Wireless Network Security and Privacy

Link layer threats & MAC misbehavior

Xiaoyu Ji 冀晓宇

Department of Electrical Engineering
Zhejiang University

2025 Autumn

Outline

- Basic link layer security considerations
- WLAN/WiFi security
- WiFi vulnerabilities
- MAC misbehaviors

Wireless Links

Link Layer Functionality

- The wireless link layer is primarily responsible for establishing and managing point-to-point links between neighboring nodes

- Also, passing data frames to/from the PHY and the network layers

Wireless Link Types

- WiFi: AP ↔ host
- Telecom: mobile ↔ BTS
- V2I: vehicle ↔ RSU
- V2V: vehicle ↔ vehicle
- V2C: vehicle ↔ cat
  - Not really…?
- D2D: device ↔ device

- And so on…

Service Breakdown

- **Establishing the link:**
  - Neighbor discovery
  - Addressing
  - Channel setup / sync
  - Authentication / authorization
- **Managing the link:**
  - Medium access control (MAC), availability
  - Confidentiality, integrity, etc.
  - Queueing & scheduling
- **Layered services:**
  - collision avoidance, carrier sensing, error correction, signaling, etc.

Link Layer Threats

Essentially, every service at the link layer has corresponding threats

Discovery Threats

- Discovery can be affected by malicious devices actively preventing benign devices from finding and connecting to each other
- Examples:
  - In WiFi, a malicious device can spoof the WiFi access point, attracting unsuspecting users to attach to the attacker instead of the intended network
  - In MANET/VANET, a Sybil attacker can present multiple network identities, attracting connection-limited devices to waste space in look-up tables

Network Access Threats

Network access can be affected in two ways:

- 1) preventing access by valid devices and
- 2) gaining access from invalid devices

- Examples:
  - Preventing access by DoS, forced disconnection, etc.
  - Unauthorized access or elevated access level, achieved by crypto-based attack, session hijacking, session take-over during hand-off, etc. based on authentication / authorization protocols

InfoSec Threats

- Secrecy / confidentiality can be compromised by attacking the crypto or security protocols used to protect the data in flight
  - Exp. if weak crypto is used
- Integrity can be similarly compromised
  - Weak crypto or unfortunate integrity protocol design

Availability Threats

- Availability can be threatened in different ways from discovery or access, namely an attacker can let you discover and connect, but get no or poor service
  - PHY-layer threats like interference/jamming can affect connection mgmt. with a discovered AP
  - Cheating is often possible at the MAC layer due to assumptions that everyone plays well together
    - More on this later

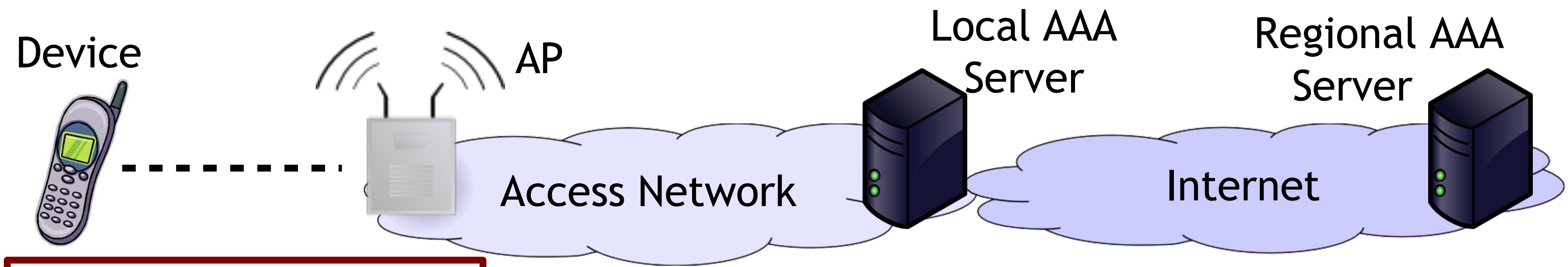
Privacy Threats

- Device/user privacy may be at risk due to the inherent exposure/exchange of identifying information in link formation and mgmt.

- Examples:
  - In WiFi (and most others), devices are required to broadcast a MAC address that identifies them
    - Even if the MAC isn't linked to a personal identity, subsequent messages/locations can be correlated

Let's go into more detail about WiFi

Private WiFi Networks



Device needs to discover available AP to connect to

Network servers store credentials, identity, etc.

Device authenticates to AAA server

Server provides cryptographic material to AP

Device ↔ AP secure channel

AP ↔ Server / Internet secure channel

AAA: authentication, authorization, and accounting (AAA) services.

WiFi Discovery

- In order for a client device to connect to an AP, it needs to discover its presence/existence
- Two ways to do this:
  - AP can announce itself to all surrounding devices
    - Can't do this very often, so devices need to wait – also need to check multiple channels, since APs can move → slow
  - Client can call out for known APs - "WiFi Probing"
    - If the client has connected before, it knows how the AP is/was configured, so can find it very quickly
    - But, WiFi probing can expose your privacy

WiFi Probing Issues

Filter: (wlan.fc.type_subtype == 0x04) ▼ Expression...

Time	Source	Type	SSID
401.697011000	54:26:	Probe Request	
401.707384000	Apple_	Probe Request	
401.855865000	bc:cf	Probe Request	
401.868368000	Apple_	Probe Request	
402.093322000	Apple_	Probe Request	Hooters
402.094443000	Apple_	Probe Request	Internet
402.095695000	Apple_	Probe Request	HarborLink - Buffalo Wi
402.096939000	Apple_	Probe Request	NetScout
402.098059000	Apple_	Probe Request	Rosen Guest Wireless
402.099190000	Apple_	Probe Request	Student
402.100310000	Apple_	Probe Request	Guest
402.101568000	Apple_	Probe Request	Gdaycreations
402.106317000	Apple_	Probe Request	cactusmoon_public
402.107442000	Apple_	Probe Request	NOTanIphone
402.108690000	Apple_	Probe Request	Gentleman Joes 3
402.109815000	Apple_	Probe Request	MISSION PRIVATE

SSID Based Threats

- Whenever a mobile device blasts out probe messages, we can learn its relevant *SSID set*

CMU-SV
Peets
SJCWiFi
monkeys
Univ WA

Personal Profiling

CMU-SV
Peets
SJCWiFi
monkeys
Univ WA

@ (x,y,t)

@ (x',y',t')

CMU-SV
Peets
SJCWiFi
monkeys
Univ WA

Tracking

CMU-SV
ZJU-Wlan
Starbucks
monkeys
Univ WA

CMU-SV
JI-HOTEL
SJCWiFi
monkeys
Univ WA

Social Relationships

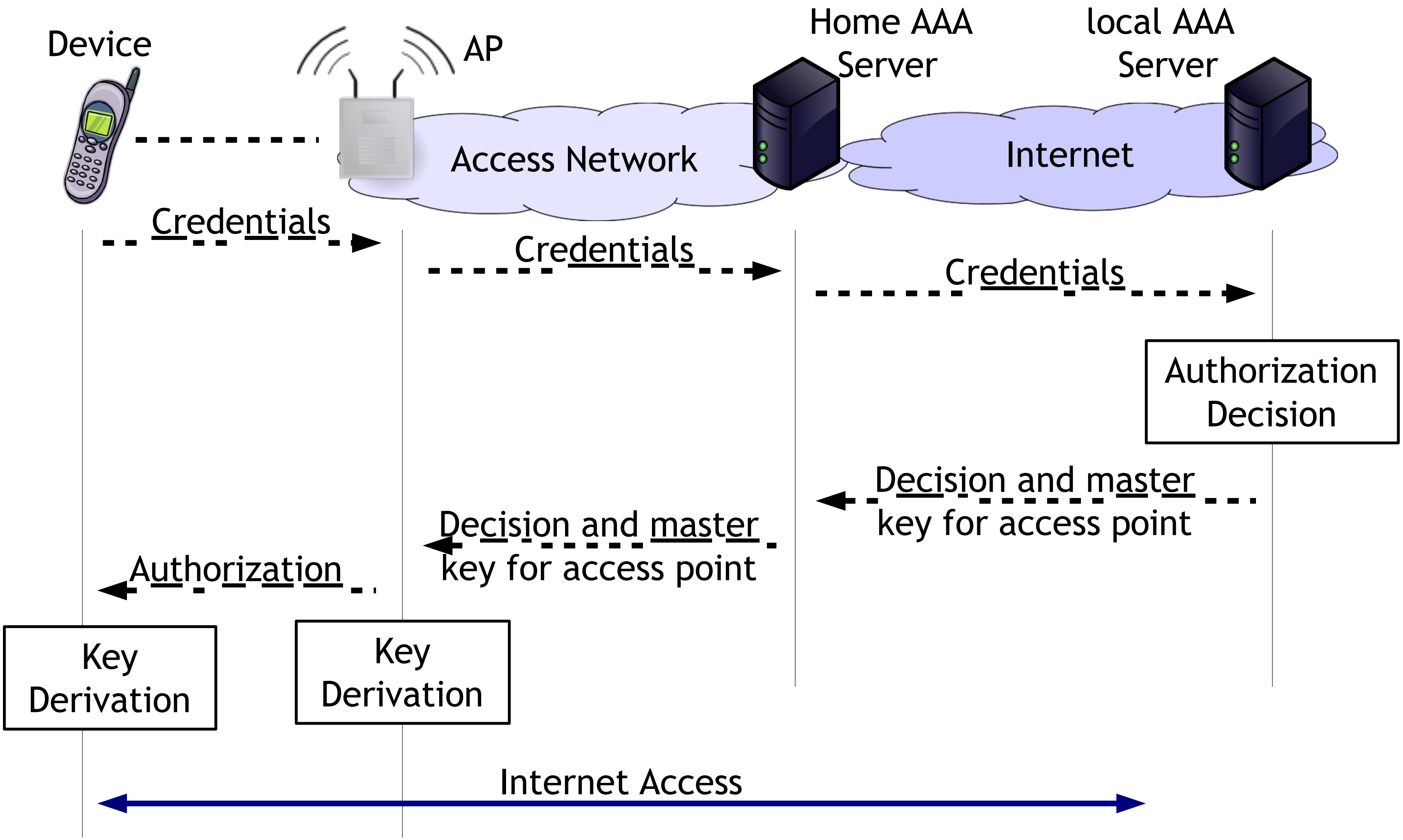
Potential Fixes

- Since many threats are based on MAC-SSID pairs, MAC pseudonymy can help
  - Implies there's a trusted third party to handle pseudonyms, requires pre-existing relationship
- MAC or SSID info can be encrypted
  - Requires computation or search on mobile and/or AP to discover which keys should be used to decrypt, requires pre-existing relationship
- Don't use direct probing
  - Slow

