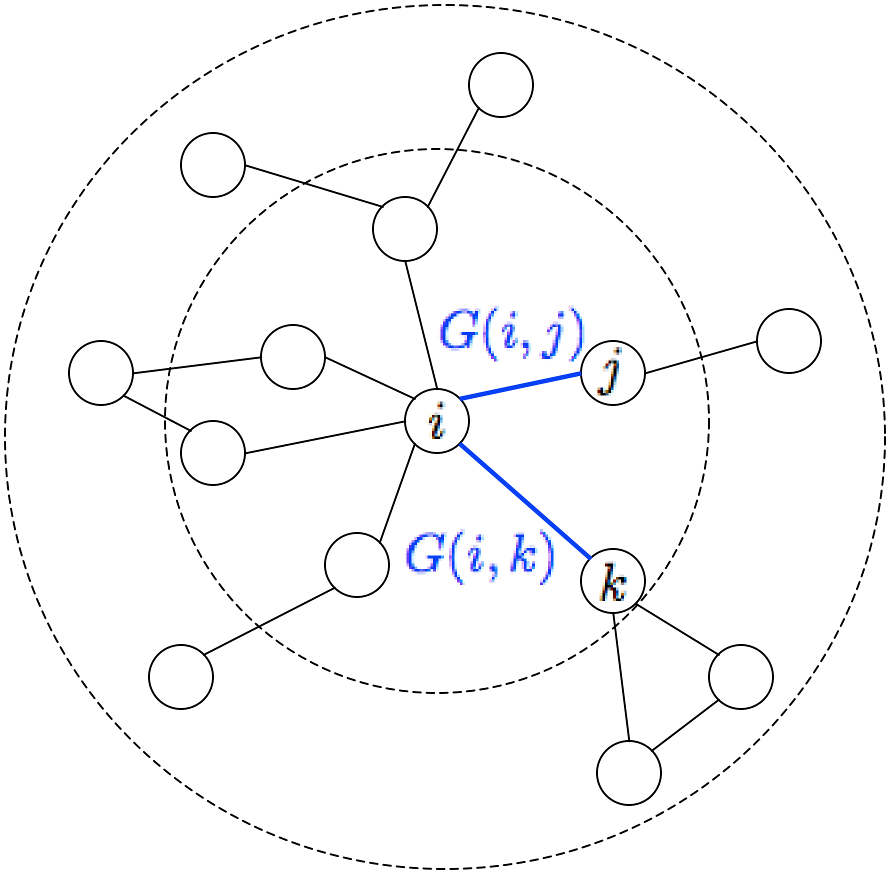
- A game with each one-hop neighbor
- Not necessarily independent
- Games themselves depend on the nodes’ interests

Games with Neighbors



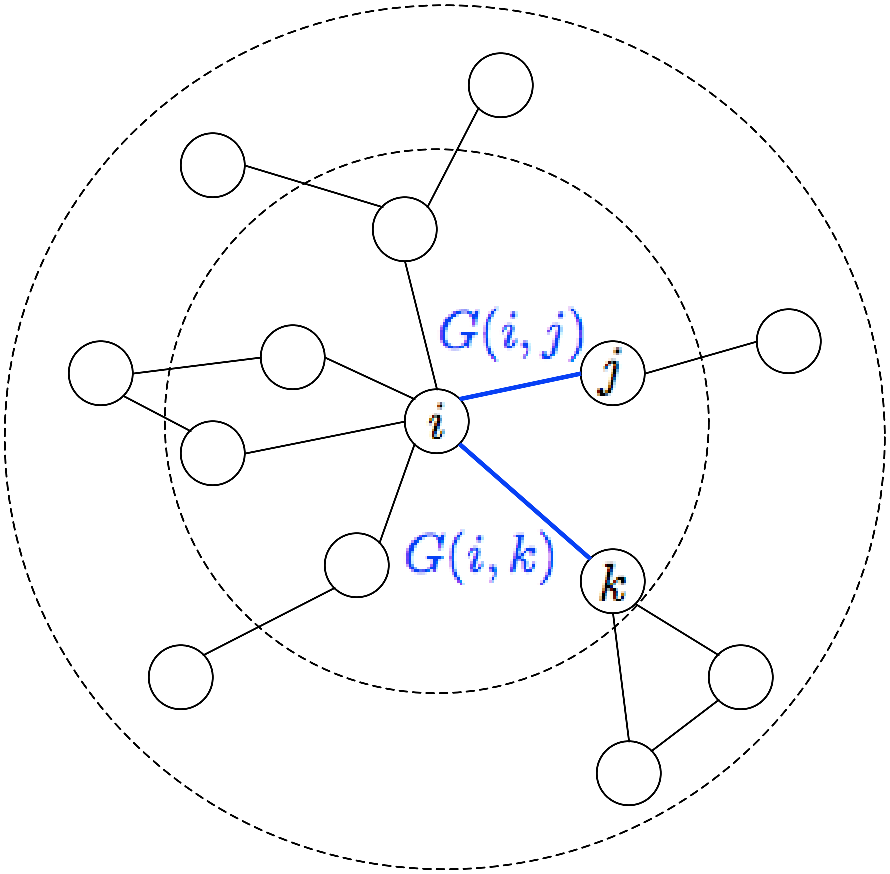
- A game with each one-hop neighbor
  - Cooperate (forward)
  - Defect (don't)
- Not necessarily independent
- Games themselves depend on the nodes' interests

