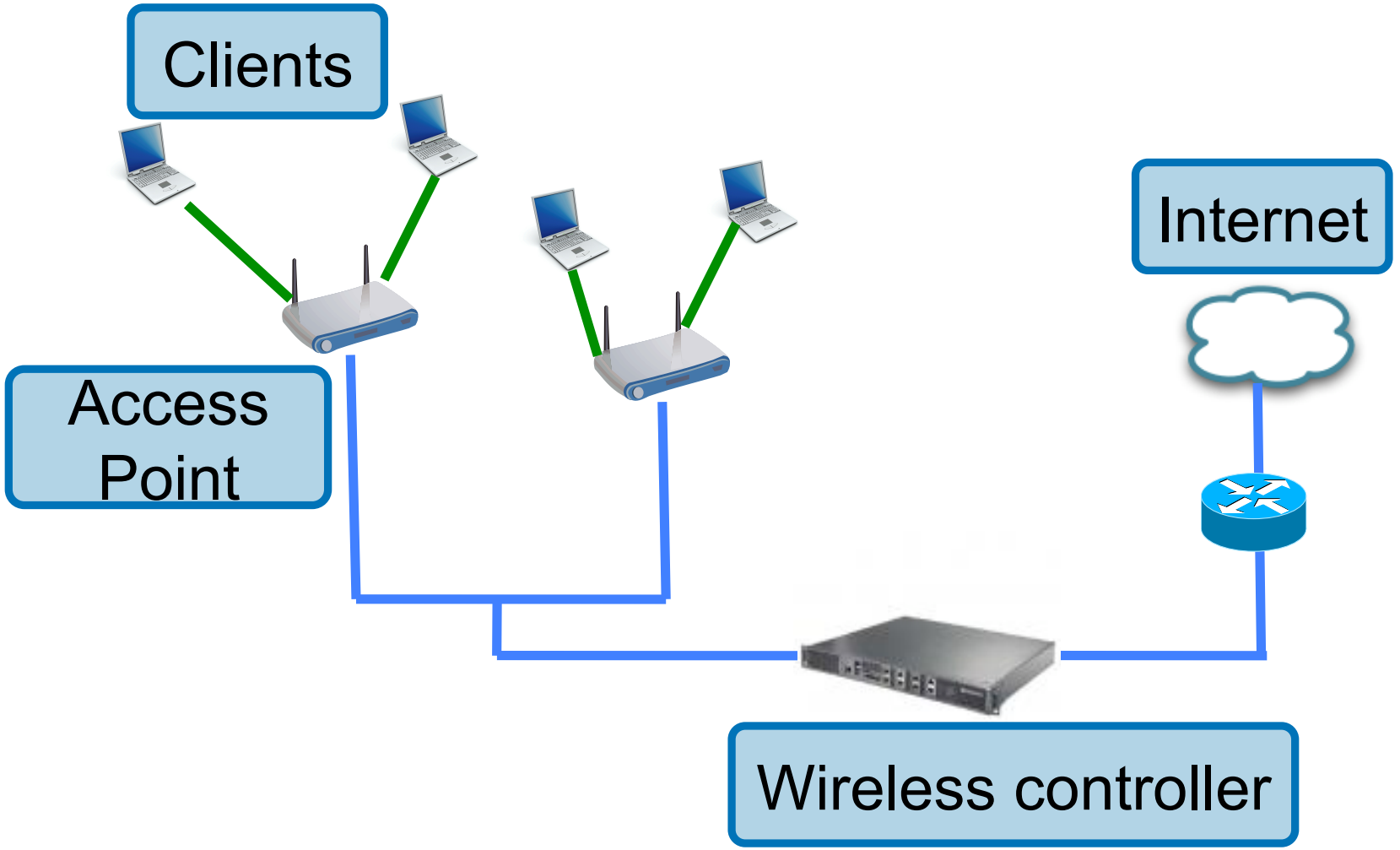
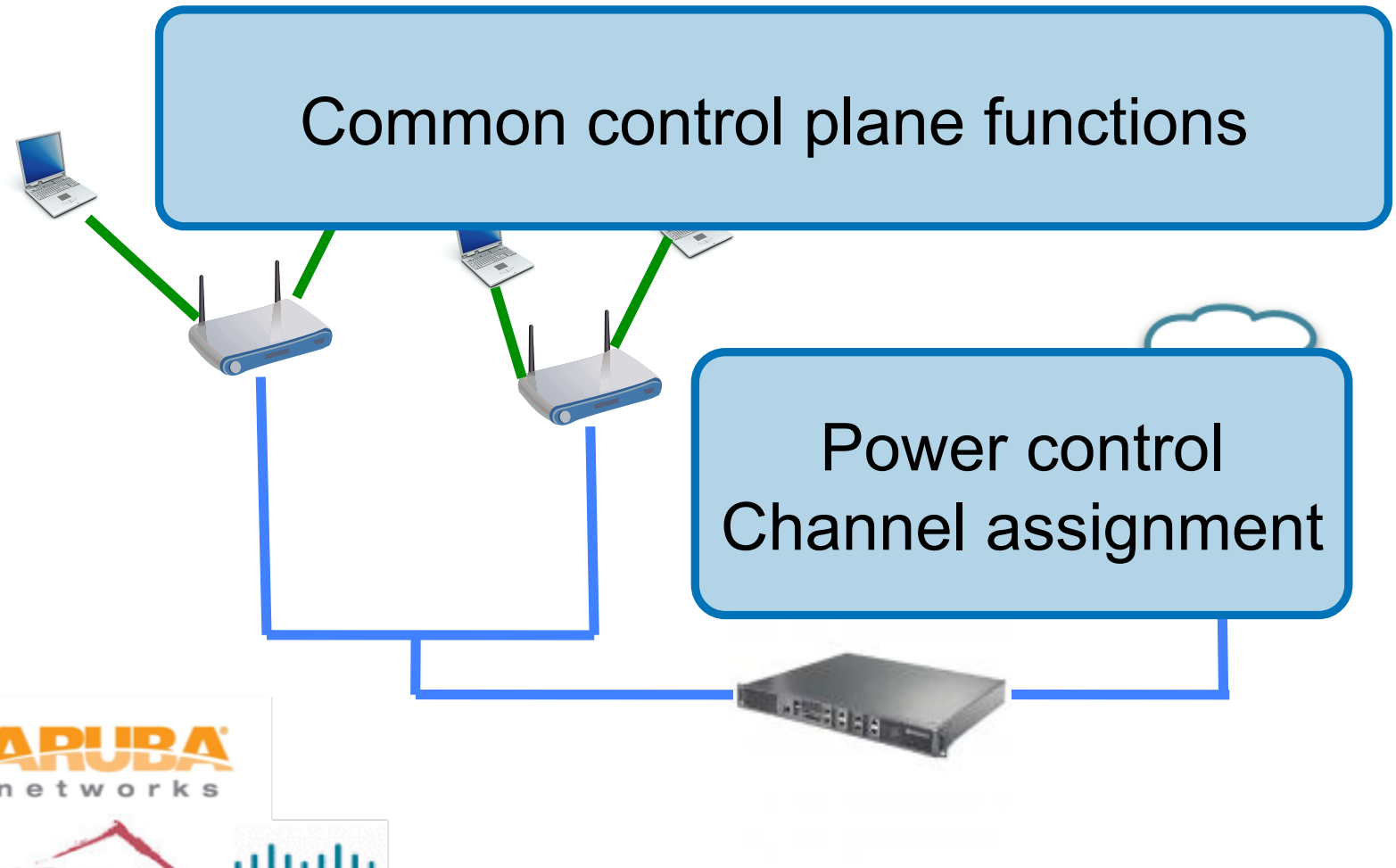
# CENTAUR: Realizing the Full Potential of Centralized WLANs Through a Hybrid Data Path

Suman Banerjee
(talk originally prepared by Vivek Shrivastava)

# Centralization of Enterprise WLANs



# Centralization of Enterprise WLANs

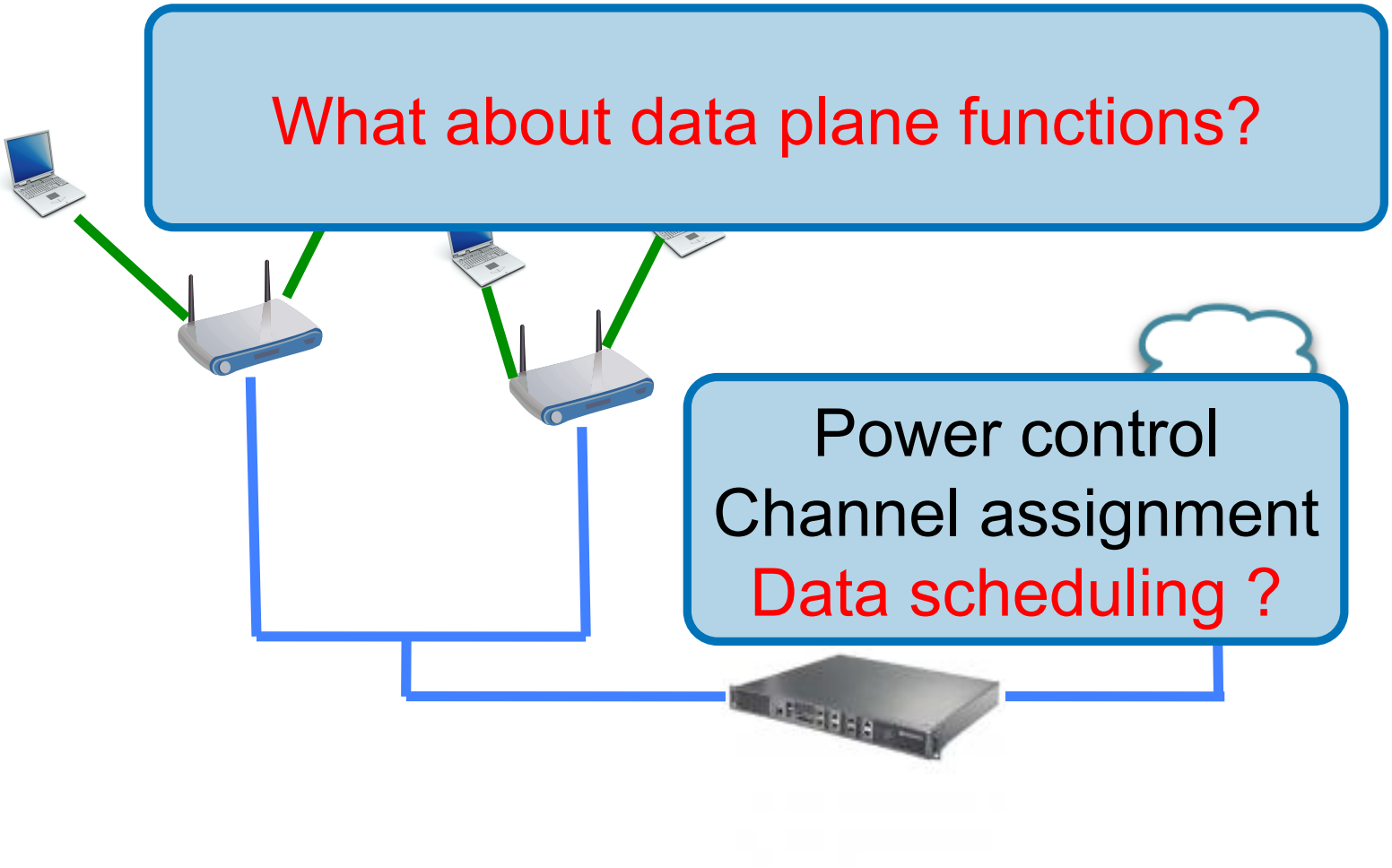



ARUBA networks

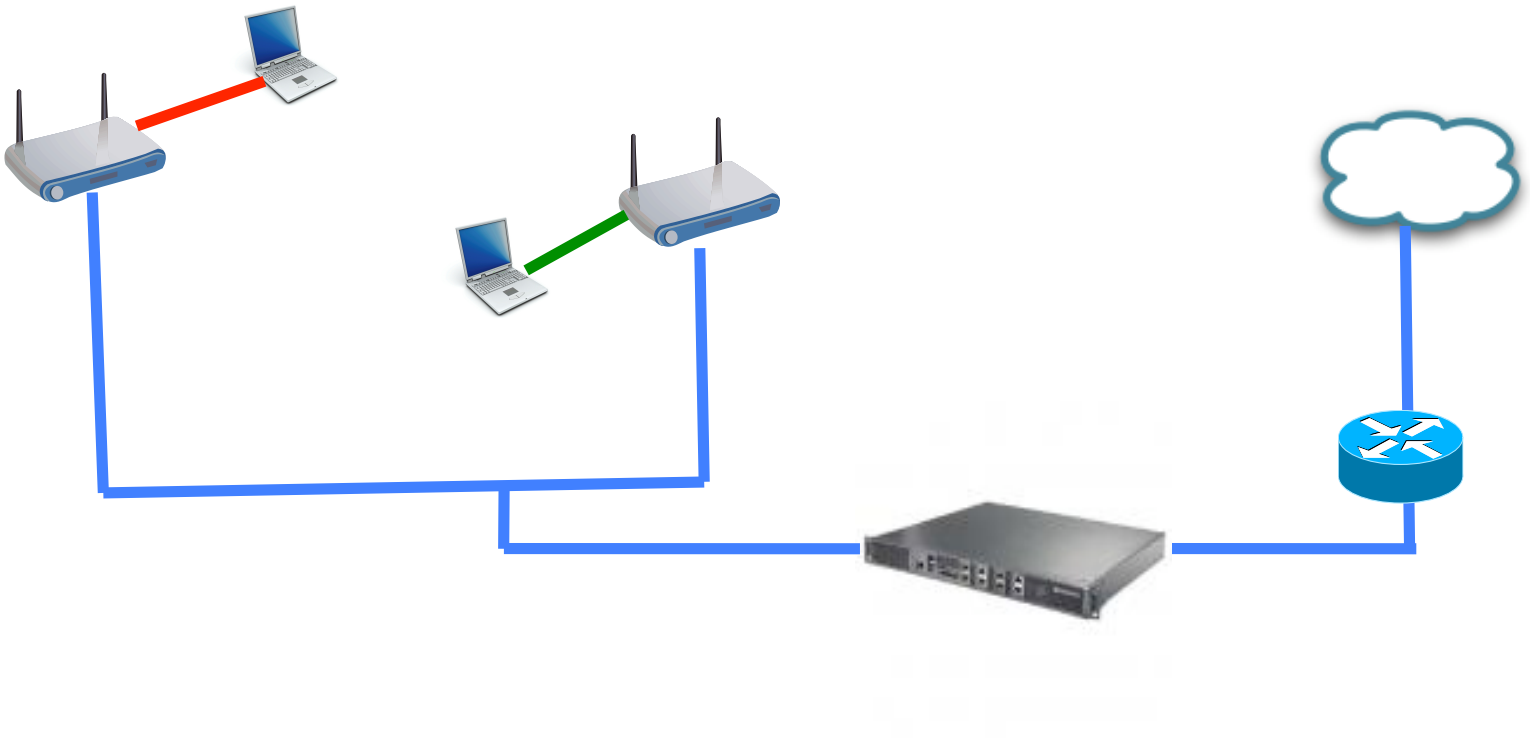
MERU NETWORKS

CISCO

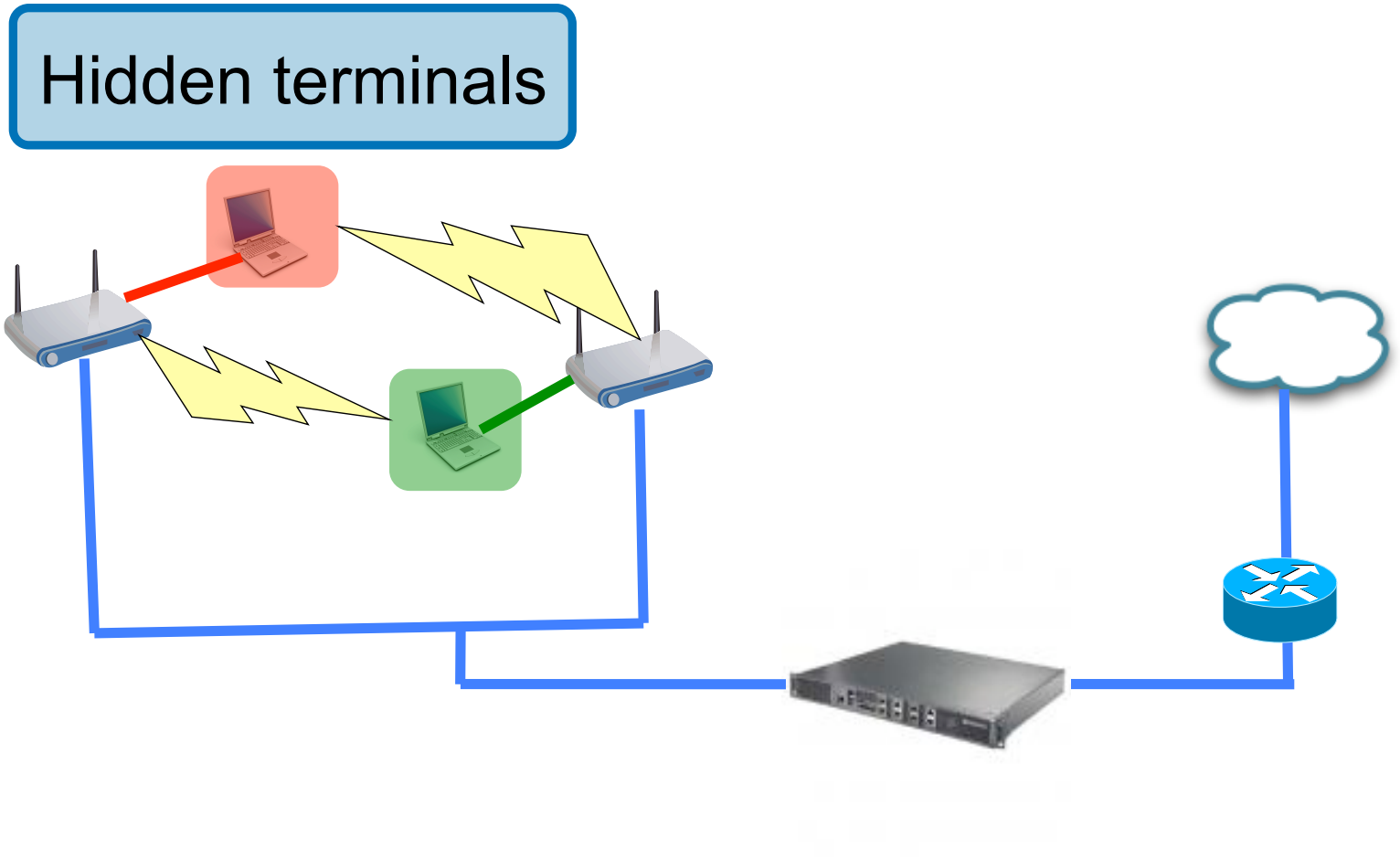
# Centralization of Enterprise WLANs



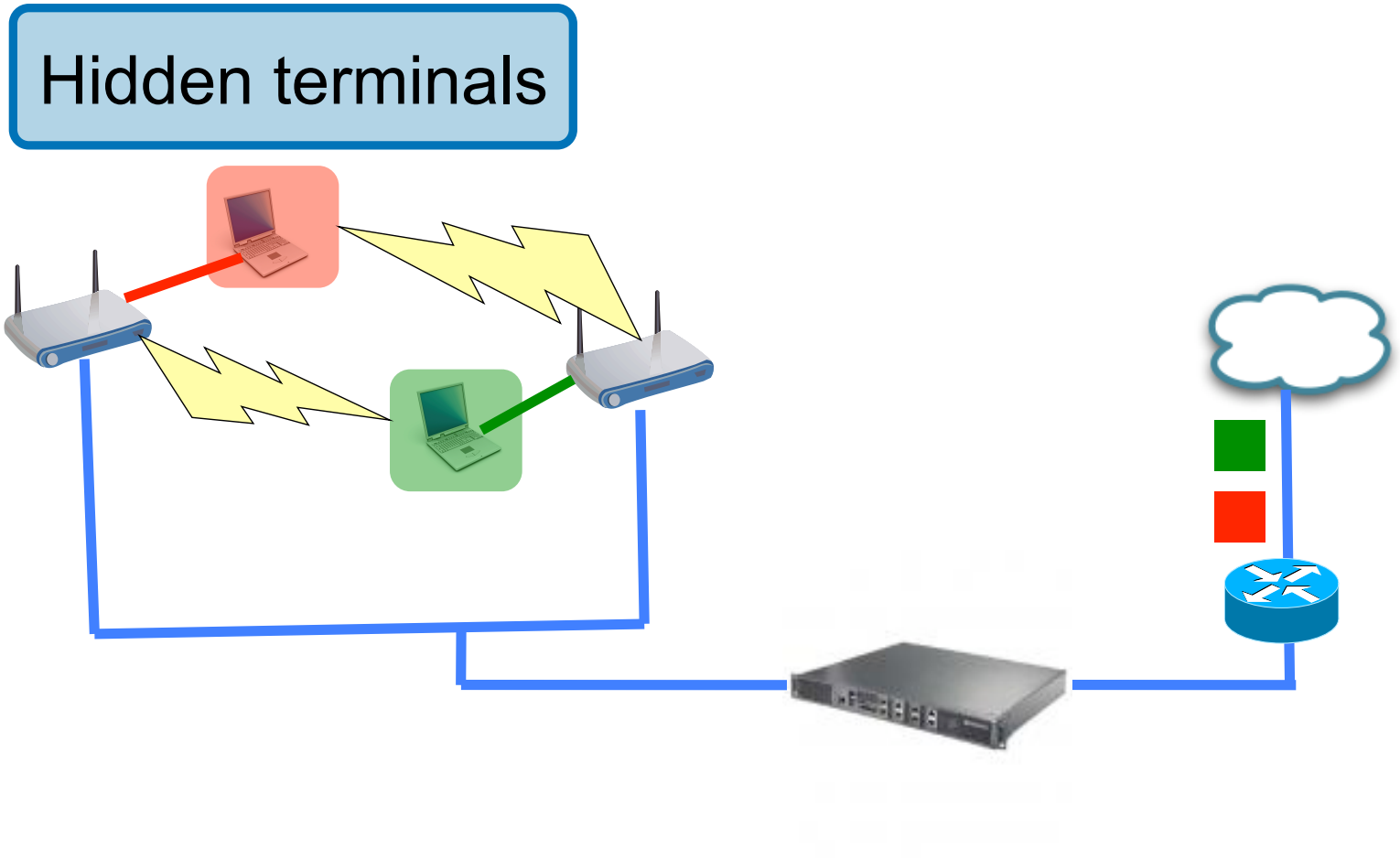
# Can centralized scheduling help?