WiFi Link Security

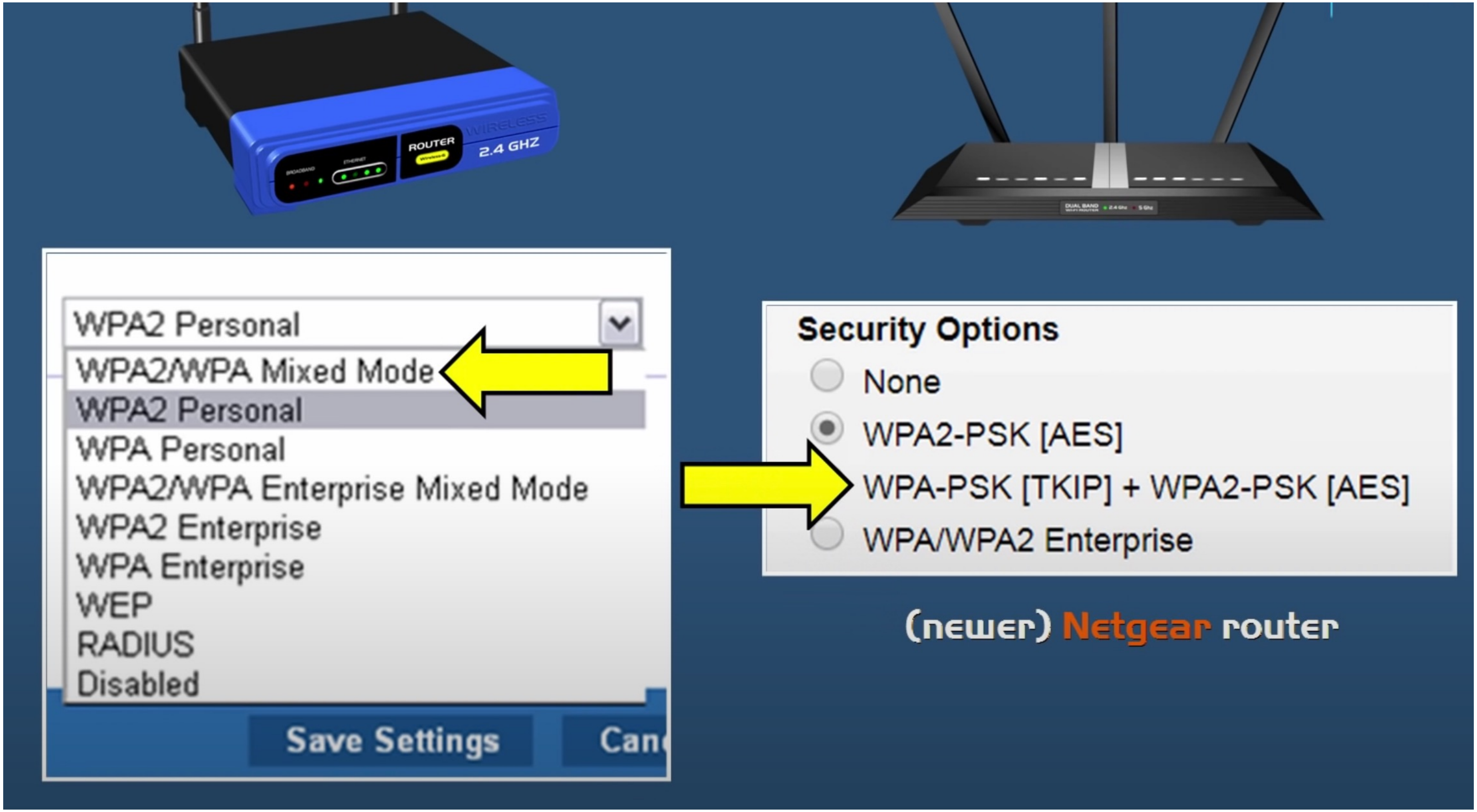
- WiFi link security focuses primarily on access control and encryption
  - In private WiFi systems, access is controlled by a shared key, identity credentials, or proof of payment
  - Most often, authentication is of user/device only, but mutual authentication may be desired/required by some users/devices, especially for IoT devices
  - Confidentiality and integrity over the wireless link
  - Shared medium among untrusted WiFi users

Private WiFi Networks



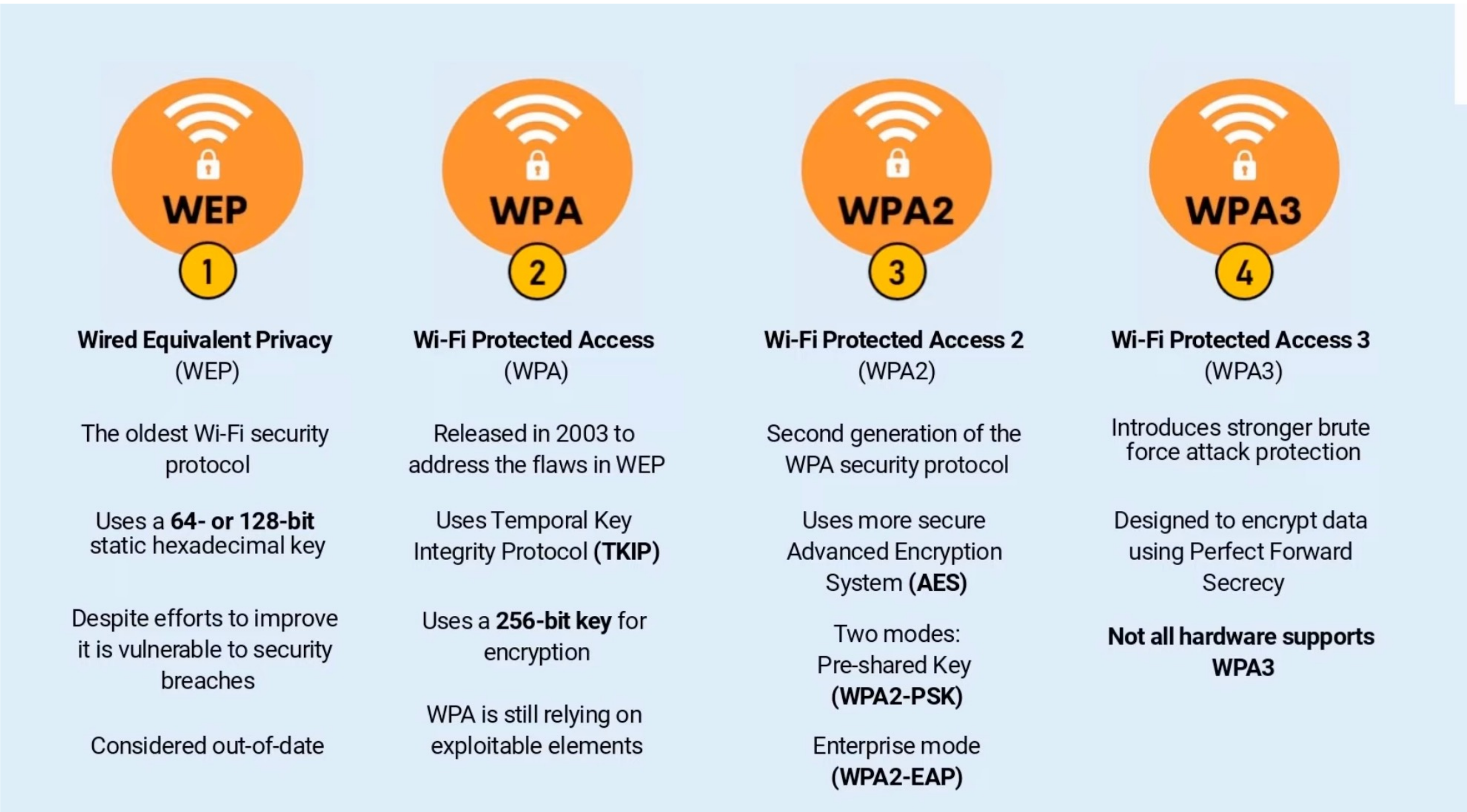
How is WiFi secured?

WiFi security



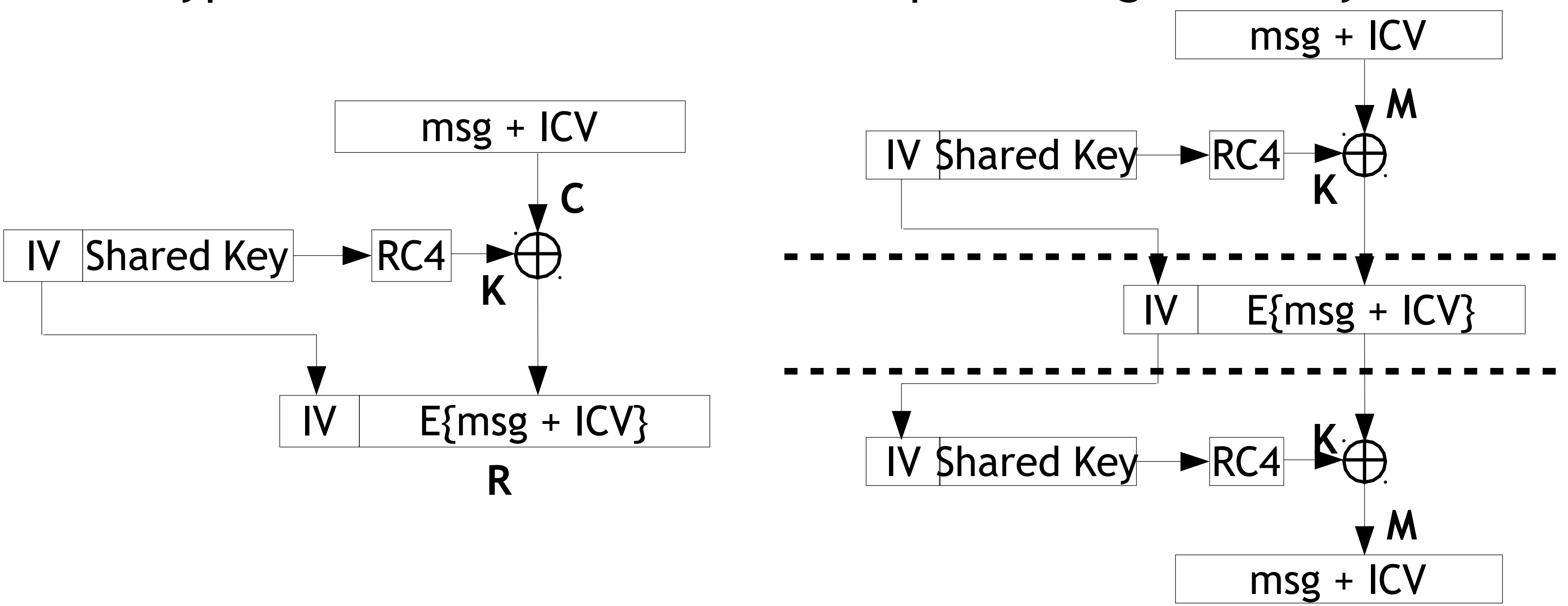
WEP/WPA/WPA2/WPA3

Video: https://youtu.be/jErjdGfbgoE



Wired Equivalent Privacy

- As name suggests, WEP(有线等效协议) aims to make the easy task of accessing WLAN much more difficult, as in wired
- WEP provides encryption and authentication
- Authentication is challenge-response to prove knowledge of a shared secret key
- Encryption is based on RC4 stream cipher using same key



WEP Authentication

- Challenge-response authentication w/ XOR
  - Issue 1: auth is not mutual
  - Issue 2: auth + enc. use same secret key
  - Issue 3: auth only occurs on initial connection
  - Issue 4: RC4 40-bit cipher be broken

- Threats: replay, brute-force attack

So, WEP is completely broken.

How did we solve the WEP problem?