Games with Neighbors



- A game with each one-hop neighbor
  - Cooperate (forward)
  - Defect (don't)
- Not necessarily independent
  - Capacity constraints: Can only cooperate so much
  - Now: infinite-capacity
- Games themselves depend on the nodes' interests

Games with Neighbors

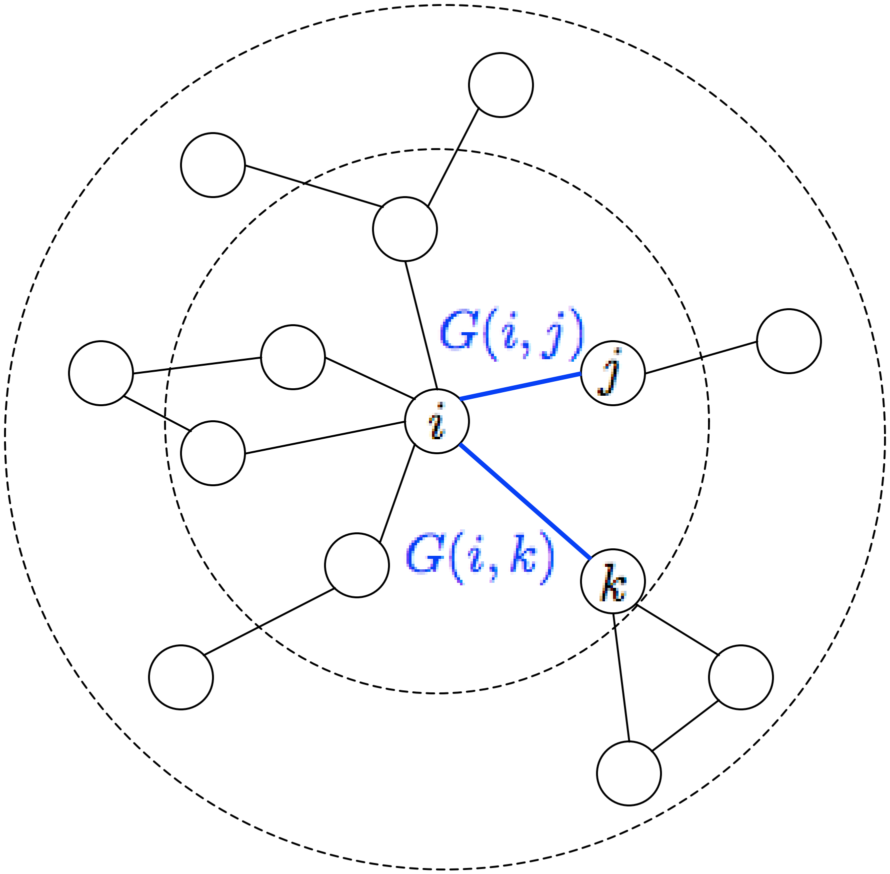


- A game with each one-hop neighbor
  - Cooperate (forward)
  - Defect (don't)
- Not necessarily independent
  - Capacity constraints: Can only cooperate so much
  - Now: infinite-capacity
  - Outcome of $G(i,j)$ = $\mathcal{U}_i^t(j)$
  - Total payout *not*

$$\sum_{j \in \mathcal{N}^1(i)} \mathcal{U}_i^t(j)$$

- Games themselves depend on the nodes' interests

Games with Neighbors

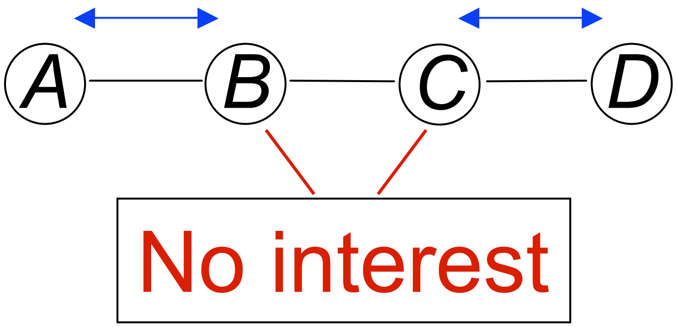


- A game with each one-hop neighbor
  - Cooperate (forward)
  - Defect (don't)
- Not necessarily independent
  - Capacity constraints: Can only cooperate so much
  - Now: infinite-capacity
  - Outcome of $G(i,j)$ = $\mathcal{U}_i^t(j)$
  - Total payout *not*