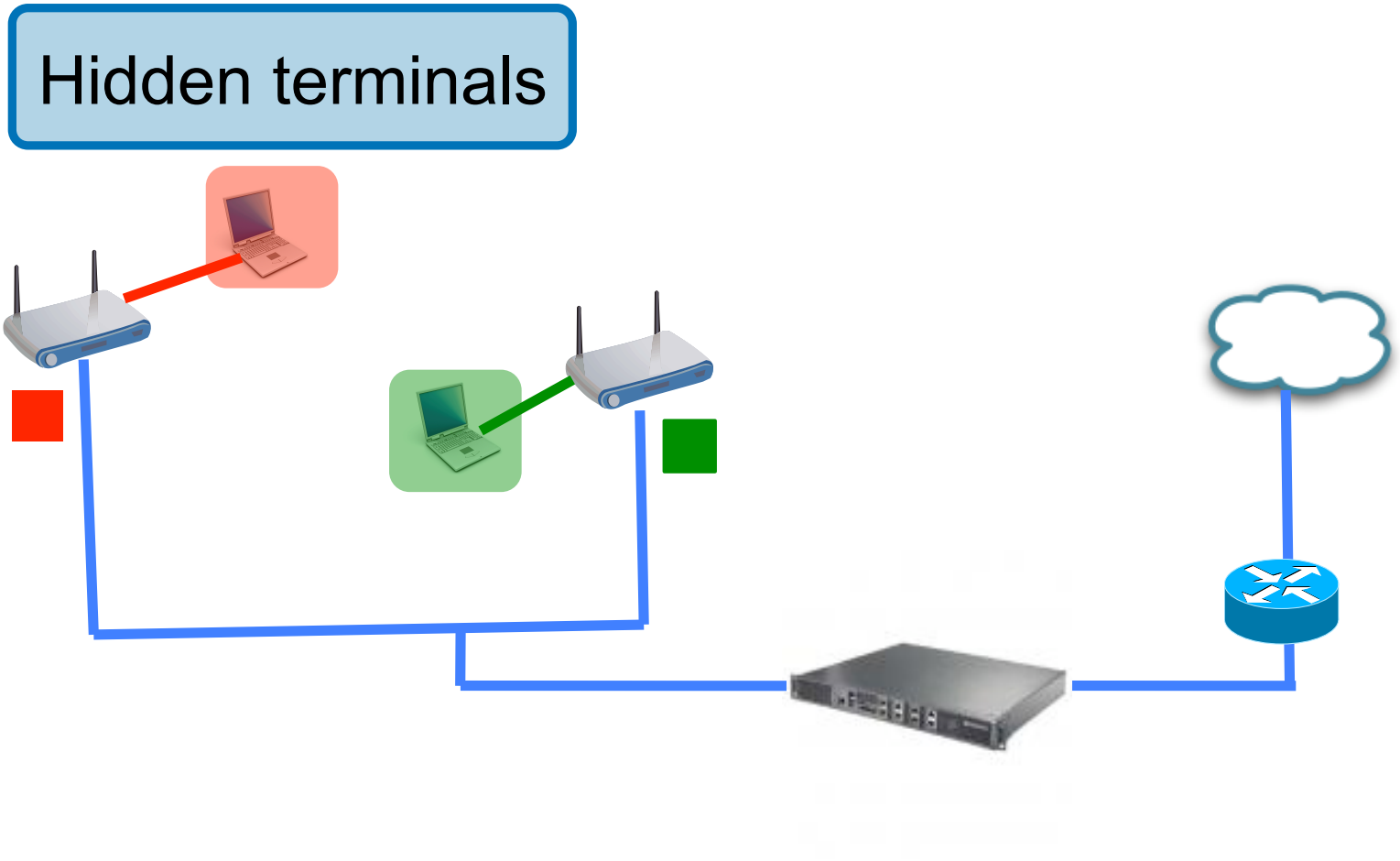
# Can centralized scheduling help?



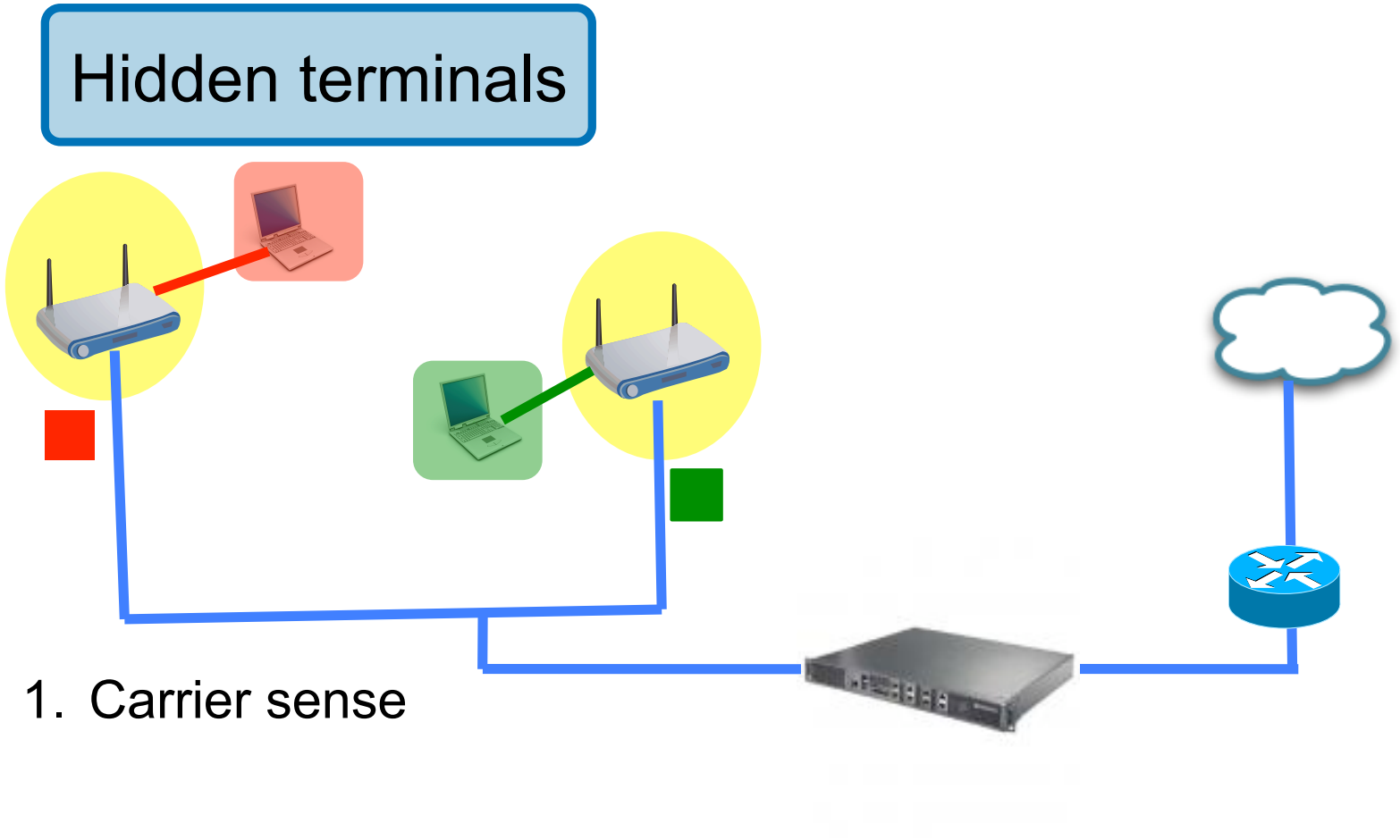
# Can centralized scheduling help?



# Can centralized scheduling help?



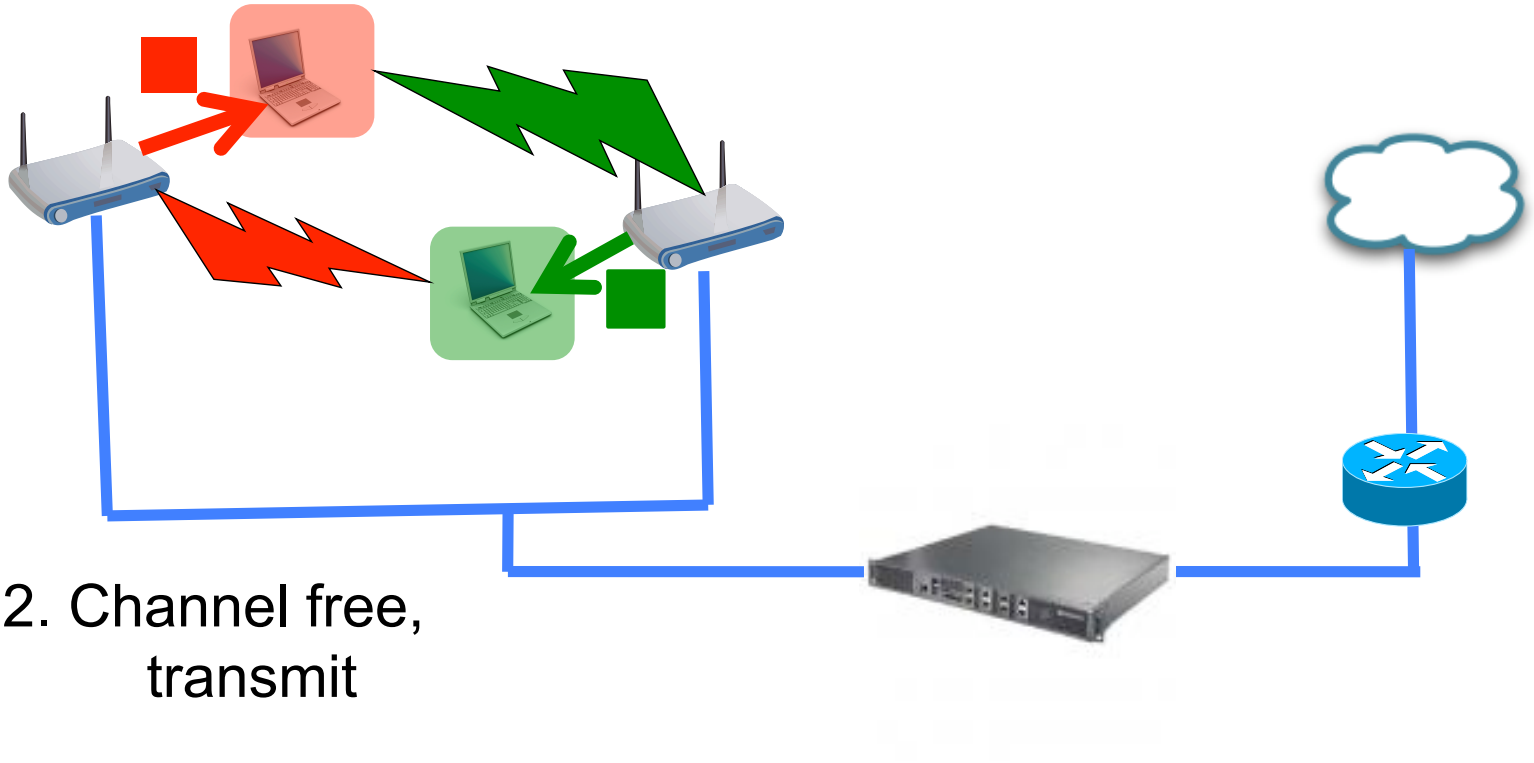
# Can centralized scheduling help?




