Wireless Network Security
Spring 2016

Patrick Tague

Class #9 – MAC Misbehavior

Reminder: Assignments

- Assignment #2 is due today
  - 11:59pm PST
- Assignment #3 posted today, due March 3
  - It's based on today's material

Class #9

- IEEE 802.11 MAC layer
- Misbehavior in 802.11 MAC
- A few other MAC threats (time permitting)
- OMNET++ Tutorial II → next week

IEEE 802.11

- Infrastructure mode
  - Many stations share an AP connected to Internet
    - Distributed coordination function (DCF)
    - Point control functions (PCF)
      - Rarely used due to inefficiency, vague standard specification, and lack of interoperability support
- Ad hoc mode
  - Multi-hop, no infrastructure, no Internet
  - Never really picked up commercially
- Mesh mode (using 802.11s)
- WiFi Direct

802.11 MAC

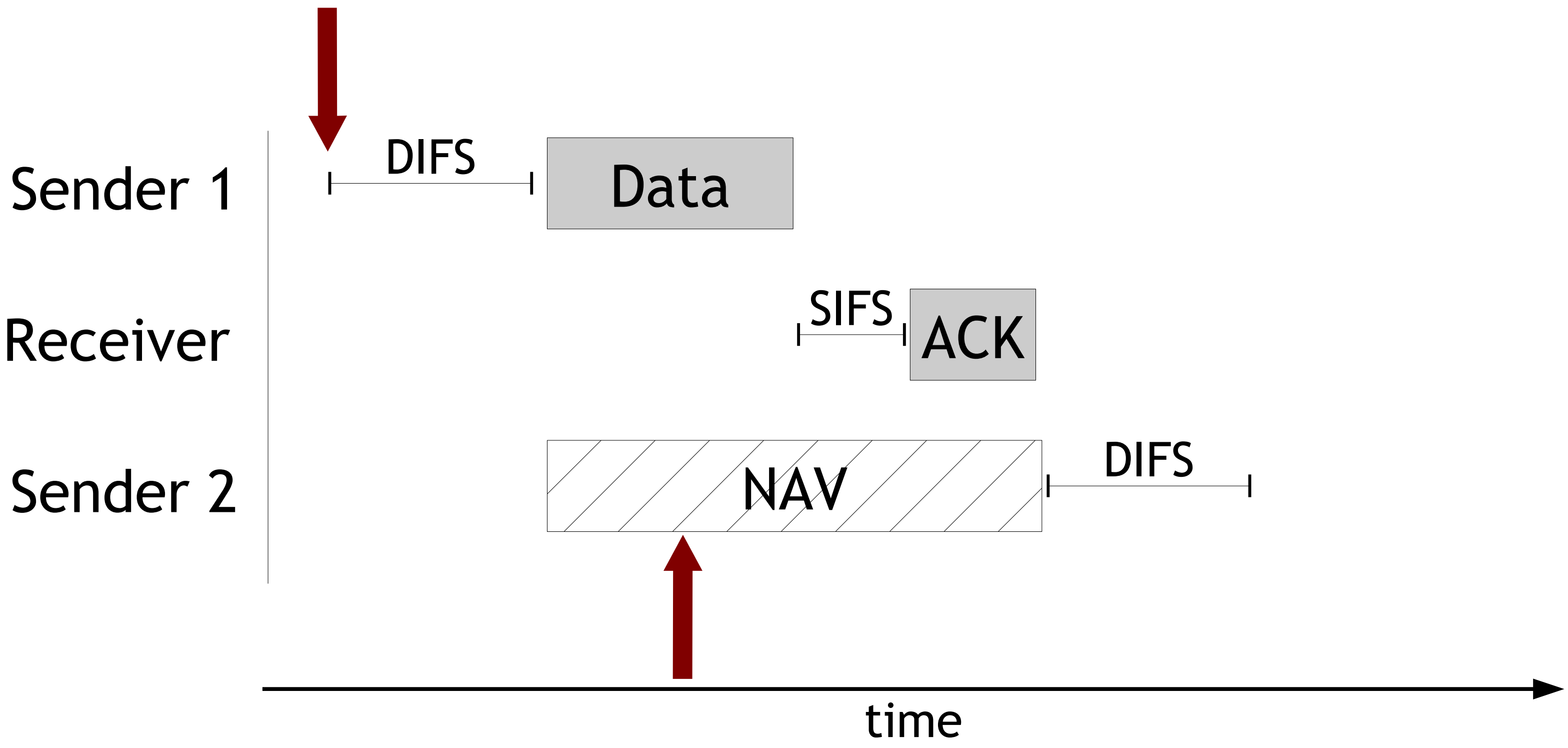
- Responsibilities of the MAC layer
  - Logical responsibilities
    - Addressing
    - Fragmentation
    - Error detection, correction, and management
  - **Timing responsibilities**
    - Channel management
    - Link flow control
    - Collision avoidance

- Today, we focus on timing-based vulnerabilities

CSMA

- Carrier Sense Multiple Access
  - Listen to the channel before transmitting
  - If channel is quiet, transmit
    - After a short delay (DIFS = DCF Inter-Frame Spacing)
  - If channel is busy:
    - Wait until it's quiet for a DIFS period
    - Wait for random backoff period
    - Send if still quiet
  - Wait for ACK or retransmit using random backoff

DCF Operation using CSMA

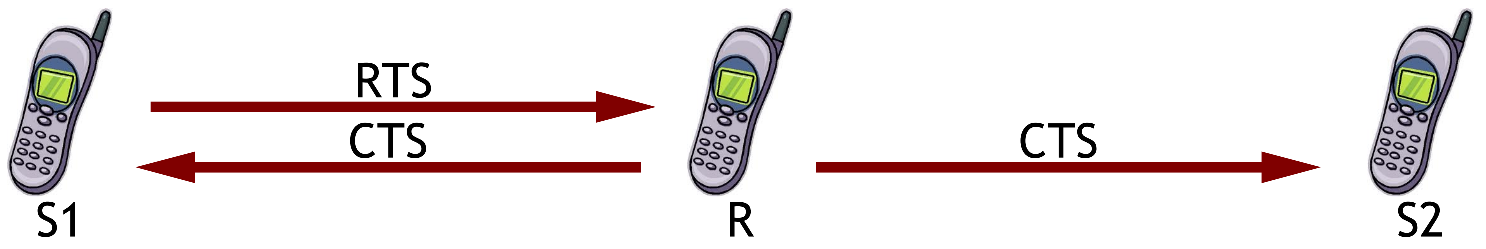


Random Backoff

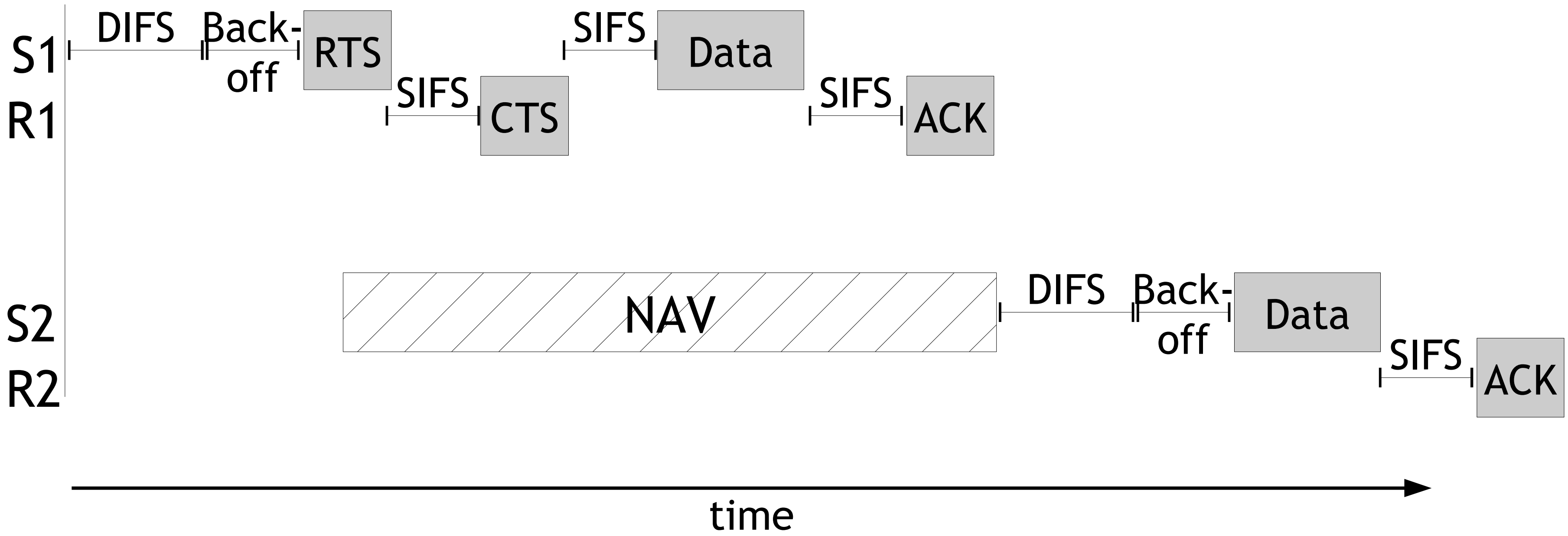
- Reduce the chance of collisions
  - Each device must wait a random duration depending on past contention – use "contention window" CW
  - If medium is busy:
    - Wait for DIFS period
    - Set backoff counter randomly in CW
    - Transmit after counter time expires
  - After failed retransmissions:
    - Increase CW exponentially
    - 2^n-1 from CW_{min} to CW_{max}, e.g., 7 → 15 → 31