IEEE 802.11i

- IEEE specification for robust network security
  - Authentication and access control based on 802.1x
  - Integrity protection and confidentiality mechanisms based on AES to replace RC4

But, RC4 and AES were implemented in hardware, so the upgrade couldn't happen overnight

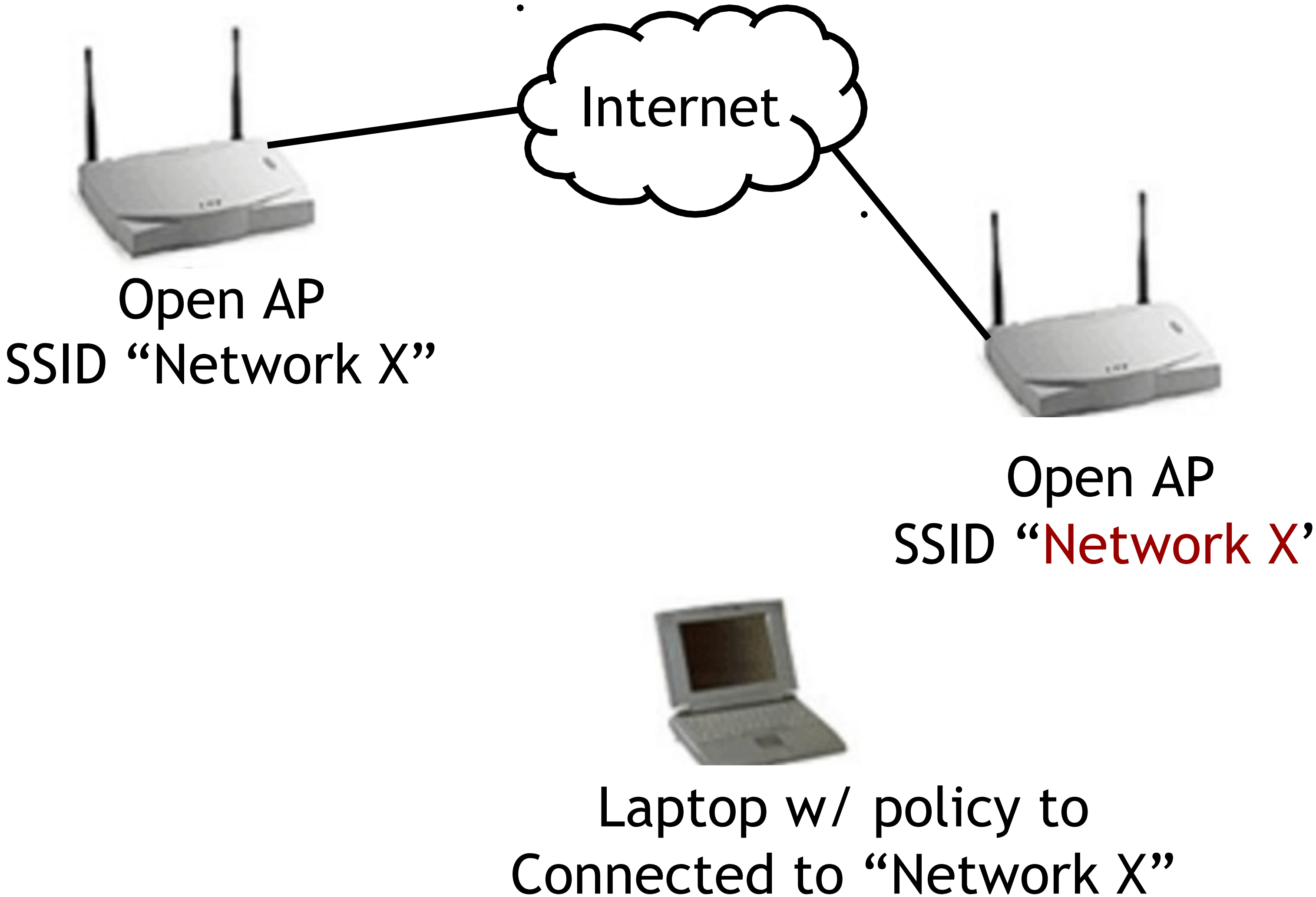
WiFi Protected Access

- **TKIP: Temporal Key Integrity Protocol**
  - TKIP ← 802.11i using RC4 instead of AES
  - Immediate firmware upgrade allowed for use of TKIP
  - WPA is the subset of 802.11i supported through TKIP
    - Auth and access control in WPA and 802.11i are the same
    - Integrity and confidentiality are TKIP-based

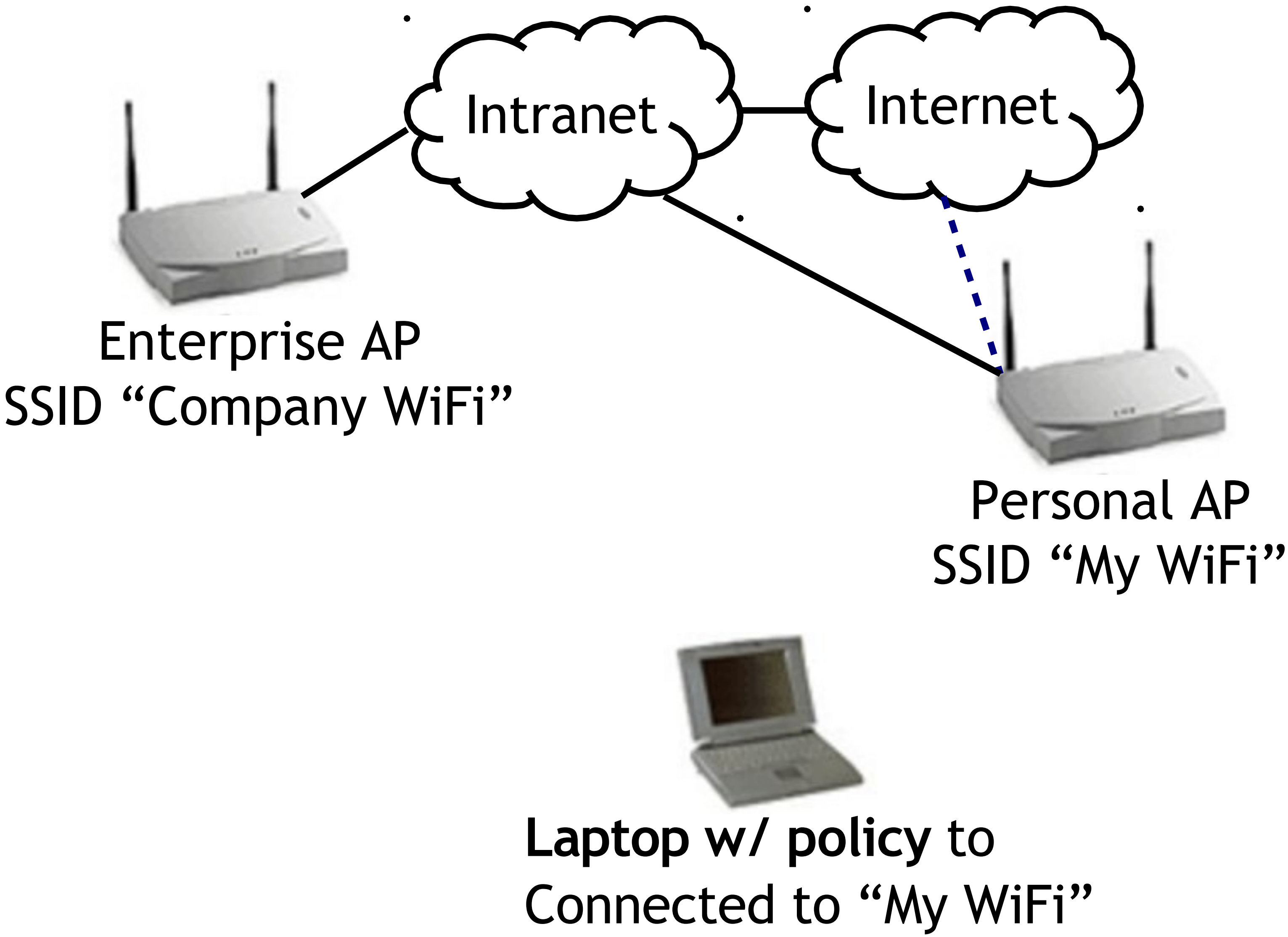
- WPA2 is full 802.11i implementation
  - But, WPA2 still has some weaknesses.
  - Read: *Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2, CCS'17*

So what kind of attacks are possible?

Fake AP Threats



Fake AP Threats in Enterprise



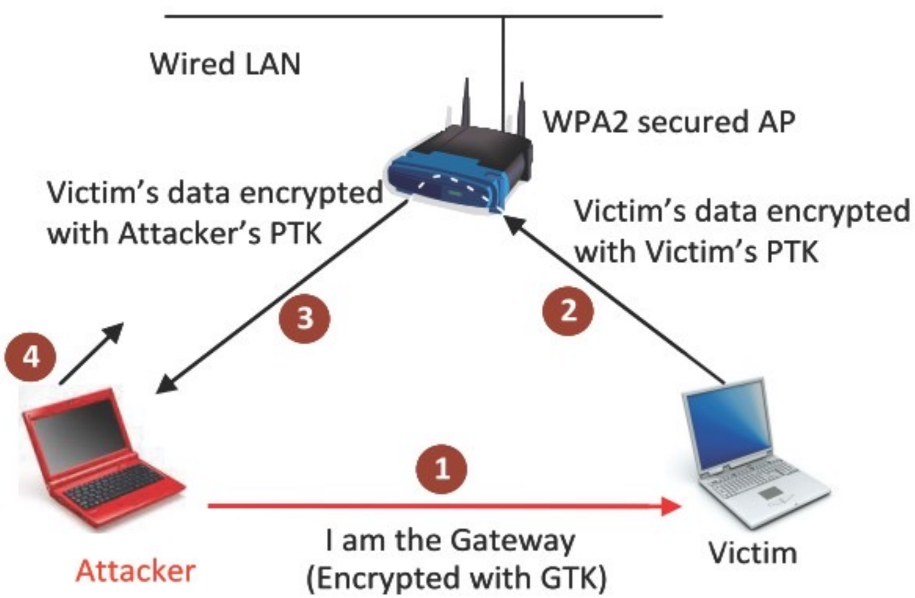
Another Interesting Attack

- Inverse Wardriving [Beetle & Potter, shmoo.com]
  - Wardriving is using a WiFi client to find open APs to get free service to the Internet
  - Inverse Wardriving is using a Fake AP to find WiFi clients that will connect to it
    - What if the client has an unpatched vulnerability?
    - IW can be used to locate vulnerable clients and exploit them
    - E.g., infect them with a worm
  - Creating a Fake AP is very easy, especially using tools like Airsnarf or similar
- KARMA attack = probe sniffing + Inverse Wardriving

What about insider threats?

Hole196 Vulnerability

- Attack against WPA2 Enterprise
- 2010 by Md. Sohail Ahmad of AirTight Security
  - Named for the page number in IEEE 802.11-v2007
  - Malicious insider can misuse the GTK(Group Temporal Key)
    - Example: the insider advertises itself as the gateway, tricking them into redirecting their data to the insider via the AP



[Image from AirTight Networks whitepaper]

Summary

WiFi security is fairly mature, but still not completely understood, partially due to ubiquity and partially due to complexity

Wireless Network Security and Privacy

MAC misbehavior

Xiaoyu Ji 冀晓宇

Department of Electrical Engineering
Zhejiang University

2025 Autumn

Outline

- IEEE 802.11 MAC layer
- Misbehavior in 802.11 MAC
- A few other MAC threats (time permitting)