1. Carrier sense

# Can centralized scheduling help?

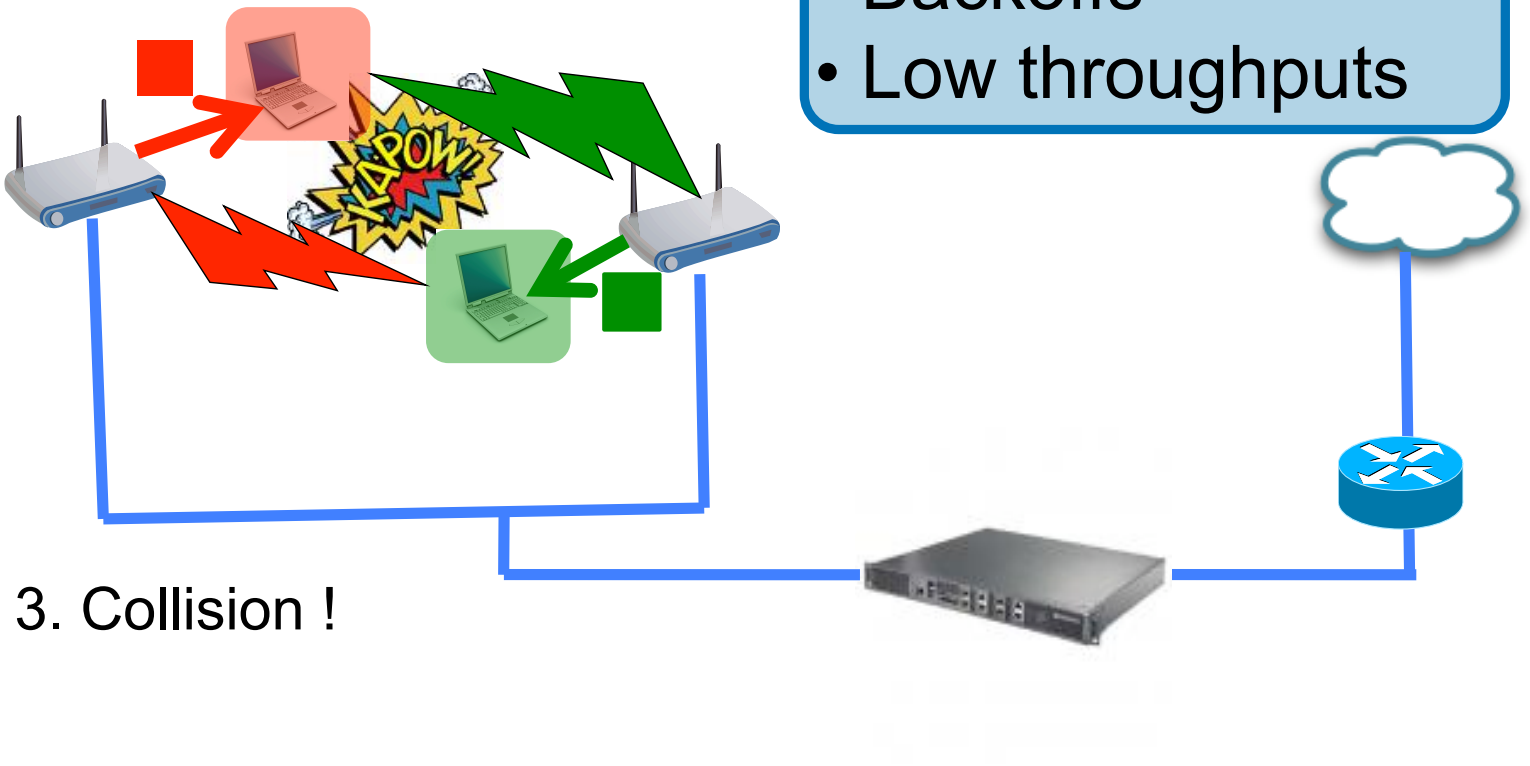
Hidden terminals



2. Channel free, transmit

# Can centralized scheduling help?

- Collision!
- Backoffs
- Low throughputs



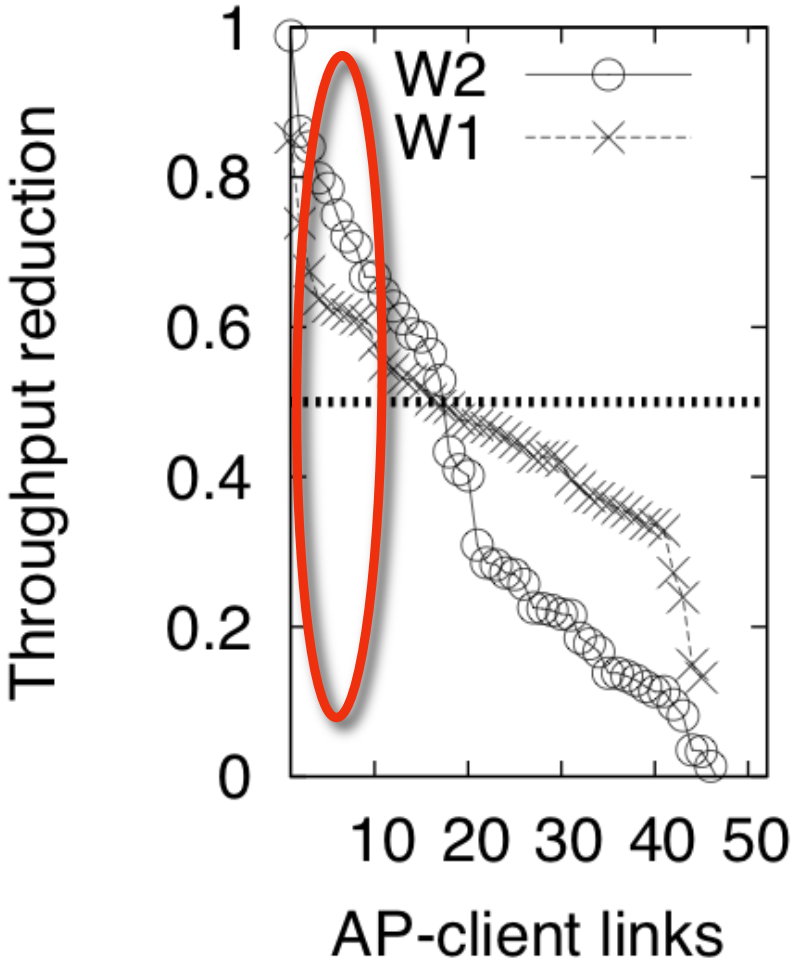

3. Collision !

# How bad is it ?

Experiments on production building-wide WLANs

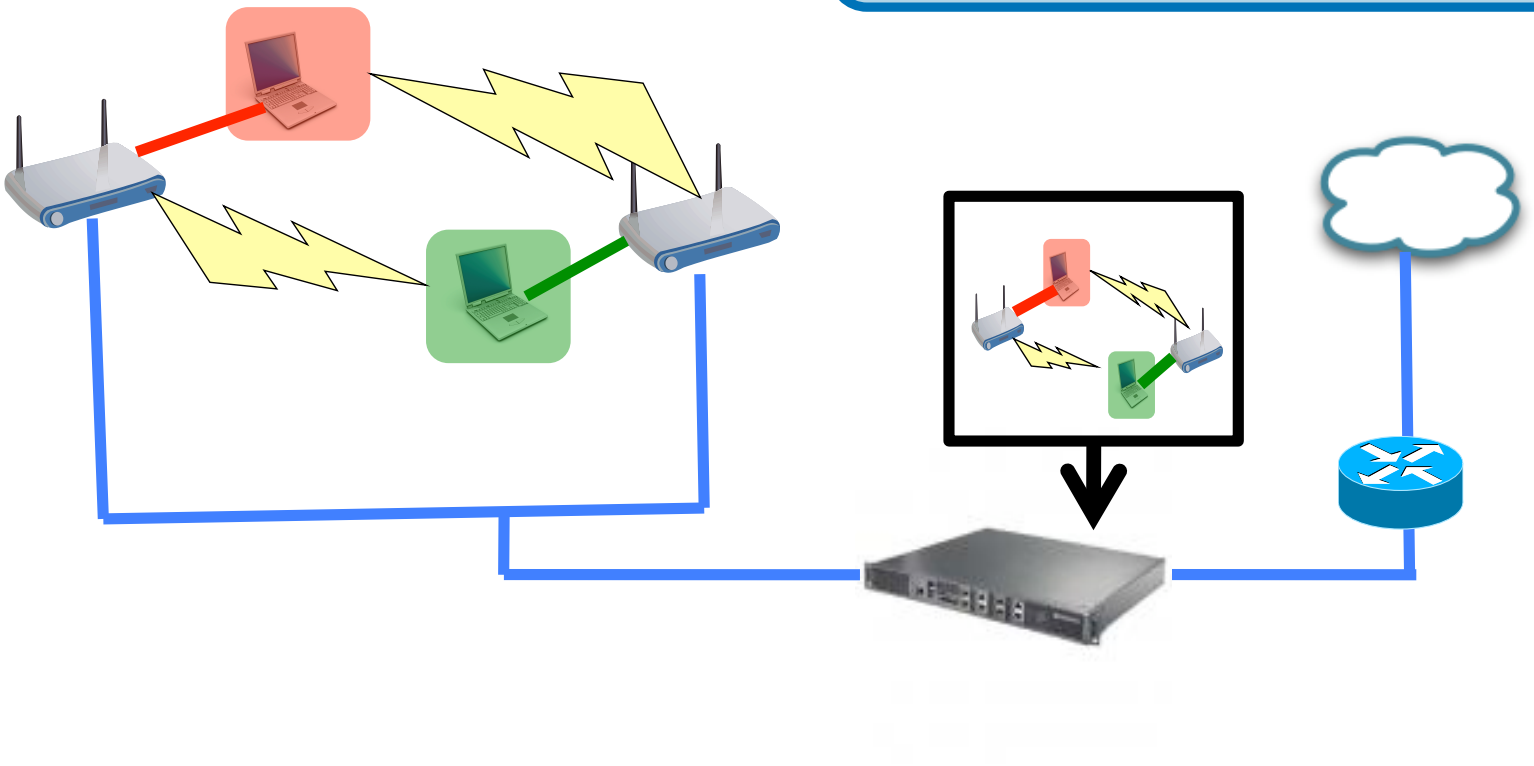
- W1: 5 floors
  9 APs, 45 clients
- W2: 1 floor
  21 APs, 51 clients

10% links suffer more than 70% throughput reduction



# A lost opportunity ?

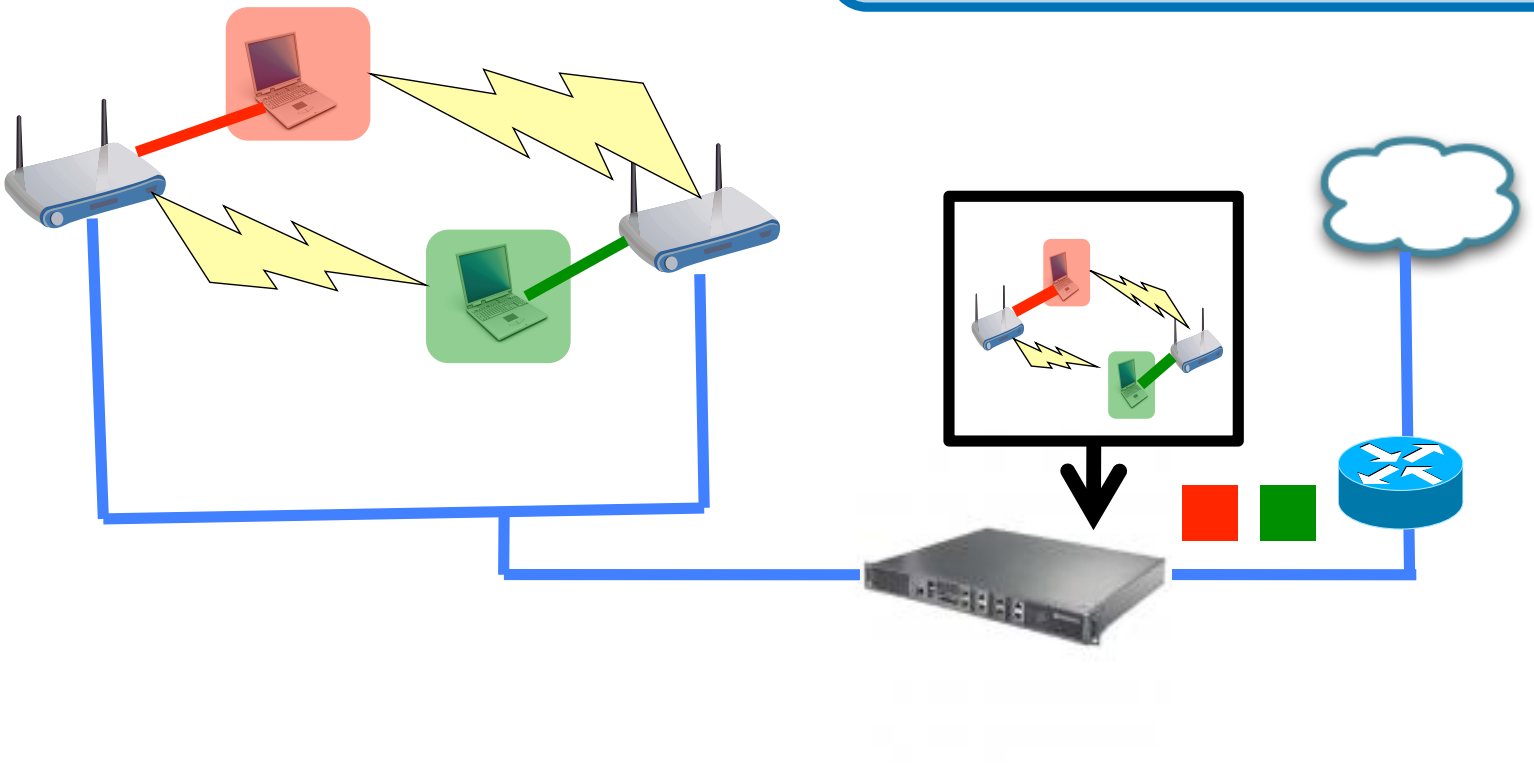
Suppose infrastructure can gather conflict data



# A lost opportunity ?

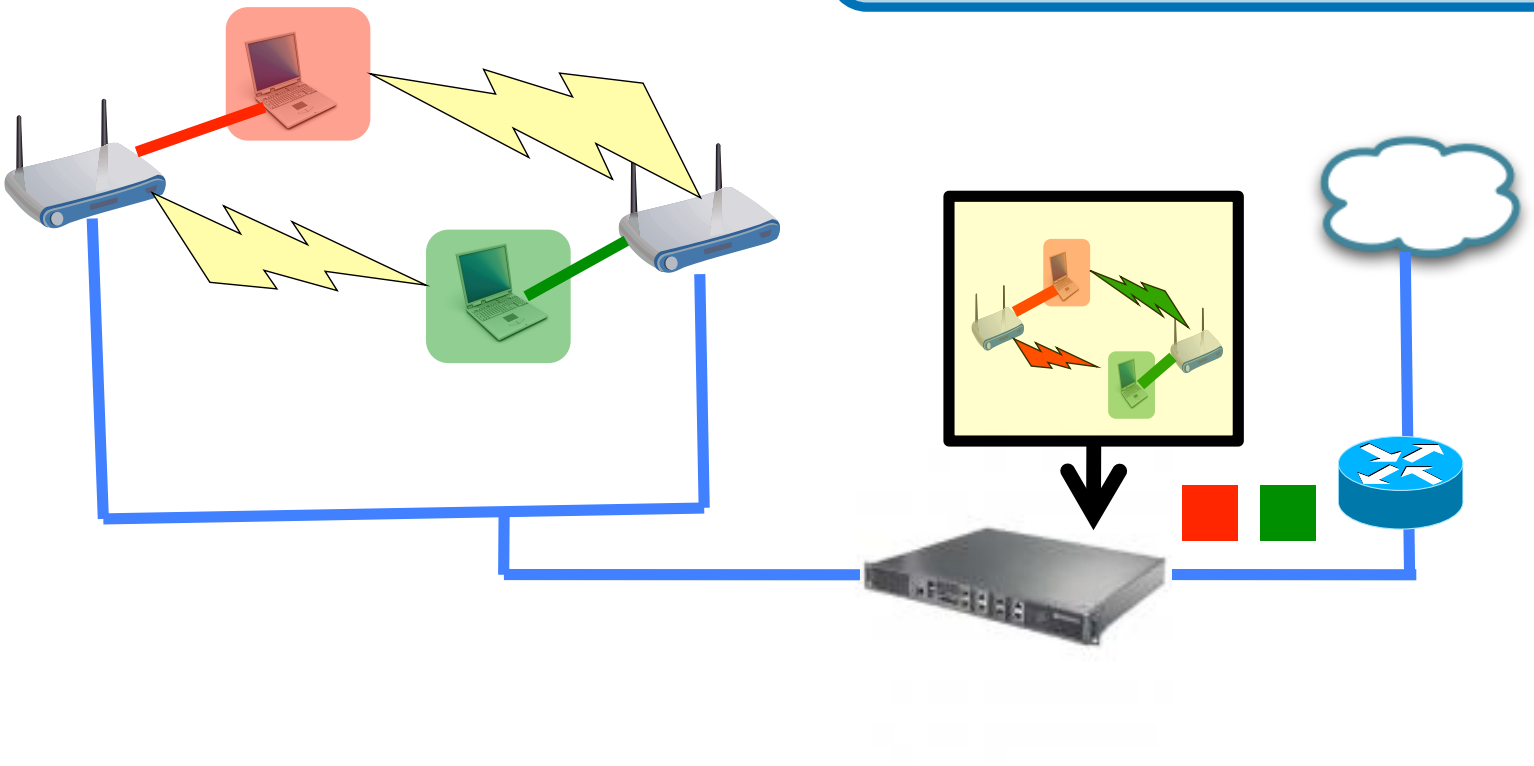
And when packets arrive

…



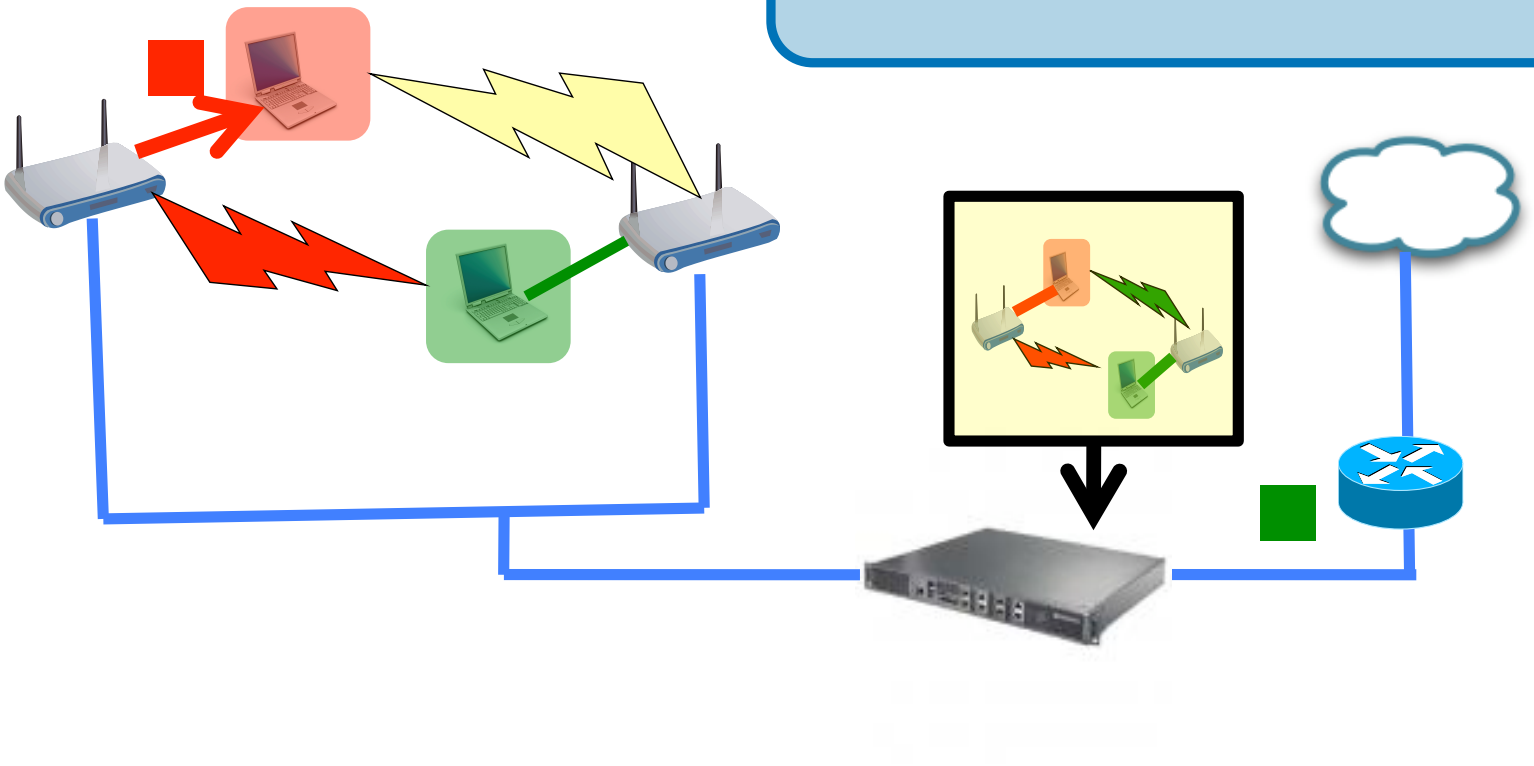
# A lost opportunity ?

… realize interference will happen …



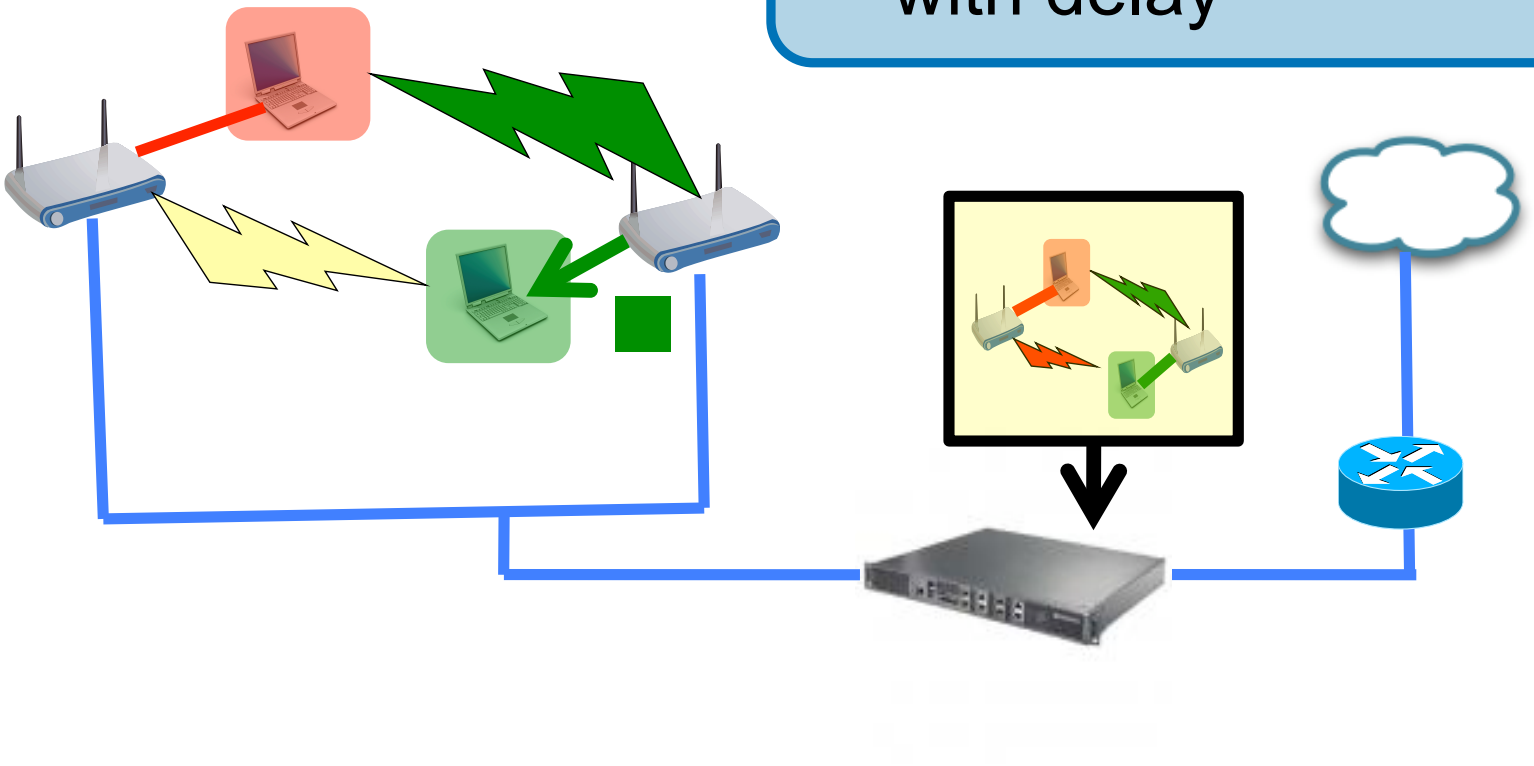
# A lost opportunity ?