Collision Avoidance

- Attempt to make channel reservation to avoid collisions by other senders
  - Request to Send (RTS)
    - Before transmitting data, sender transmits RTS
  - Clear to Send (CTS)
    - Receiver transmits CTS to tell sender to proceed
  - RTS and CTS use short IFS (SIFS < DIFS) to give priority over data packets



RTS/CTS Usage



- RTS/CTS is not required
  - S1-R1 use RTS/CTS, S2-R2 do not

MAC Layer Misbehavior

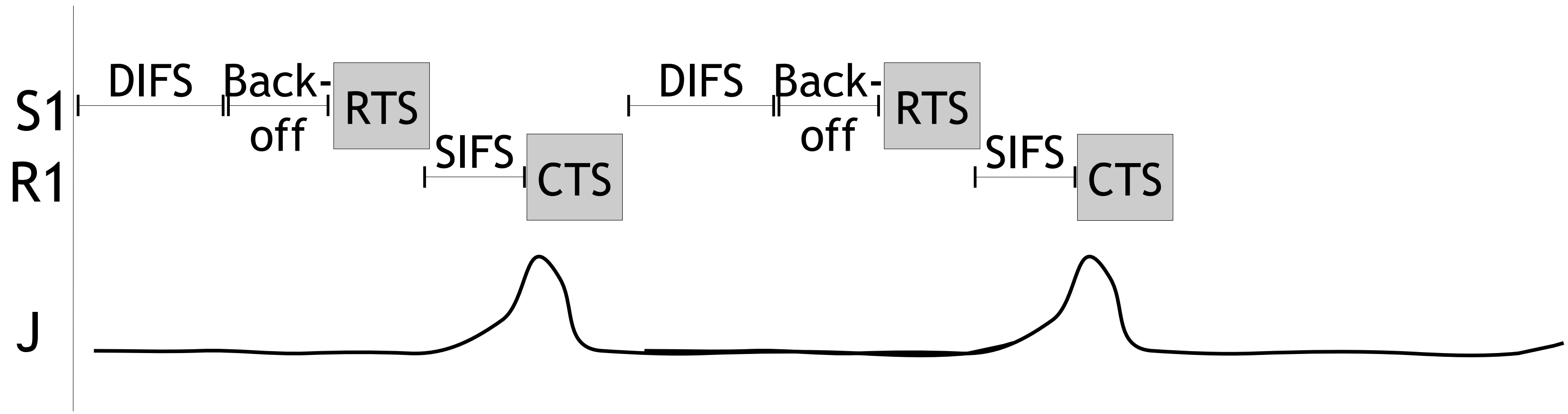
- 802.11 DCF works well under the assumption that everyone plays nicely together
  - This may have been a reasonable assumption when MAC protocols were hardware-bound

- However, selfish and malicious nodes are free to arbitrarily break the rules
  - Software MAC makes this very easy to do

What are some of the different ways to misbehave at the MAC layer?

MAC Jamming

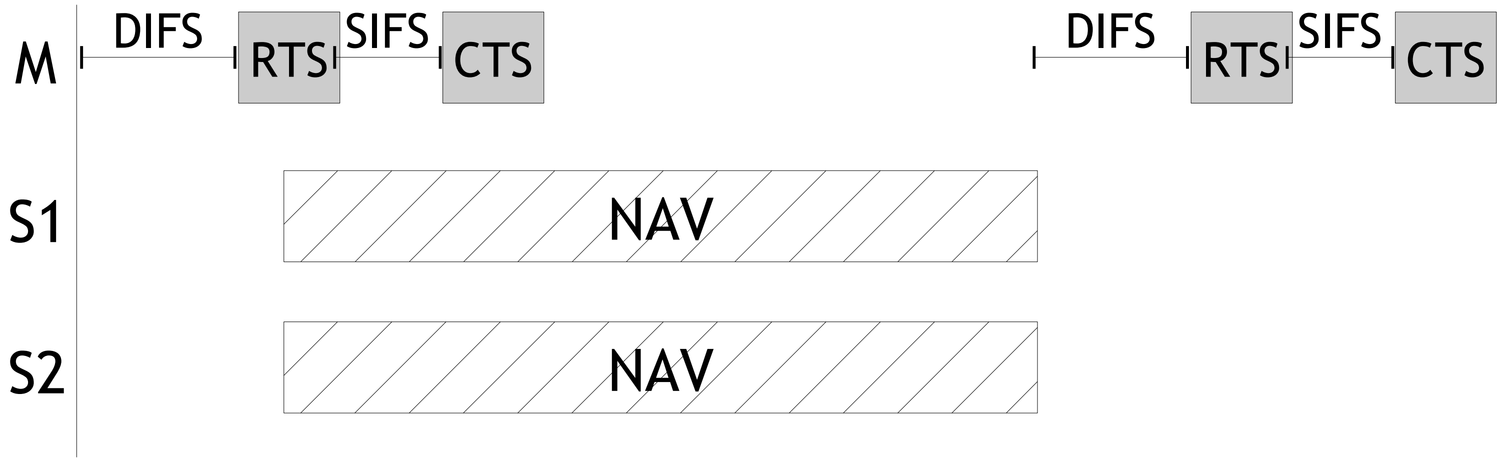
- DCF structure and behavior gives advantages to jamming attackers
  - Jamming after RTS (and SIFS period) blocks CTS (prevents data flow) and occupies channel (prevents other senders from using it)
    - Low duty-cycle attack → order-of-magnitude efficiency gain



Carnegie Mellon University

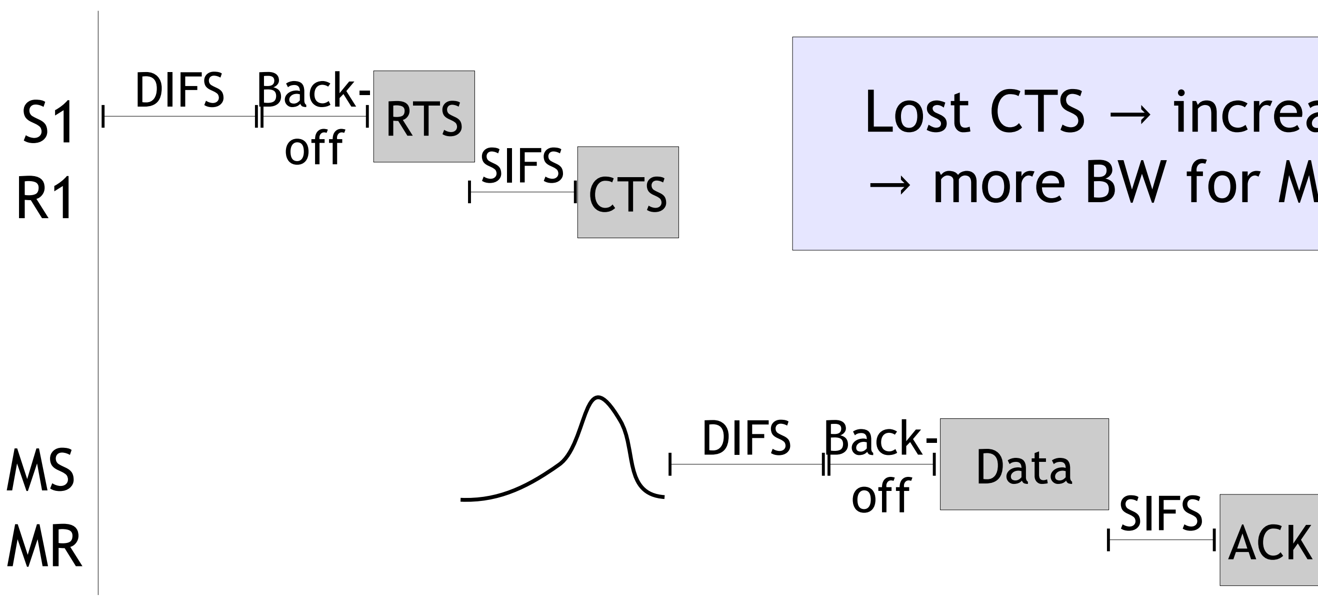
MAC Blocking

- DCF structure and behavior gives advantages to other DoS attackers
  - RTS/CTS “flooding” - repeated sending of RTS/CTS exchanges while other senders obey the rules