$$\sum_{j \in \mathcal{N}^1(i)} \mathcal{U}_i^t(j)$$

- Games themselves depend on the nodes' interests

Ad Hoc Routing Game

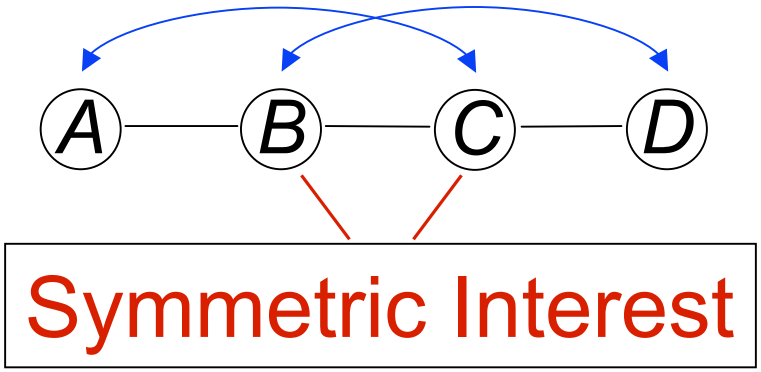


No interest

B \ *C*	~~Coop~~	Defect
~~Coop~~	~~0 , 0~~	~~0 , 0~~
Defect	~~0 , 0~~	0 , 0

- Neither *B* nor *C* will ask one another to forward
- Only one outcome: (Defect, Defect)

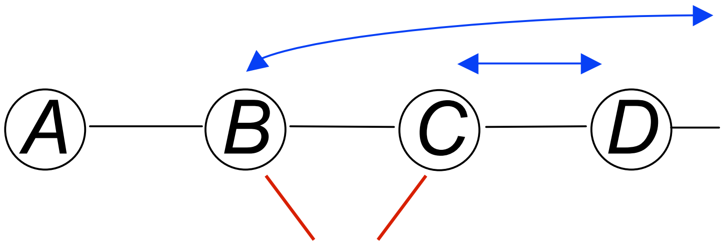
Ad Hoc Routing Game



B \ C	Coop	Defect
Coop	1 , 1	-1 , 2
Defect	2 , -1	0 , 0

- Both want each other to forward for them
- Iterated prisoner’s dilemma
- Outcome using TFT: (Coop, Coop)

Ad Hoc Routing Game



Asymmetric Interest

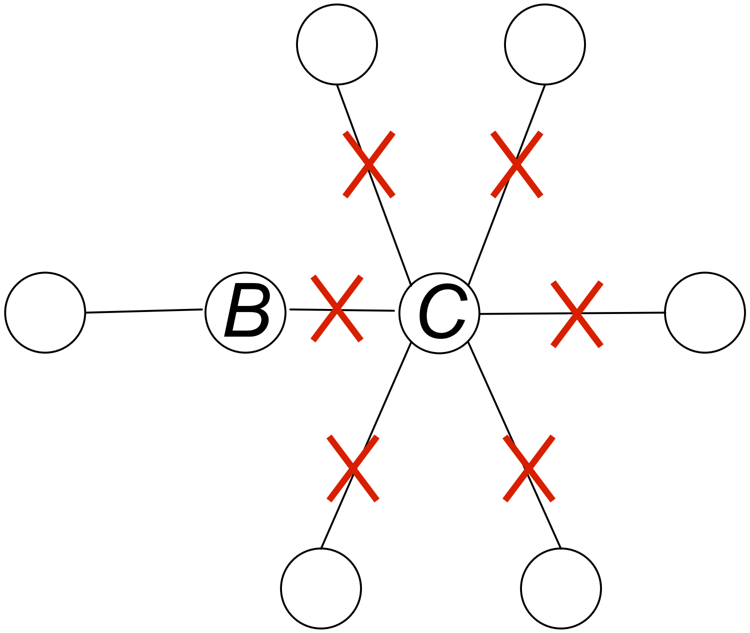
B \ *C*	Coop	Defect
Coop	2 , -1	0 , 0
Defect	2 , -1	0 , 0