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

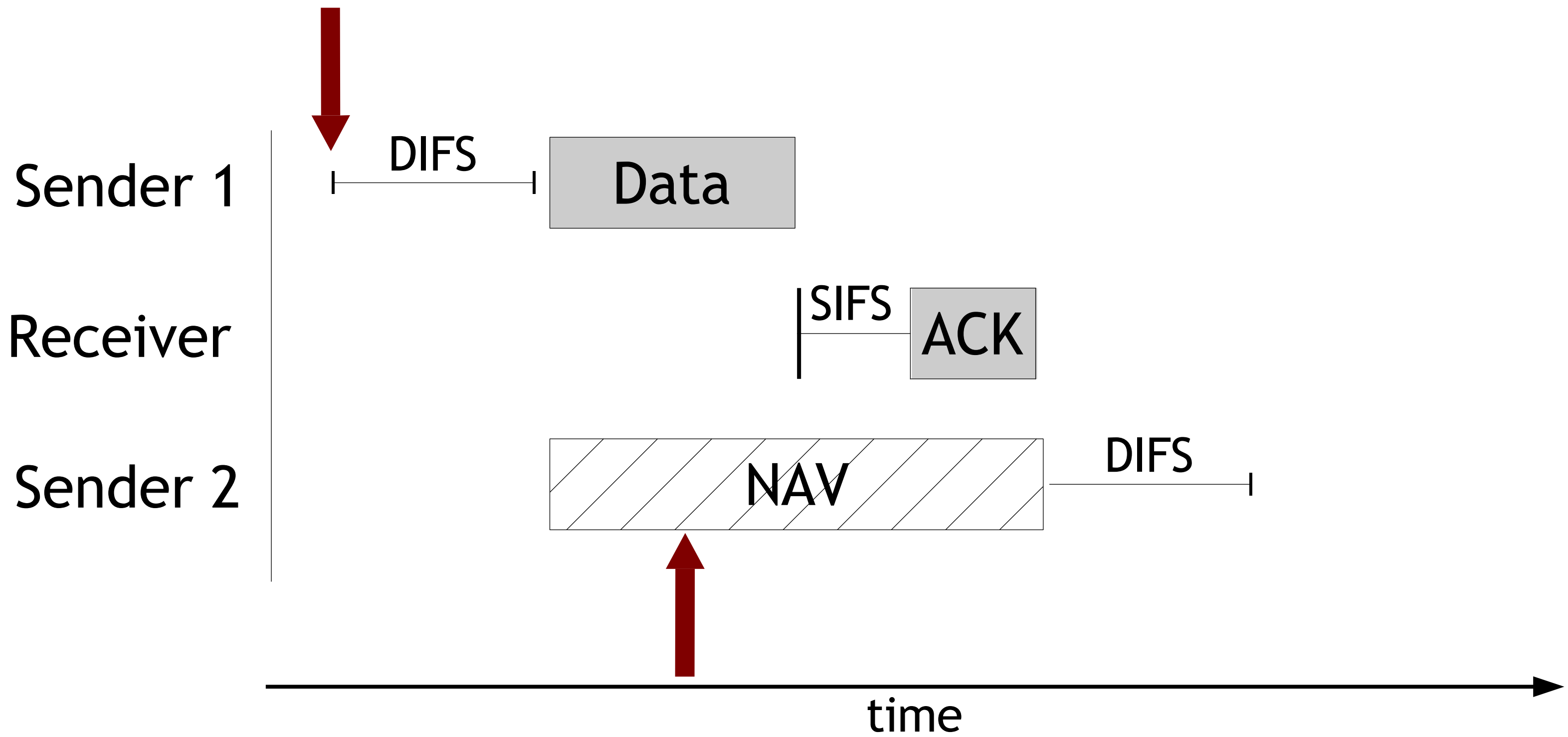
- Responsibilities of the MAC layer
  - Logical responsibilities
    - Addressing
    - Fragmentation
    - Error detection, correction, and management
  - **Timing responsibilities**
    - Channel management
    - Link flow control
    - Collision avoidance

- 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



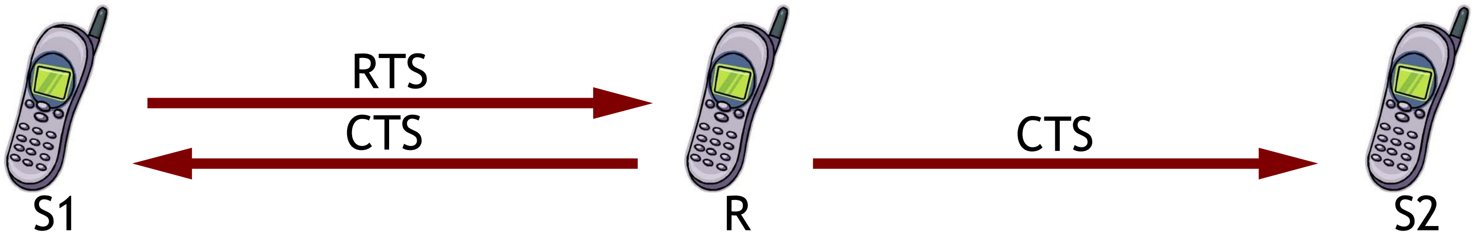
DIFS: DCF Interframe Space(DIFS)
SIFS: Short Interframe Space(SIFS)
NAV: Network Allocation Vector

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 $\rightarrow$ 15 $\rightarrow$ 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



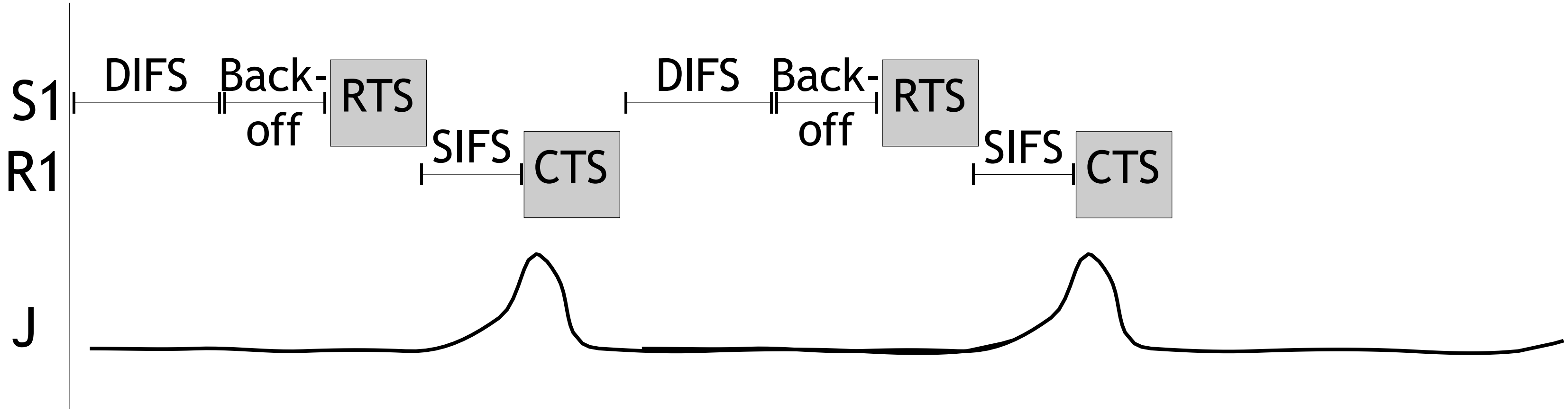
MAC Layer Misbehavior

- 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?

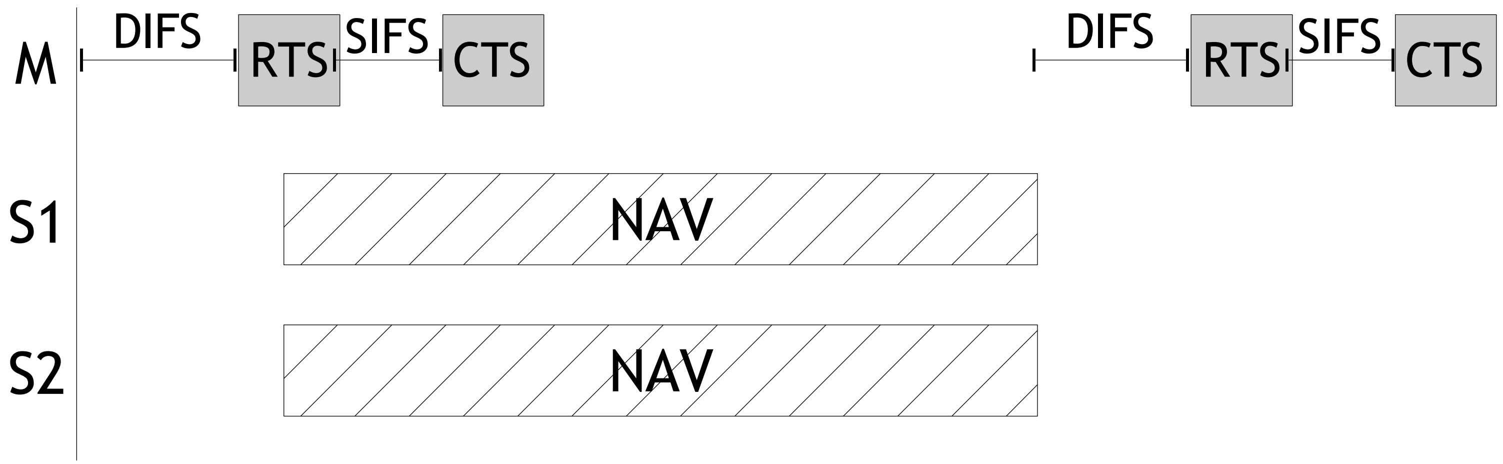
MAC Jamming

- 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



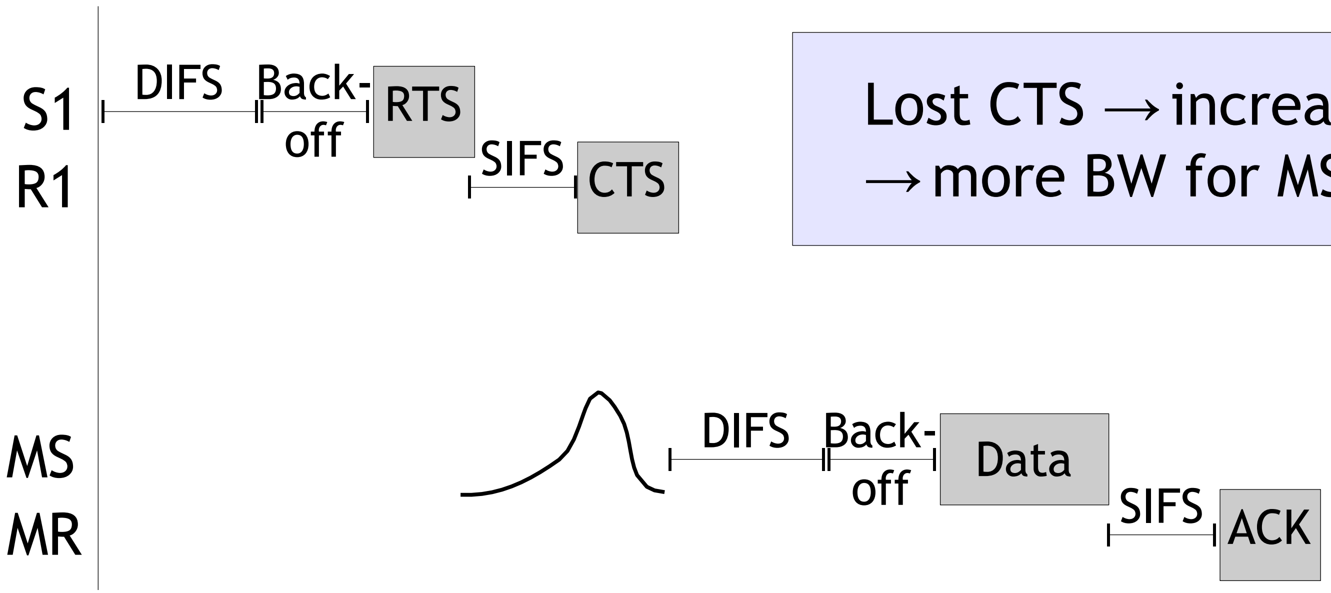
MAC Blocking

- 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

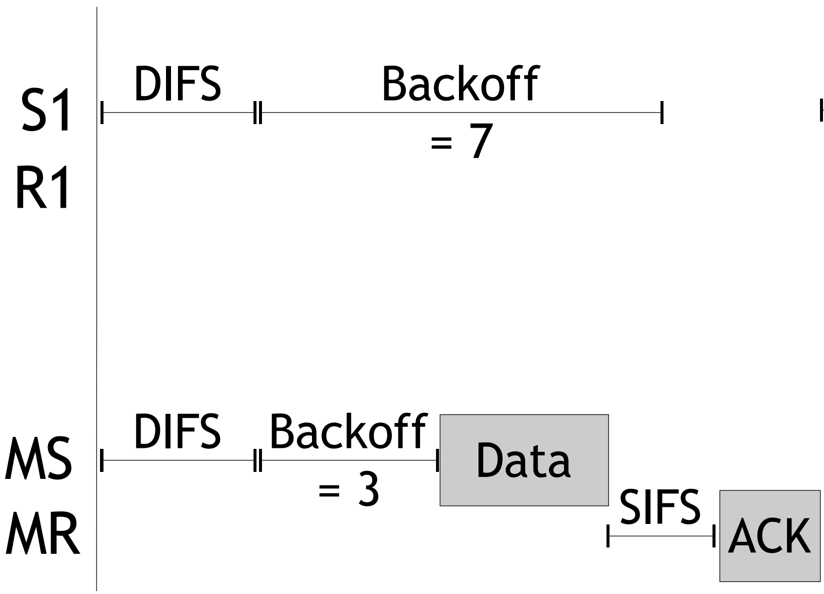
- Greedy/malicious sources can block or collide with other sources, causing their sending rates to decrease
  – Gives more opportunity to greedy source