MAC Greed w/ Jamming

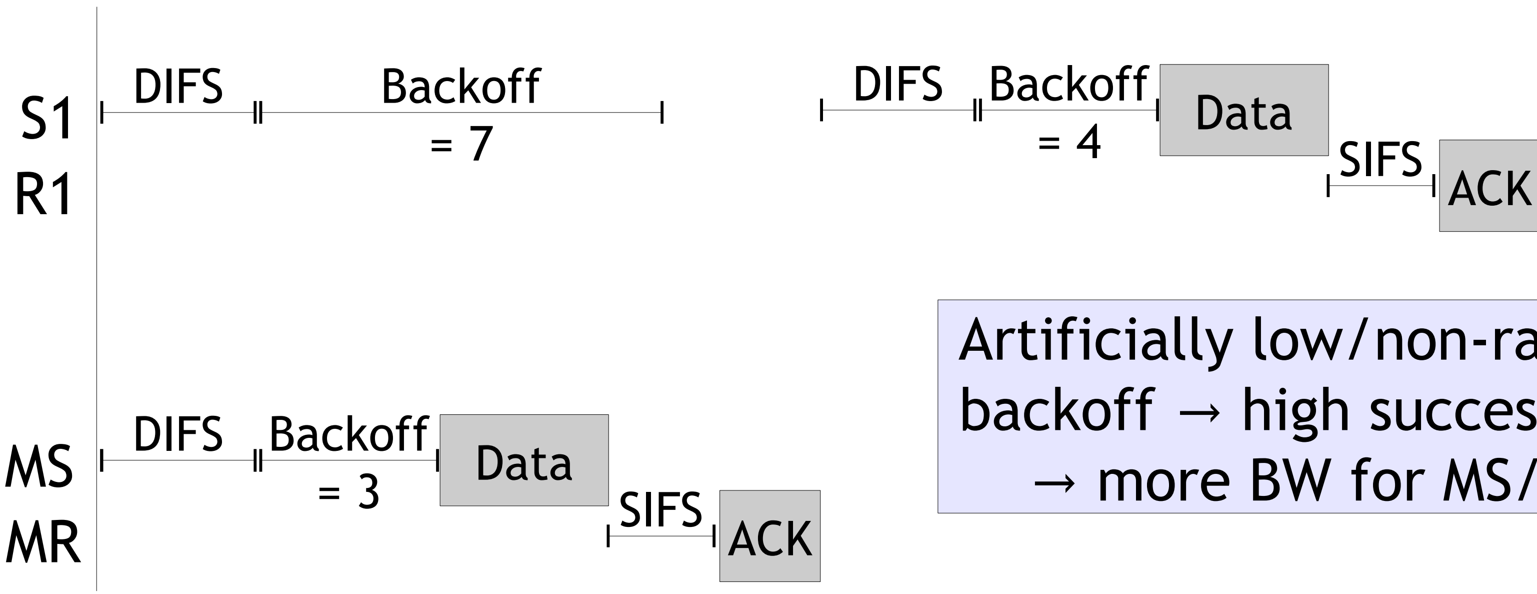
- Greedy/malicious sources can block or collide with other sources, causing their sending rates to decrease
  - Gives more opportunity to greedy source



Lost CTS → increase CW → more BW for MS/MR

MAC Greed w/ Parameters

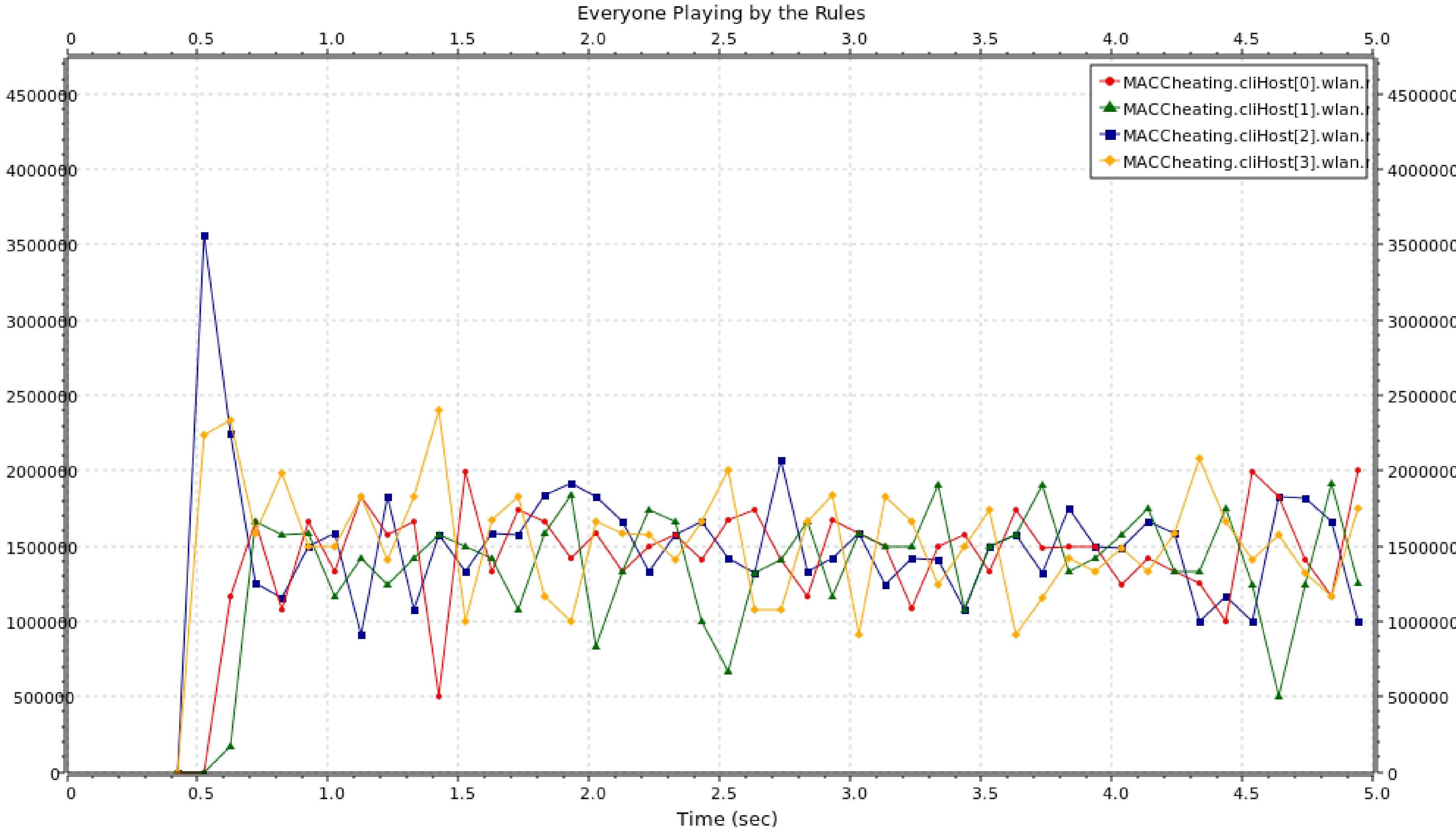
- Greedy/malicious sources can manipulate protocol parameters for unfair resource usage



Artificially low/non-random backoff → high success rate → more BW for MS/MR

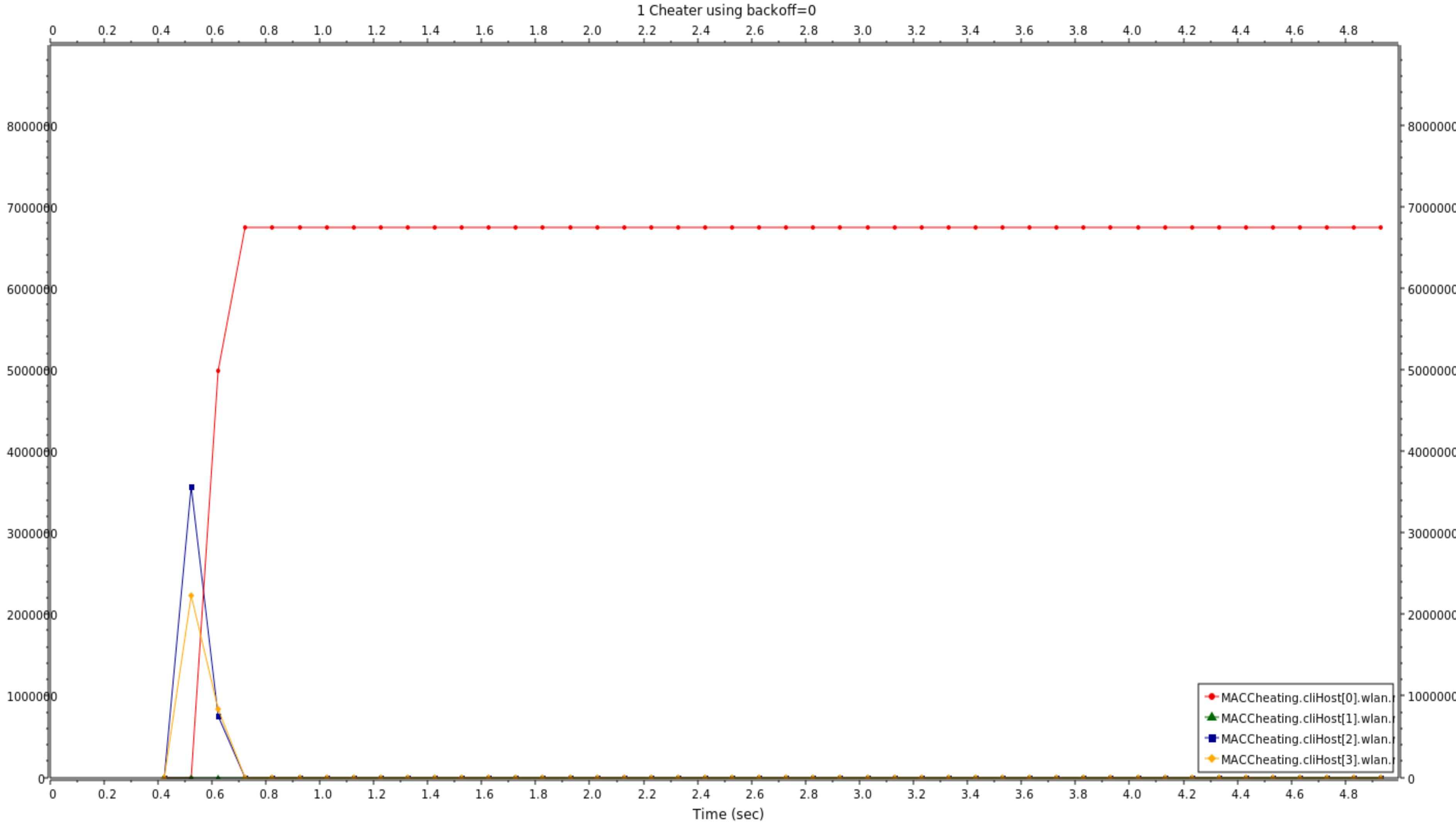
Example

- 4 clients, all cooperating (using OMNET++)



