www.networks.imdea.org

The Effects of Frame Collisions in 802.11-based Mesh Networks

Vincenzo Mancuso
Staff Researcher

instituto madrileño de estudios avanzados

Problem

Wireless network utilization is not efficient

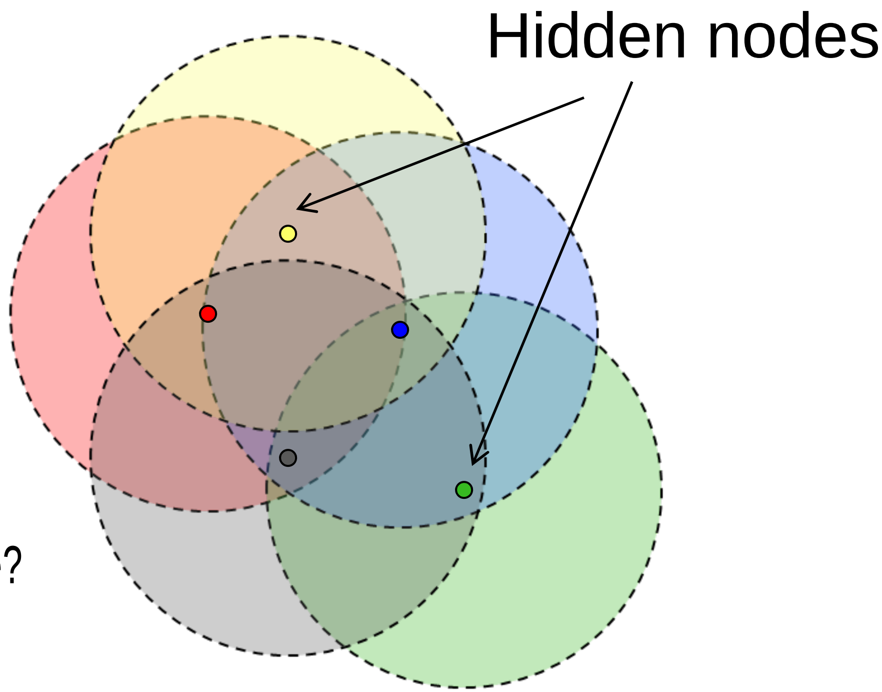
- 802.11 networks are designed for 11 to 54 Mbps
  - 108 Mbps with Turbo Mode (use two channels)
- 802.11-based Mesh Networks can hardly reach few Mbps

Why? --- lack of coordination (no scheduling)

- *Uncoordinated nodes generate many collisions*
  - Unresponsive traffic flows (UDP) do not slow down
    - Packet dropping
  - Reactive traffic flows (TCP) slow down *unfairly*
    - Some nodes have better performance than others

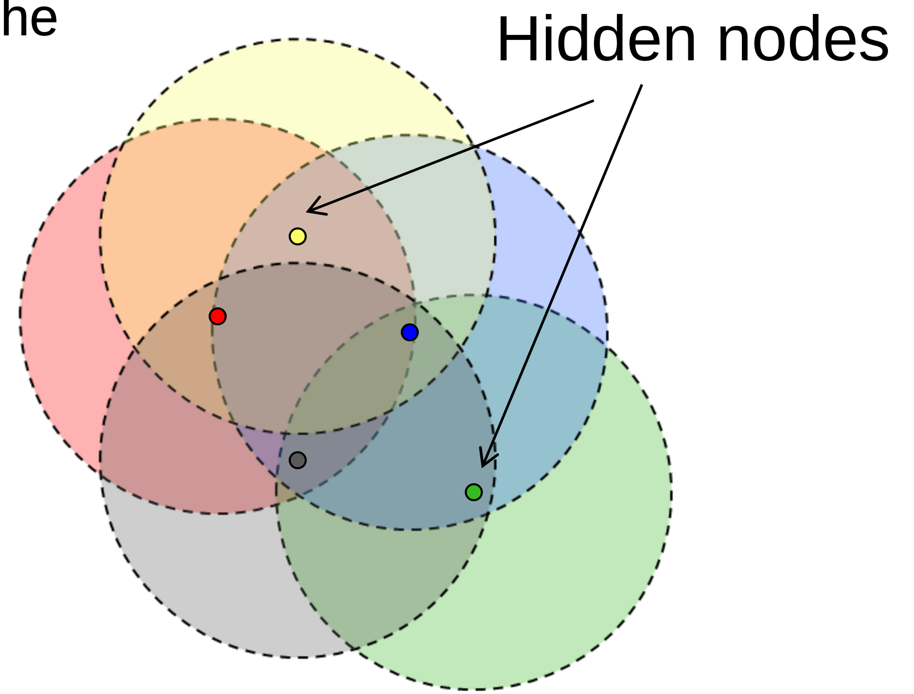
Who should transmit?

- A receiver can listen to multiple transmitters…
  - … but not in parallel!
- Collisions happen if wireless nodes are not able to coordinate
- CSMA random access (802.11):
  - A node listens to the channel
  - If the channel is idle, the node transmits with some probability *p*
- What if two nodes are not in range?
  - *Hidden nodes*
  - **E.g., there is a problem if hidden nodes want to transmit towards the same destination**

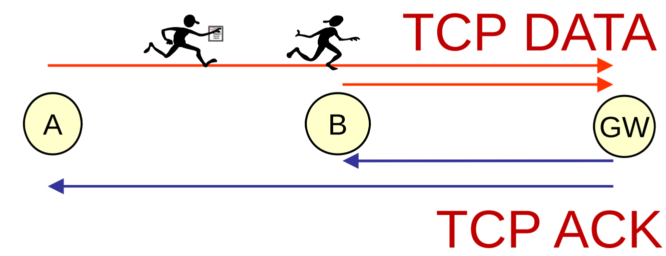


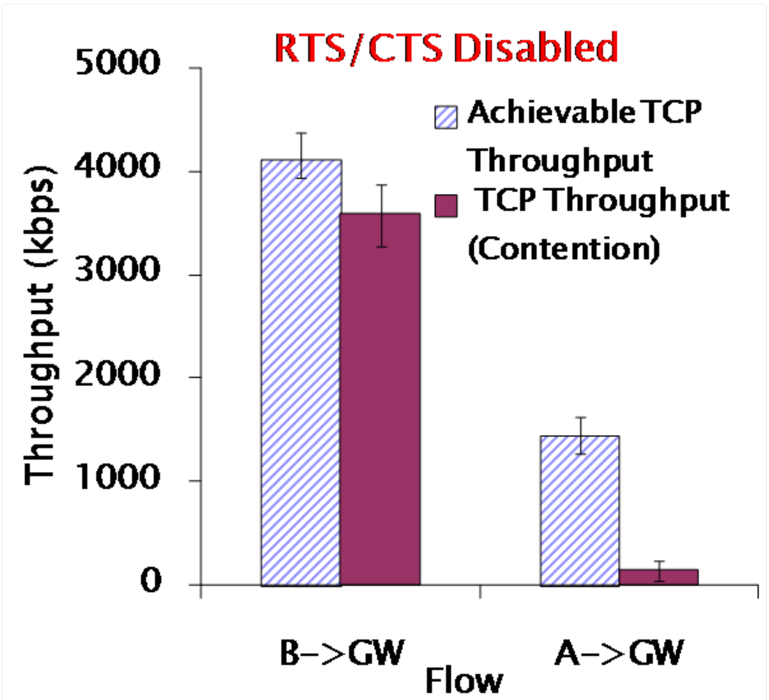
Who should transmit?

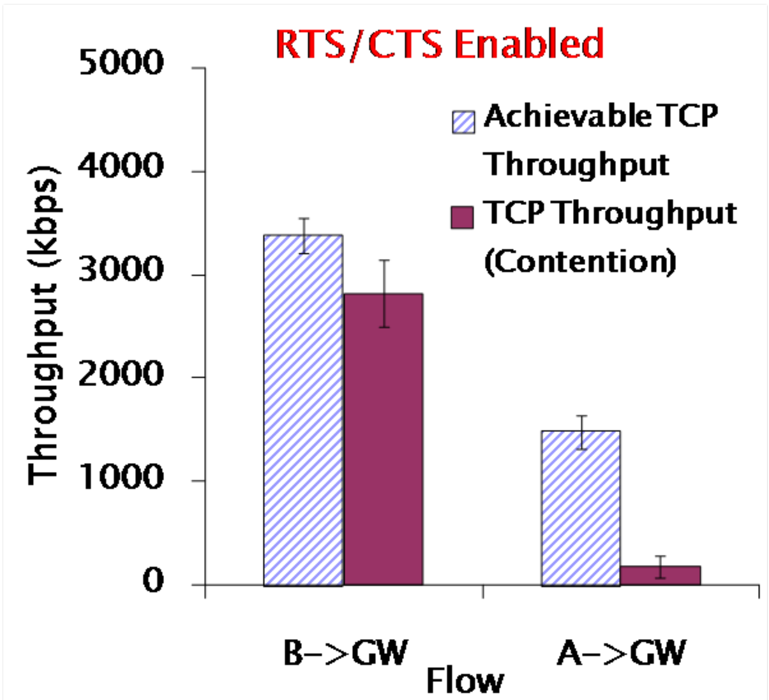
- Many works focused on modeling the optimal value for *p*
  - Graph-theory
  - Markov chains
  - ...
- Many works on experiments are available
  - Simulations
  - Real testbeds
- Results strongly depends on
  - Topology
  - Traffic matrix
  - Protocols (MAC and TRANSPORT)



Is there any real problem?