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

MAC Greed w/ Parameters

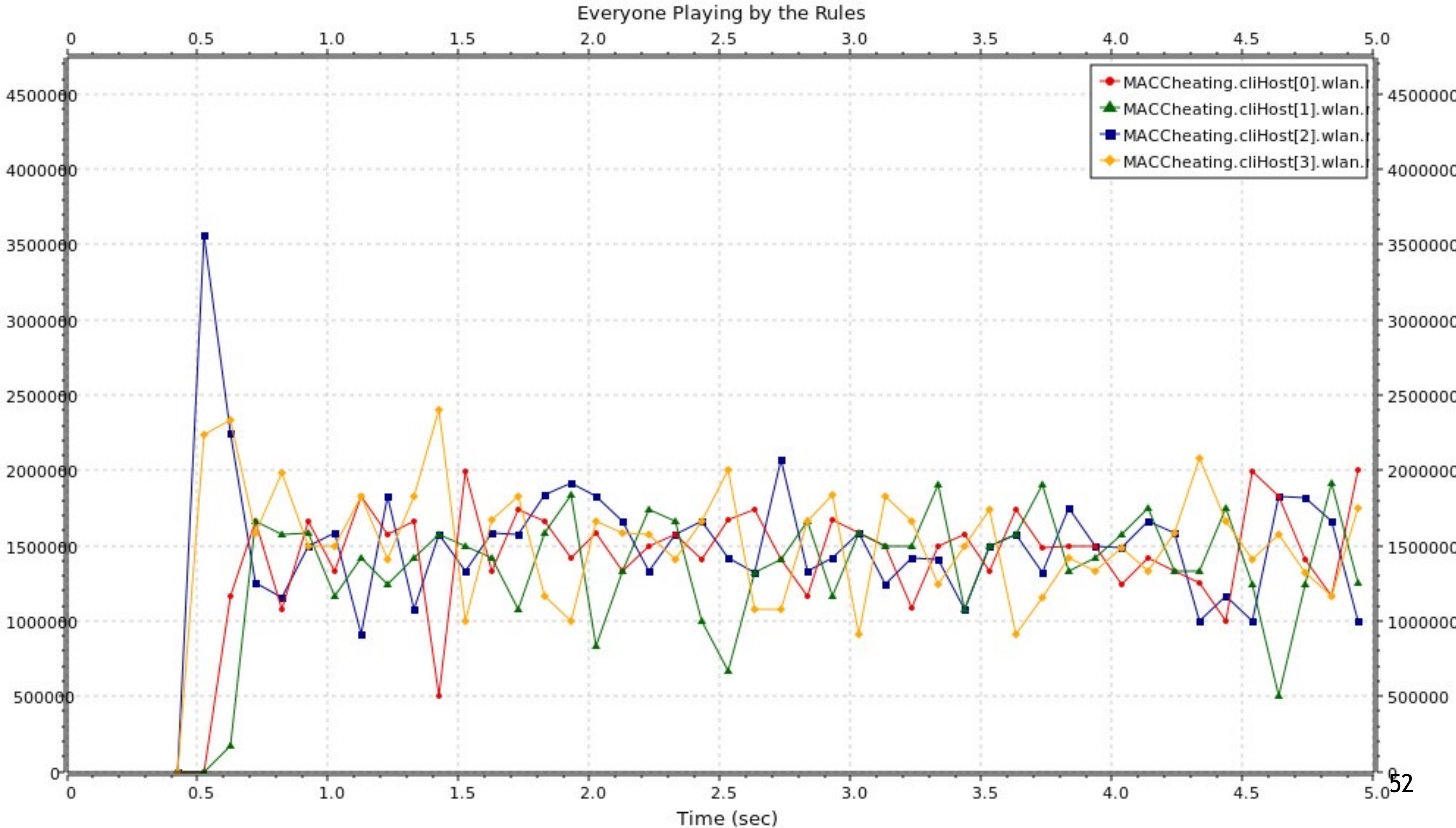
- Greedy/malicious sources can manipulate protocol parameters for unfair resource usage