- *B* wants *C* to forward
- *C* does not need *B*

- *C*'s dominant strategy: Defect

This is what we're trying to solve

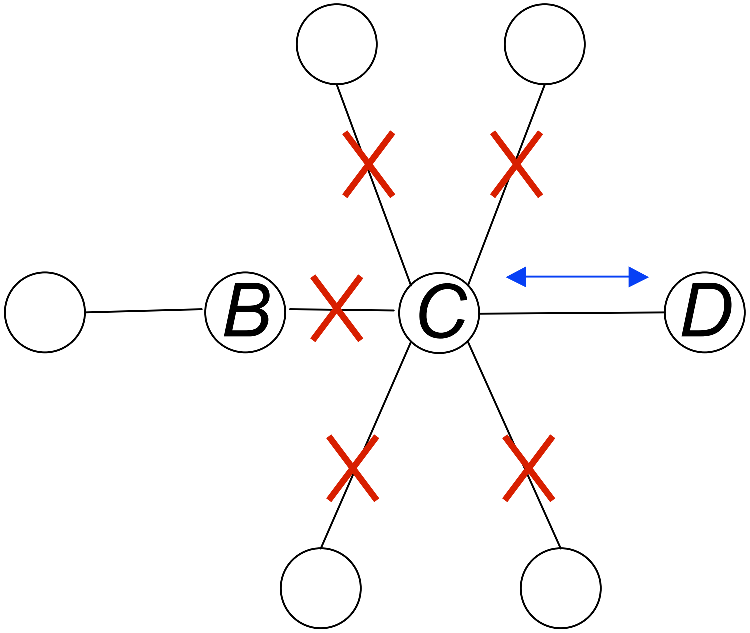
Punishing via Isolation



C's utility at most 0

- *C* does not forward for *B*
- *B* detects this
  - Watchdog [Marti et al]
- And gets *C* isolated
- Requests all $i \in N^1(C)$ to play Defect in $G(i, C)$
  - Catch [Mahajan et al]

Does Isolation Always Work?



- *B* tries to get *C* isolated
- *D* has a connection to *C*
- *Why would D isolate?*
  - And cut off its own connection?
- To avoid punishment?
  - Hide their connection