1. Transmit first packet



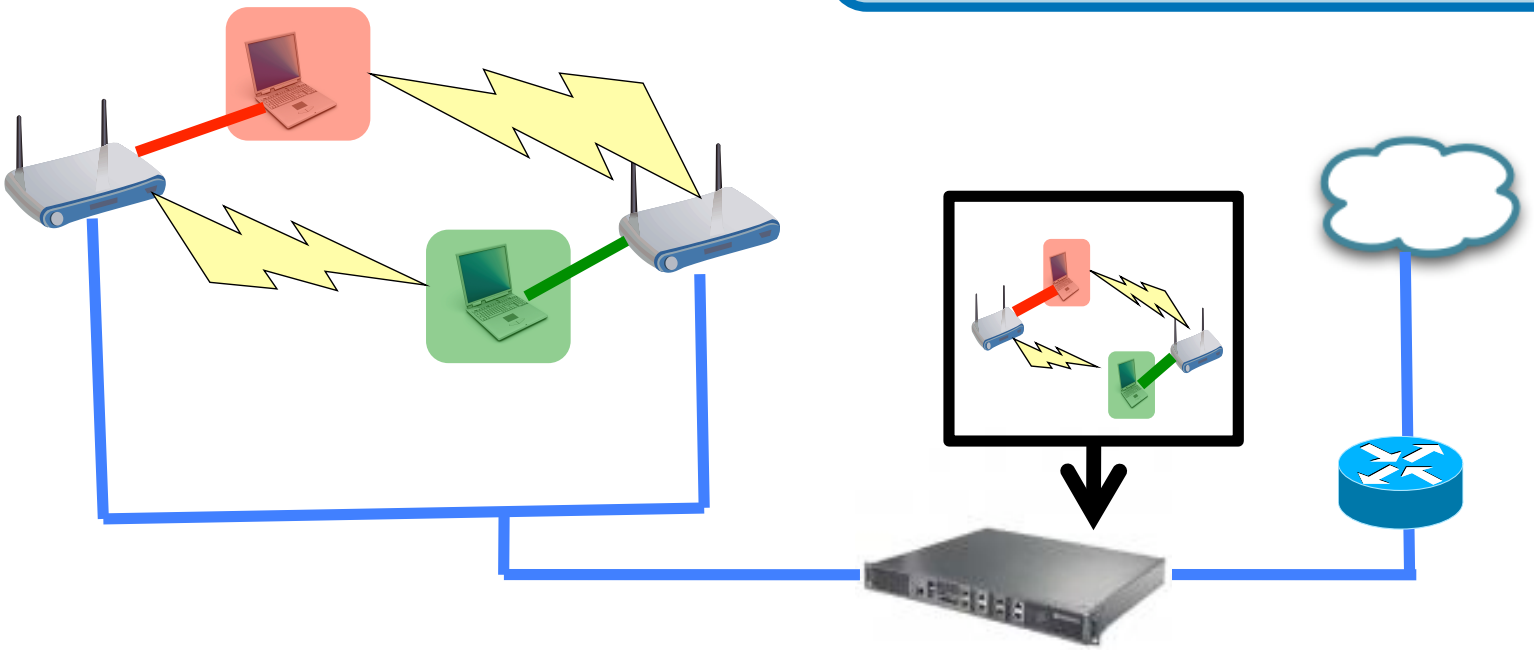
# A lost opportunity ?

1. Transmit first packet
2. Transmit second packet with delay



# Use an in-band scheduler

Simple FIFO schedule with interference avoidance



Scheduling functionality

# What about exposed terminals ?

- In our experiments, about 41%of link pairs suffer from exposed terminal interference
- Disabling carrier sense to solve the problem can be dangerous for uplink, non-enterprise traffic
- We will show how centralization can help even here

# Challenge

- Centralization has obvious overheads
- How to make this feasible and useful under
  - Real applications
  - Common large-scale wireless environments
  - Presence of uplink and non-enterprise traffic
  - Requirements of no client modifications

# Contributions

1. Design CENTAUR, a hybrid (partly centralized, partly distributed) scheduling approach
   - Resolves hidden and exposed terminals
   - Requires no client modifications
   - Requires no carrier sense disabling
2. Evaluate CENTAUR on two WLAN testbeds with real-world traffic traces
   - 1.48x greater throughput for bulk data traffic
   - 1.38x reduction in web-transaction times

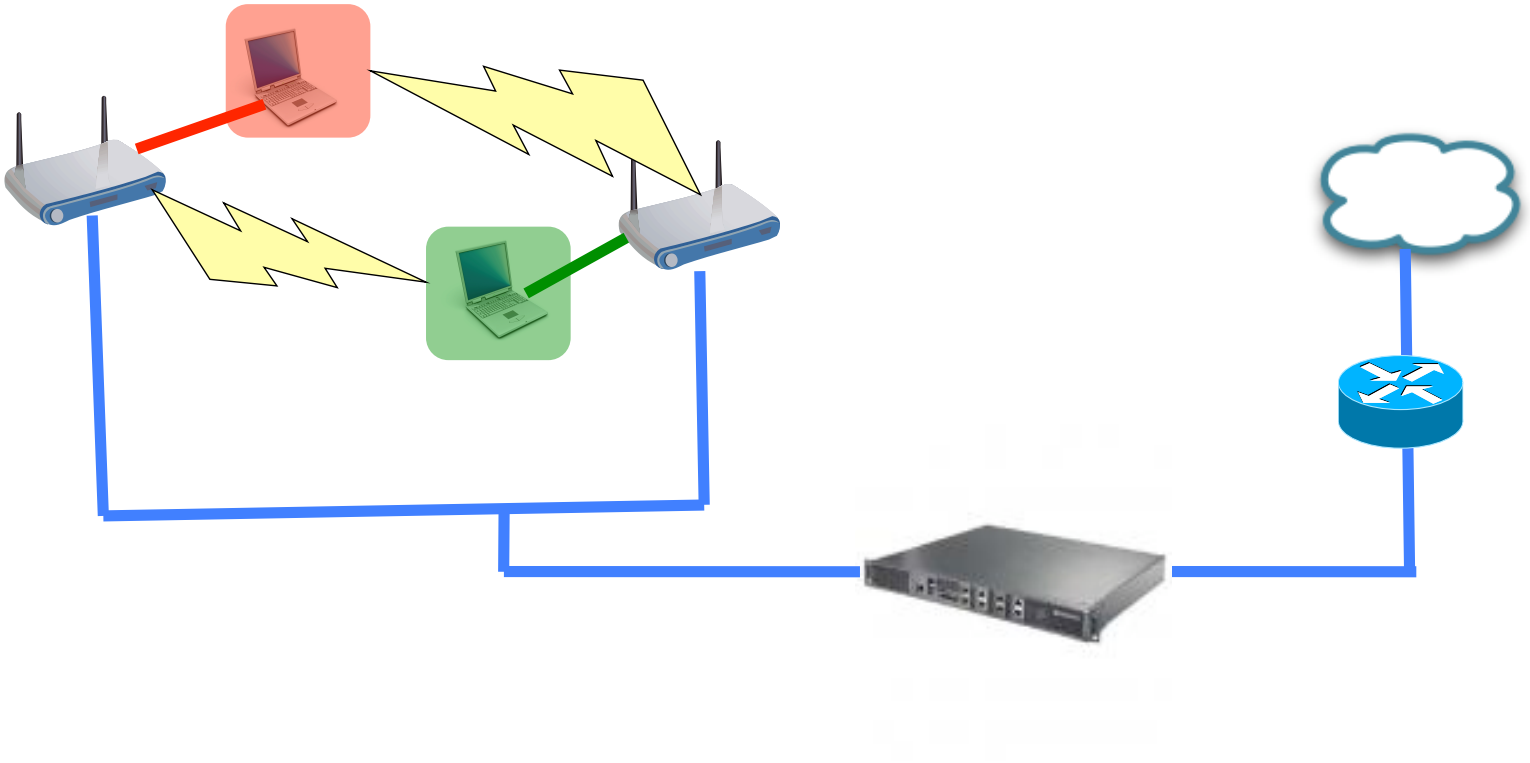
# Outline

- A naïve attempt at centralized scheduling (DET)
- Our hybrid centralized scheduler (CENTAUR)
- Evaluation
- Related Work
- Summary

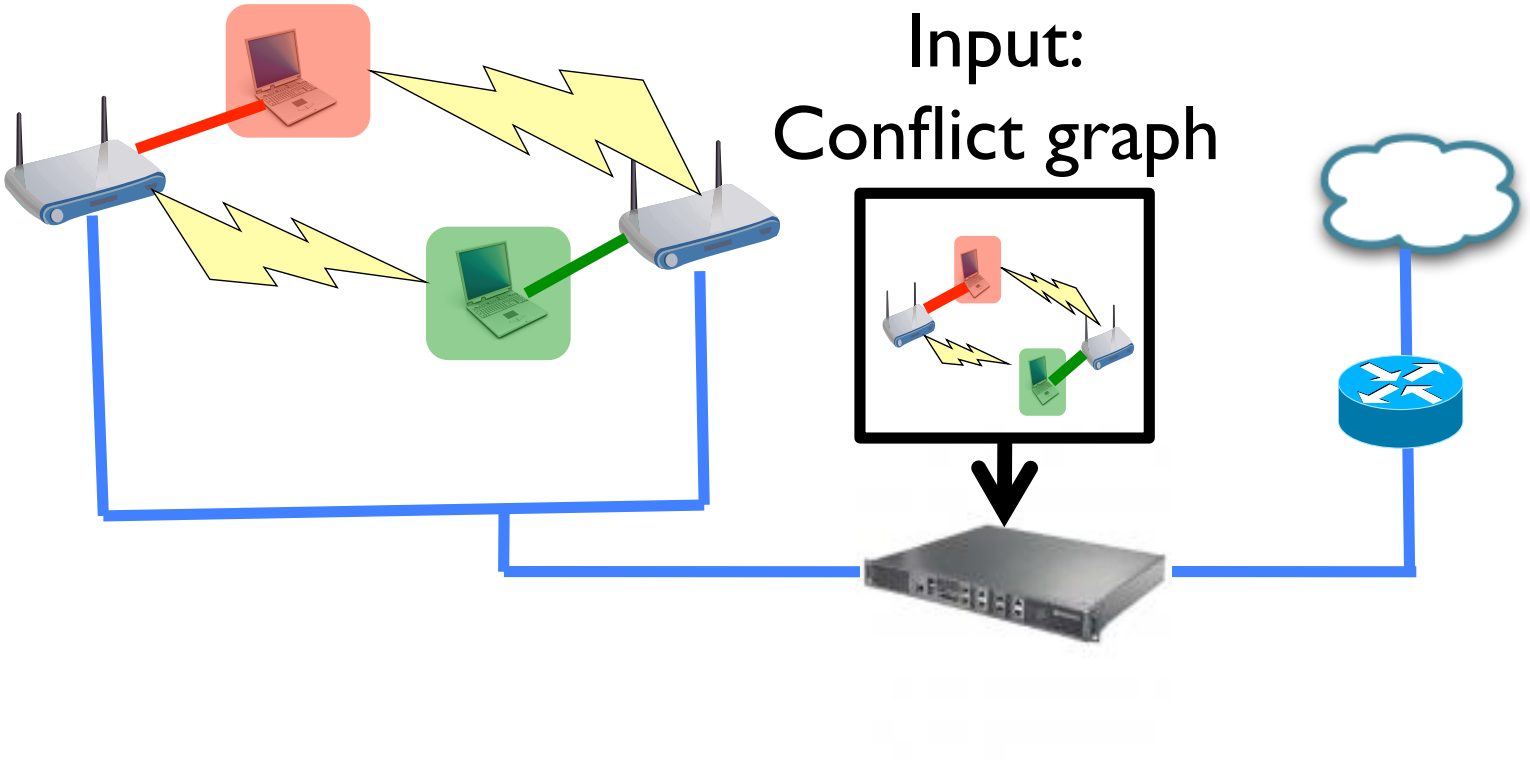
# Outline

- A naïve attempt at centralized scheduling (DET)
- Our hybrid centralized scheduler (CENTAUR)
- Evaluation
- Related Work
- Summary