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

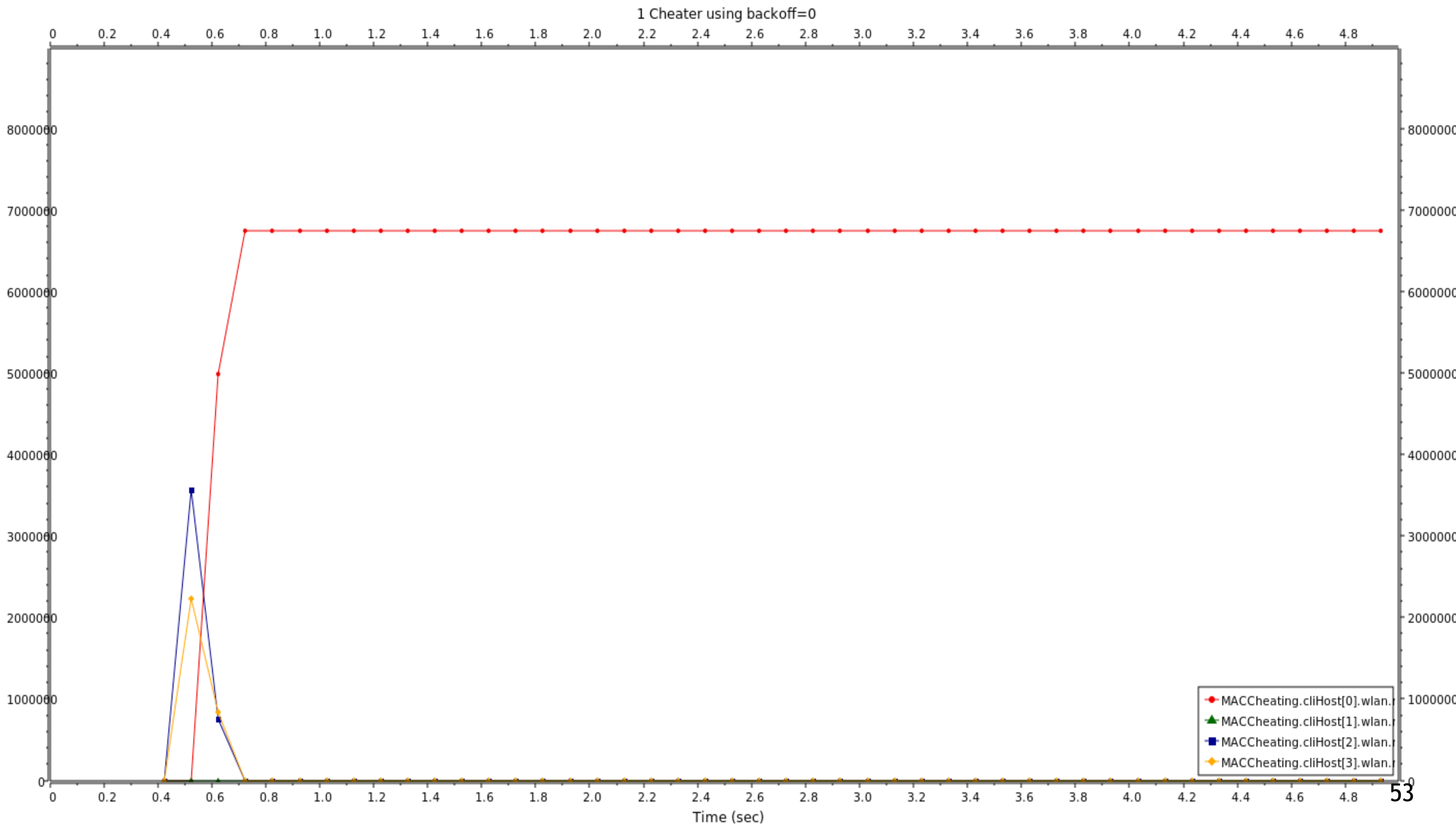
Example

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



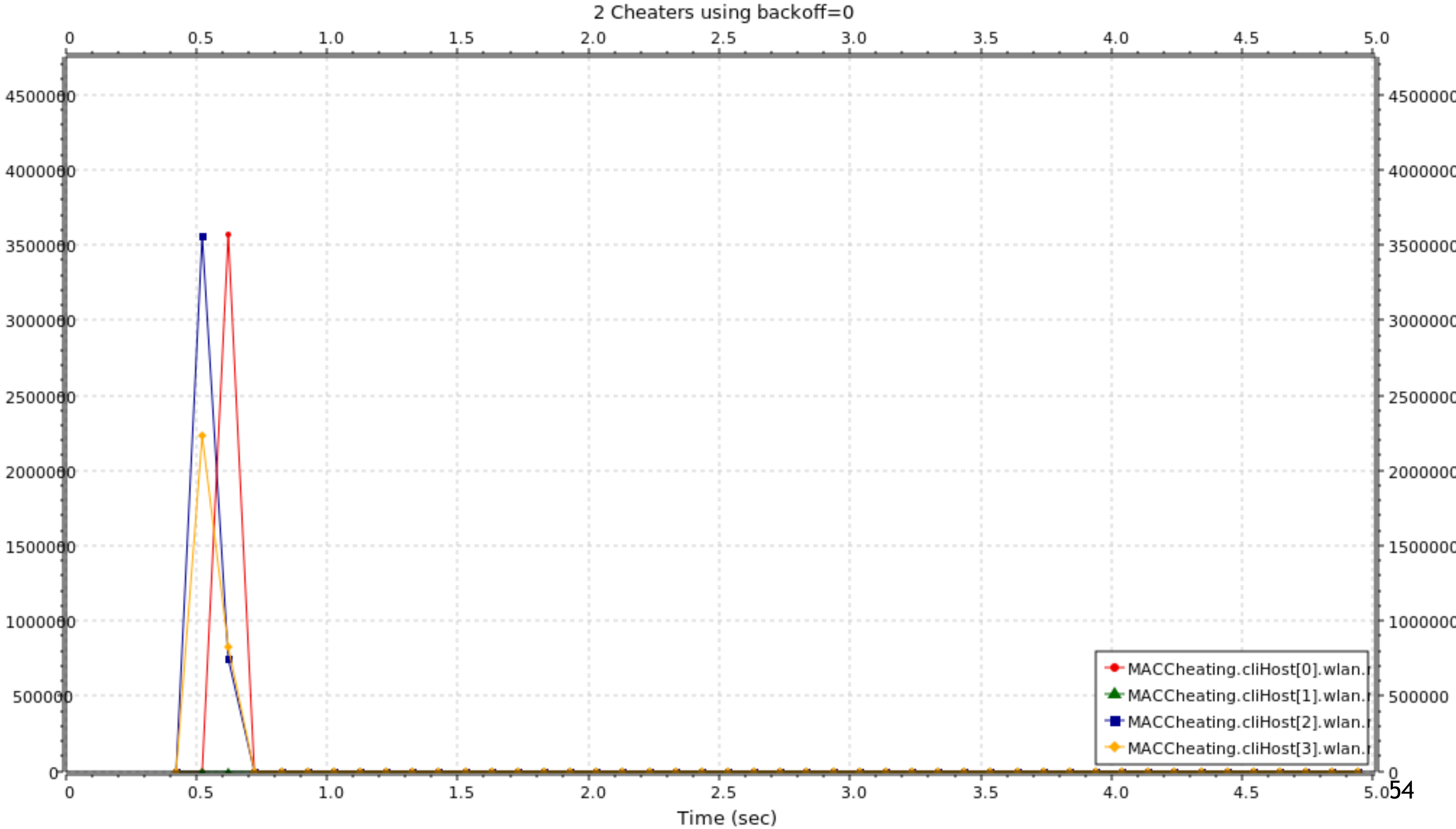
Example

- 4 clients, 1 using backoff = 0



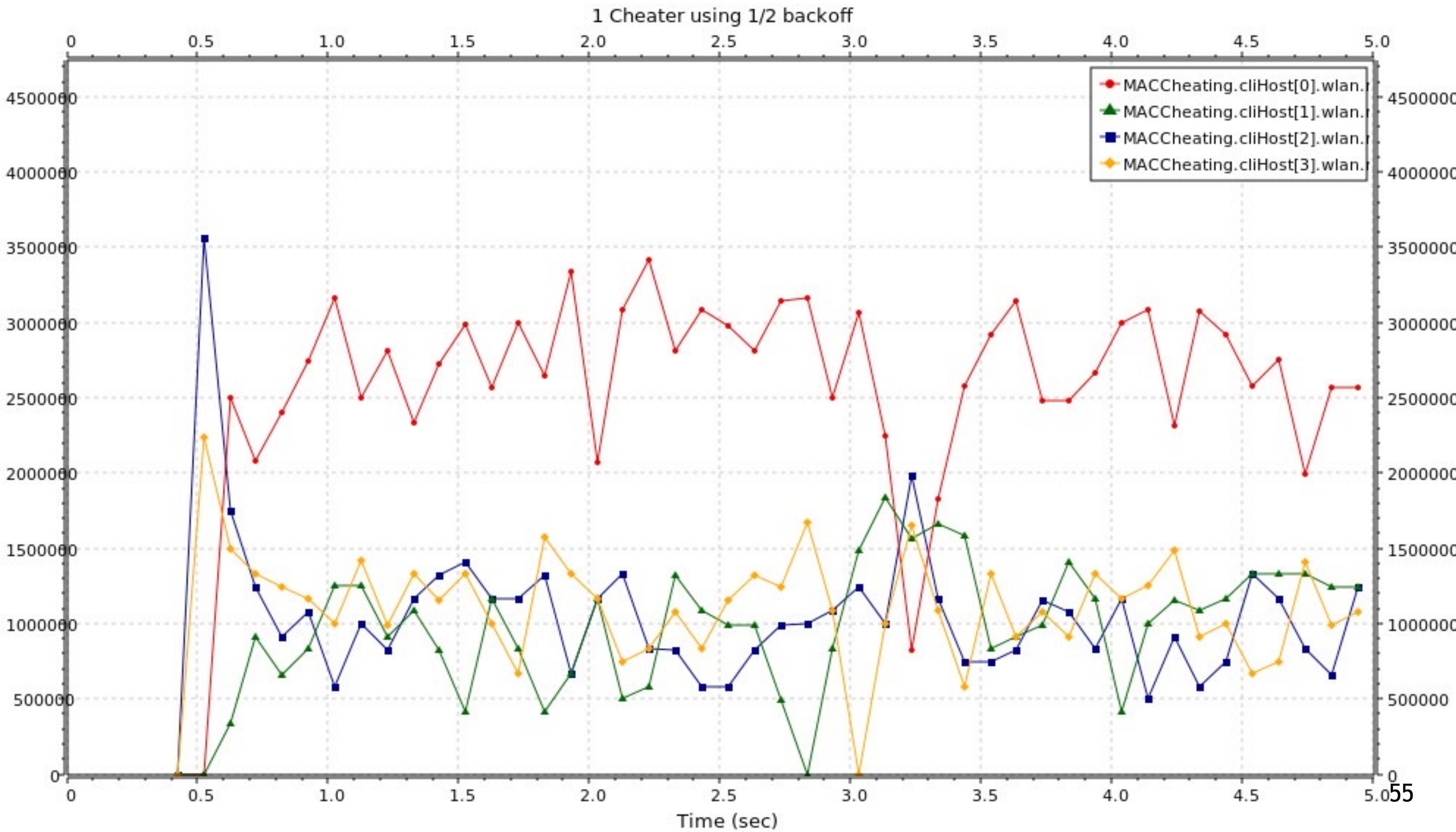
Example

- 4 clients, 2 using backoff = 0



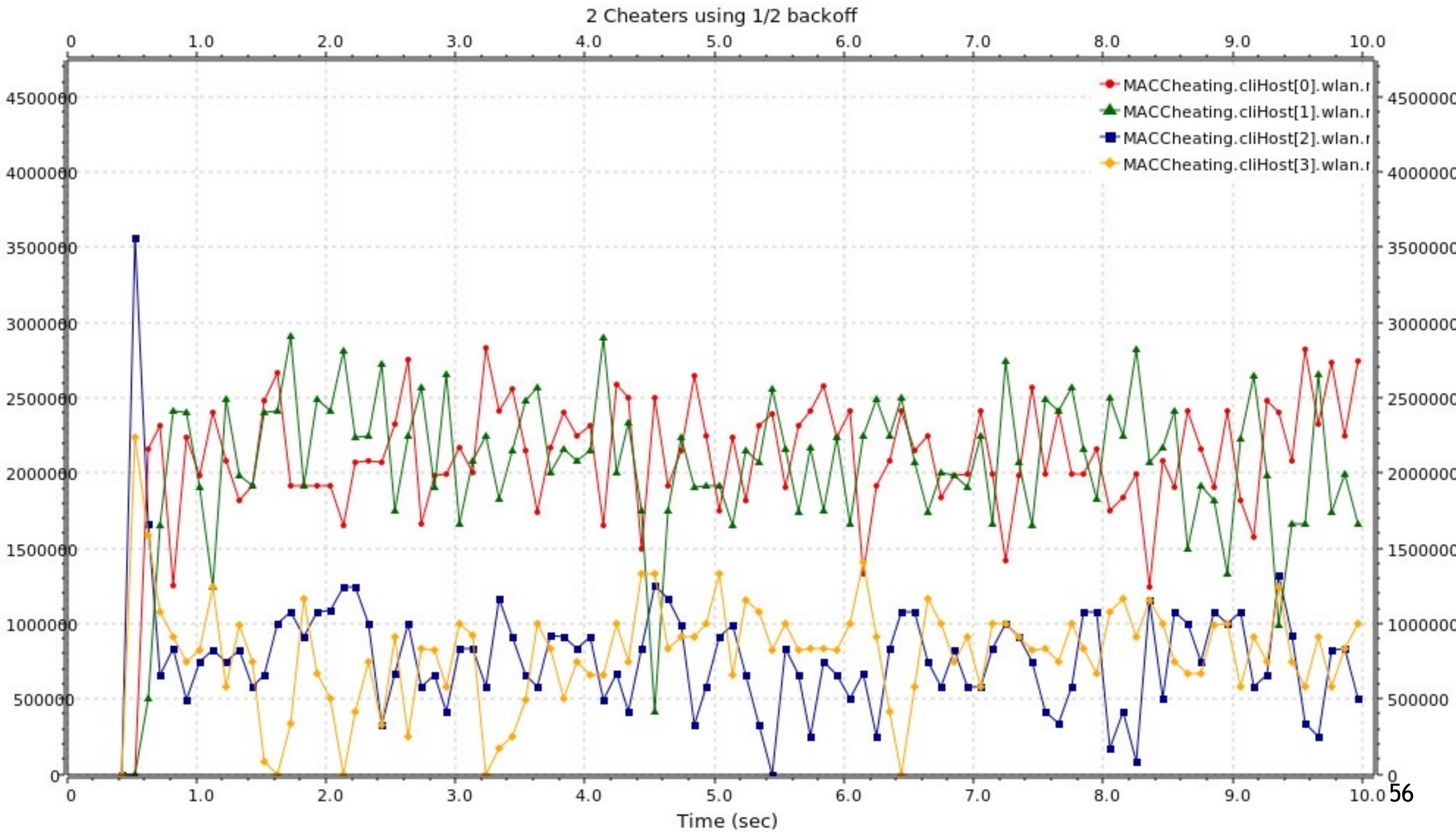
Example

- 4 clients, 1 using backoff / 2



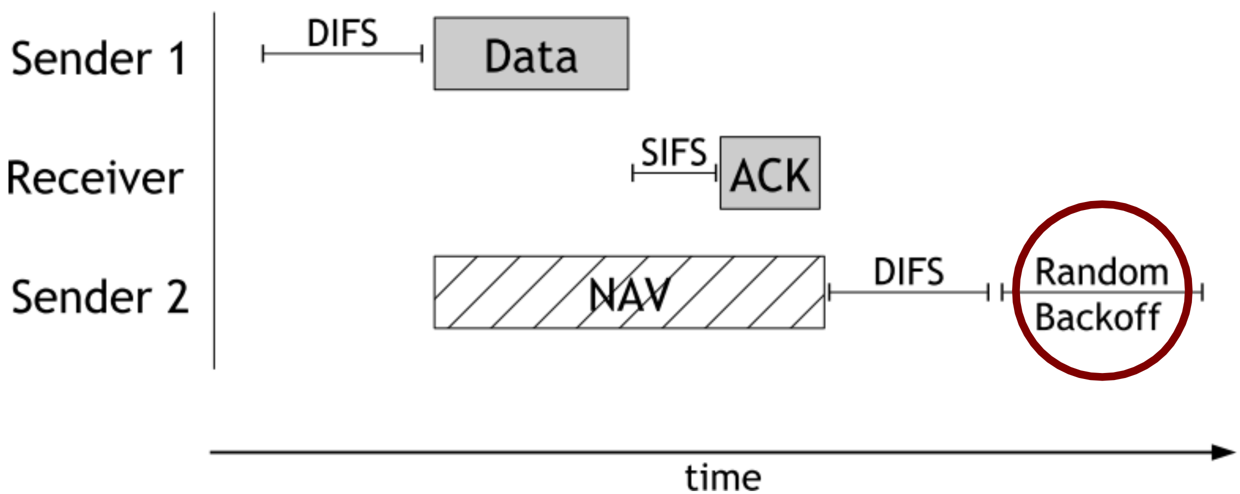
Example

- 4 clients, 2 using backoff / 2



Cheating in CSMA/CA

- "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

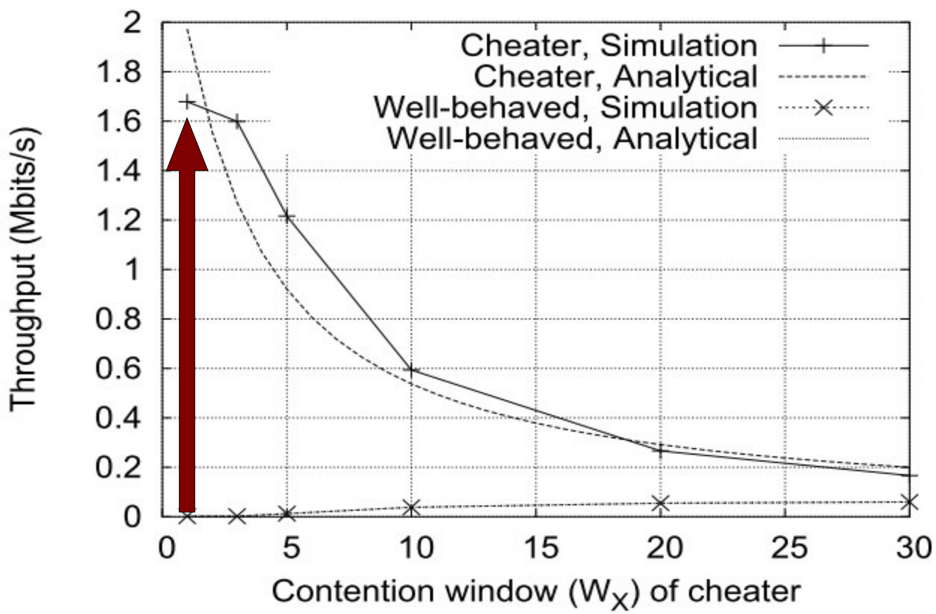


System Game Model

- *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

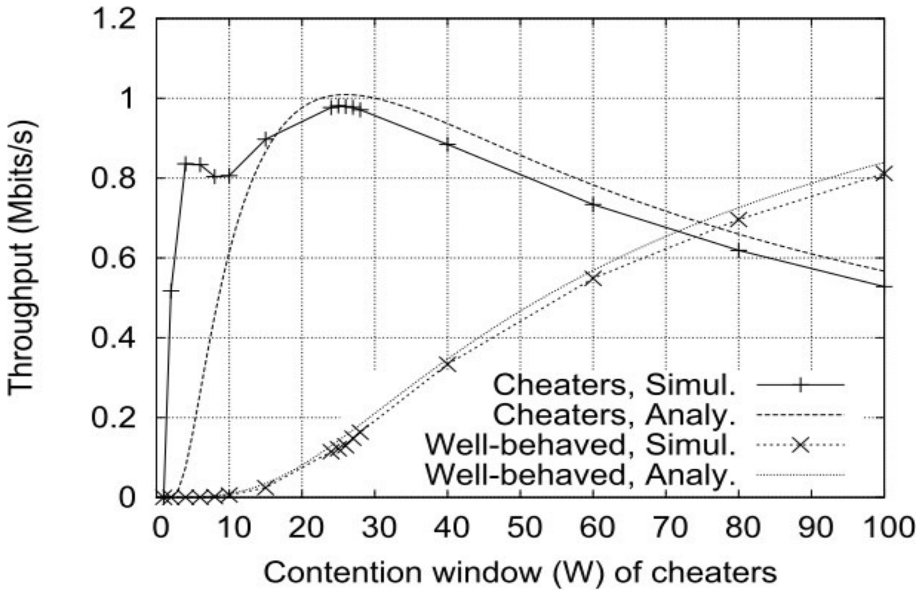
Single Static Cheater

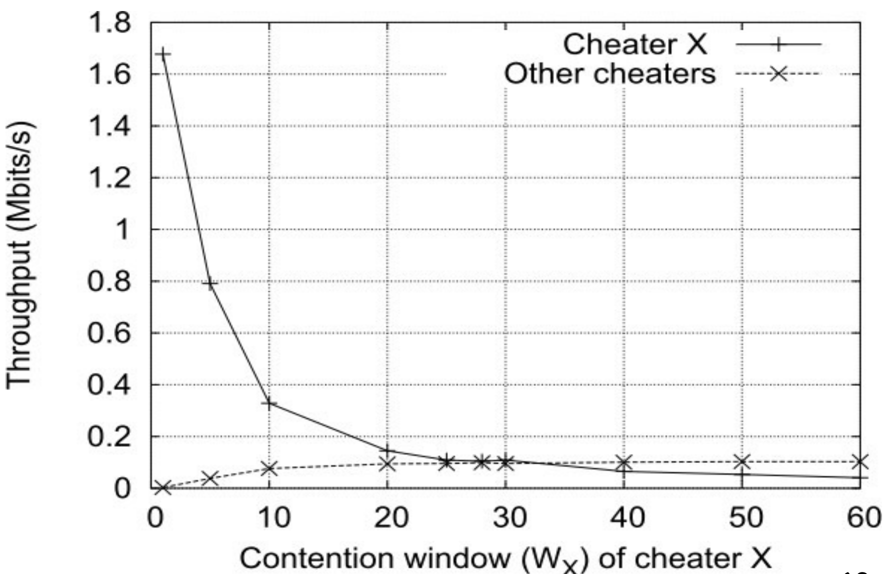
- **First case**: a single cheater with a fixed strategy (i.e. makes a decision and sticks with it)