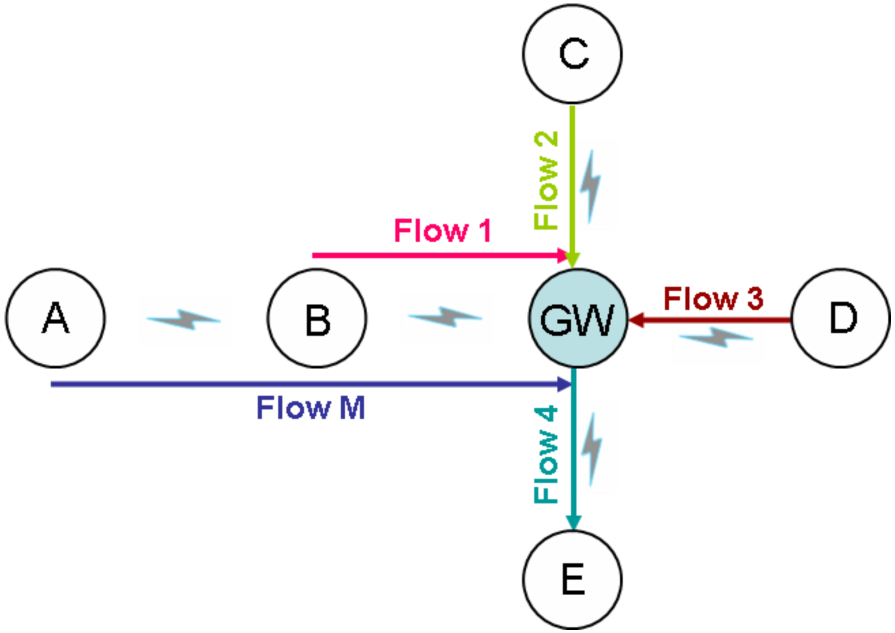




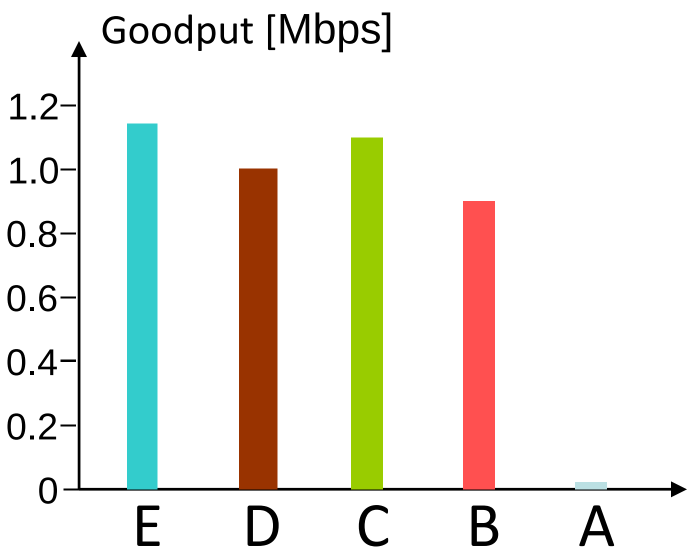


TCP Starvation in 802.11-based mesh networks!!!!

Non-fully backlogged TCP flows



Achievable goodput ~4.5Mbps



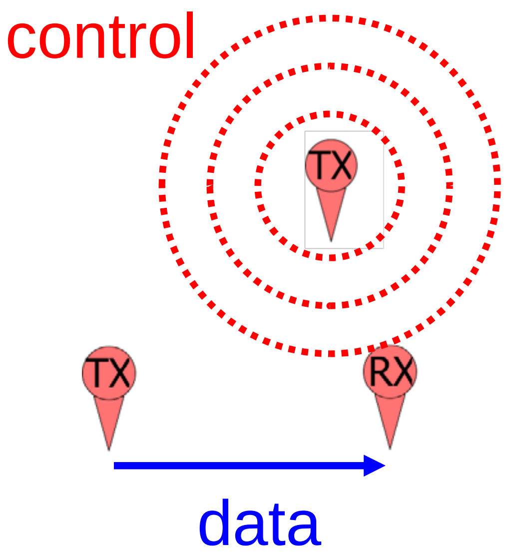
Aggregate one-hop load ≥ GW capacity

⇒Severe Throughput Imbalance

Location matters!

One-hop nodes have almost strict priority

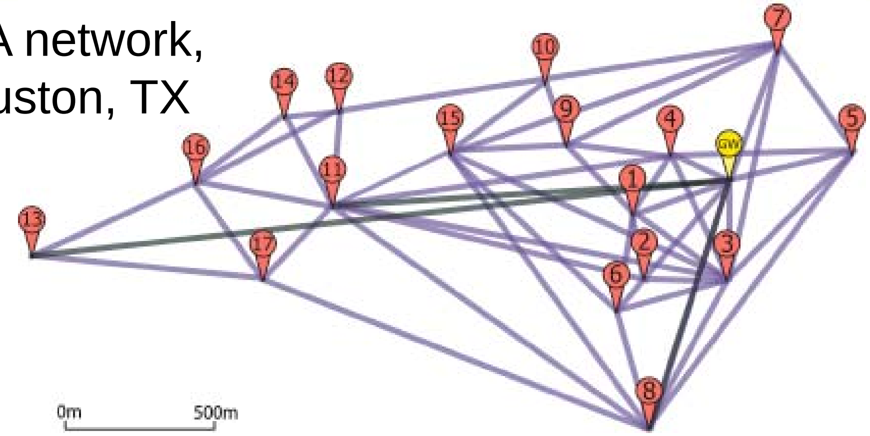
What if collisions involve uncorrelated traffic?
(e.g., data flows and control flows)



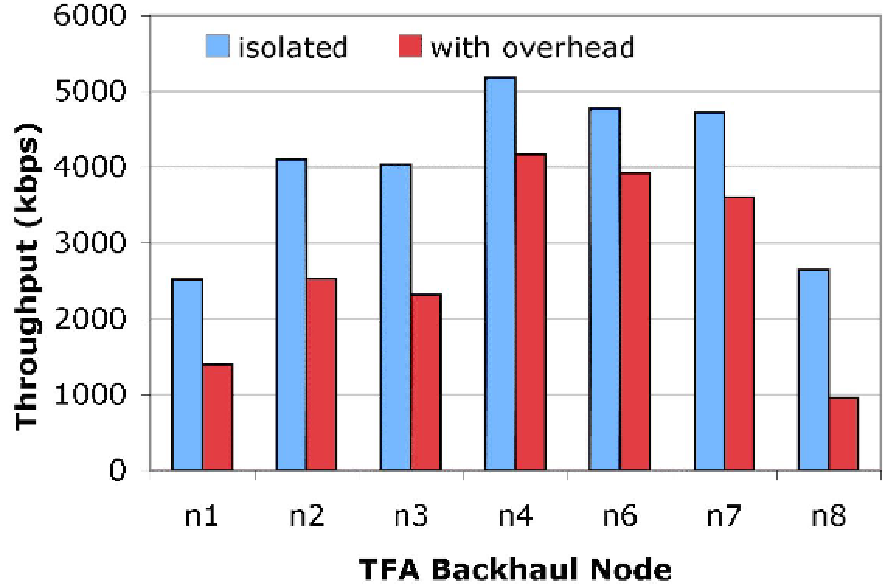
- Small control packets collide with big data packets
- This way, overhead (OH) can cause a throughput reduction witch is NOT proportional to the OH traffic!