# DET: A Simple Deterministic Scheduler

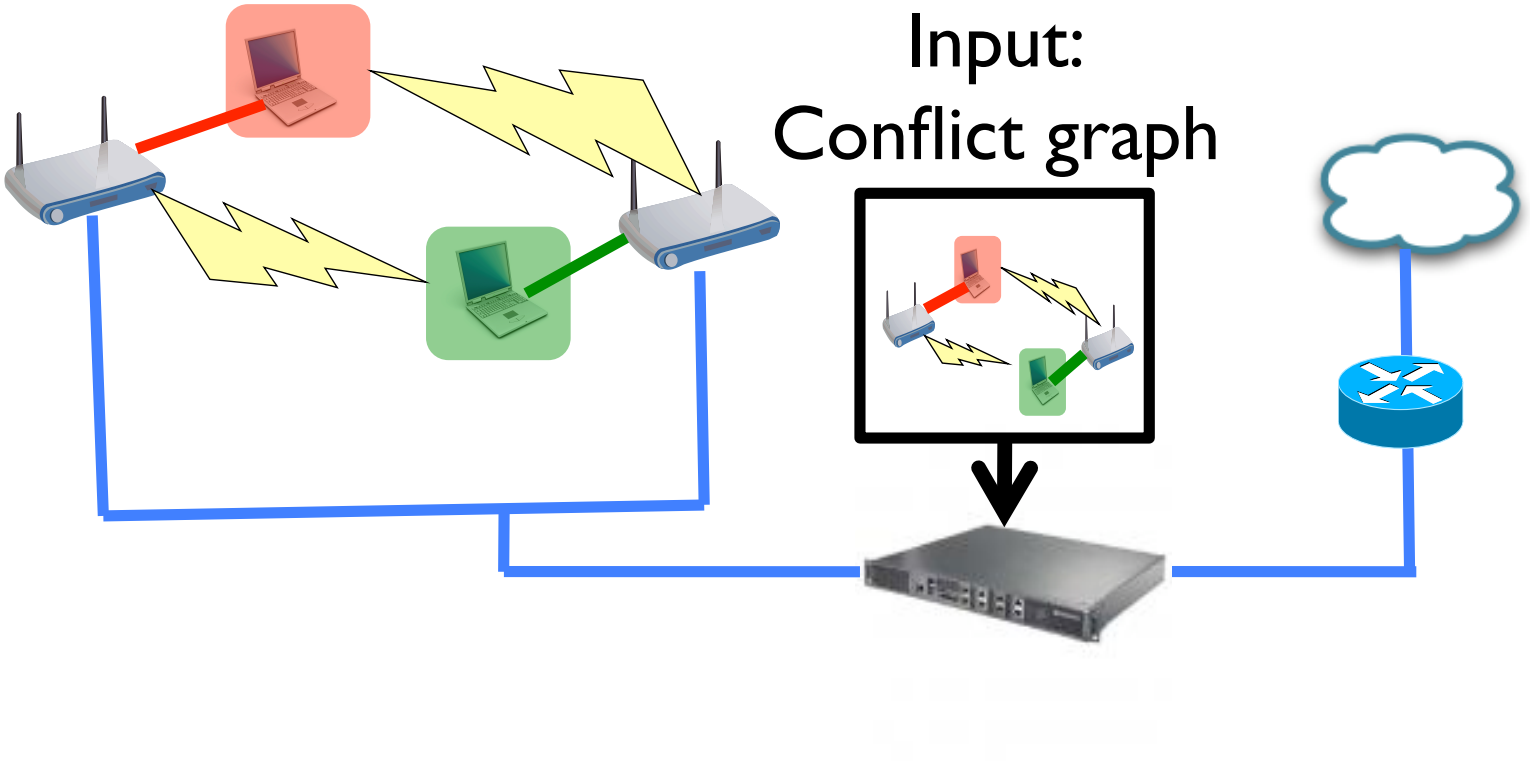


# DET: A Simple Deterministic Scheduler



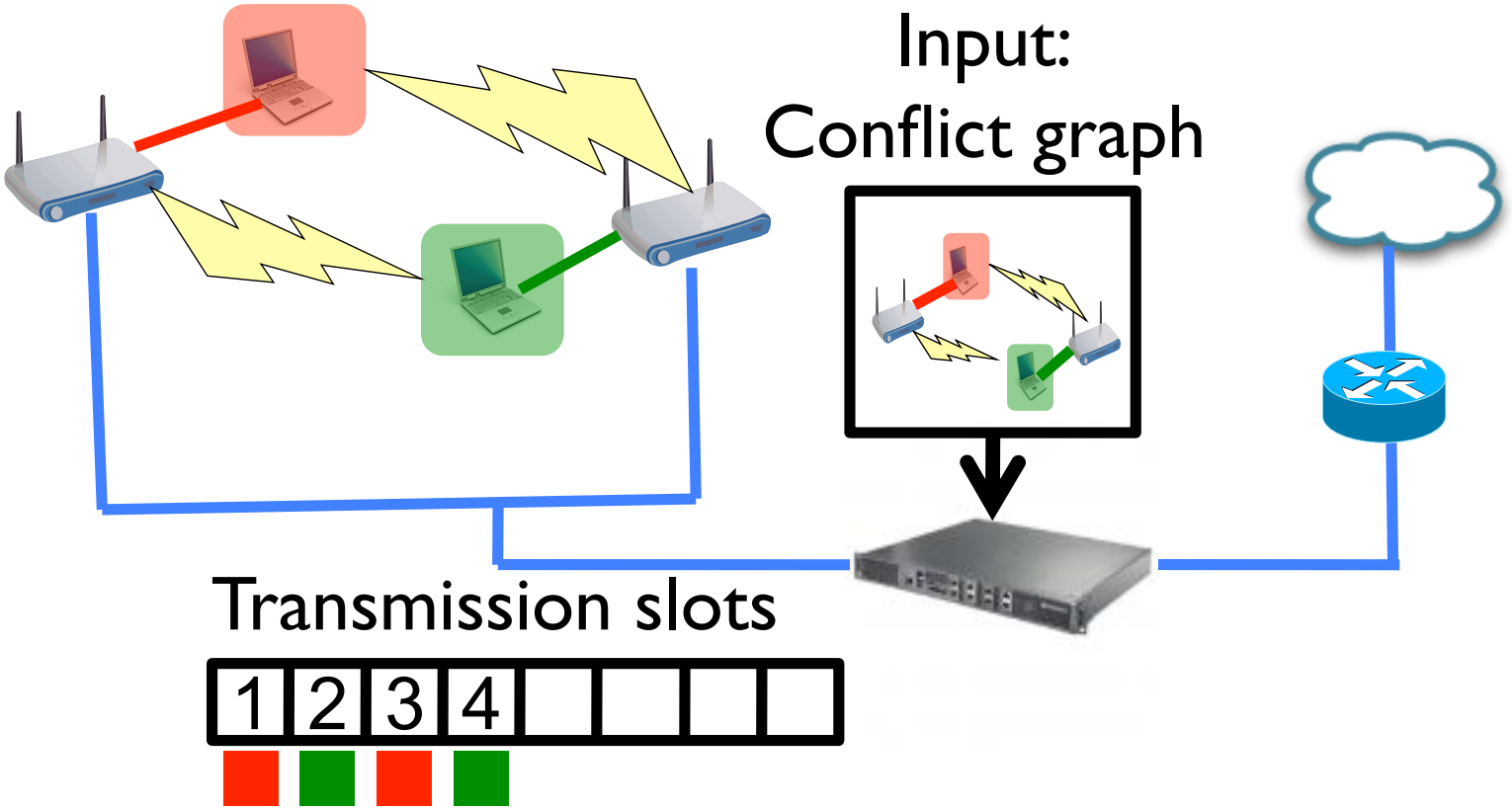
# DET: A Simple Deterministic Scheduler

Goal:
Schedule each incoming downlink packet



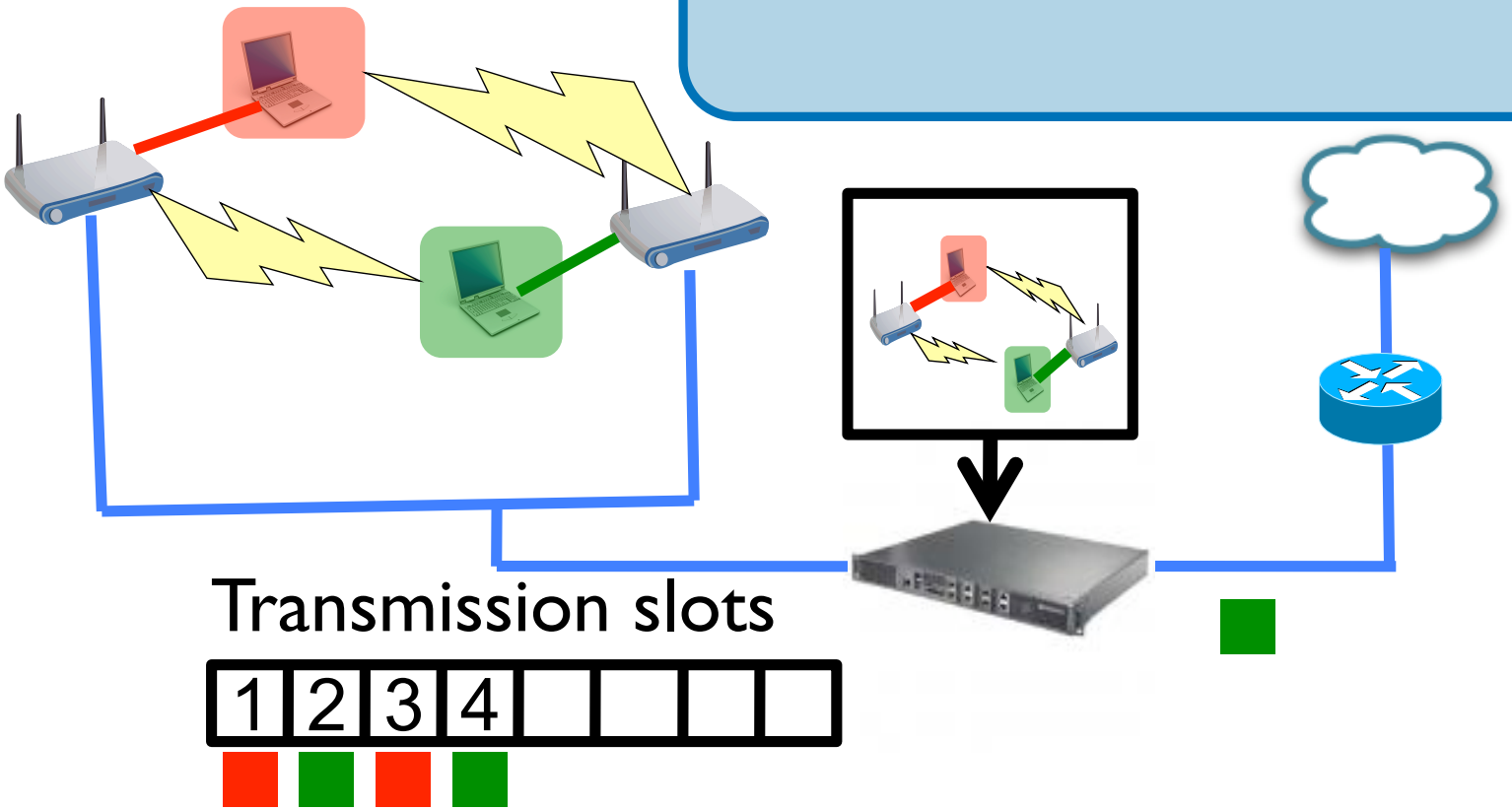
# DET: A Simple Deterministic Scheduler

Goal:
Schedule each incoming downlink packet



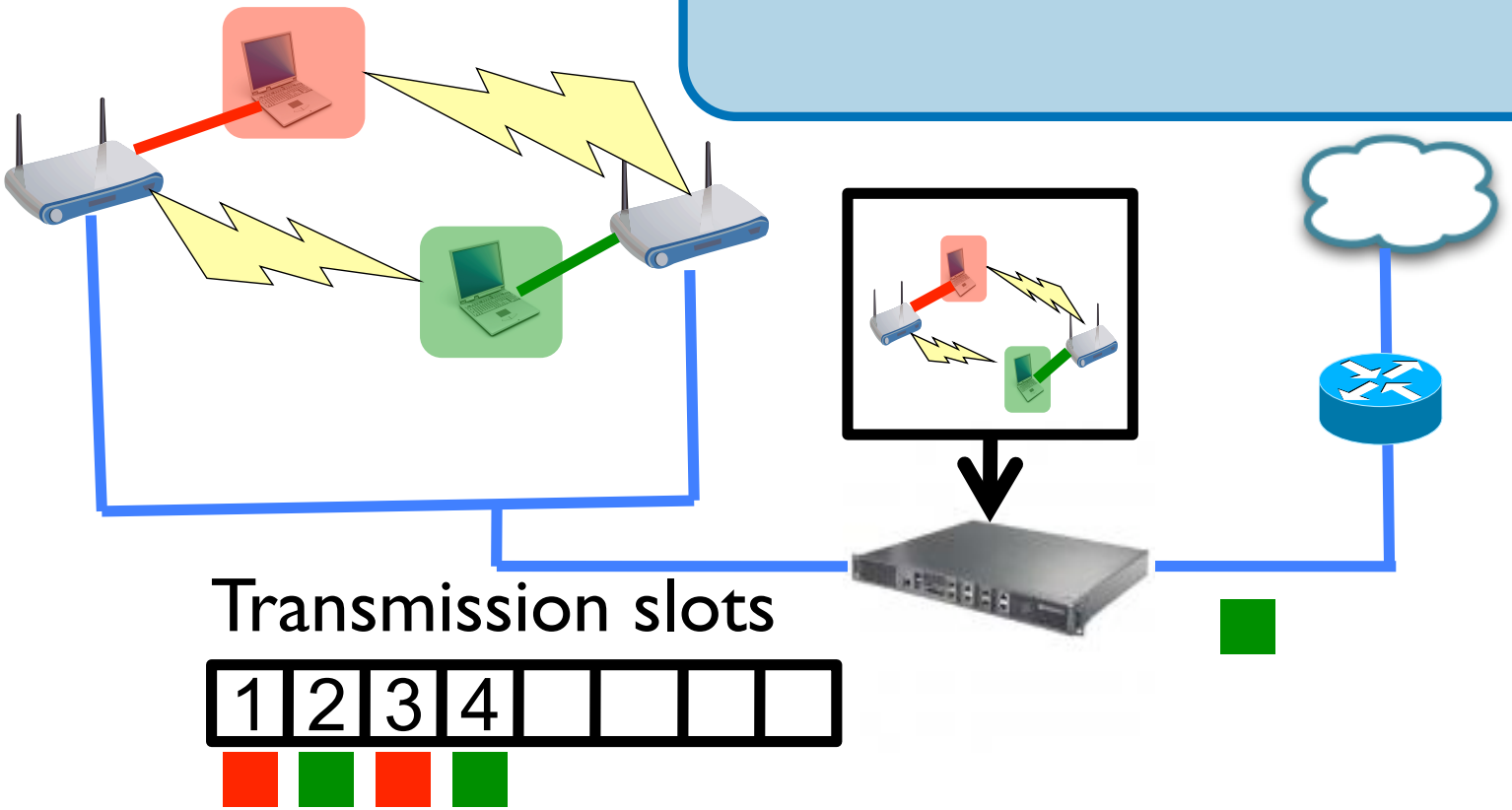
# DET: A Simple Deterministic Scheduler

1. New packet arrives



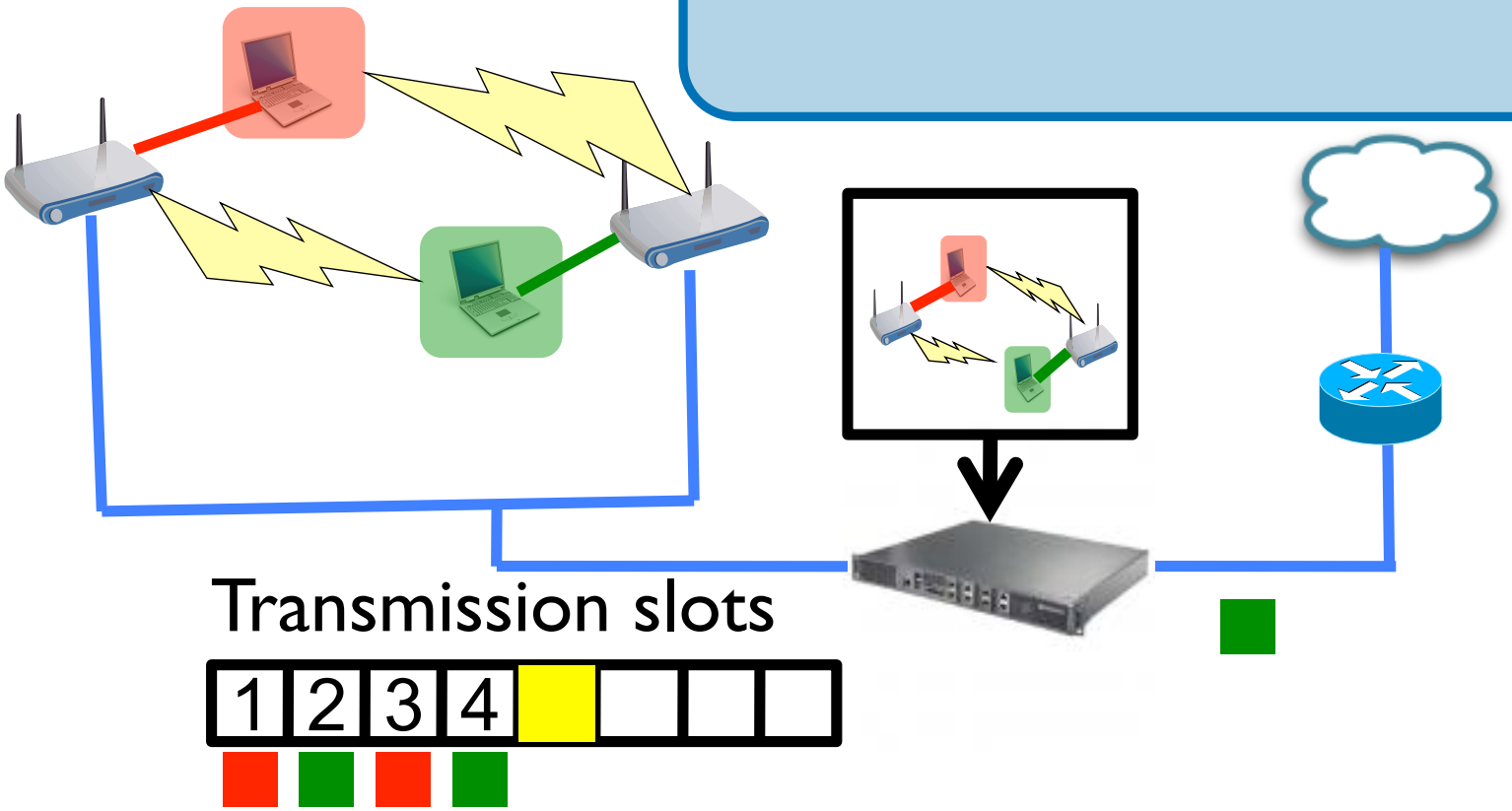
# DET: A Simple Deterministic Scheduler

1. New packet arrives
2. Find the earliest conflict free slot



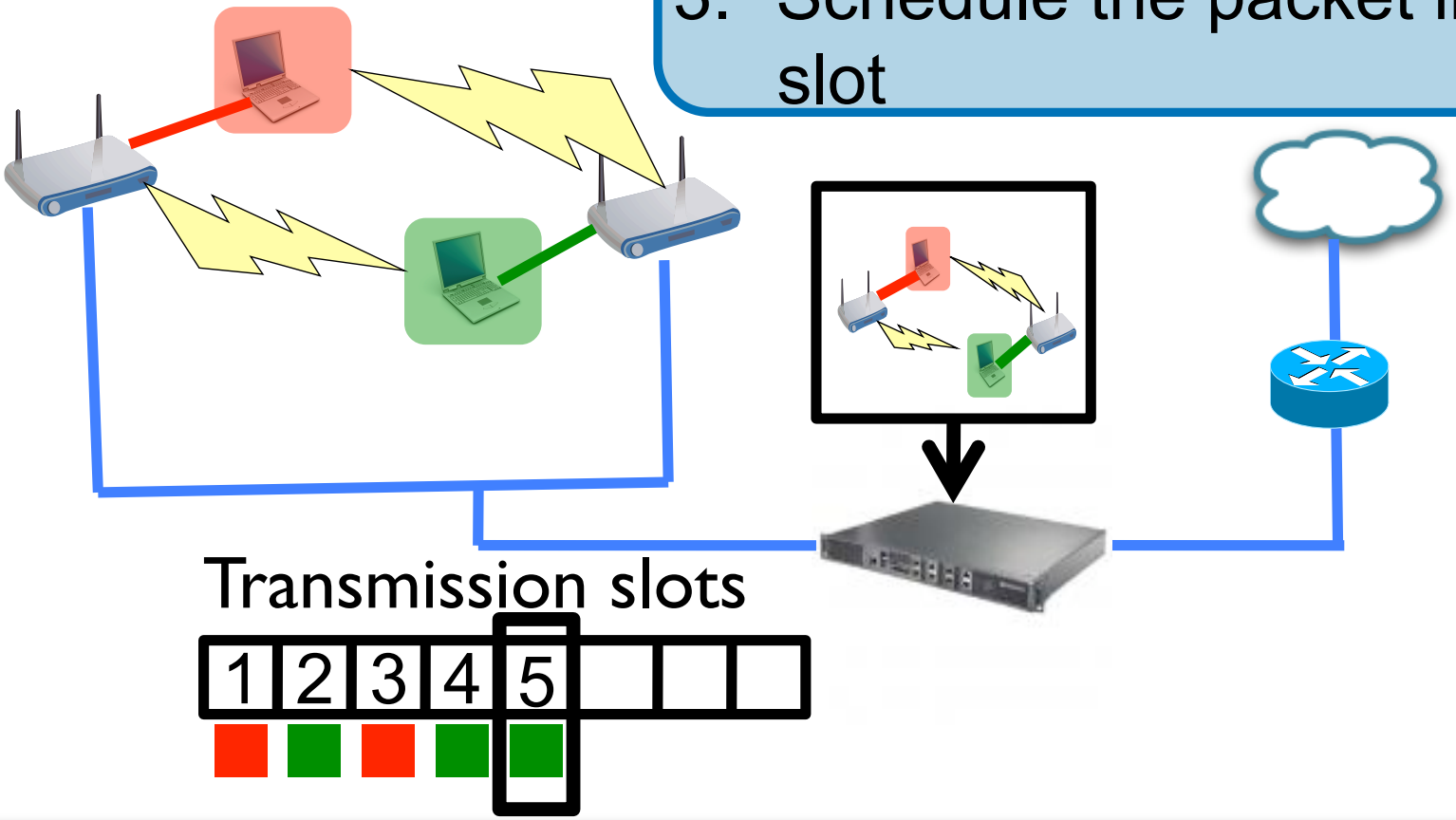
# DET: A Simple Deterministic Scheduler

1. New packet arrives
2. Find the earliest conflict free slot

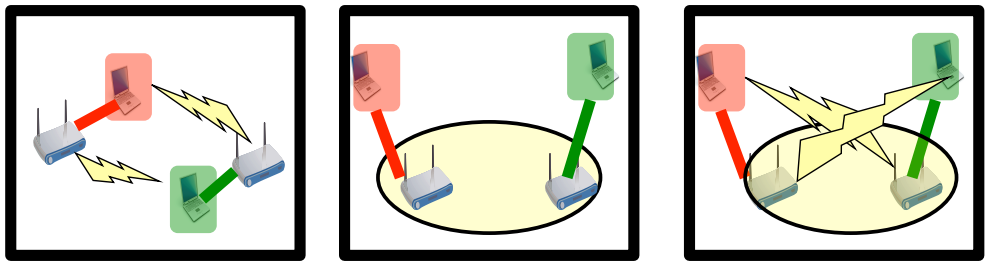


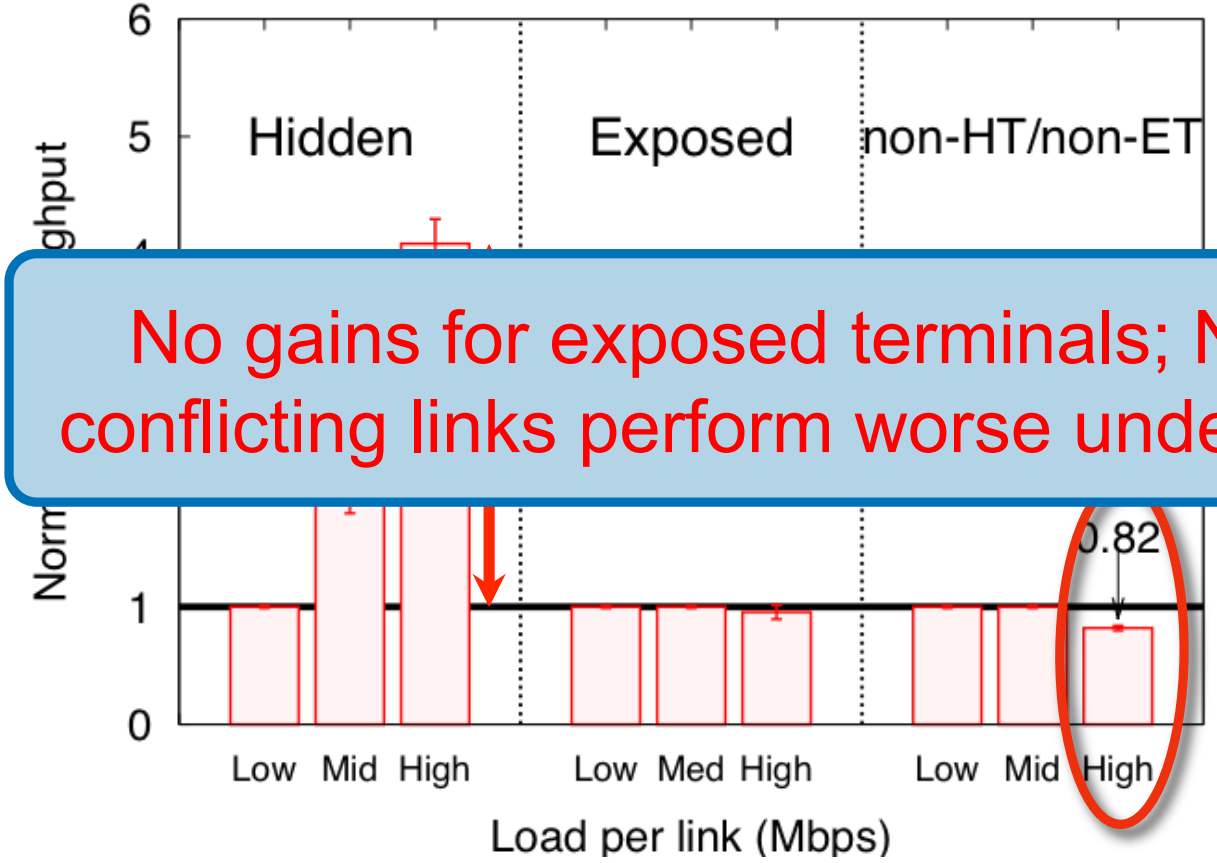
# DET: A Simple Deterministic Scheduler

1. New packet arrives
2. Find the earliest conflict free slot
3. Schedule the packet in that slot



# Performance of DET






No gains for exposed terminals; Non-conflicting links perform worse under load

# Outline

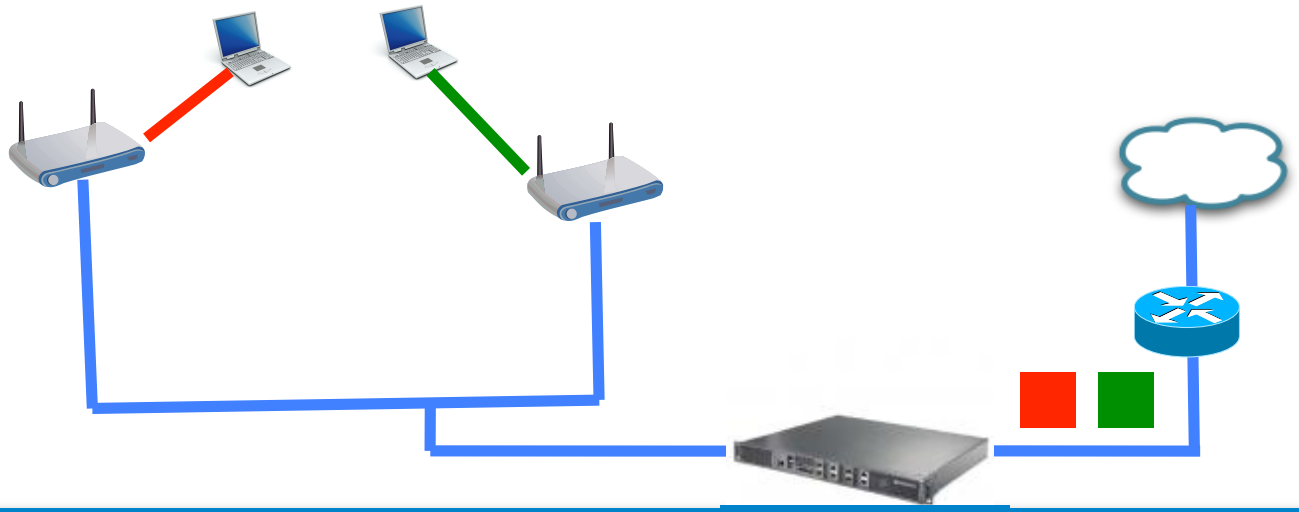
- A naïve attempt at centralized scheduling (DET)
- Our hybrid centralized scheduler (CENTAUR)
- Evaluation
- Related Work
- Summary