- A single cheater gets best throughput at $W_i=1$
- In fact, $W_i=1$ is the Nash Equilibrium for the static game with $C=1$



Multiple Static Cheaters

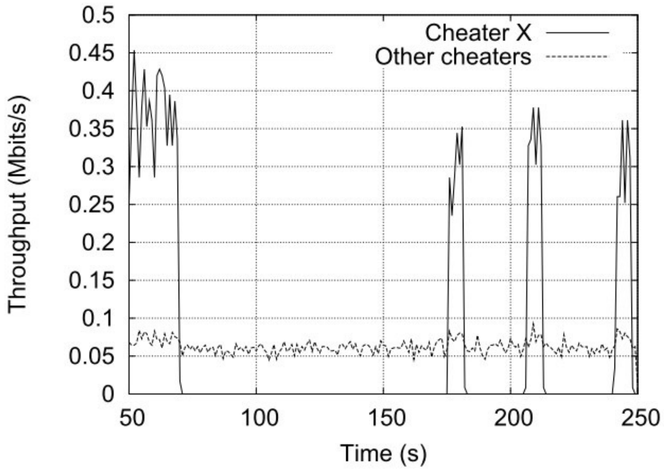
- **Second case:** many cheaters with fixed strategy
  - 2.1 Cheaters don't know about each other
  - 2.2 Cheaters are aware of cheater v. cheater competition in forming strategies
- Window size $W_i=1$ is no longer optimal

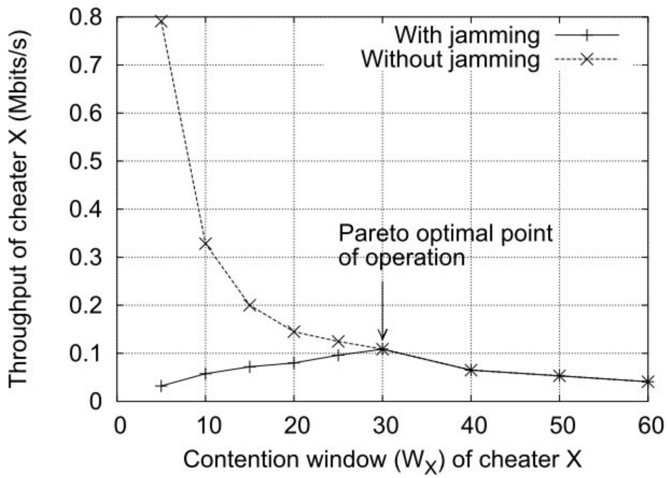




Dynamic Cheating Game

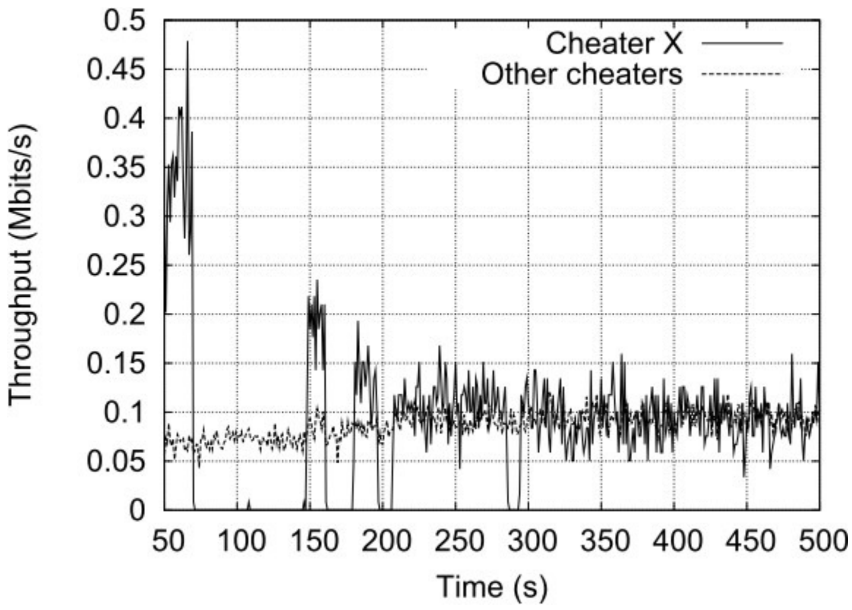
- In the dynamic game, cheaters can change their strategy in response to other players (including other cheaters)
  - A penalty is enforced on the utility function, so cheaters converge to the optimal operating point
  - "Cooperative cheaters" can inflict the penalty on "non-cooperative cheaters" by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