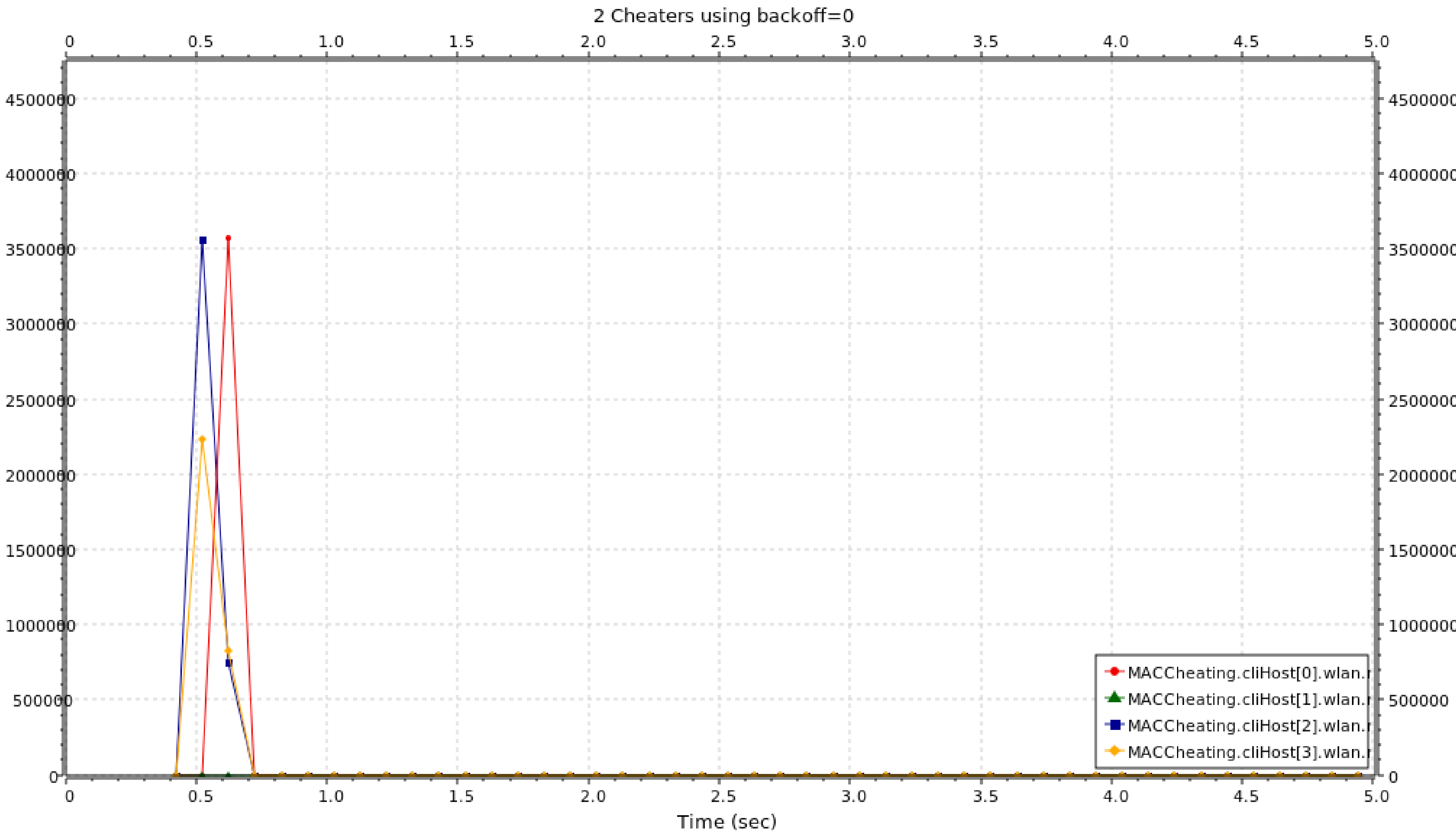
Example

- 4 clients, 1 using backoff = 0



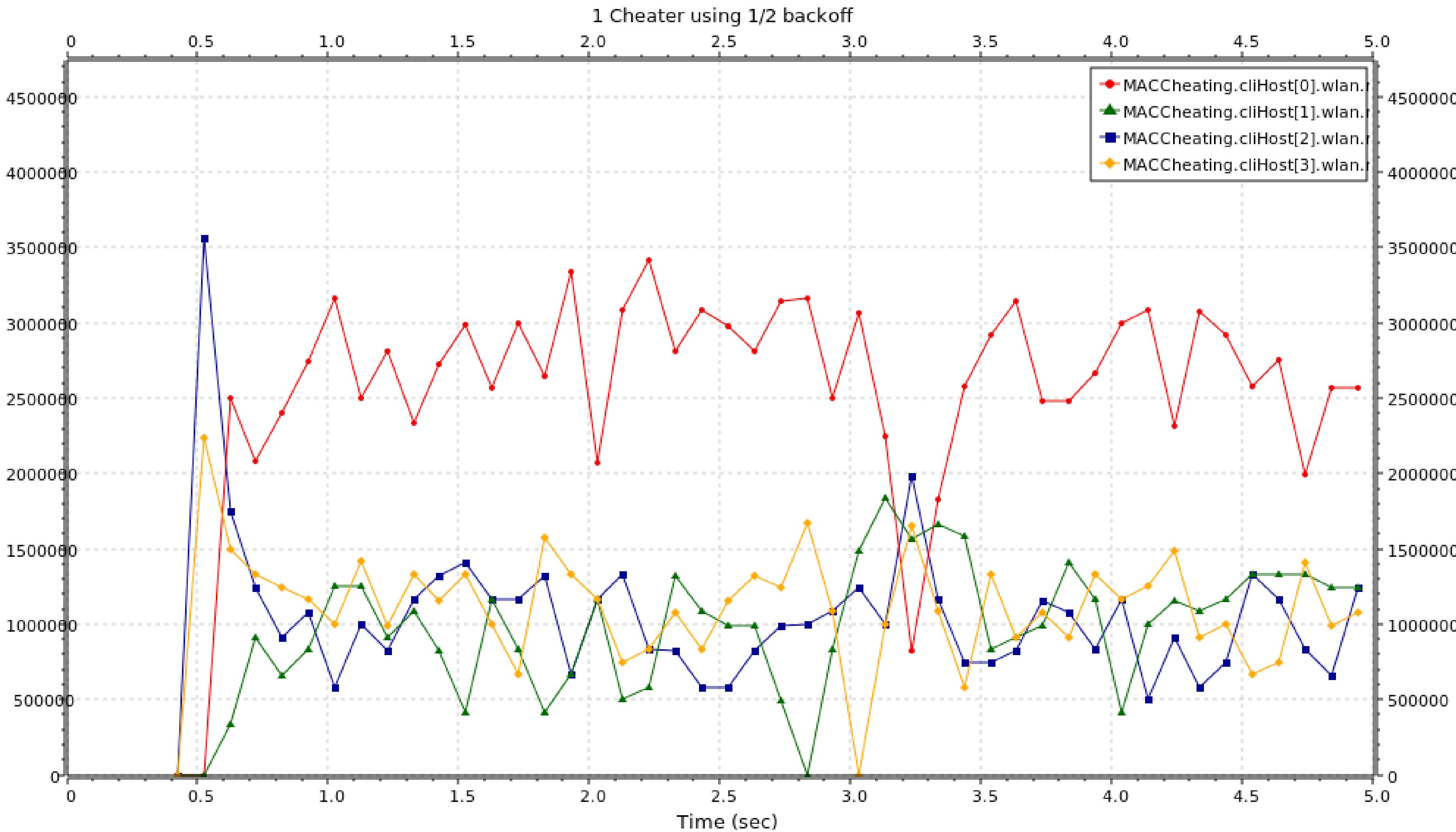
Example

- 4 clients, 2 using backoff = 0



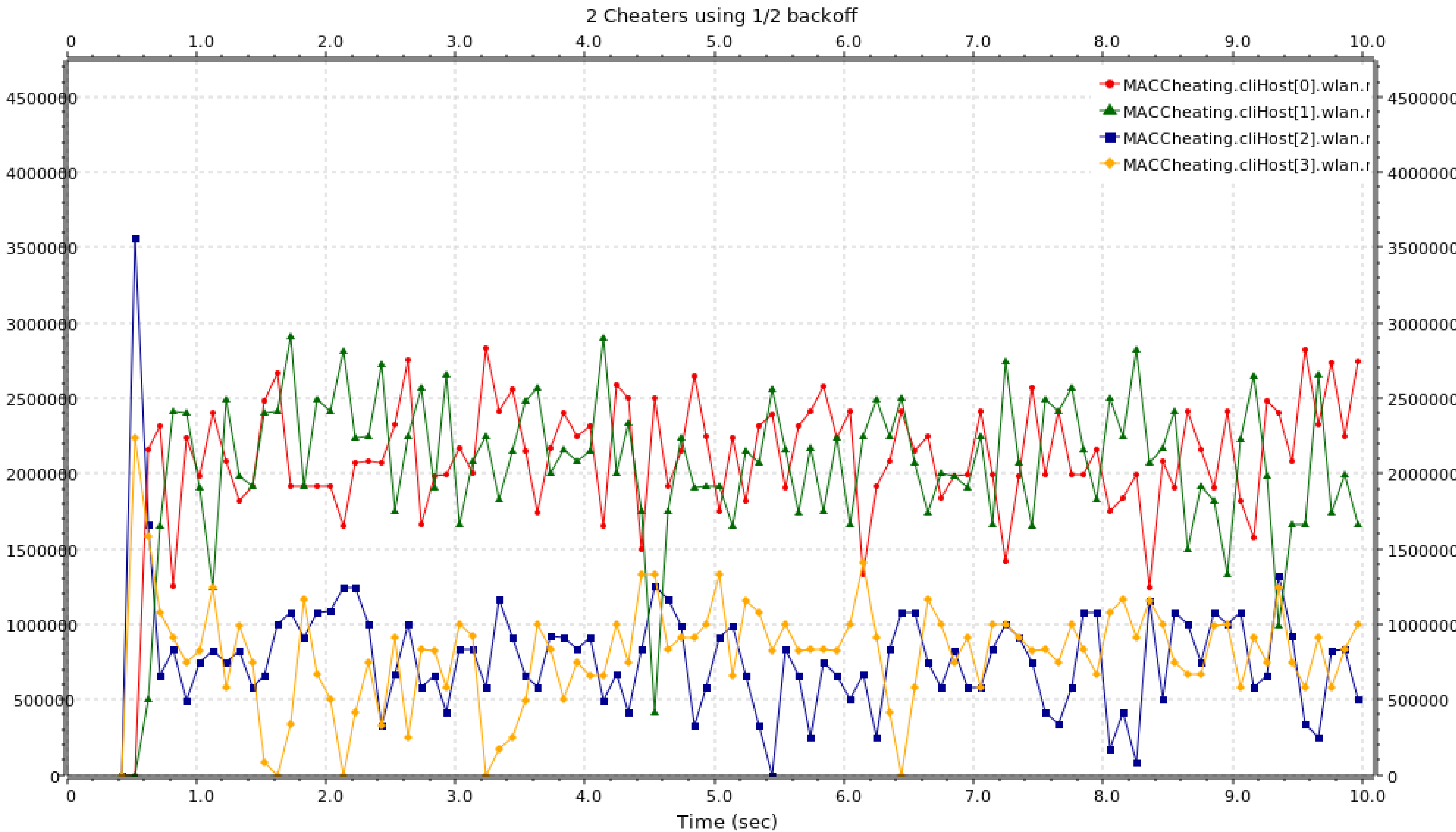
Example

- 4 clients, 1 using backoff / 2



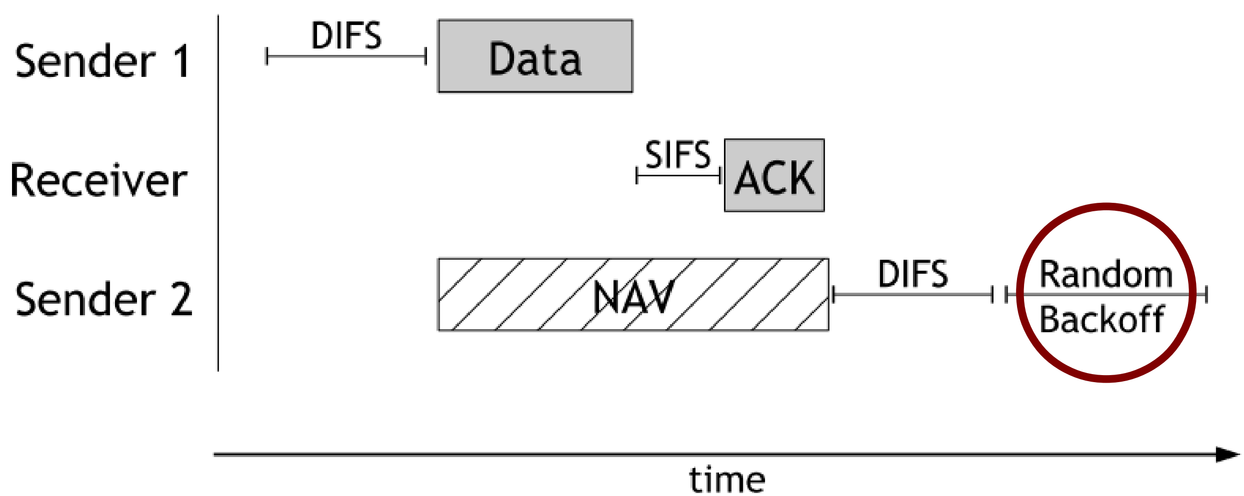
Example

- 4 clients, 2 using backoff / 2



Cheating in CSMA/CA
[Čagalj et al., 2004]

- "CSMA/CA was designed with the assumption that the nodes would play by the rules"
  - MAC cheaters deliberately fail to follow the IEEE 802.11 protocol, in particular in terms of the contention window size and backoff



System Game Model

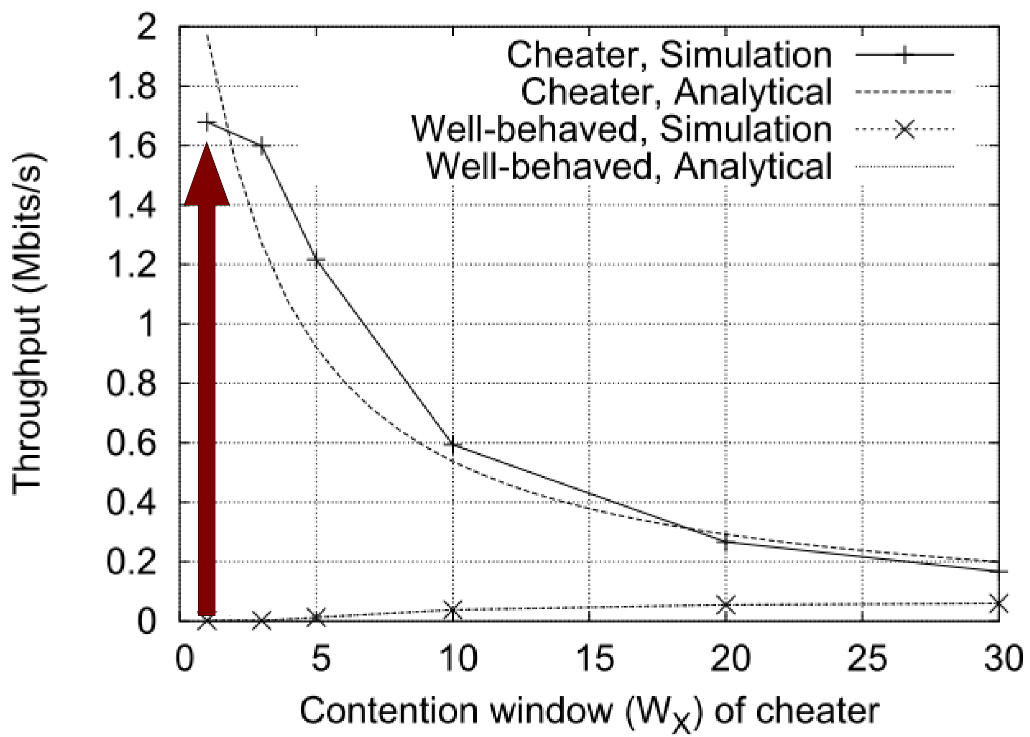