# Overview of CENTAUR

- Incorporate basic DET scheduler
- Tackle DET's shortcomings:
  - Amortize scheduling overhead
  - Improve performance for exposed links
  - Avoid degrading normal links
- Coexist with non-enterprise and uplink traffic

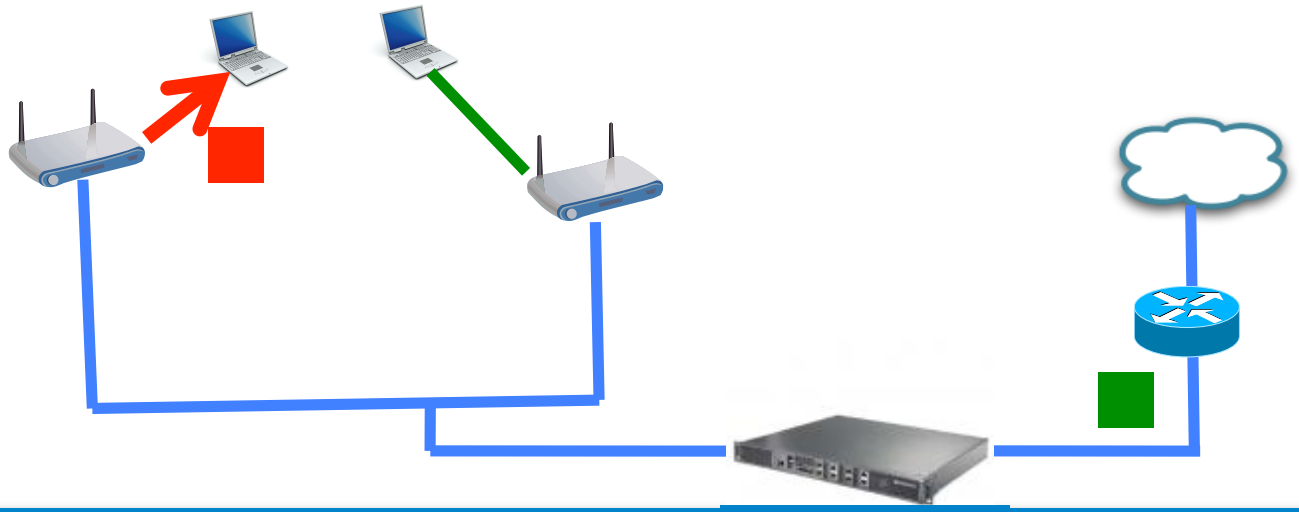
# (1) Avoid Scheduling Overheads

- Problem: Per-packet scheduling performs poorly under high network loads
- Solution:
  - Schedule packets in batches (or epochs)



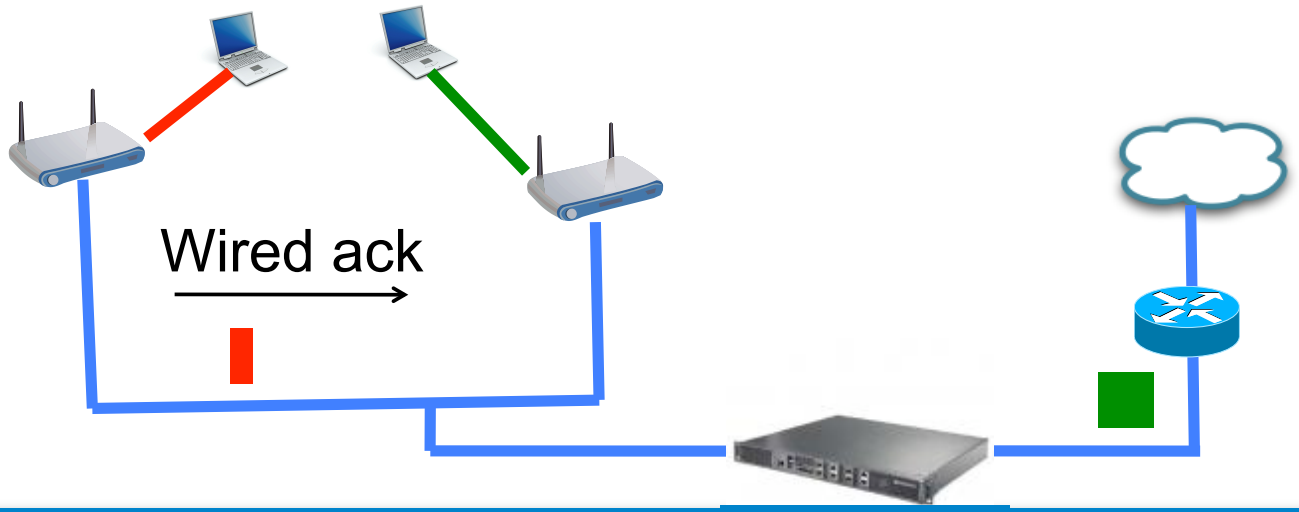
# (1) Avoid Scheduling Overheads

- Problem: Per-packet scheduling performs poorly under high network loads
- Solution:
  - Schedule packets in batches (or epochs)



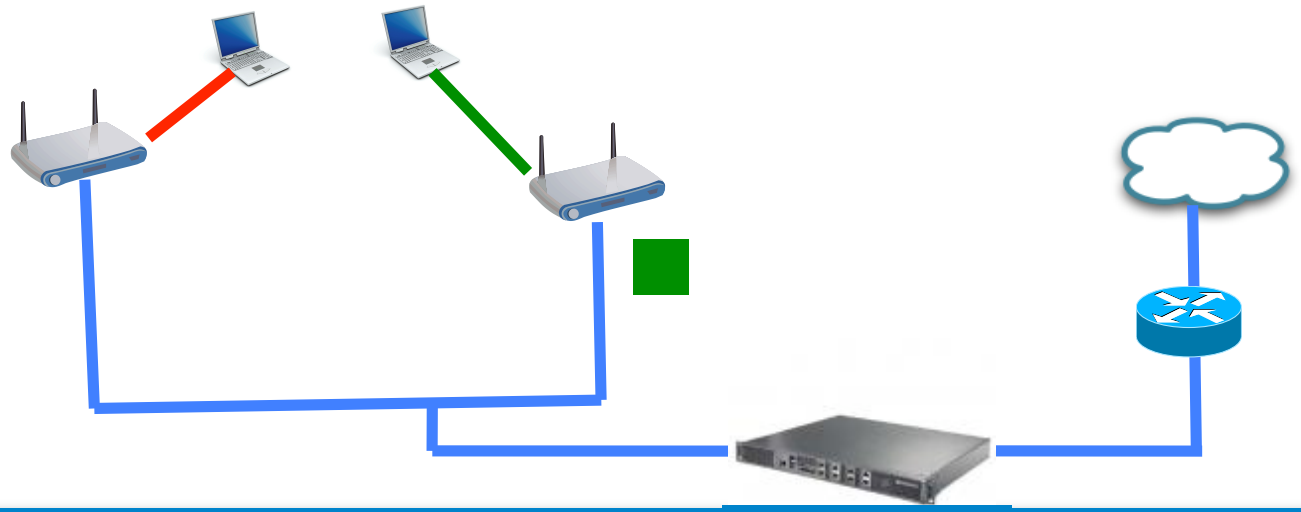
# (1) Avoid Scheduling Overheads

- Problem: Per-packet scheduling performs poorly under high network loads
- Solution:
  - Schedule packets in batches (or epochs)



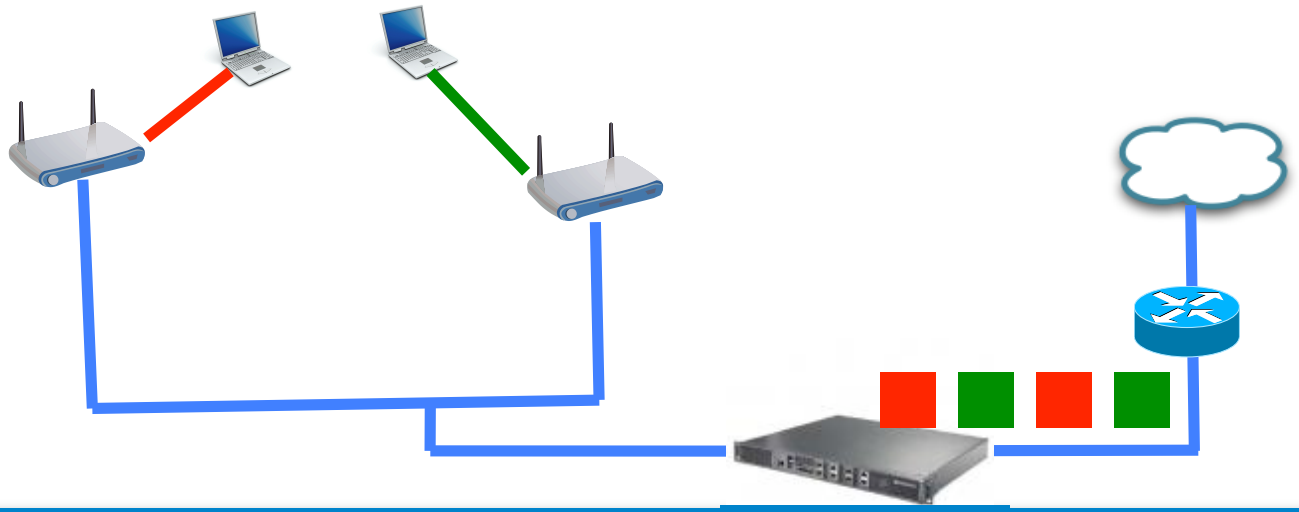
# (1) Avoid Scheduling Overheads

- Problem: Per-packet scheduling performs poorly under high network loads
- Solution:
  - Schedule packets in batches (or epochs)



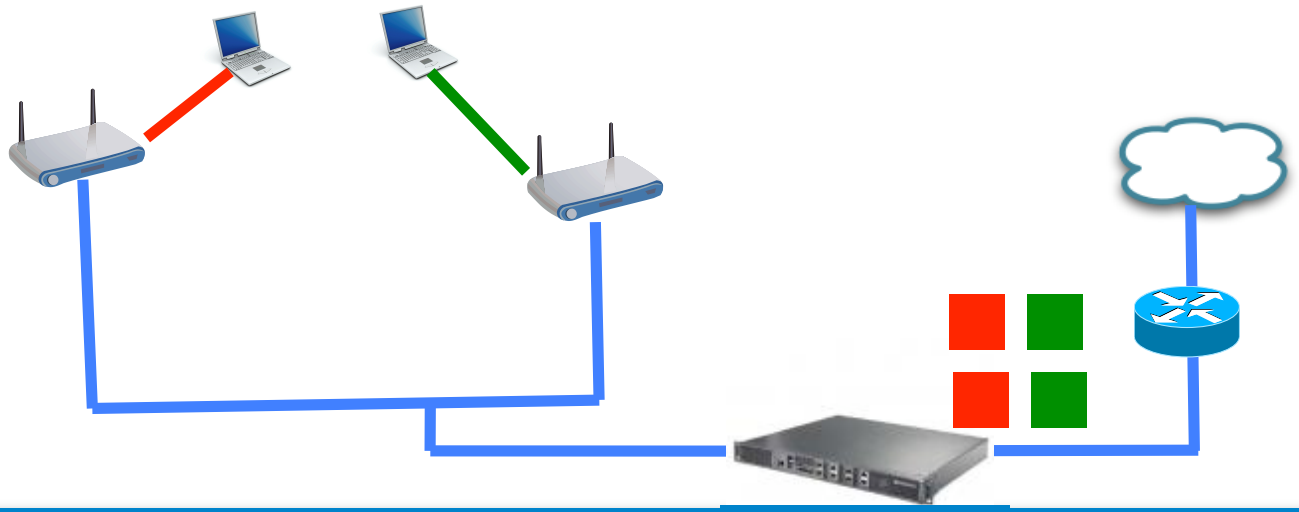
# (1) Avoid Scheduling Overheads

- Problem: Per-packet scheduling performs poorly under high network loads
- Solution:
  - Schedule packets in batches (or epochs)



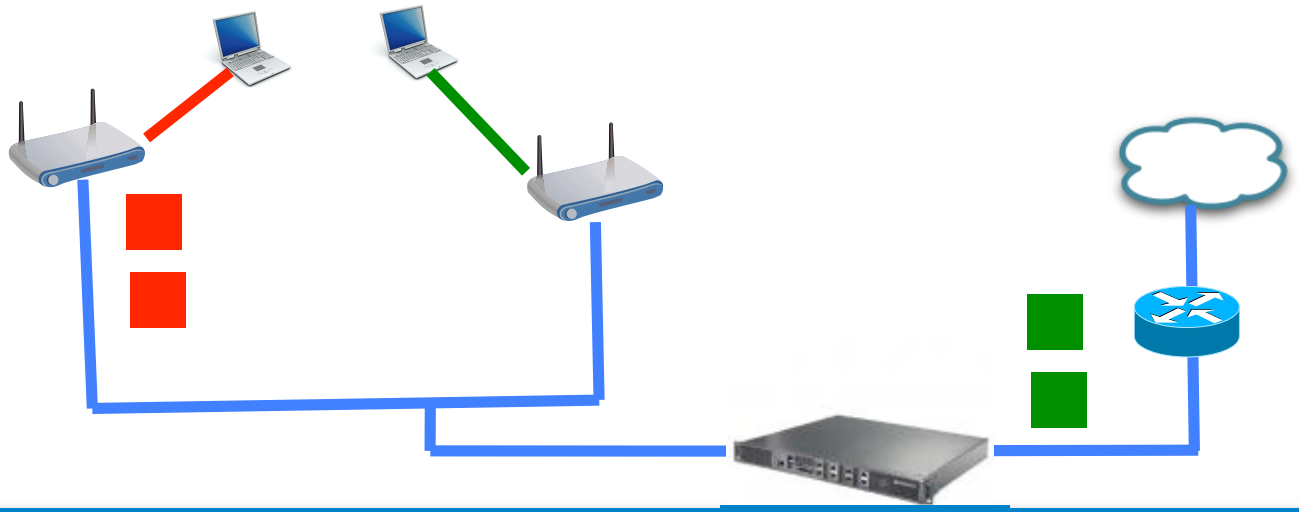
# (1) Avoid Scheduling Overheads

- Problem: Per-packet scheduling performs poorly under high network loads
- Solution:
  - Schedule packets in batches (or epochs)



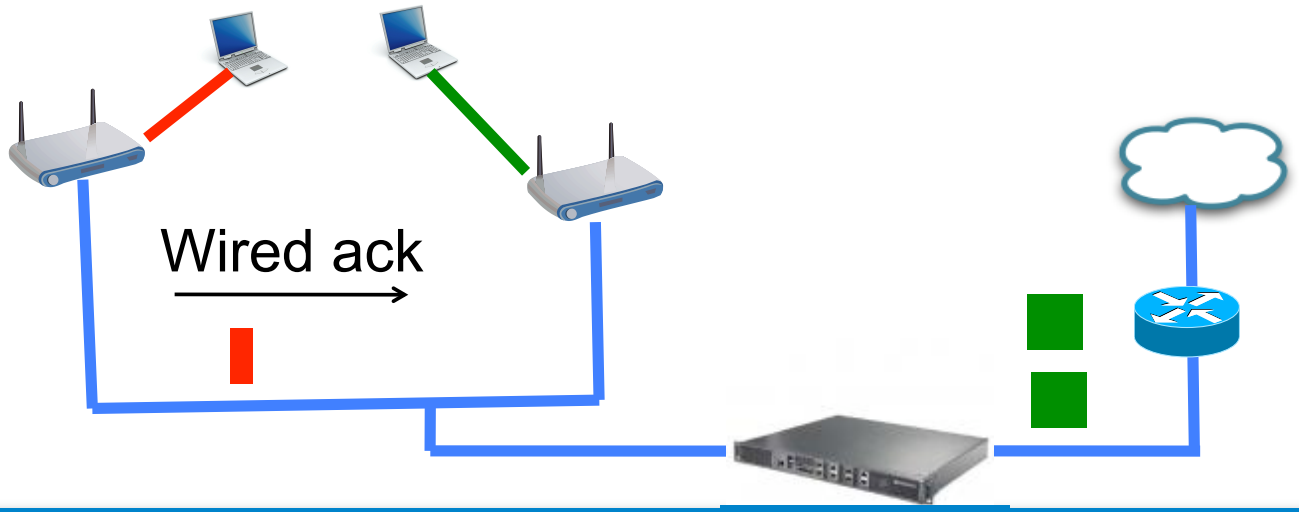
# (1) Avoid Scheduling Overheads

- Problem: Per-packet scheduling performs poorly under high network loads
- Solution:
  - Schedule packets in batches (or epochs)



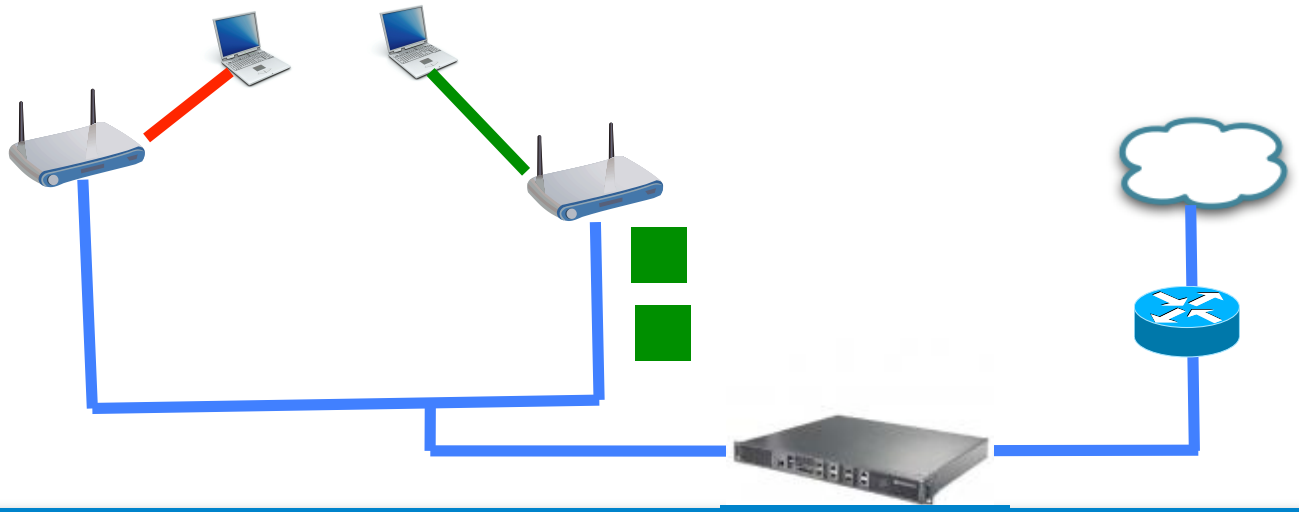
# (1) Avoid Scheduling Overheads

- Problem: Per-packet scheduling performs poorly under high network loads
- Solution:
  - Schedule packets in batches (or epochs)