Overhead Effects

TFA network, Houston, TX



- Experiment on a 802.11b operational mesh network
- Overhead of 80 kbps (approx. 10 kbps/node)
- Vastly different performance with and without overhead
  - 800 to 1800 kbps degradation
  - *10-20 times* injected overhead
  - Heterogeneity of effect mainly due to hidden node presence (*capture effect* and *autorate fallback*)

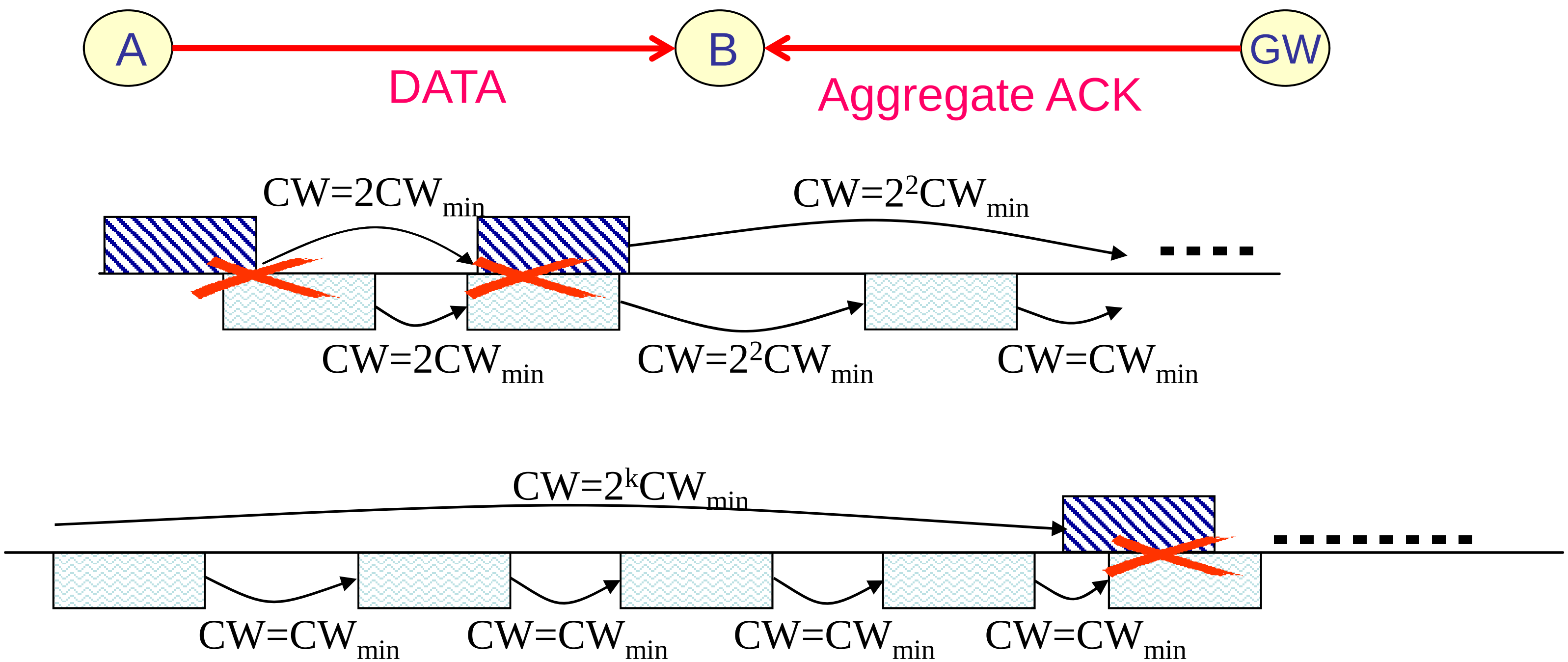


Understanding spatial bias

Compounding effect of many factors:

- OH reduces the capacity in a heterogeneous manner
  - Basically depends on hidden nodes and capture effect
- Considering the remaining capacity, MAC and TCP can not cooperate very well…
  - (i) The collision avoidance in medium access protocol induces bi-stability in which pairs of nodes symmetrically alternate in capturing system resources
  - (ii) The congestion control in transport protocol induces asymmetry in the time spent in each state and favors the one-hop flow
  - (iii) High penalty due to cross-layer effects in terms of loss, delay, and consequently, throughput, in order to re-capture system resources

(i) 802.11-MAC bi-stability (symmetric effect)

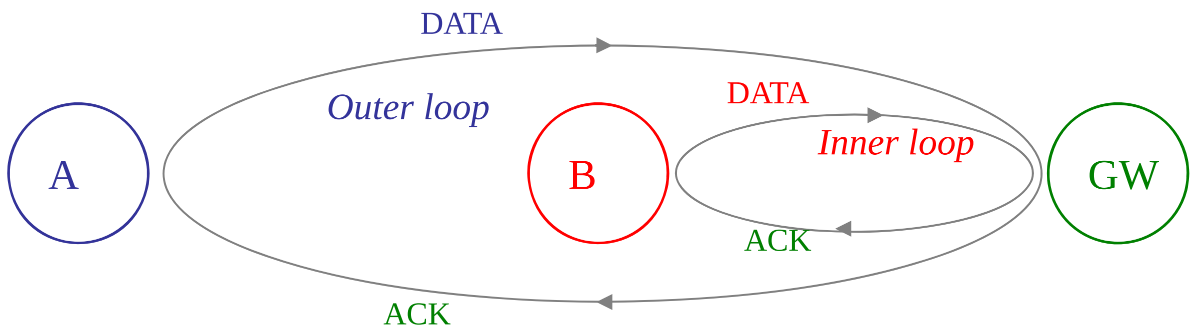


Due to lack of coordination:

- Bi-stable state: either A transmits and GW is in high backoff, or GW transmits and A is in high backoff
- *Success* state and *fail* state alternate