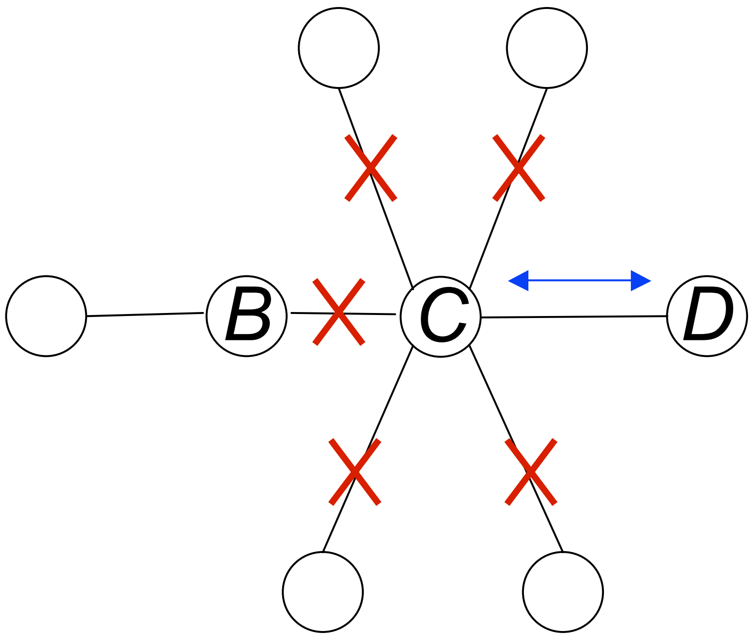
C's utility not affected

Ensuring Cooperation

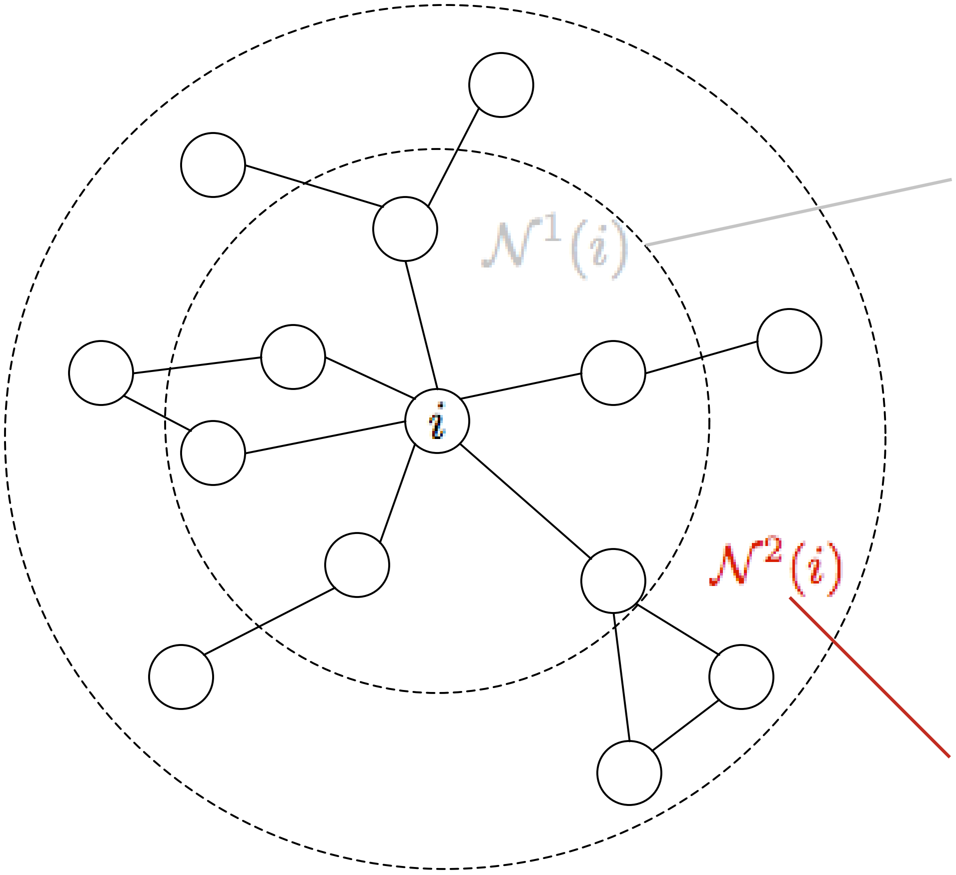
- Why didn’t isolation work?
  - Collusion!

- Would any other isolation system work?

- *B* needs a way to break the *C* - *D* connection



Recall



Transmission range

All 1-hop nbrs hear i's transmissions

Carrier Sense Range

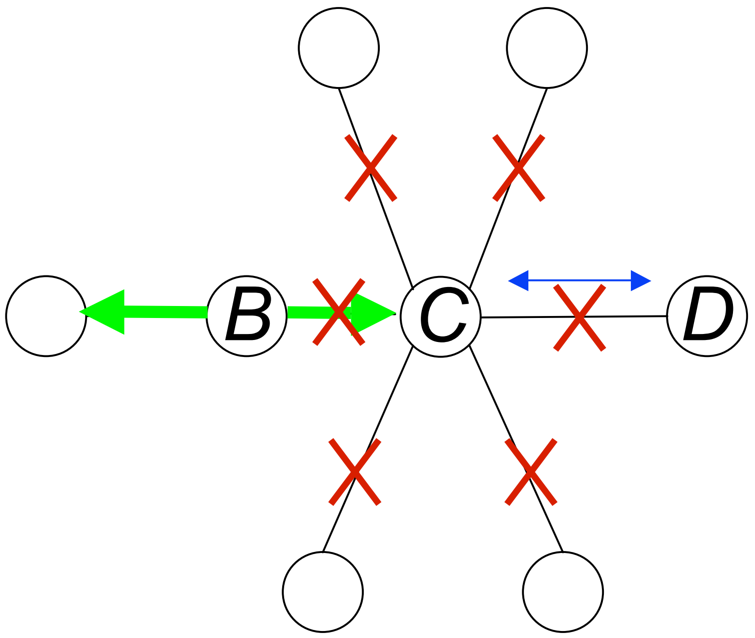
No one within 2 hops can send/recv while i is transmitting

Assured Punishment

- While *B* is sending
  - *C* cannot send or recv
  - The *C* - *D* connection is broken

- If *B* has nothing to send?
  - Send garbage
  - Jamming

- *C* can collude all he wants



B can always punish *C*

Node Preferences w/ Jamming

- Connectivity is best

  Utility C
- Disconnected is bad

  $C > D$
- Forwarding costs more

  $C > D > F < 0$
- But connectivity overcomes

  $C + F > D$
- Costs more to Jam than Fwd

  $0 > F > J$

Nodes can be disconnected and forward

w.l.o.g.:

$C = 2$

$D = 0$

$F = -1$

$J = -2$

Payoffs with Jamming

- Recall: Games $G(i,j)$ and $G(i,k)$ not independent
  - Jamming affects all nearby games

- If node i is playing Cooperate for $f_i(t)$ neighbors:

$$U_i(t) \overset{\text{def}}{=} \begin{cases} -2 & \text{if } i \text{ is Jamming} \\ \boxed{-f_i(t) \qquad \exists j \in \mathcal{N}^2(i) \text{ Jamming}} & \\ \boxed{\sum_{j \in \mathcal{N}^1(i)} \mathcal{U}_i^t(j) \qquad \text{otherwise}} & \end{cases}$$

≤ 0

Normal payoffs

Is Jamming Viable?