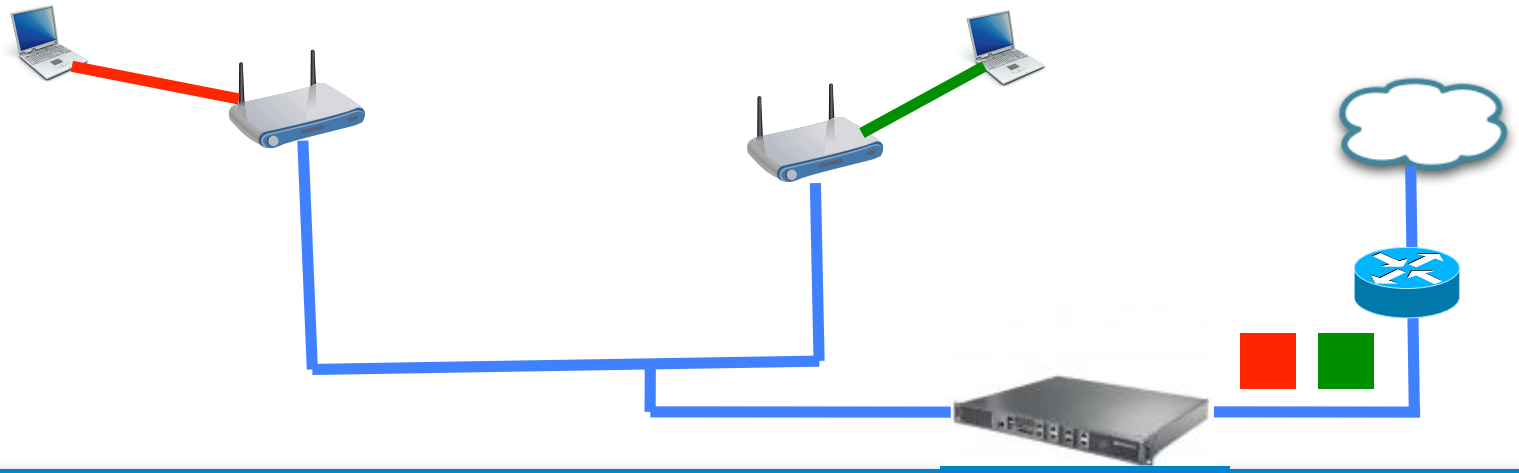
# (1) Avoid Scheduling Overheads

- Problem: Per-packet scheduling performs poorly under high network loads
- Solution:
  - Schedule packets in batches (or epochs)



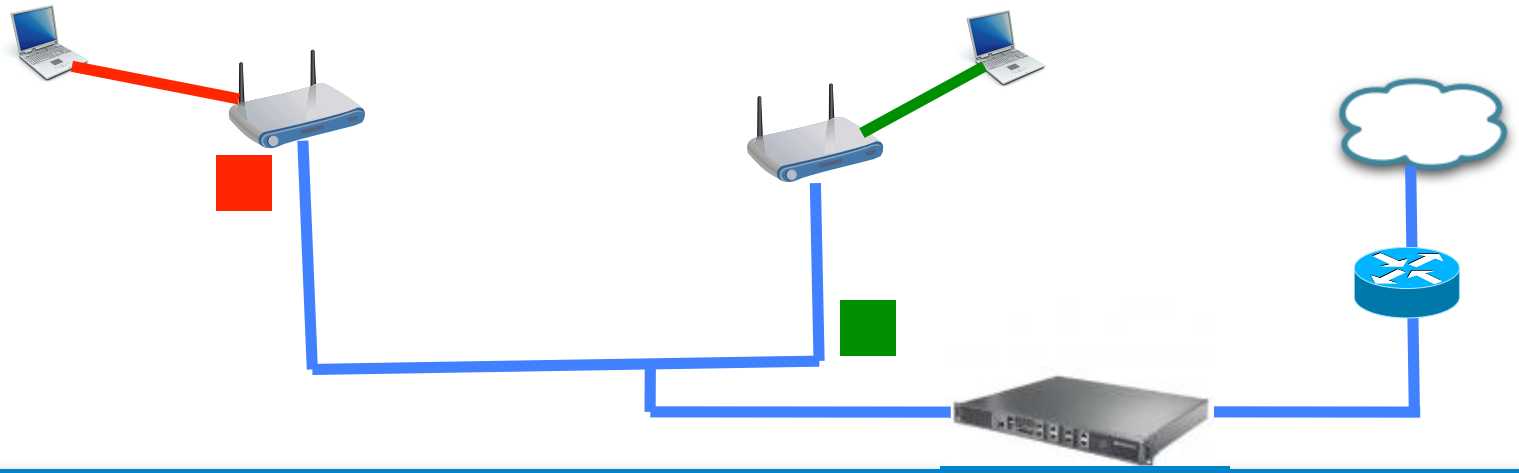
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



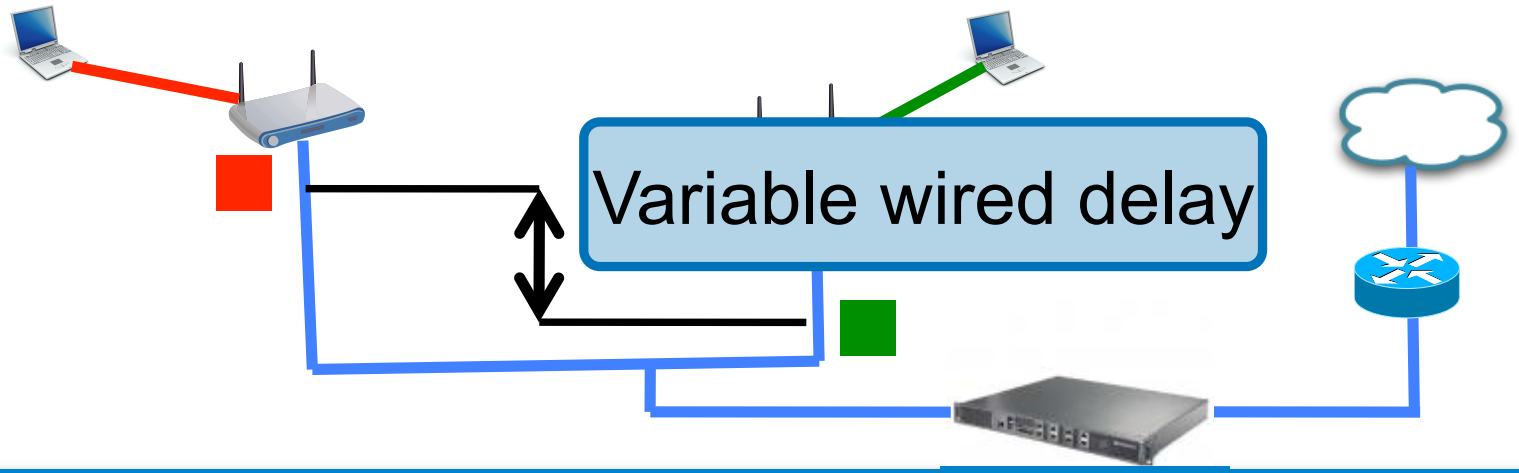
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



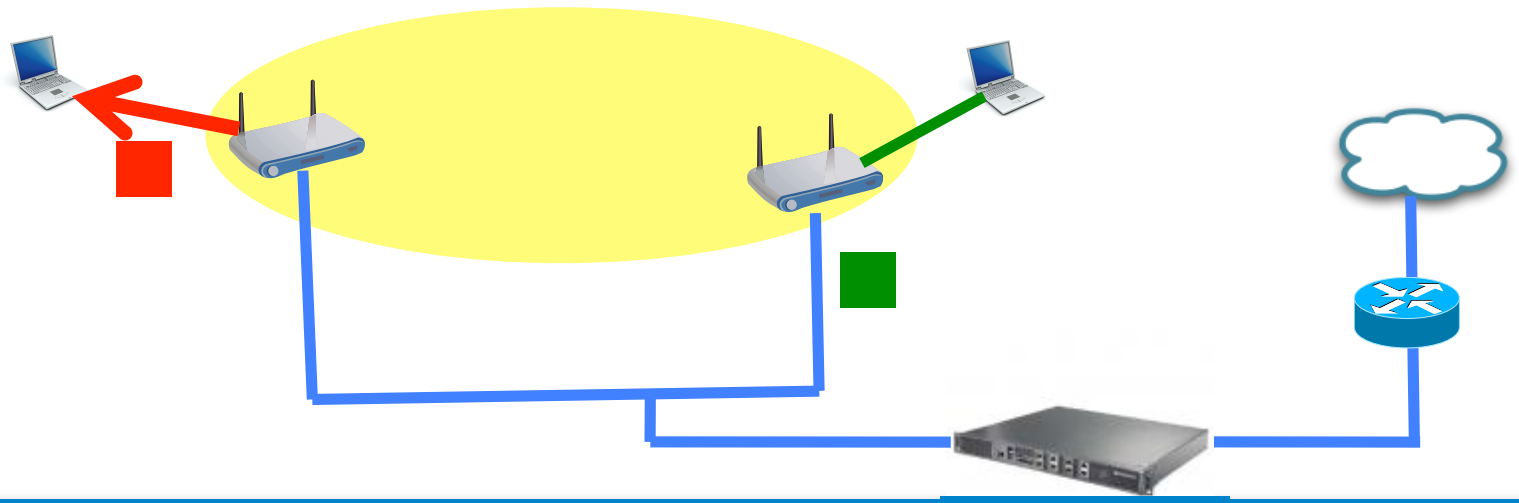
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



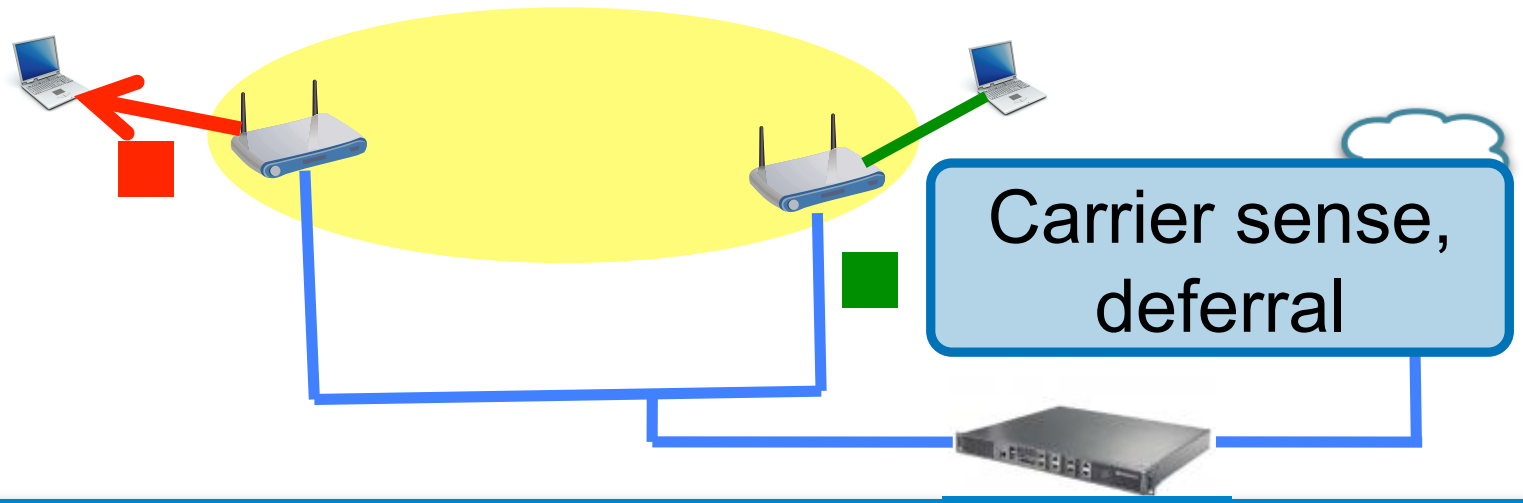
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



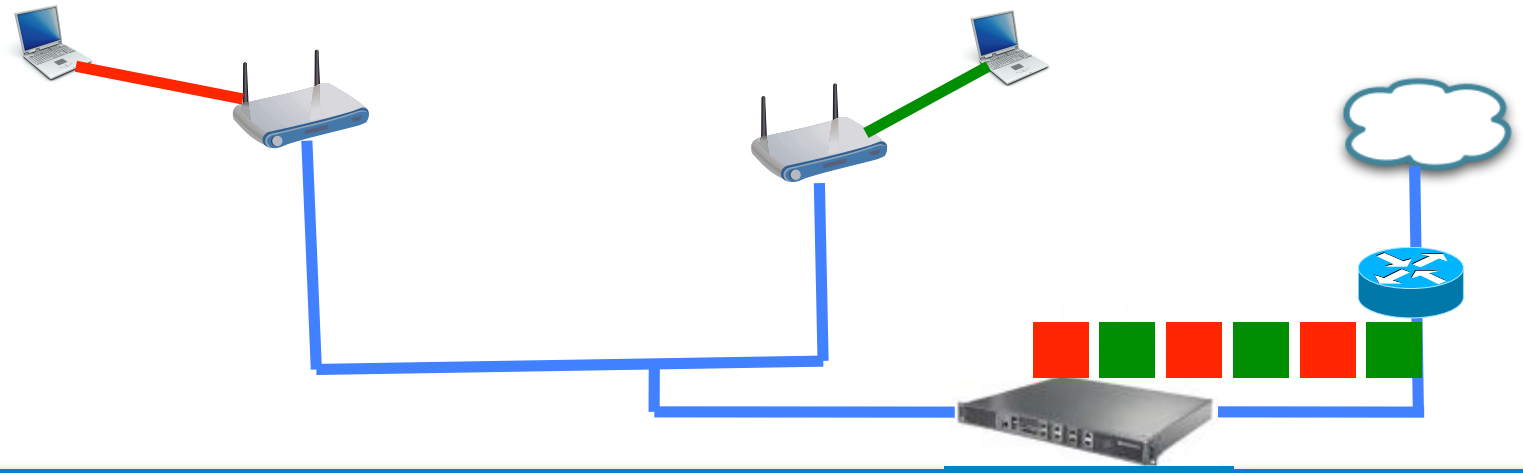
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



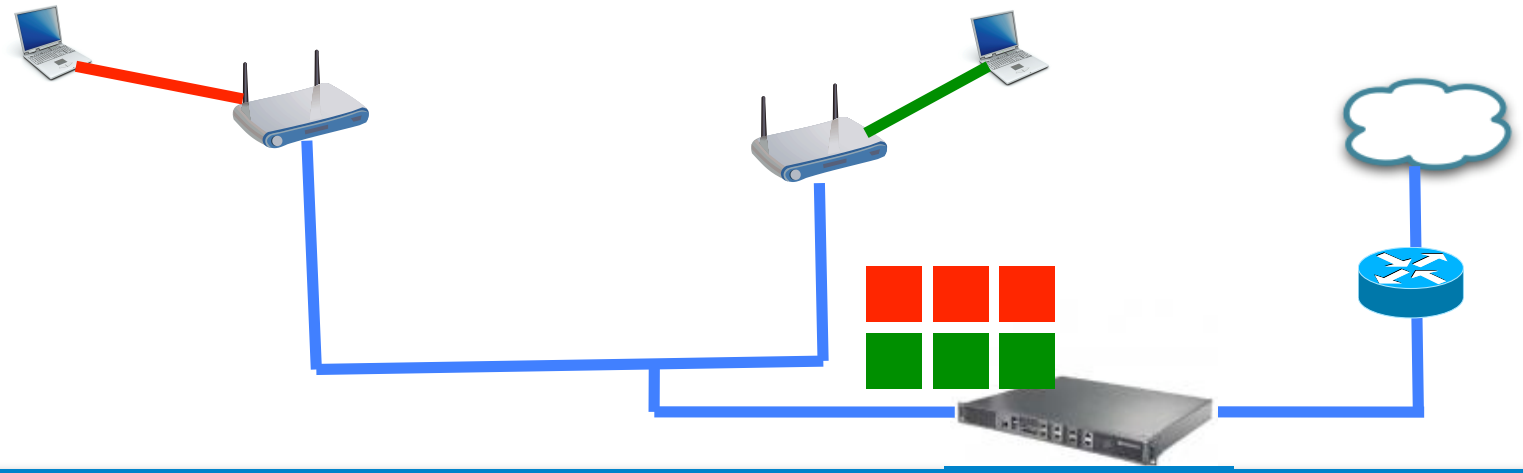
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



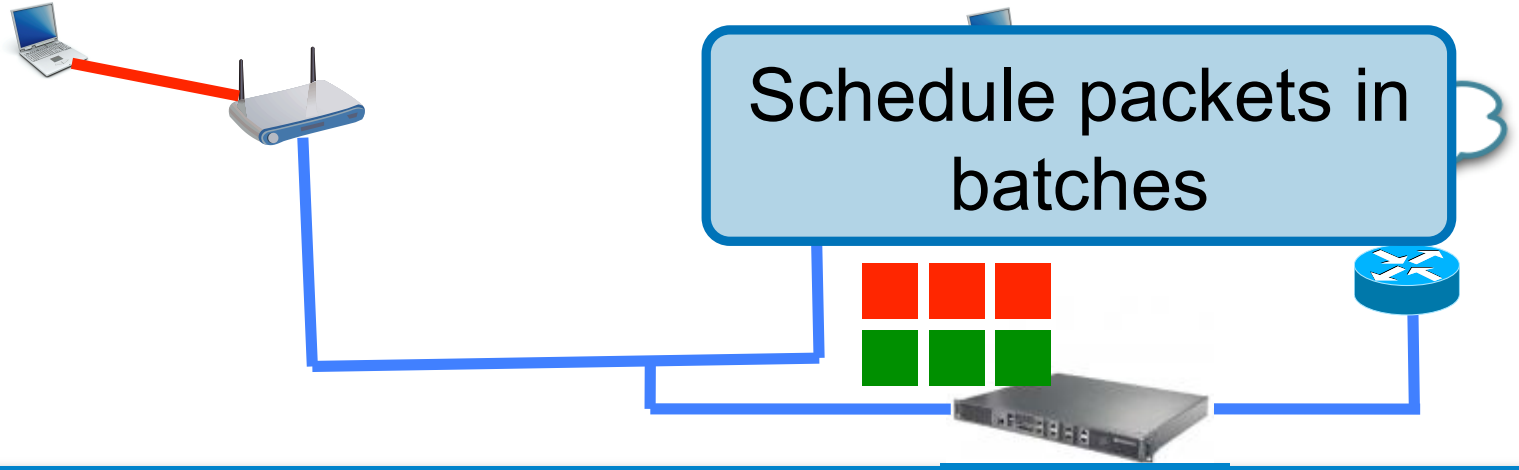
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



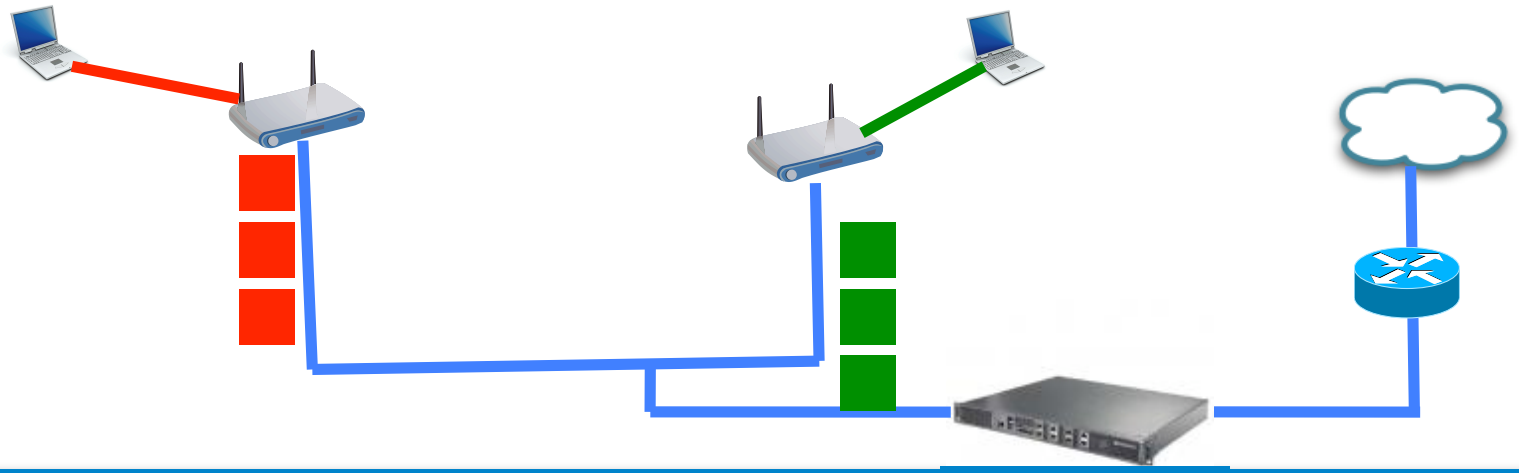
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