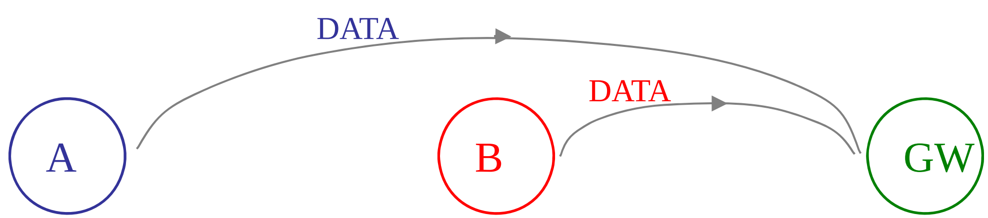
(ii) Asymmetry induced by TCP

- Two nested transport loops and sliding windows

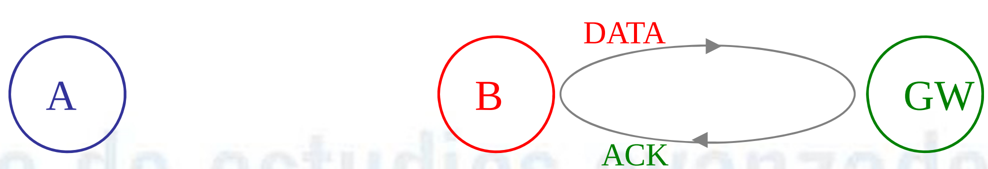


- *Asymmetric impact* of multipacket capture

- (A, B) burst:
  the burst size is limited by:
  - TCP window size

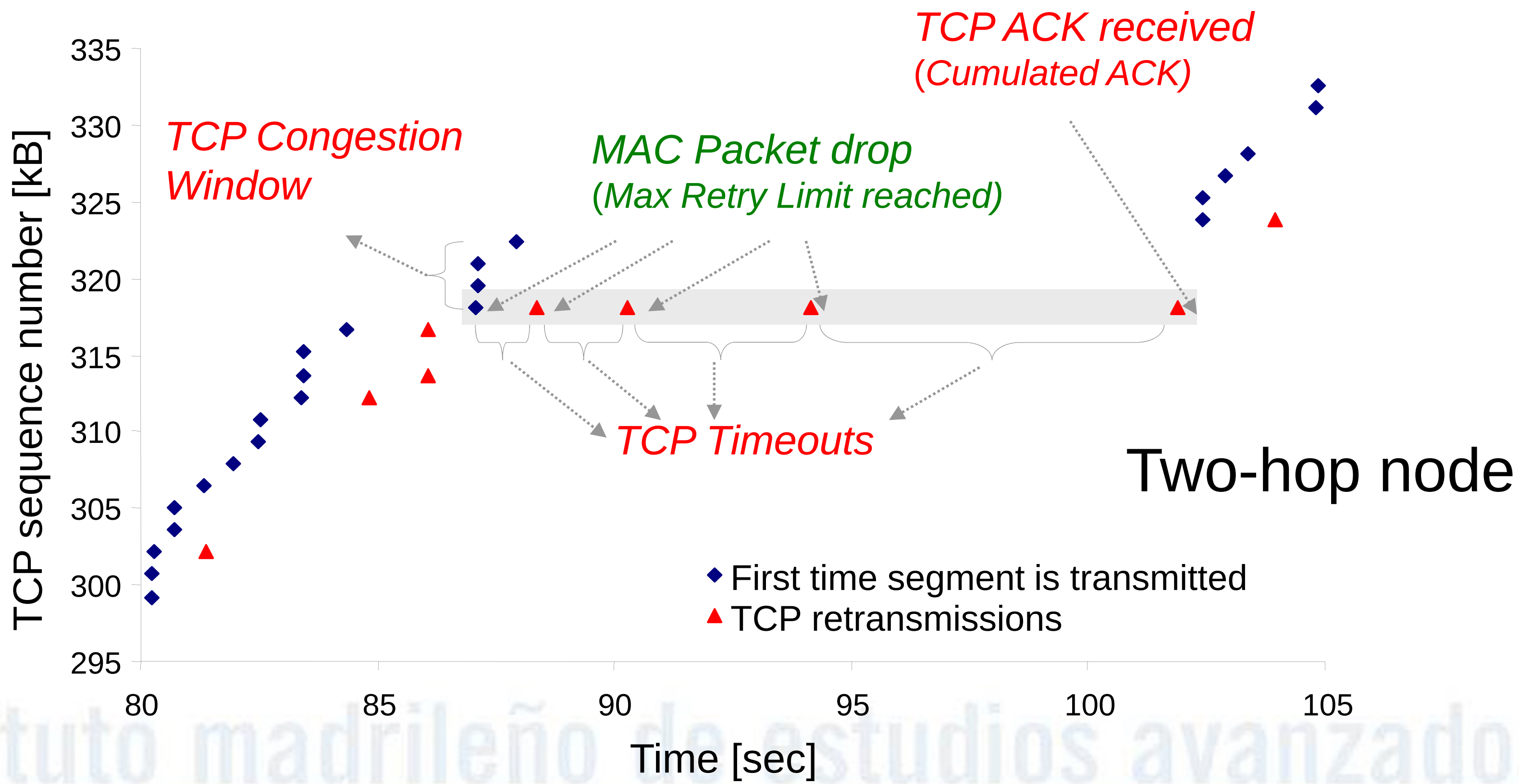


- (GW, B) burst:
  self-sustaining loop:
  - TCP ACK are generated



(iii) Penalties (asymmetric)

- Node GW incurs small penalty: short duration of fail state *but* long packet bursts
- Node A incurs high penalty: long duration of fail state *and* low offered load, high backoff & multiple TCP timeouts



Does spatial biasing trigger other mechanisms?

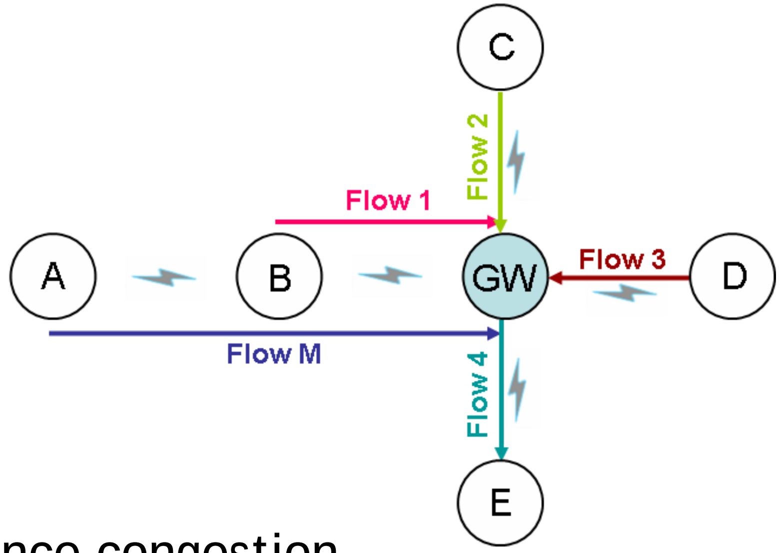
Three properties of a mesh:

1- *Distributed Approximate Priority:*

One-hop nodes have Access Priority

2- *Congestion Indication*:

Any congested link indicates congestion around the gateway



- gateway *airtime* must be saturated
- gateway congested → all flows experience congestion

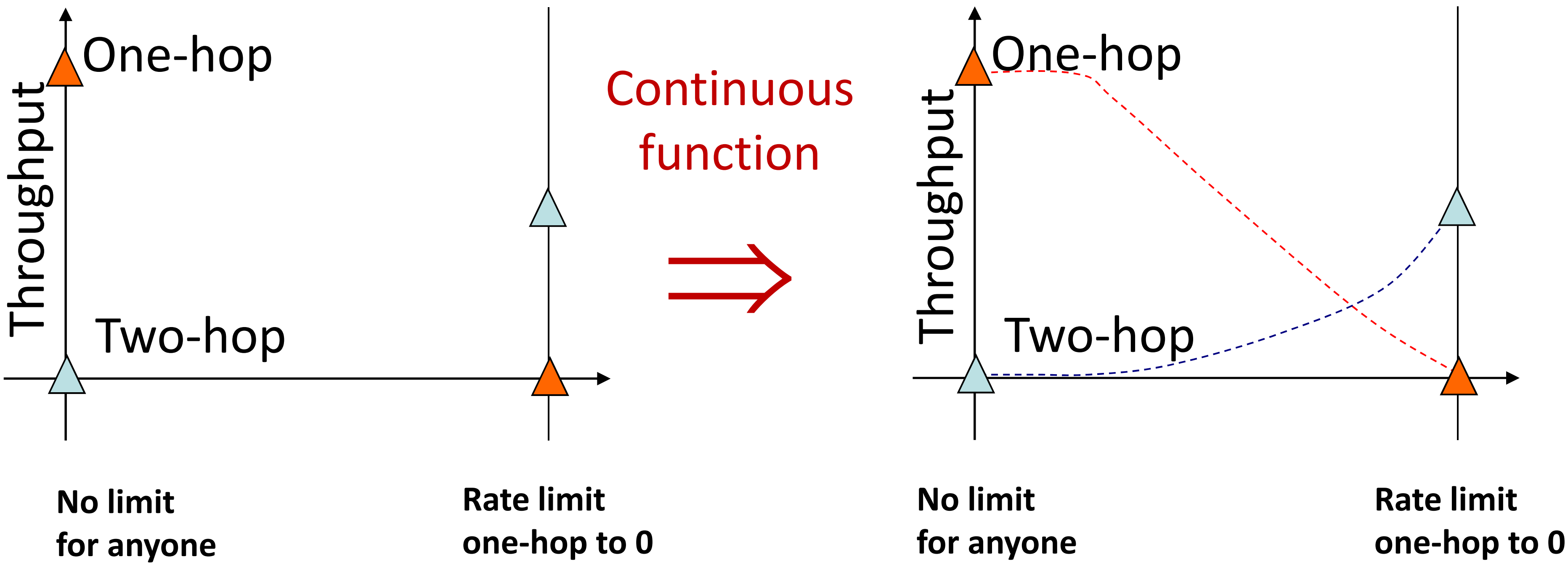
3- *Control by proxy*: Enforcing free airtime in the gateway neighborhood gives multi-hop nodes transmission opportunities

- one-hop controls multi-hop
- also spatial reuse enhanced

Counter-bias policy

- All nodes that are directly connected to the gateway should decrease their access probability *p*
- *E.g.: increase the contention window*
  - Simple to implement- no overhead or message exchange between nodes
  - Compliant with IEEE 802.11e EDCA
- Or *use rate limiting*

Rate limiting at one-hop?