- When *B* Jams:
  - The best *C* can do is stop forwarding

$$U_C(t) = -f_C(t) = 0$$

  - *C*'s minmax payoff

$$U_i(t) \overset{\text{def}}{=} \begin{cases} -2 & \text{if } i \text{ is Jamming} \\ \boxed{-f_i(t) \qquad \exists j \in \mathcal{N}^2(i) \ \text{Jamming}} \;\rule[0.5ex]{3em}{0.4pt}\; \le 0 & \\ \sum_{j \in \mathcal{N}^1(i)} \mathcal{U}_i^t(j) & \text{otherwise} \end{cases}$$

Is Jamming Viable?

- When *B* Jams:
  - The best *C* can do is stop forwarding

$$U_C(t) = -f_C(t) = 0$$

  - *C*'s minmax payoff

- Folk theorem: There exist SPNE with
  - Payoffs greater than disconnectedness
  - Jamming used as punishment

Jamming *Is* Viable

- It works
- In fact:
  - When there is no packet loss (noise) Every participant will always forward
- What happens when there is noise?
  - Free-riding vs. Packet loss

Modeling Noise

Depend on interest

- Packets dropped with probability *p*
- Expected gain from my neighbor's Cooperation: $(1-p)U_c + pU_d$
- Expected gain from my neighbor's Defection: U_d

Symmetric Game

B \ *C*	Coop	Defect
Coop	1-2*p*, 1-2*p*	-1 , 2-2*p*
Defect	2-2*p* , -1	0 , 0

Asymmetric Game

B \ *C*	Coop	Defect
Coop	2-2*p* , -1	0 , 0
Defect	2-2*p* , -1	0 , 0

Modeling Noise

- For any $p < 1$
  - Minmax payoff from (Jam, Defect)
  - Jamming will yield a SPNE

Symmetric Game

B \ *C*	Coop	Defect
Coop	$1-2p$, $1-2p$	-1 , $2-2p$
Defect	$2-2p$, -1	0 , 0

Asymmetric Game

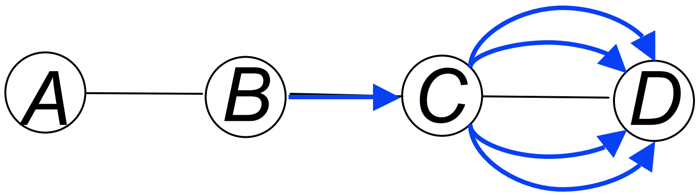
B \ *C*	Coop	Defect
Coop	$2-2p$, -1	0 , 0
Defect	$2-2p$, -1	0 , 0

Behaviors with Noise

- Greedy nodes may try to free-ride
  - – Looks like packet loss
- But what about risk-averse nodes?
  - – May try to avoid even packet loss

“Forward Error Correction”

- Nodes may forward a given packet more than once
  - To make sure their neighbors see it