- N tx-rx pairs in a single collision domain, using 802.11, C of N are cheaters with control of MAC layer parameters
- Cheaters want to maximize avg. throughput r_i
- As a game:
  - Each player (cheater) adjusts its contention window size W_i to maximize utility $U_i = r_i$
  - Players react to changes of remaining N-C users who play by the rules
- Authors analyze relationships between throughput and contention window sizes

Single Static Cheater

- **First case**: a single cheater with a fixed strategy (i.e. makes a decision and sticks with it)

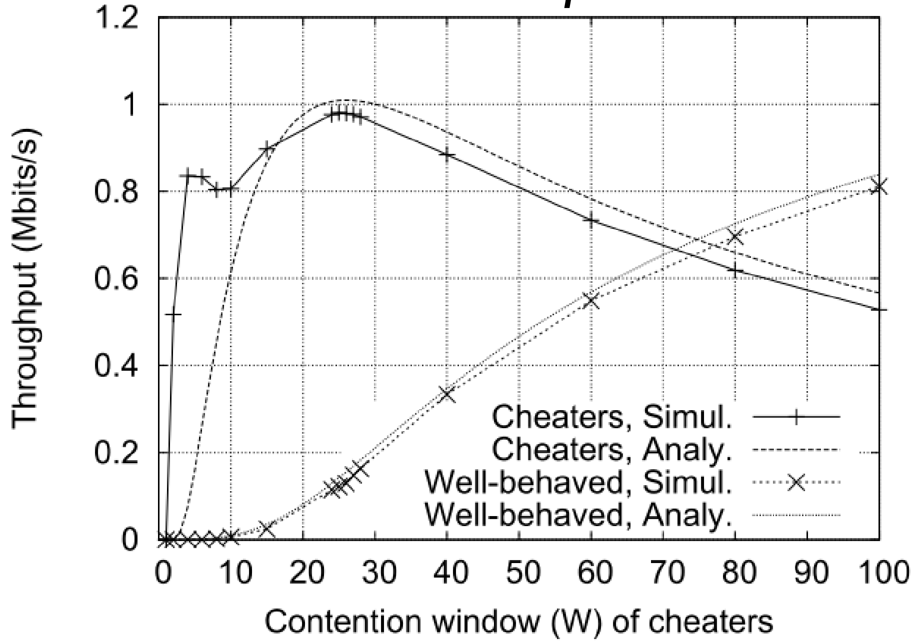
- A single cheater gets best throughput at $W_i = 1$