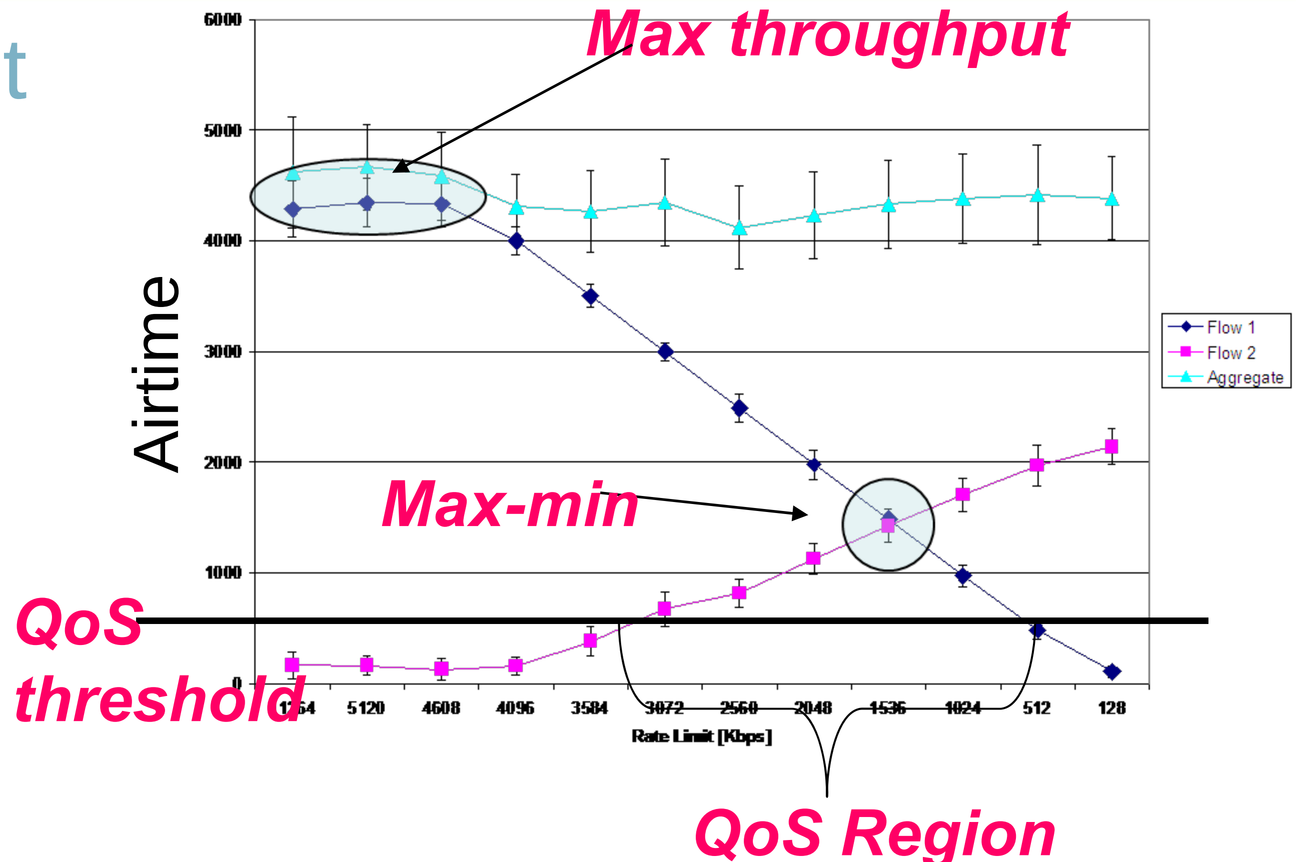


Idea: One-hop gives, two-hop takes (*not 1:1*)

⇒Based on the objective, find a working point

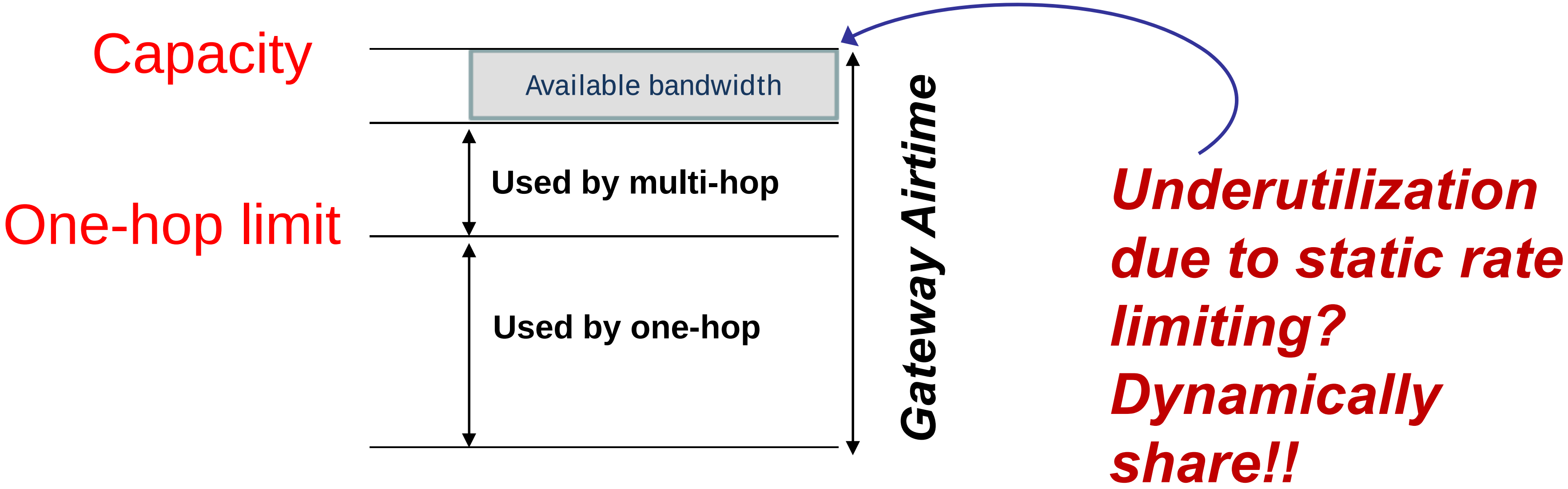
Working point

- Fairness
  - Max-min
  - Proportional
  - Time-share
- Utilization
  - Max throughput
- QoS
  - Reservation
  - Delay bound



- Static rate limiting is a good solution for fully backlogged scenarios, not for variable traffic
  - E.g., static rate limit policies can be inefficient

Gateway Airtime Partitioning (**GAP**) through Elastic Rate Limiting



- *Elastic rate limit* operates a gateway airtime *partitioning*
  - Guaranteed one-hop bandwidth
  - Multi-hop bandwidth
  - Unused bandwidth

How much of the unutilized bandwidth should be reallocated?

- If **ALL** is reallocated to one-hop nodes ⇒
  No way for multi-hop nodes to rejoin/get the bandwidth back
  - Causes collisions
  - Severe Throughput Imbalance
- If **ZERO** is reallocated to one-hop nodes ⇒ Static
  - Underutilization

Idea: make room for ***signaling*** from multi-hop nodes: just leave a small bandwidth reserved for disadvantaged nodes (a small ***gap***):

$$B_D \leq \gamma U_{max} \ \ (B_D << U_{max})$$

GAP

Must leave free air-time at the gateway to let multi-hop nodes signal their demands

- *Objectives*
  1. Ensure minimum rates that would be guaranteed under saturation load conditions
  2. Fairly share unused resources among all competing nodes
- *Constraints*
  1. Disadvantaged-flow **Signaling Bandwidth** $B_D \leq \gamma U_{max}$
     - **Aggregate**
  2. Minimum Guaranteed **One-hop Rate** $(1-\gamma)U_{max}$
     - **Per-node (or aggregate)**

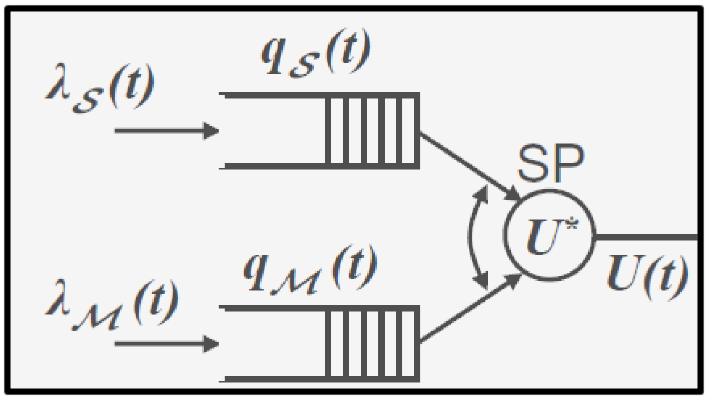
Using the GAP

- Given that
  - U_{max} is the maximum GW utilization (constant)
  - B_D is reserved to multi-hop traffic (signaling or data)
- Then only multi-hop traffic can drive the GW utilization beyond U_{max}-B_D
- Thus, one-hop nodes can *detect* multi-hop traffic by *observing* a GW utilization exceeding U_{max}-B_D

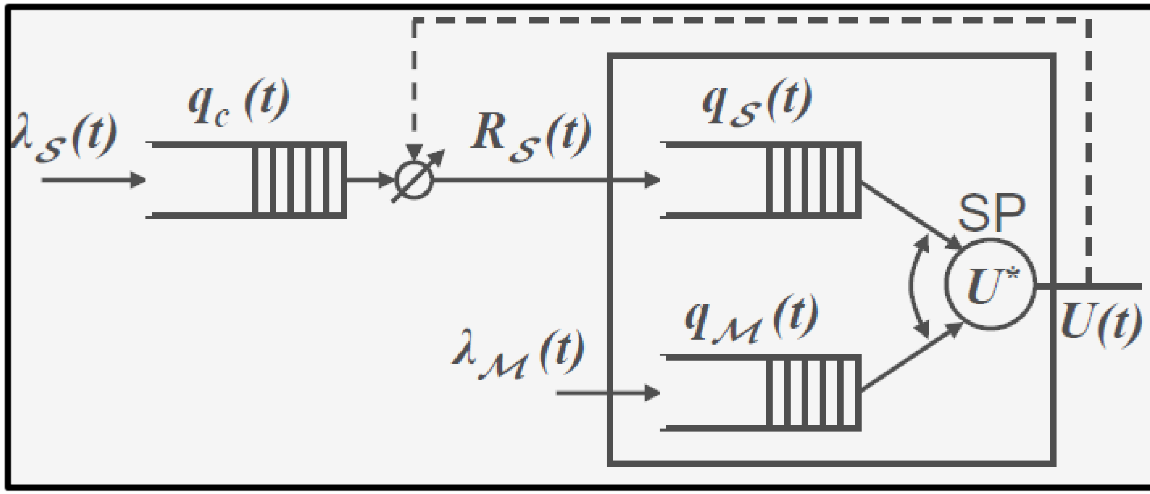
- E.g., each one-hop node estimates the GW utilization $U(t)$ and uses AIMD to adjust its rate limiting
  - **AIMD** based on the aggregate one-hop load $U(t)$ and the threshold U_{max} –B_D

if $U(t) < U_{max}$-B_D
    Increase rate limit *additively*
else
    Decrease rate limit *multiplicatively*