Probability *B* sees Defection = p^4

- Will pay the cost multiple times
  - If the punishment is strong enough

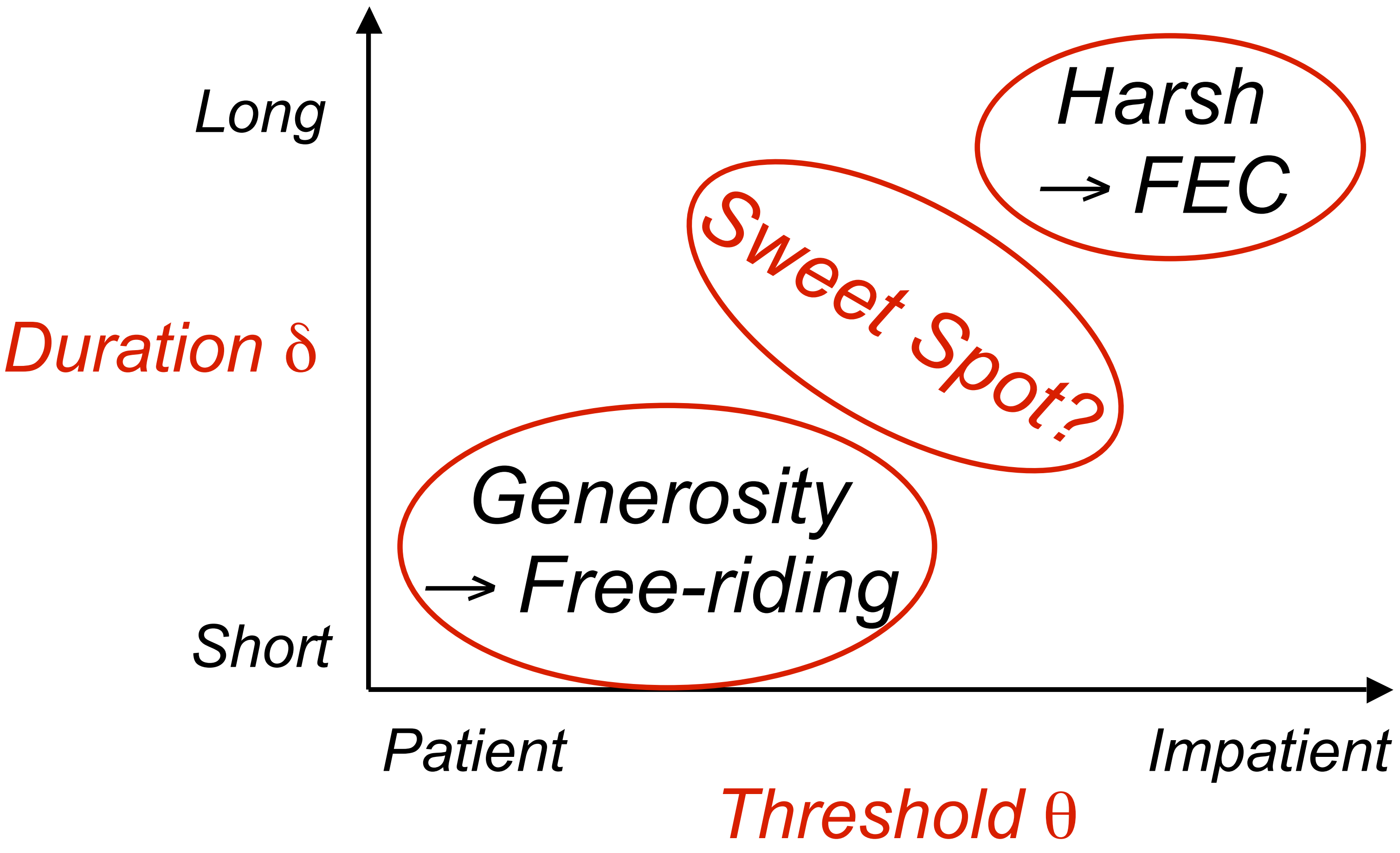
Punishment Strength

- When to punish?
  - Each node uses a watchdog
  - Punish once they forward less than threshold θ