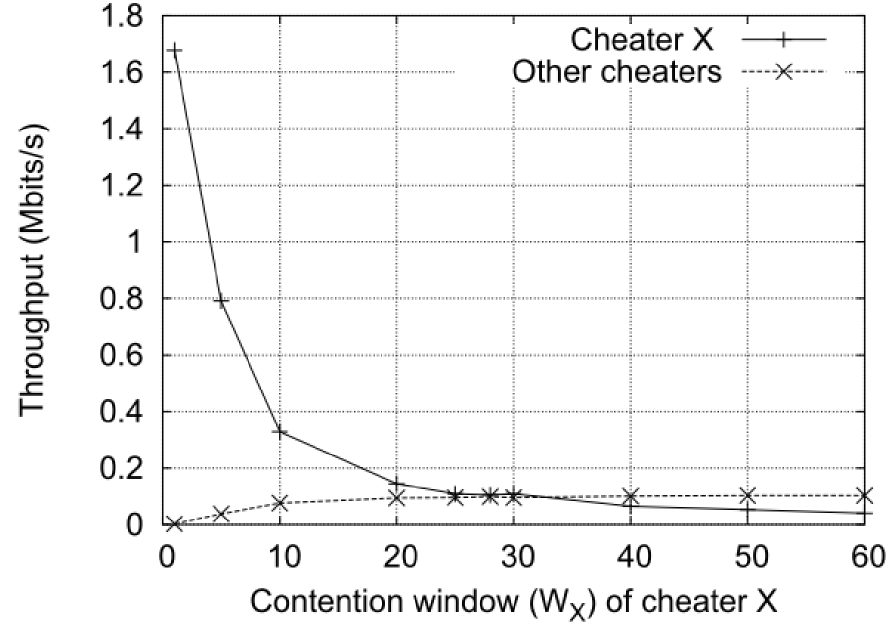
- In fact, $W_i = 1$ is the Nash Equilibrium for the static game with $C = 1$



Multiple Static Cheaters

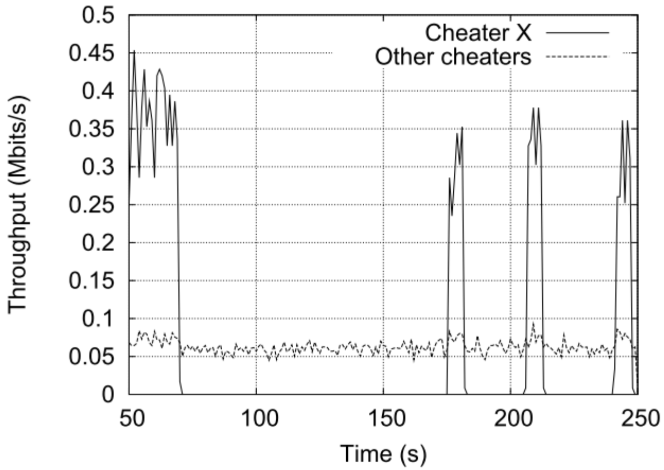
- **Second case:** many cheaters with fixed strategy
  - 2.1 Cheaters don't know about each other
  - 2.2 Cheaters are aware of cheater v. cheater competition in forming strategies
- Window size $W_i=1$ is no longer optimal

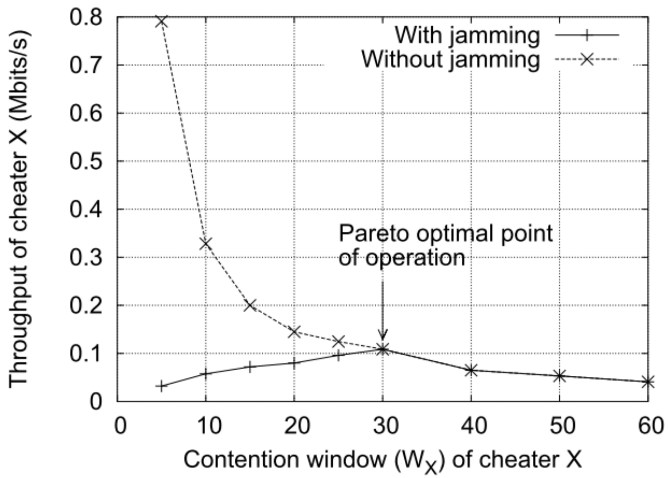




Dynamic Cheating Game

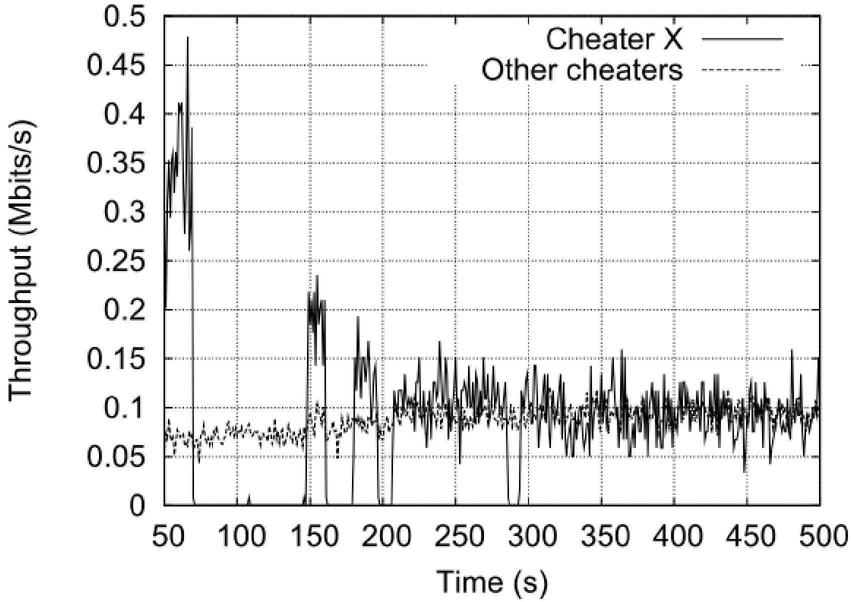
- In the dynamic game, cheaters can change their strategy in response to other players (including other cheaters)
  - A penalty is enforced on the utility function, so cheaters converge to the optimal operating point
  - "Cooperative cheaters" can inflict the penalty on "non-cooperative cheaters" by jamming their packets

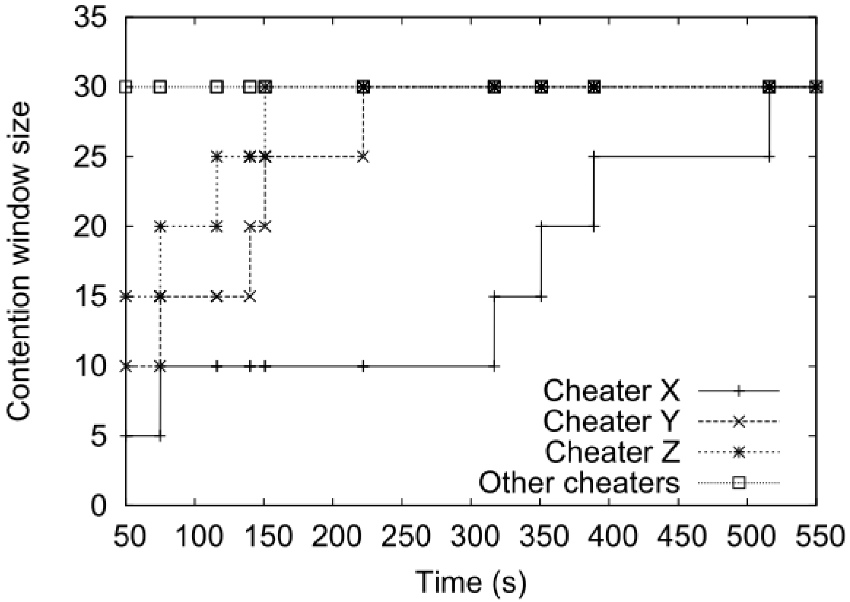




Distributed/Adaptive Cheating

- Cheaters can observe actual throughput and jamming to adapt contention window size
  - Cheaters are forced to cooperate or get lower throughput due to penalization from other cheaters