Analytical Model



Without control



Rate limiting controlled system

Theorem : The equilibrium points of the proposed rate control framework are stable

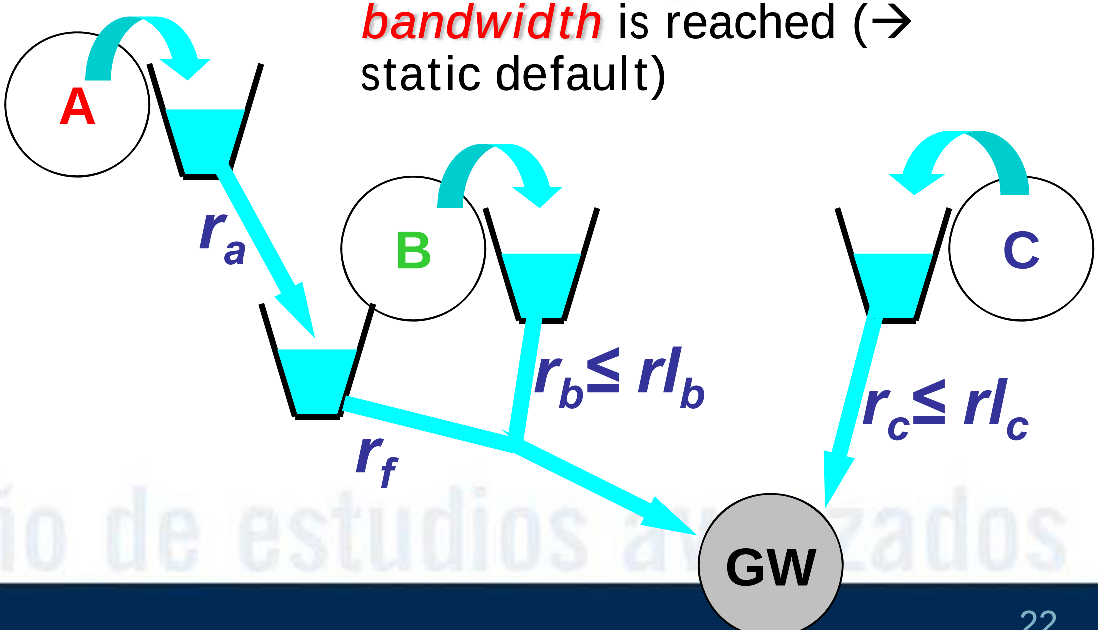
Corollary: Perturbations of the equilibrium points of the system are exponentially decaying with time with constant that is based on the equilibrium point

A fluid simulation of the fluid model

Rules of the game

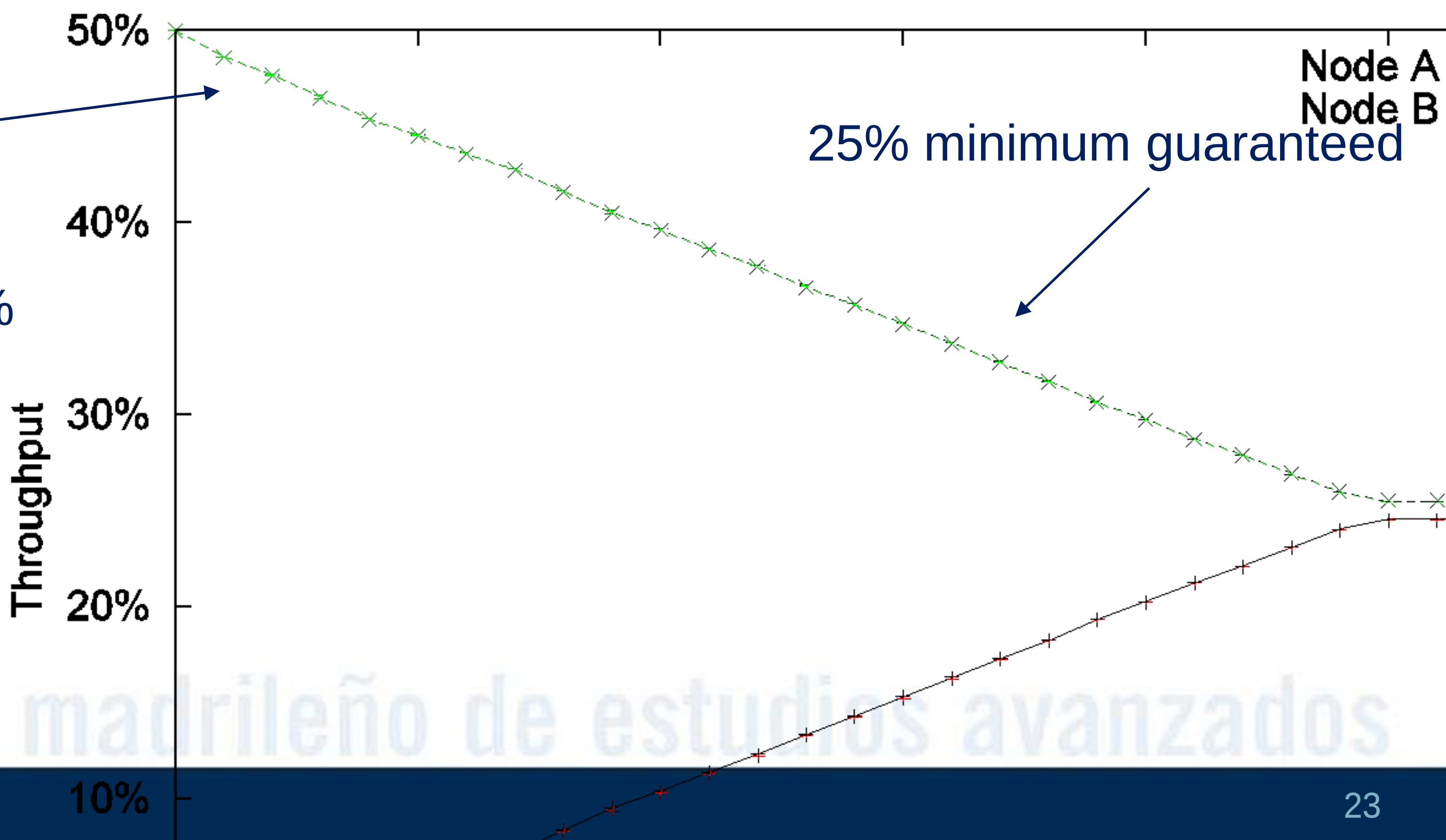
- Time is slotted
- **GW** can sink at most U_{max} units/s
- A can push to B only if B and C are idle (*strict priority*)
- Rate limiting at B and C (*elastic*)
- B + C aim at not exceeding U_{max} **-GAP** units/s
- B and C estimate the load at each slot (*with noise*)
- B's (C's) estimate is allowed to exceed U_{max}**-GAP** only if B (C) is below its guaranteed rate
- B and C adapt their rate limit at the beginning of the slot
  - Increase or decrease, unless the ***minimum guaranteed bandwidth*** is reached (→ static default)