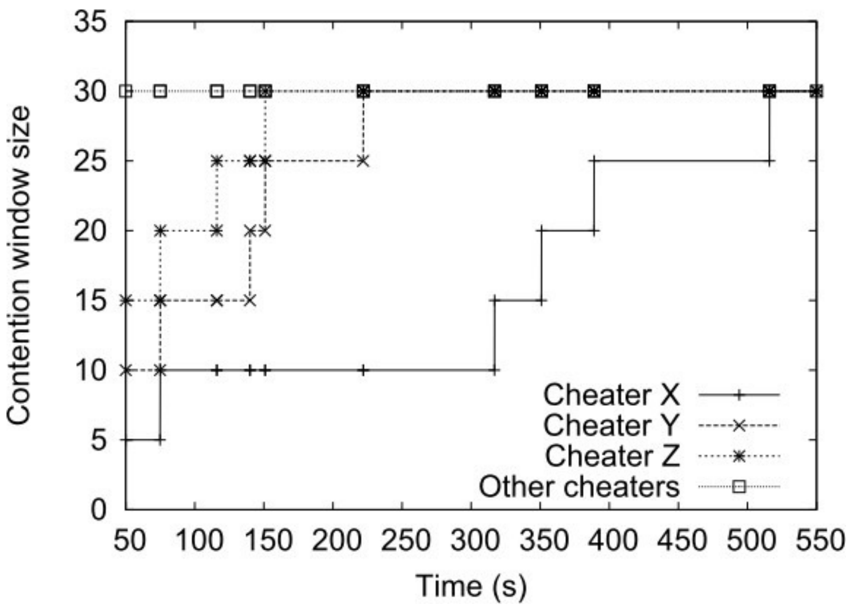




Distributed/Adaptive Cheating

- 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

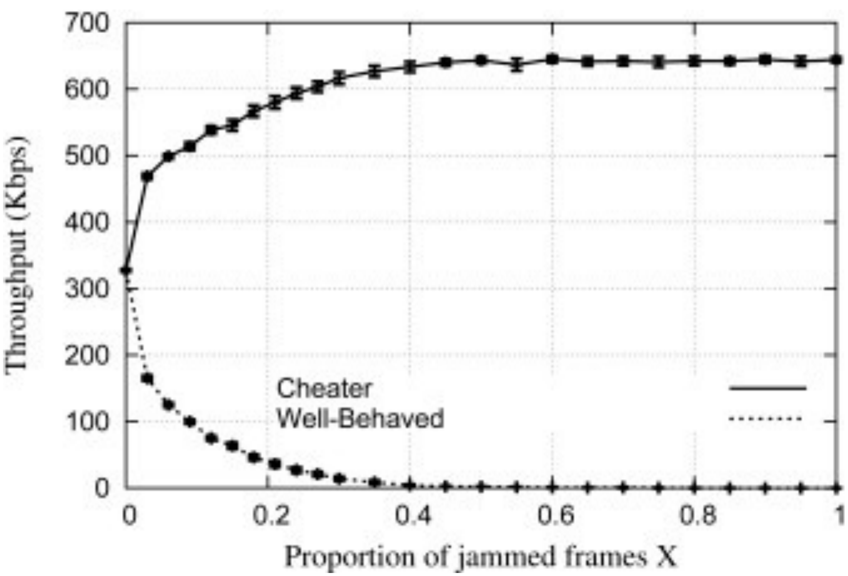


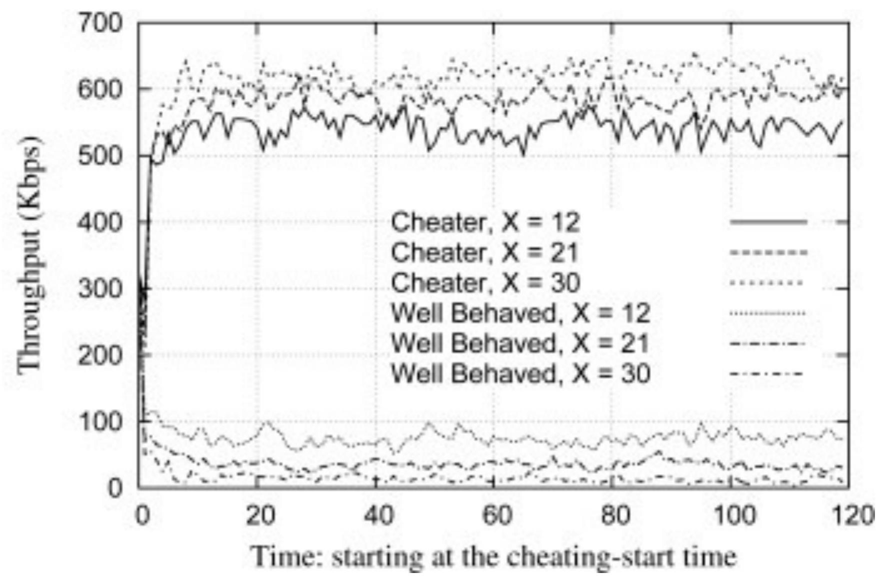


Detecting Greedy Behavior

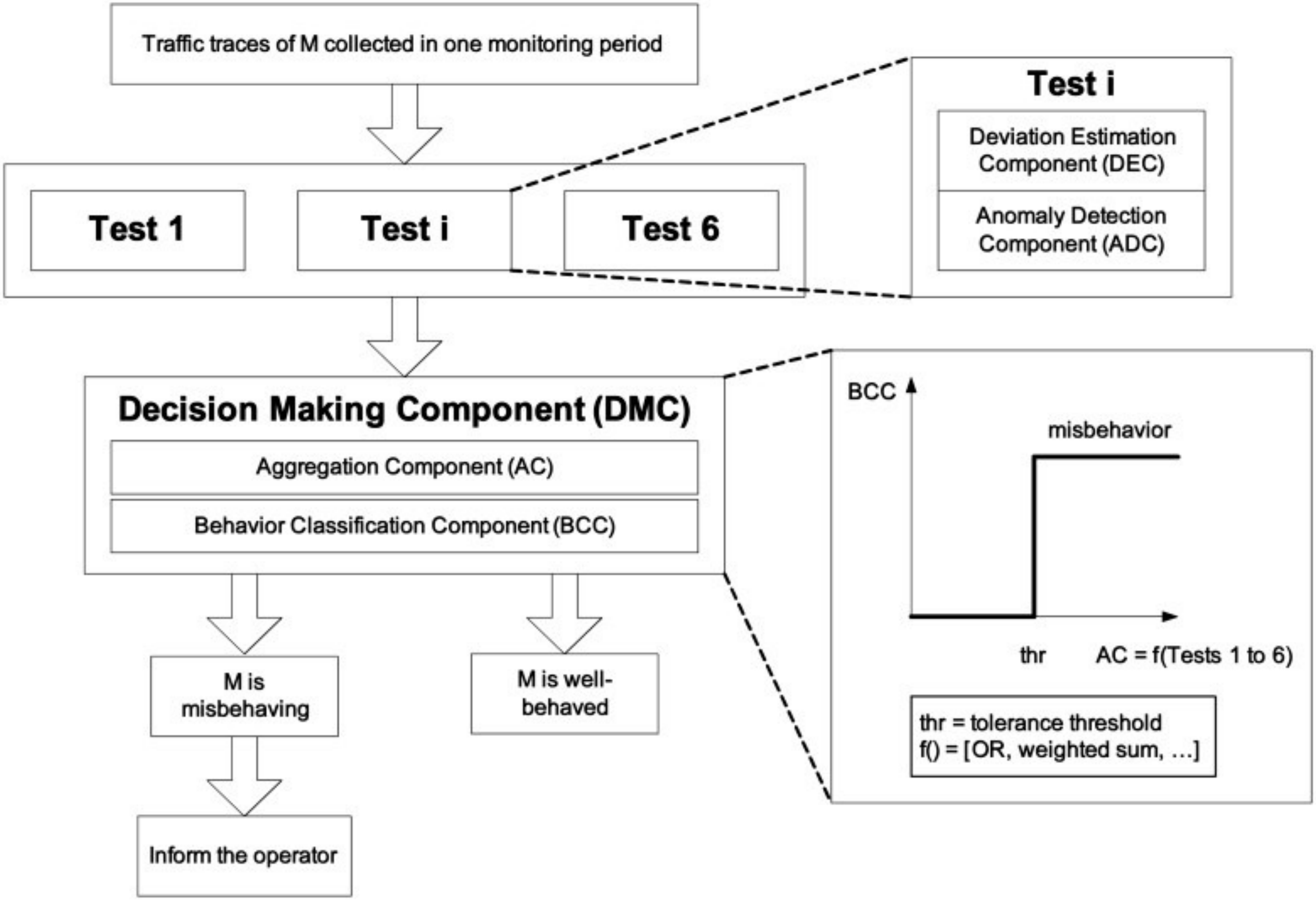
[Raya et al., 2006]

- 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



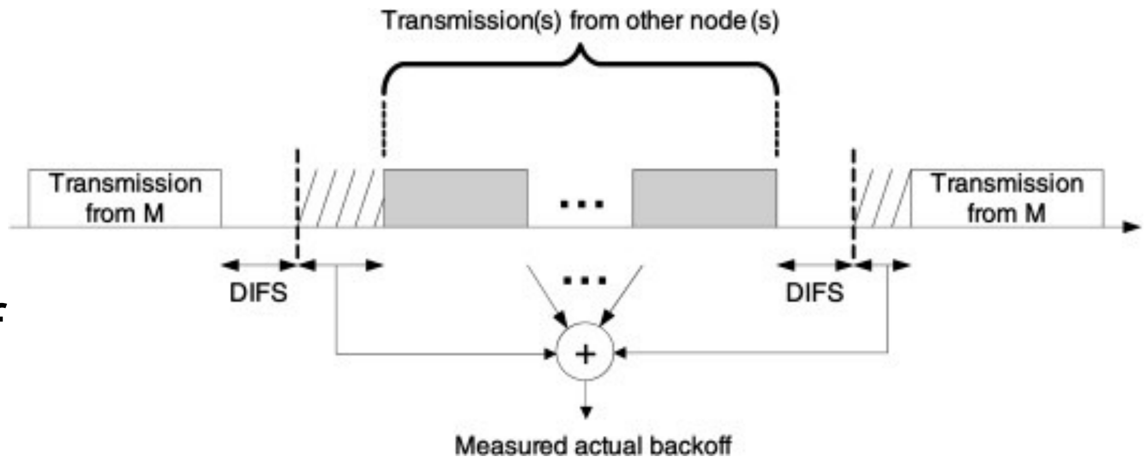


DOMINO Architecture

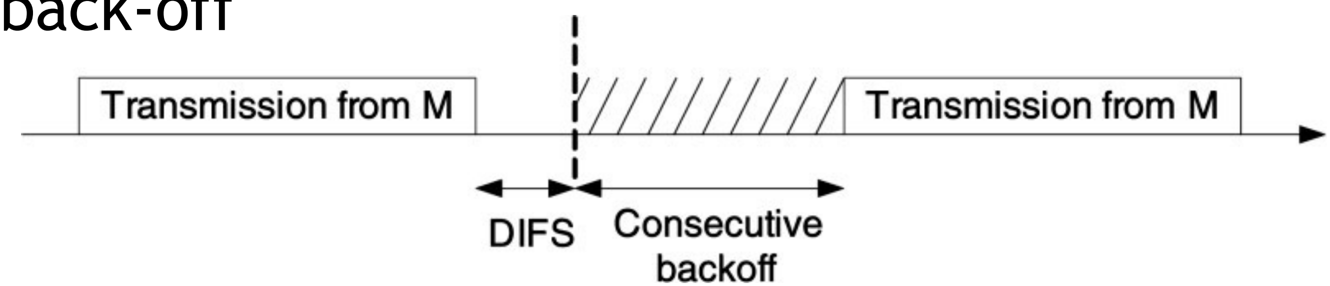


Behavior Tests

- 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



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

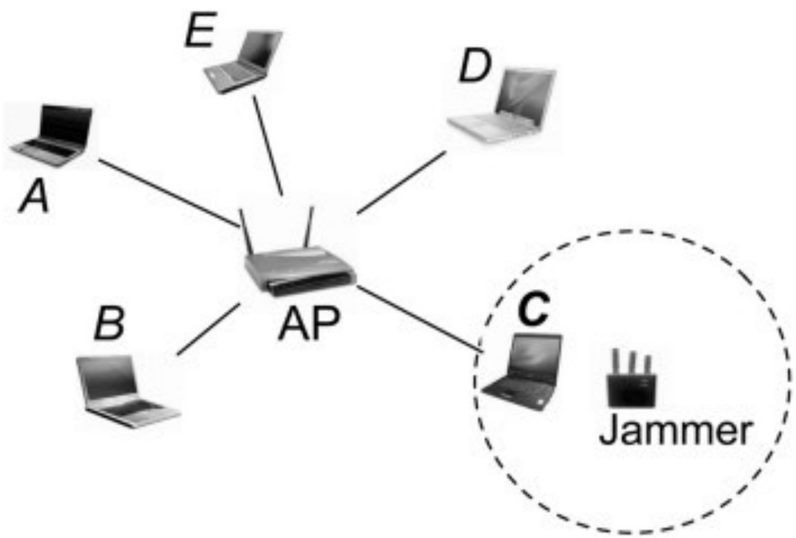
Implicit Jamming in 802.11

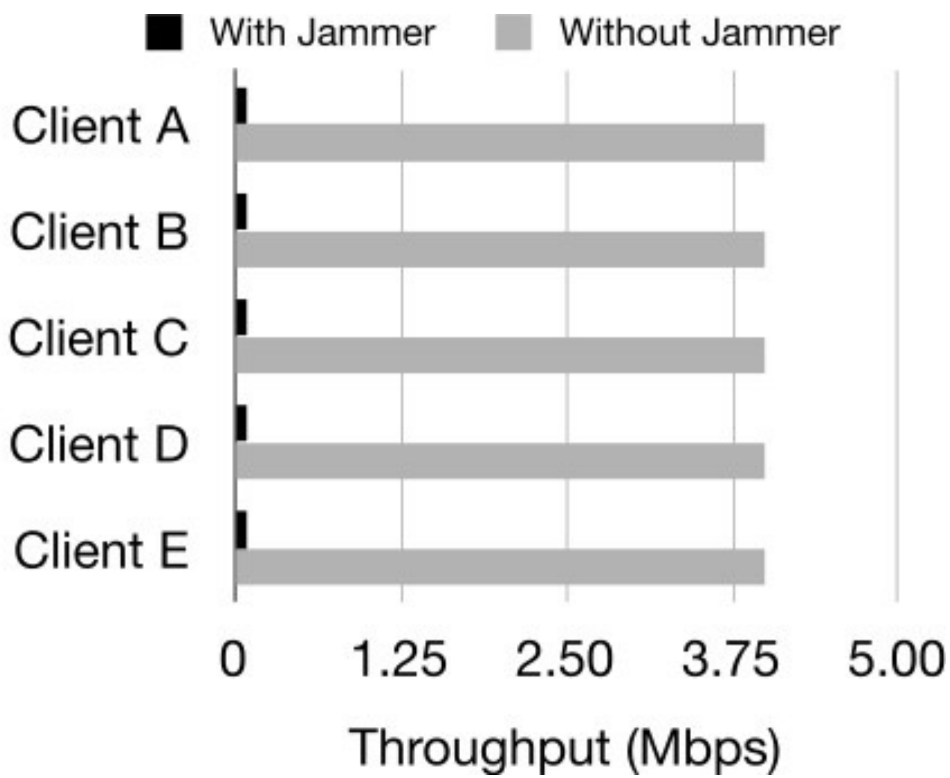
[Broustis et al., 2009]

- 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