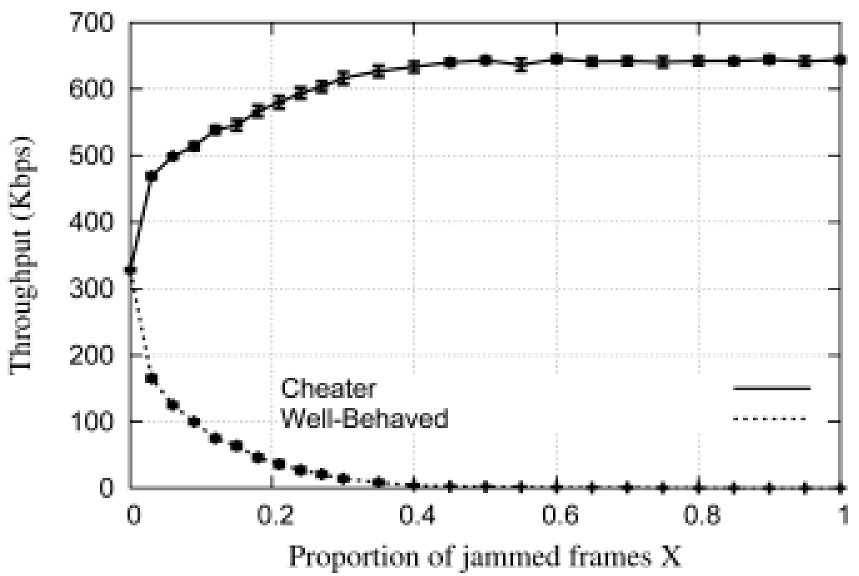


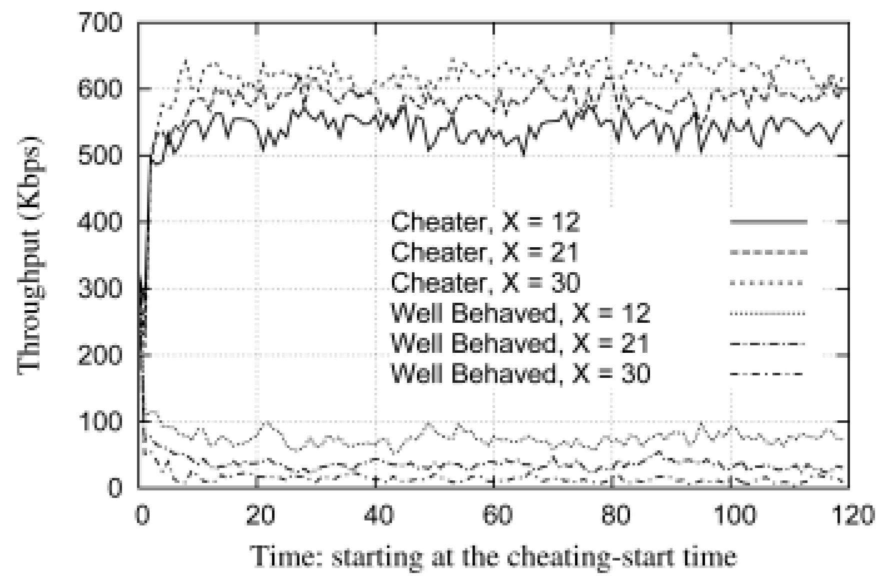


Detecting Greedy Behavior

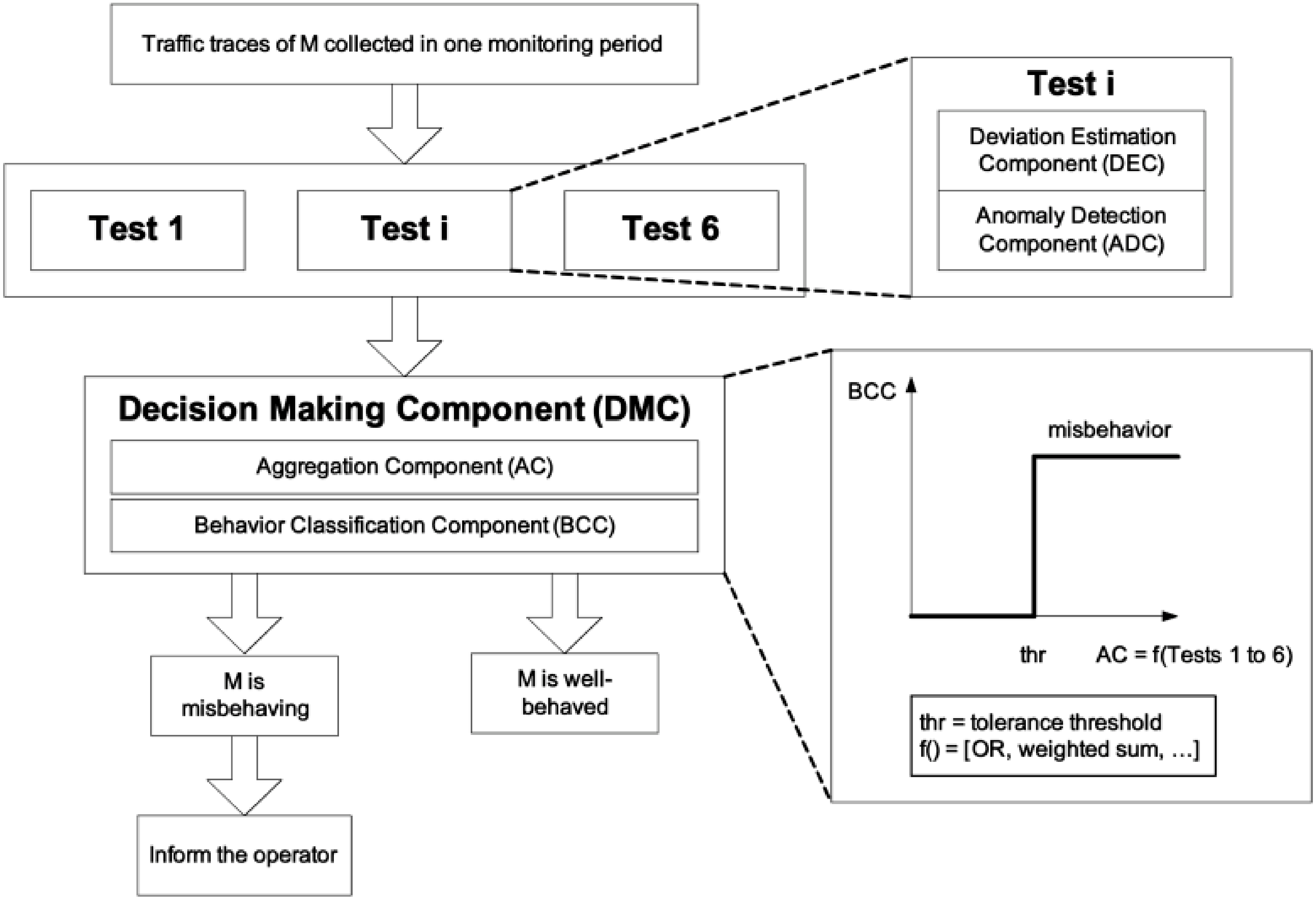
[Raya et al., 2006]

- Detection Of greedy behavior in the Mac layer of Ieee 802.11 public NetwOrks (DOMINO)
  - Software installed at/near the access point that can detect and identify greedy players
  - No changes to software of benign players



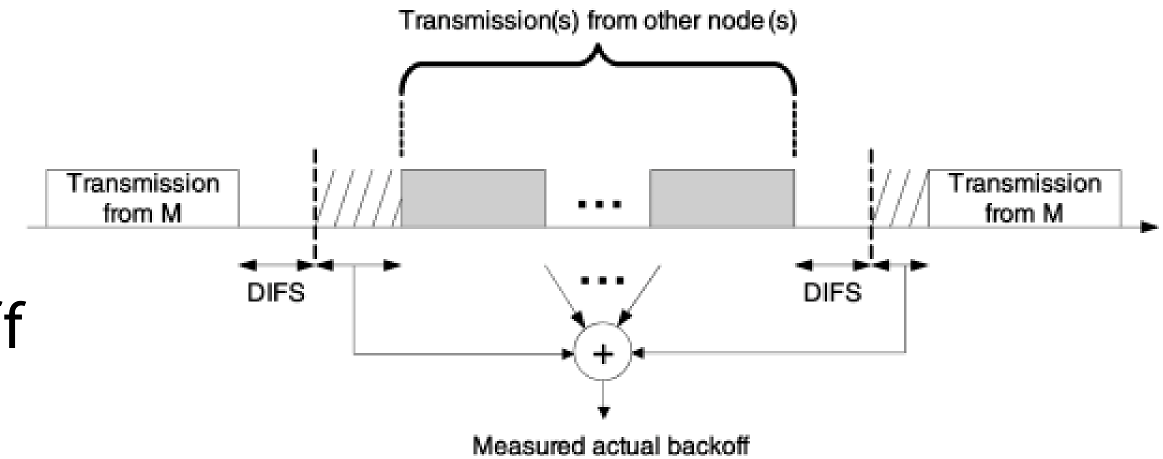


DOMINO Architecture

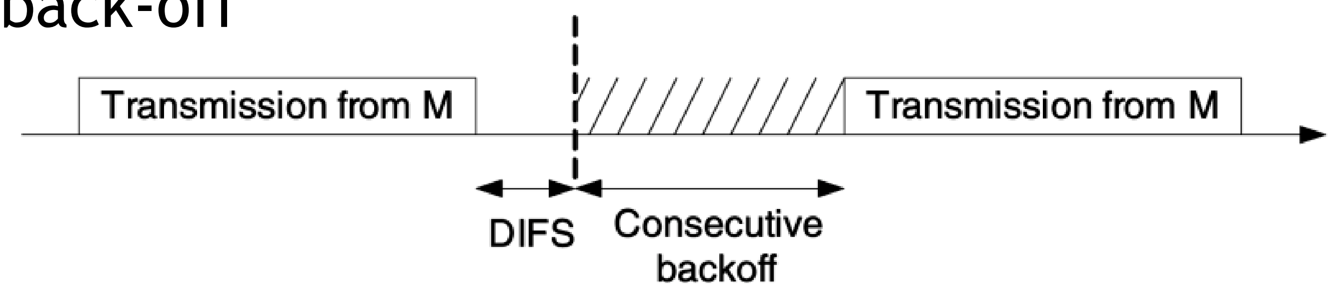


Behavior Tests

- The DOMINO-enabled AP performs a number of behavioral tests as a decision-making basis
  - Scrambled / re-transmitted frames
  - Shorter than DIFS
  - Oversized NAV



  - Observed back-off
  - Consecutive back-off



Further Discussions in Paper

- The DOMINO paper talks about a lot of different types of misbehavior
  - Jamming attacks, timing misbehavior, etc.