The rate limit ***rl*** is tuned via an AIMD mechanism



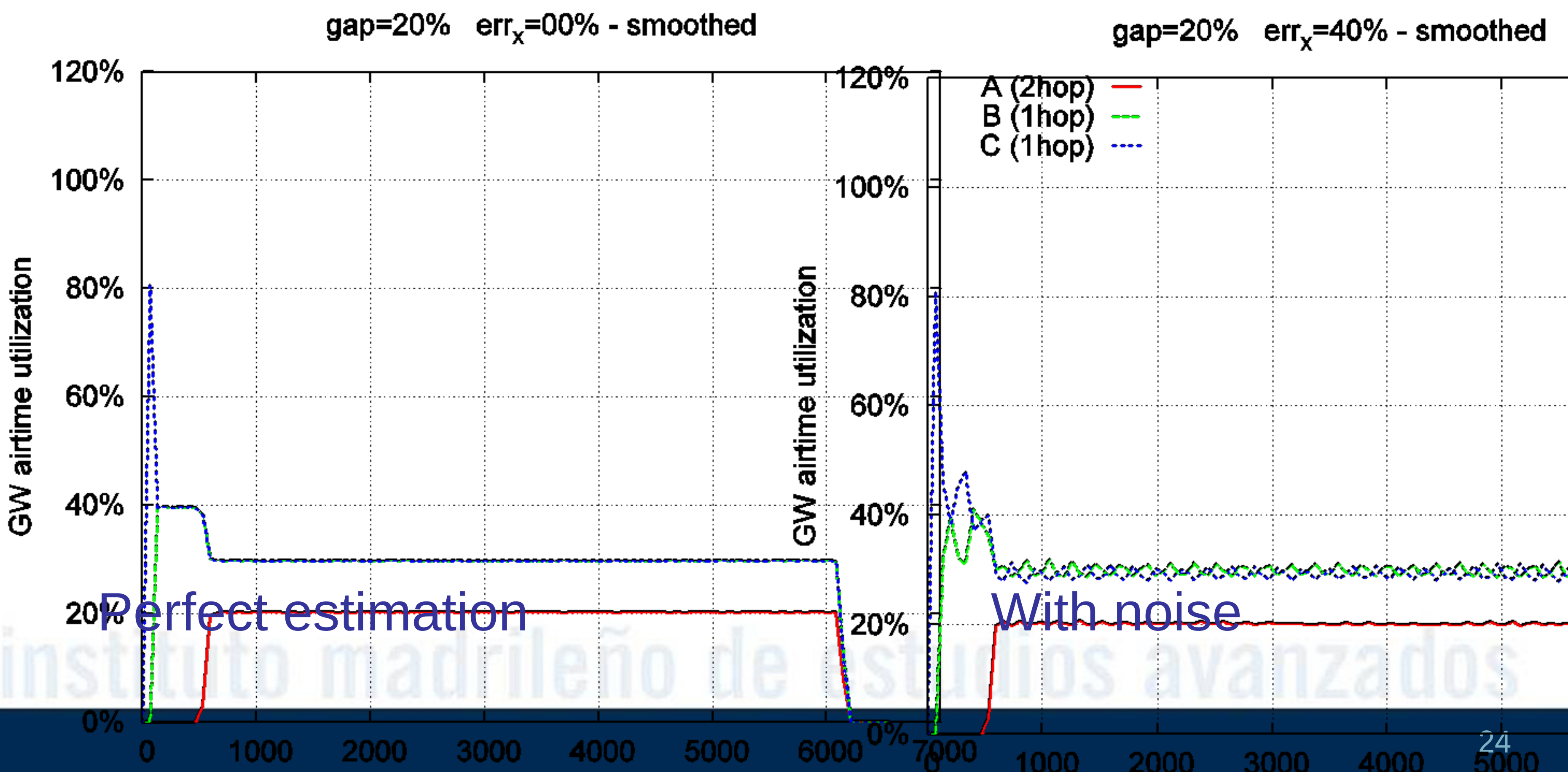
How much GAP?

$\text{throughput}_{\text{Node B}}$
=
$\text{Throughput}_{\text{NodeC}}$
⇒
One-hop nodes attain up to 100%



GAP is robust to noise

Low-pass filtered results



Is GAP better than scheduling?!

- In a distributed and low-overhead wireless scenario
  - *U(t)* estimates are prone to uncertainty
  - Distributed *U(t)* estimation is not synchronous
  - Centralized *U(t)* estimation can be delivered in different time instants to different one-hop nodes

  → distributed scheduling strategies cannot converge under these assumptions

    → V. Gambiroza, B. Sadeghi, and E. W. Knightly, "End-to-End Performance and Fairness in Multihop Wireless Backhaul Networks," in Proceedings of ACM MOBICOM, 2004
  - Conversely, GAP is robust enough for a distributed implementation and yields fairness

Practical issues in GAP implementation

- Distributed (One-hop nodes)
  - Local traffic estimations based on traffic overhearing
  - Prone to large estimate errors
- GW-operated
  - The GW counts the one-hop node traffic
    - Better quality estimate
  - AIMD triggered by GW commands
    - One bit only is needed (increase/decrease AIMD command)
  - Per-node commands
    - Use ACKs to convey commands to each one-hop node
  - Per-aggregate commands
    - Use ACKs or other control messages, e.g., BEACONS to transmit undifferentiated commands
- One-hop good also for UDP upstream
  - UDP downstream is not an issue

NO PACKET OVERHEAD

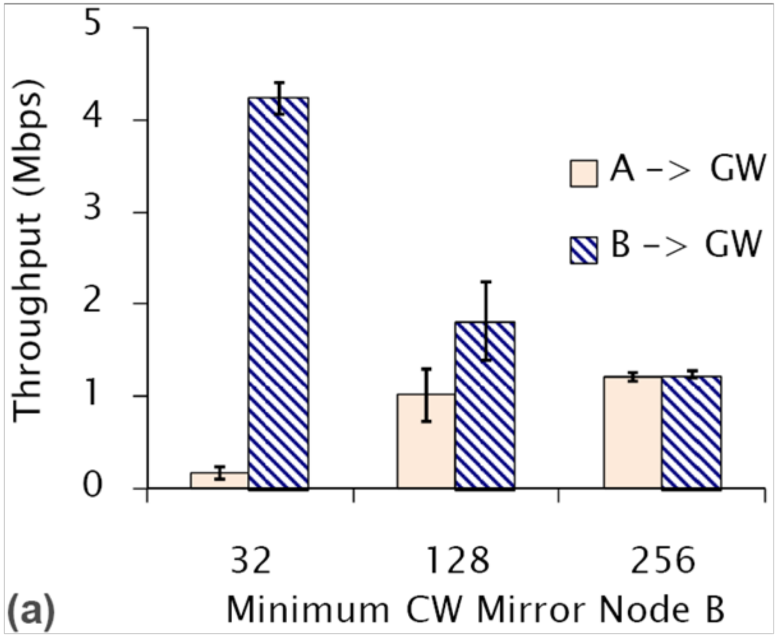
Conclusions

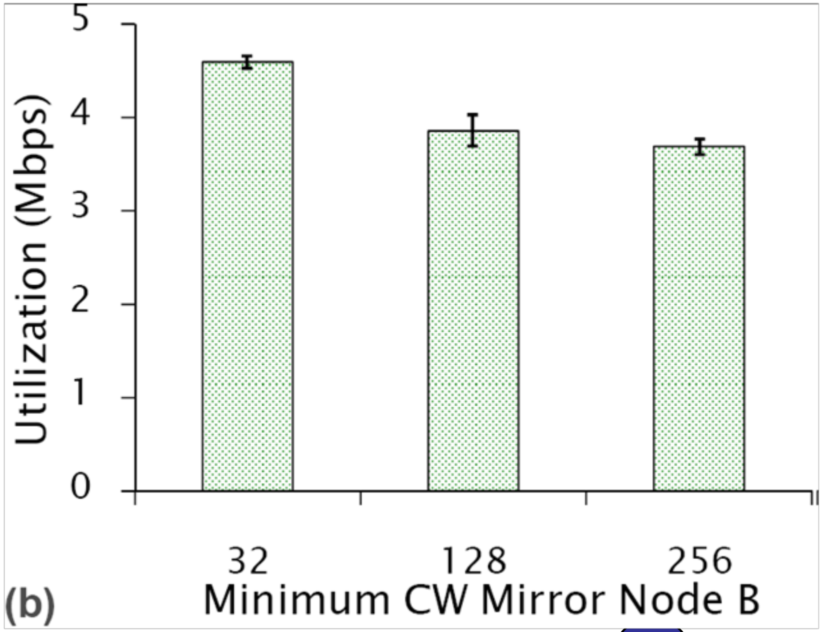
- One-hop rate limiting is enough to
  - Drastically reduce collisions
  - Avoid multi-hop starvation
  - Enable fairness
  - Control network throughput
- Elastic rate limiting is needed to better use the available resources
  - GAP protocol
- GAP performs better than scheduling
  - Robust in non-ideal scenarios
- GAP is easy to implement
  - In principle, just include an extra bit in the beacons