- How hard of a punishment?
  - Punish for duration δ

- Strength of punishment: δ / θ

Emergent Behaviors

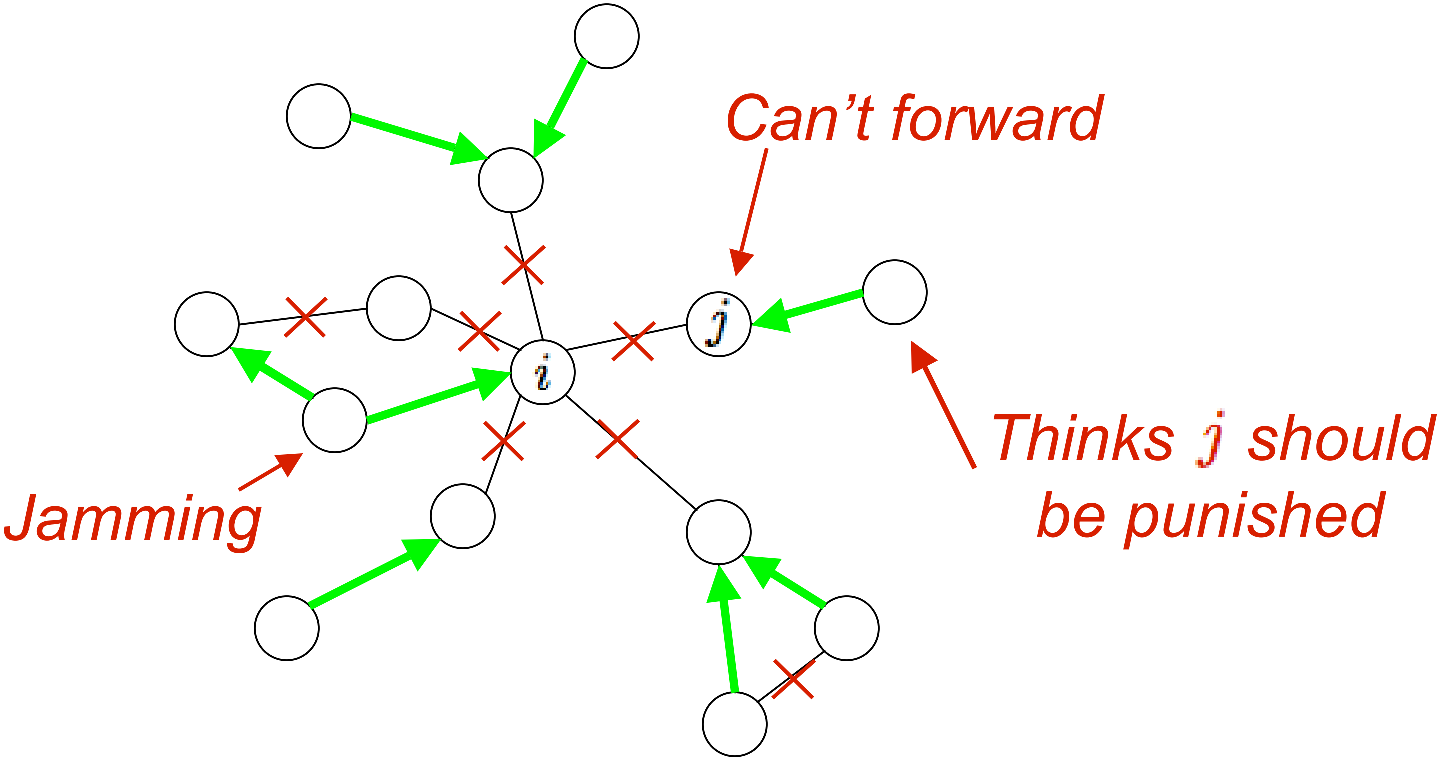


Jamming Strategy: Guidelines

- Strategy should be adaptive
  - Adjust to noise levels
  - Don’t punish nodes for not doing the impossible
- Nodes may have to share info
  - To better understand capacity
  - How to do this truthfully?
- Deal with punishment echoes

Punishment Echoes

- Jamming can echo



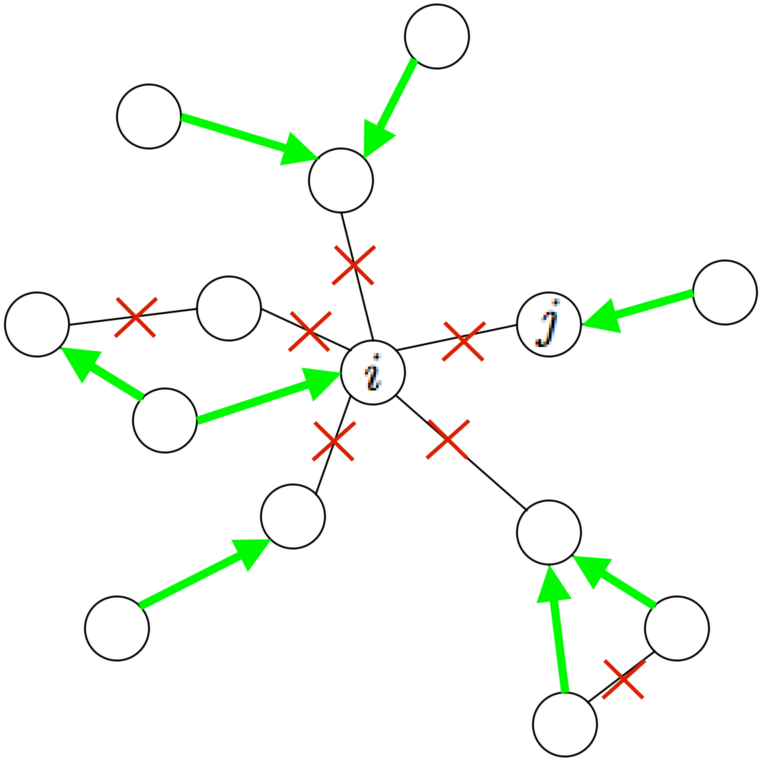
Can't forward

Thinks j should be punished

Jamming

Punishment Echoes

- Jamming can echo



Mutually assured destruction

Potential solution 1:

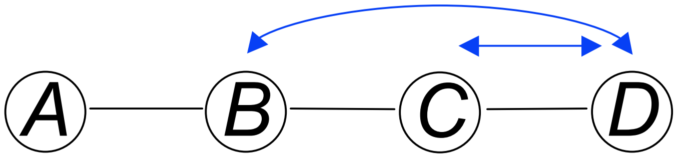
Punish with some small probability

Potential solution 2:

Tell nbrs you are being jammed

Is Jamming Worth Doing?

- Addresses the Asymmetric case



- Should we always have cooperation?
  - Should nodes be allowed to opt out?

- It works but is it worth the loss in
  - Efficiency?
  - Network lifetime?

Goal: minimize these

Conclusion

- Internal incentives easier to deploy
  - No trusted components
- Isolation does not always work
  - “Collusion” happens
- Jamming works
  - SPNE
- But not without issues
  - Echoes, efficiency loss…
- Question:

Are there easy-to-deploy, viable solutions?

Thanks

- Seungjoon Lee
- Vijay Gopalakrishnan
- Bobby Bhattacharjee