- Design of a deployable system
  - Lots of design parameters to choose
  - Analysis of numerous types of misbehavior
  - Incorporation of security mechanisms, quality of service, wireless error scenarios (e.g., hidden terminal)

Fairness in 802.11

- 802.11 incorporates various fairness mechanisms
  - Provides fairness regardless of connection quality
  - Allows low-quality connections to occupy the medium for much longer than high-quality connections

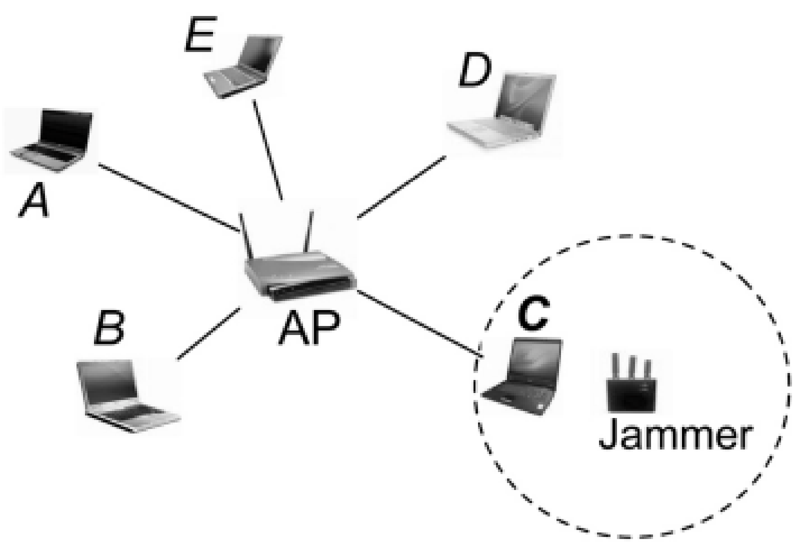
Implicit Jamming in 802.11

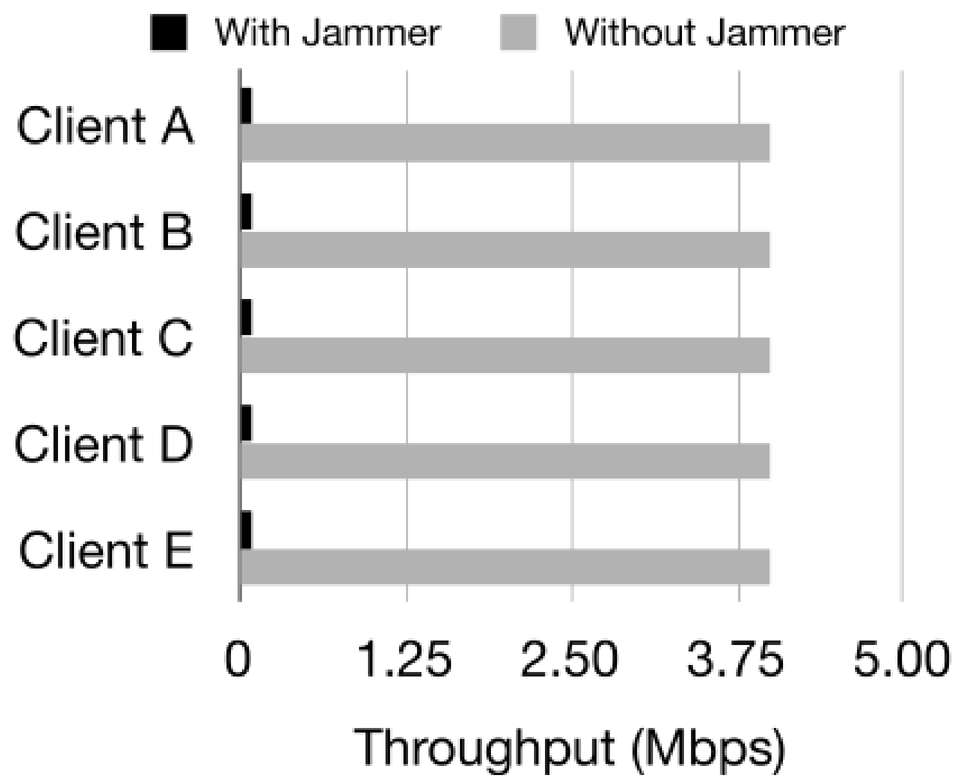
[Broustis et al., 2009]

- 802.11 has a built-in fairness mechanism that basically allows all users to get the same long-term throughput
  - A clever attacker can take advantage of this property to deny service to others by jamming a single user
  - Degradation of the single user effectively starves the other users
  - Jamming an end node is not necessarily observable by the AP, so detection is much harder

Implicit Jamming

- Low-power jammer attacks a single nearby node, degrades throughput for every user using the same AP





Mitigating Implicit Jamming

- FIJI: anti-jamming mitigation of the implicit jamming attack
  - **Goal 1**: ensure that nodes not under attack are not indirectly affected by the attack
  - **Goal 2**: ensure that the maximum amount of traffic is delivered to the node under attack, given that the node is under attack
  - Both goals rely on explicit detection of the jamming attack

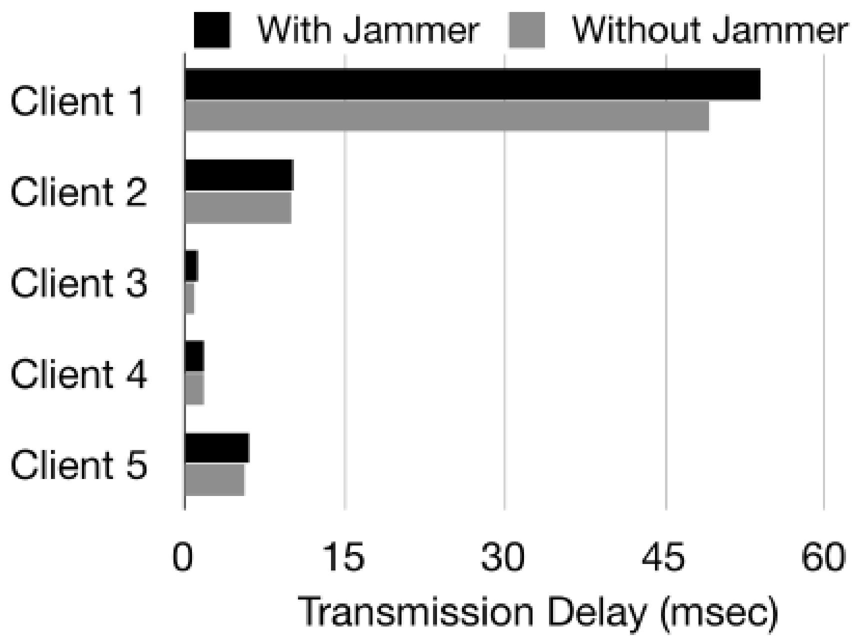
FIJI Detection Component

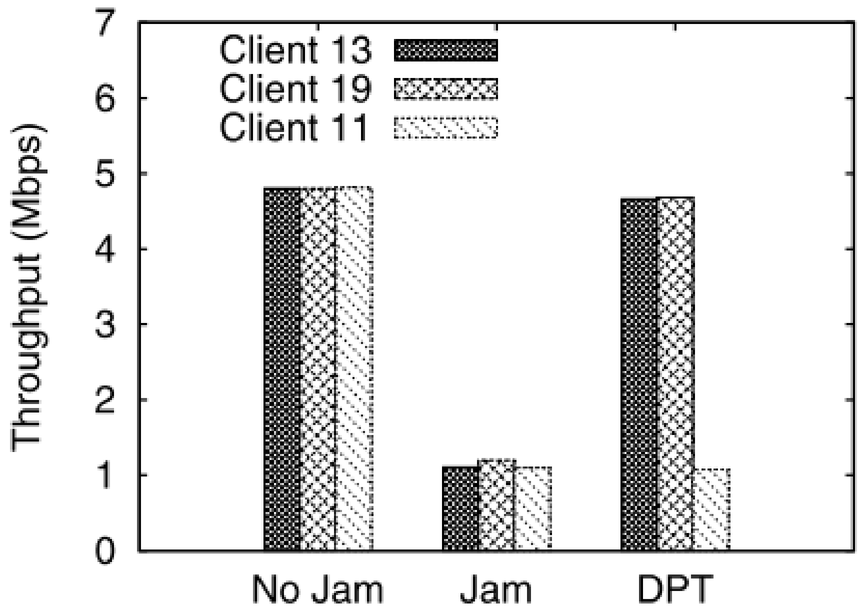
- Detection module
  - Since FIJI is run/managed entirely at the AP, detection must also take place there; not typical jamming attack detection
  - Standard jamming detection mechanisms (e.g., using RSSI+PDR) don't apply, need other metrics
  - Instead, look for changes in transmission delay
    - Very large increment in measured transaction time indicates the node is under attack

FIJI Traffic Component

- Adjust the traffic patterns to all clients based on detection events
  - Trivial solution: don't send any data to jammed clients, but this is unfair and could lead to big problems if any detection errors occur
  - Accept traffic degradation to attacked node, but keep traffic patterns constant for other nodes
  - Two approaches to deal with the attacked node:
    - Adjust the data packet size: shorter packet fragments are more likely to get through
    - Adjust the data rate: send to the jammed nodes less often

FIJI Evaluation





More OMNET++/INET

Feb 16:
Network Layer Threats;
Identity Mgmt.