BACKUP

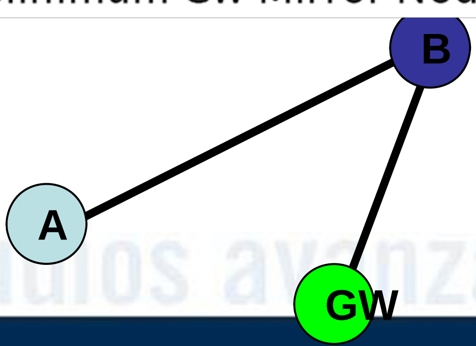
Experimental Validation

RTS/CTS Enabled

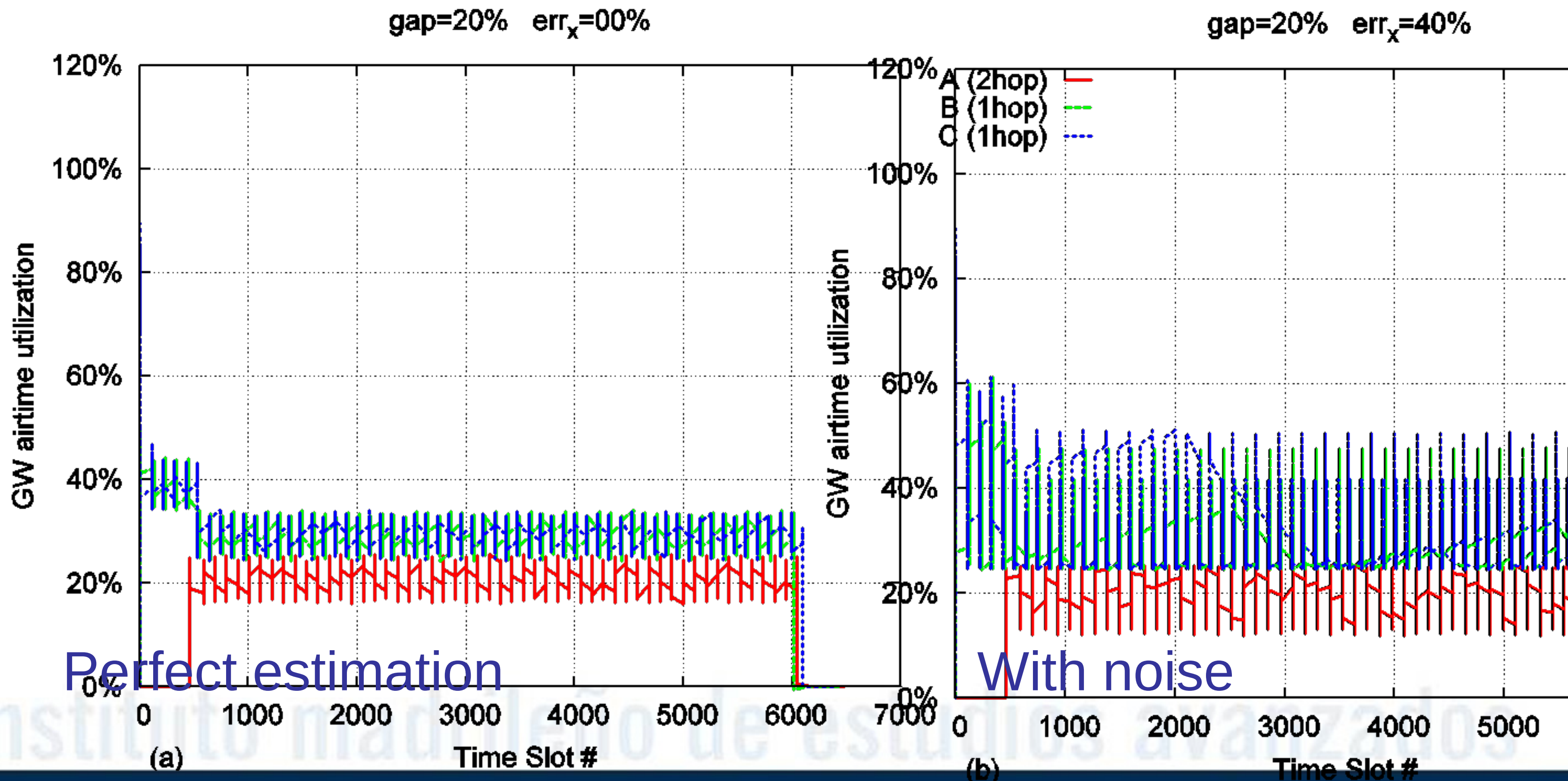




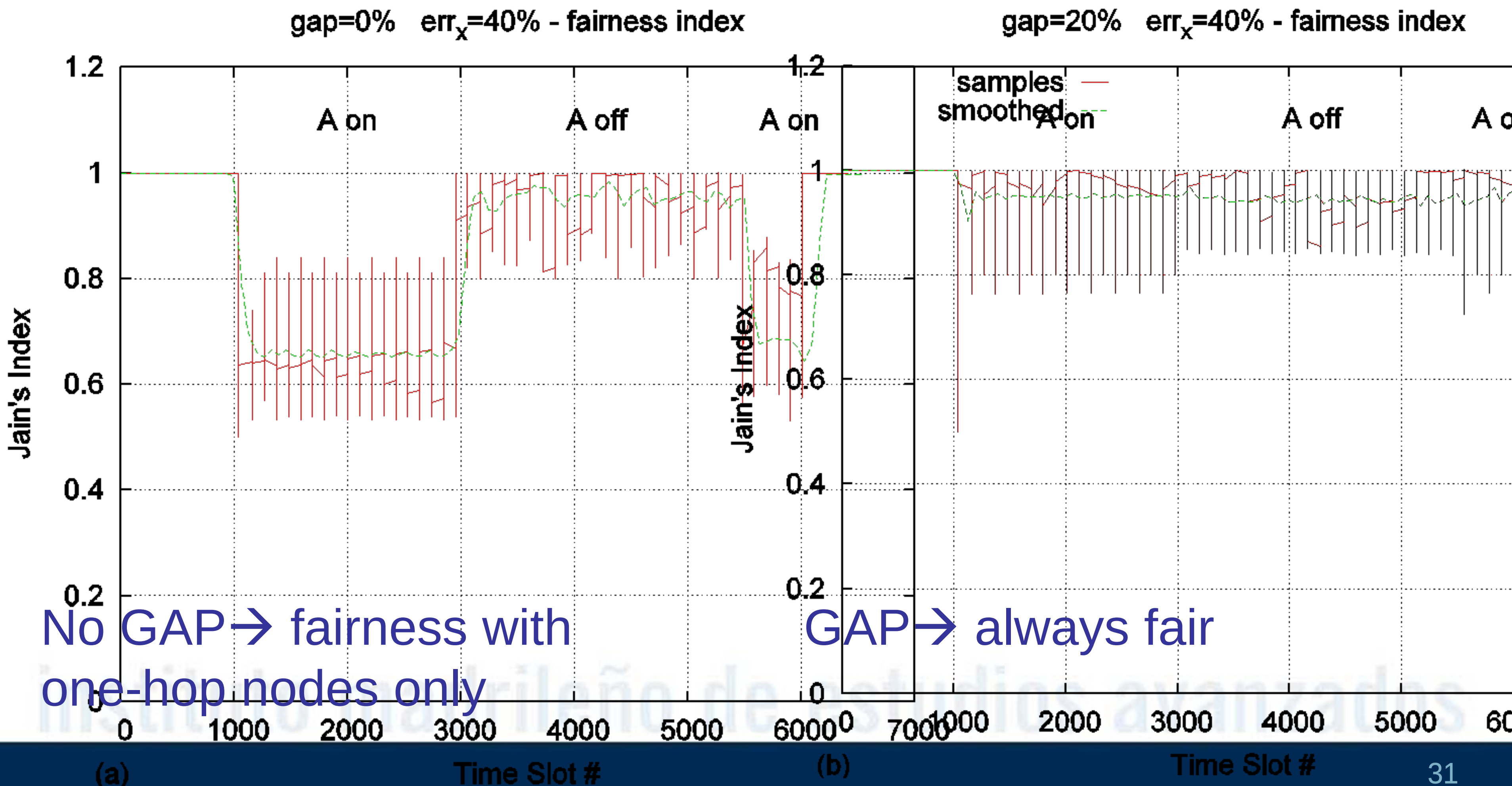
Fairness can be achieved (under different definitions)



GAP is robust to noise (1)



GAP and fairness



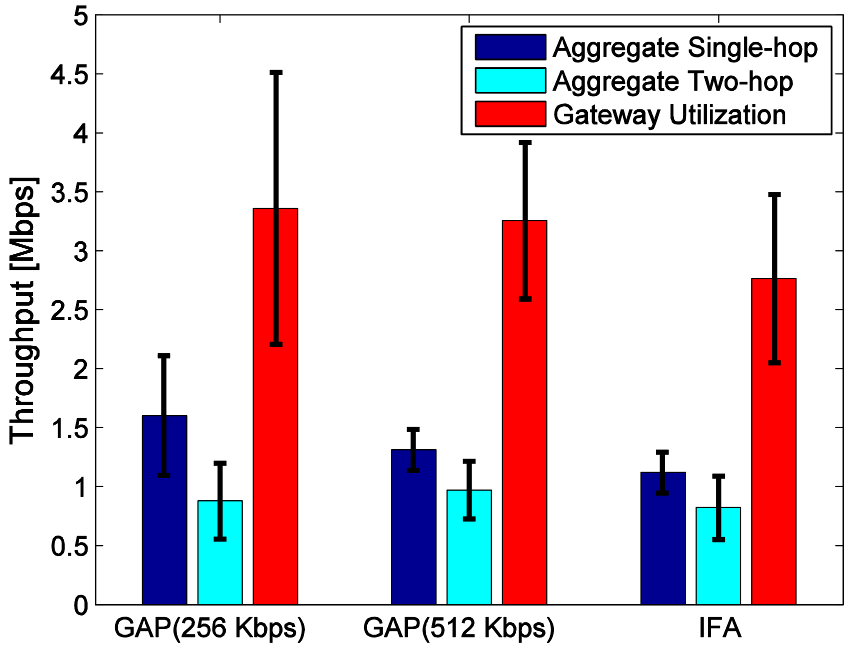
No GAP→ fairness with one-hop nodes only

GAP→ always fair

Simulation (NS2)

GAP vs. IFA
(Inter-TAP Fairness Algorithm - scheduled access)

8-branch tree



GAP reaches similar or better performances with no need of signaling message exchange