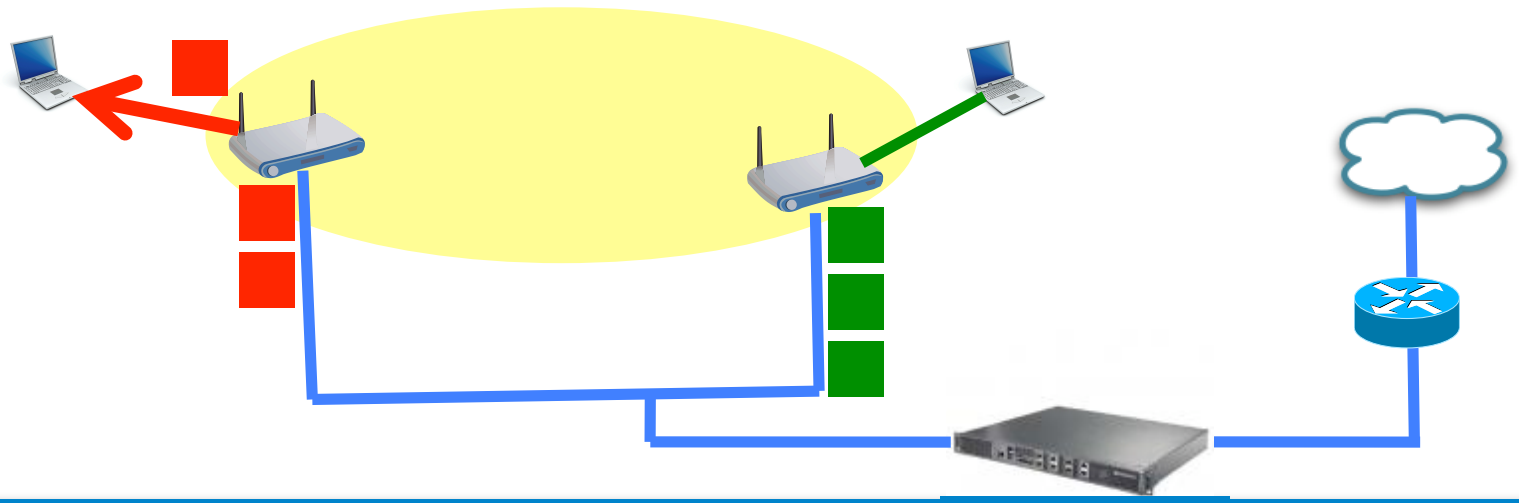
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



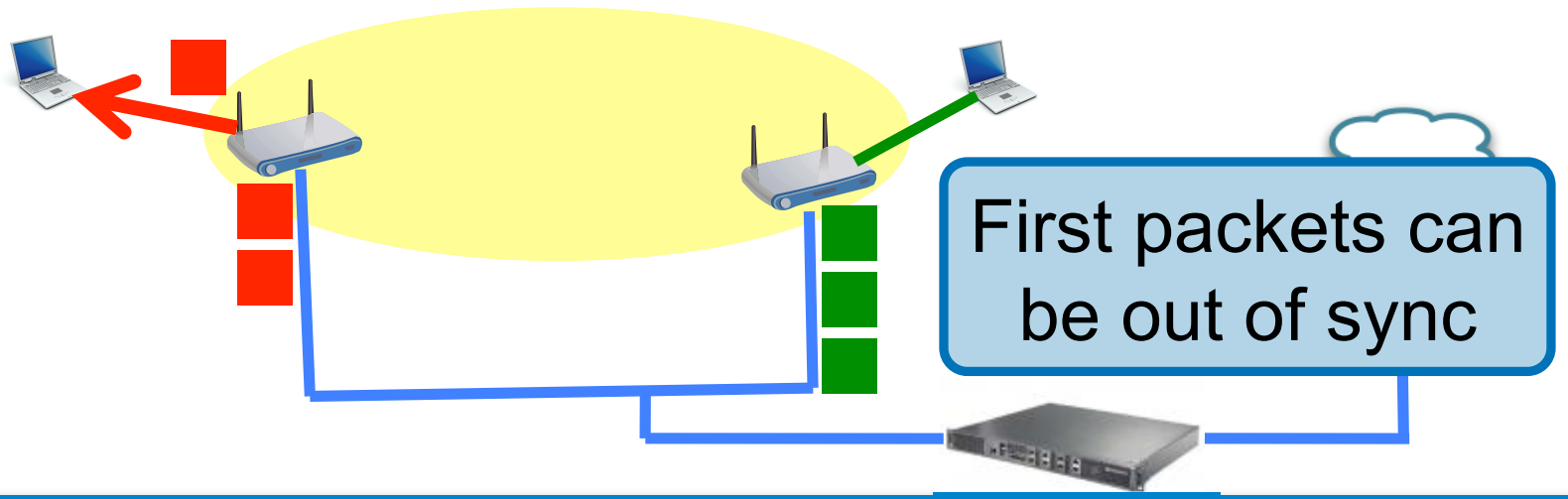
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



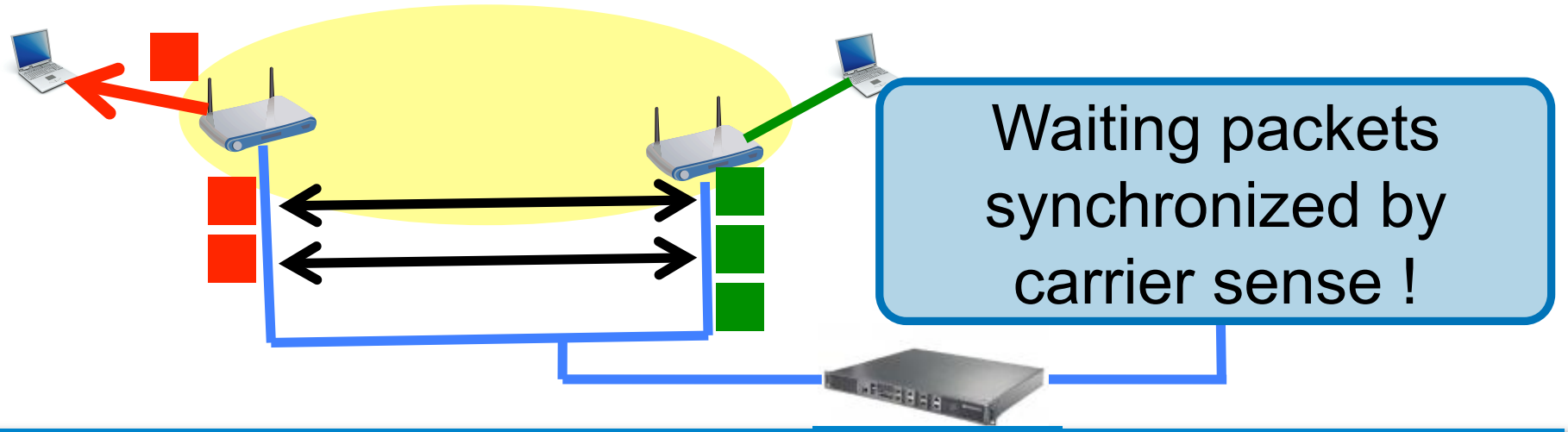
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



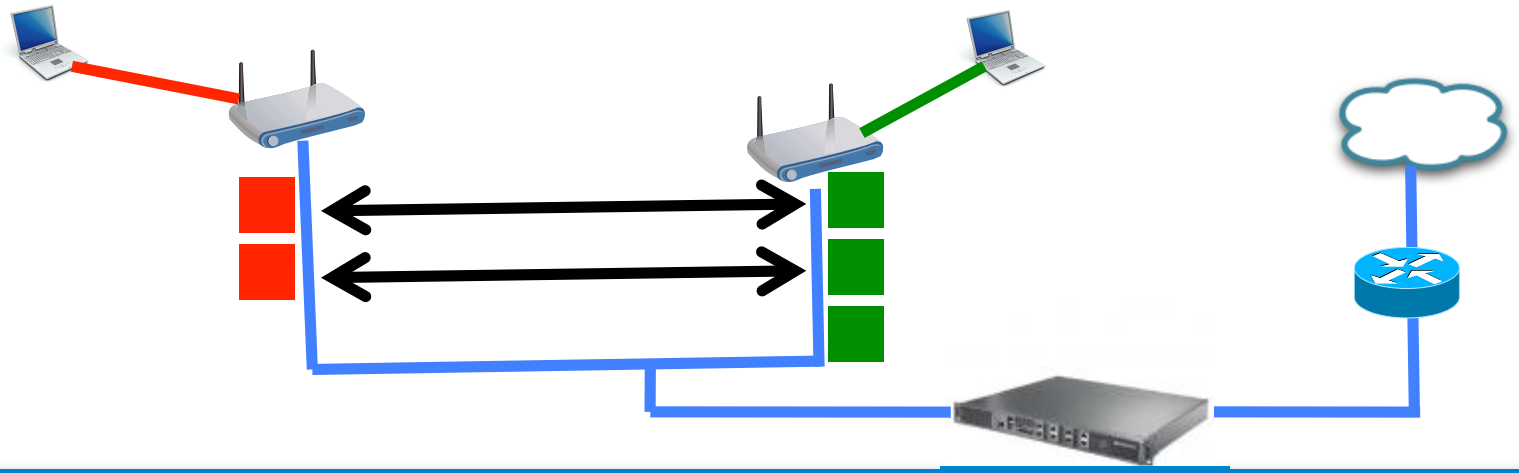
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



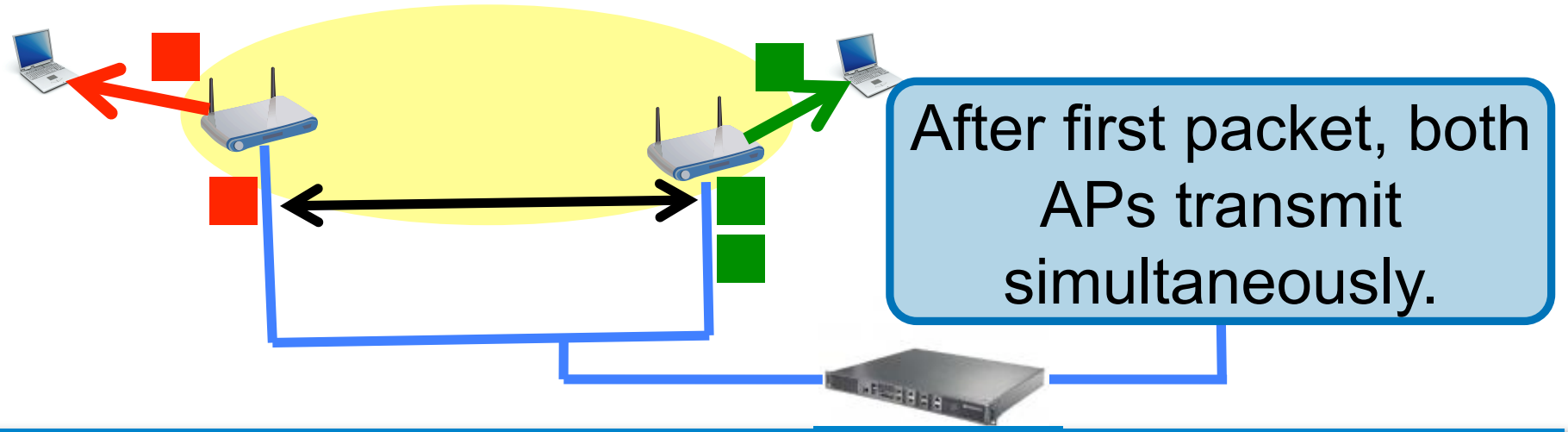
# (2) Improve Exposed Terminals

- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods

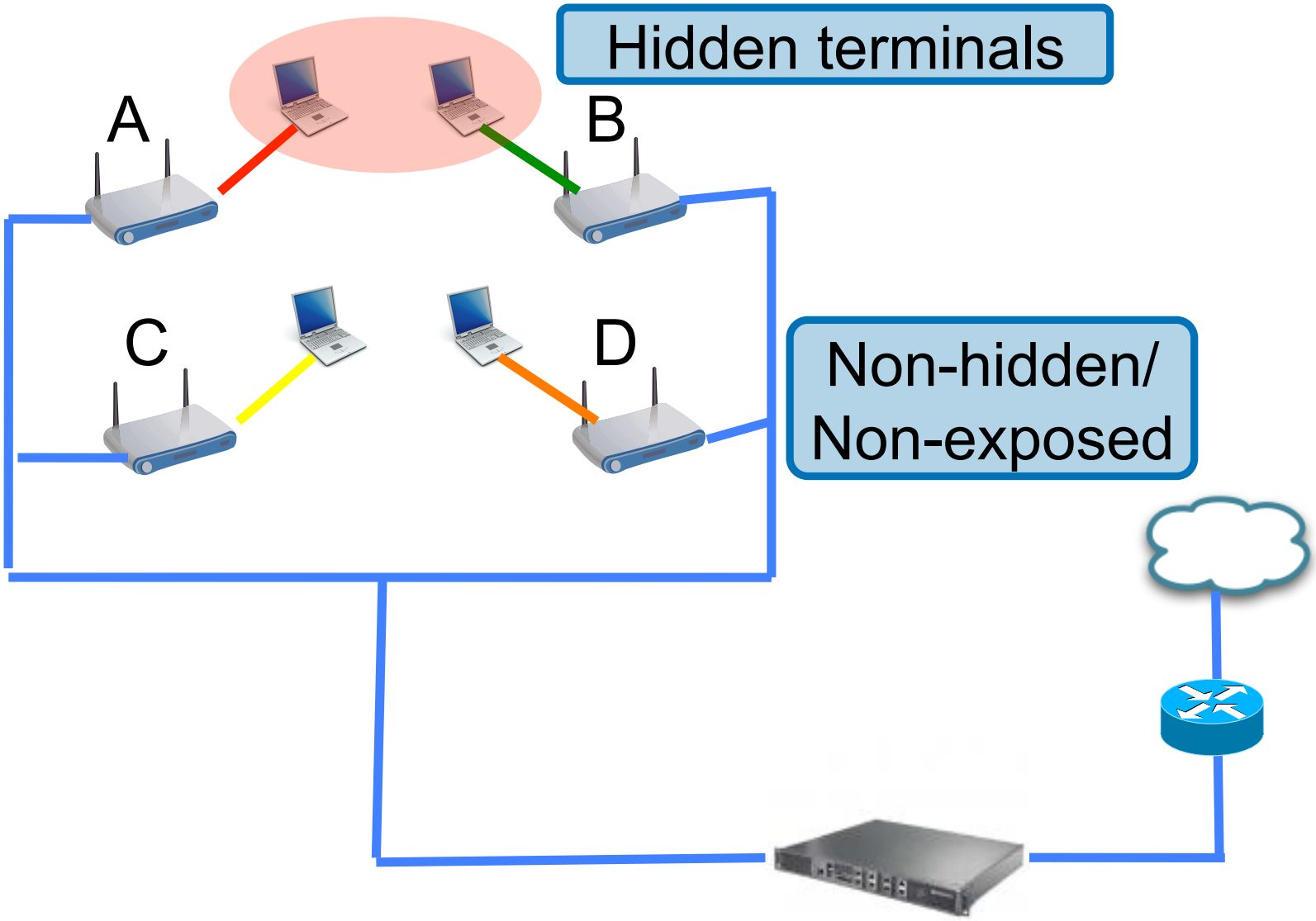


# (2) Improve Exposed Terminals

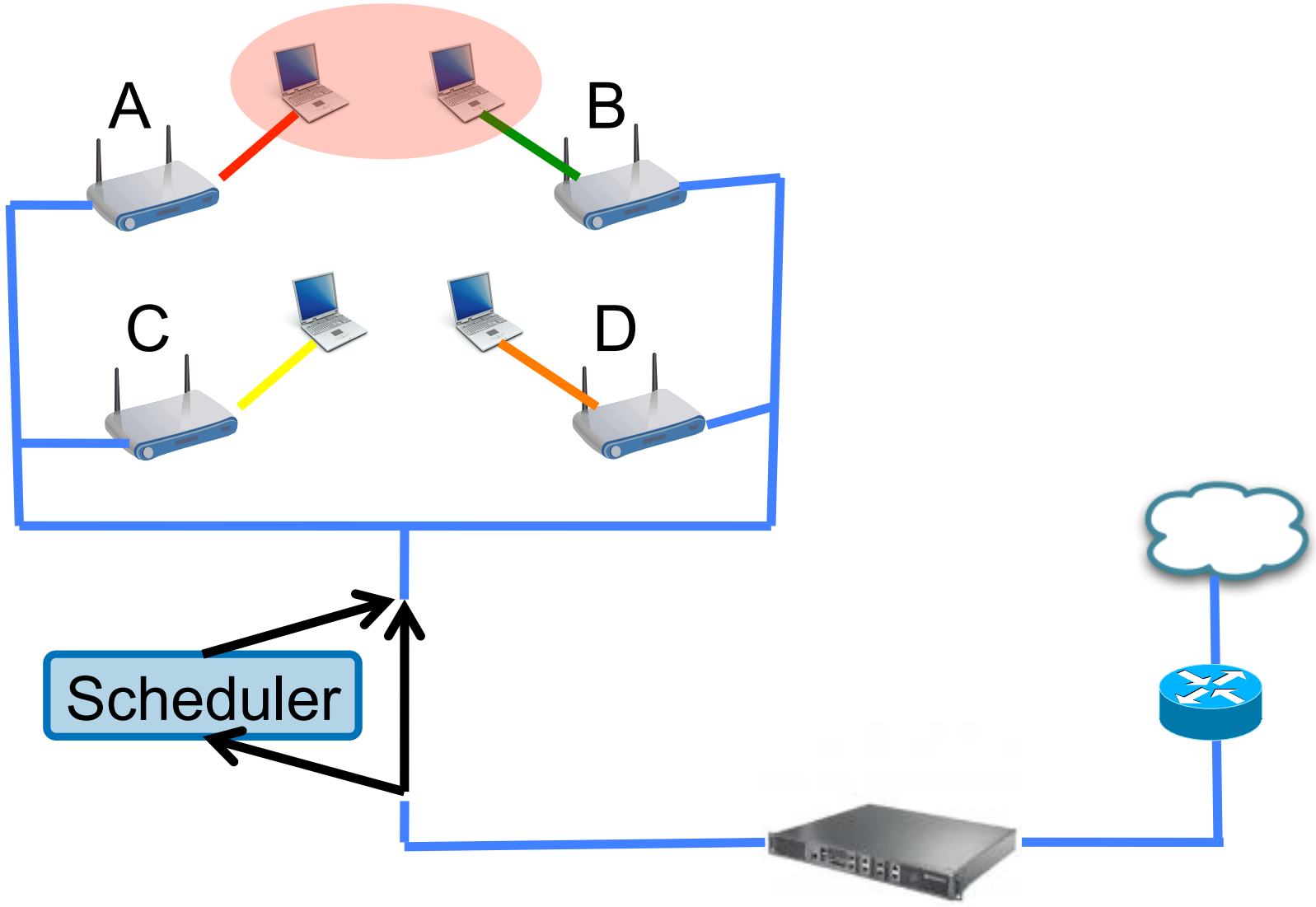
- Problem*:* Exposed links can operate in parallel but don't due to *carrier-sensing*
- Solution:
  - Schedule batch of packets, fix backoff periods



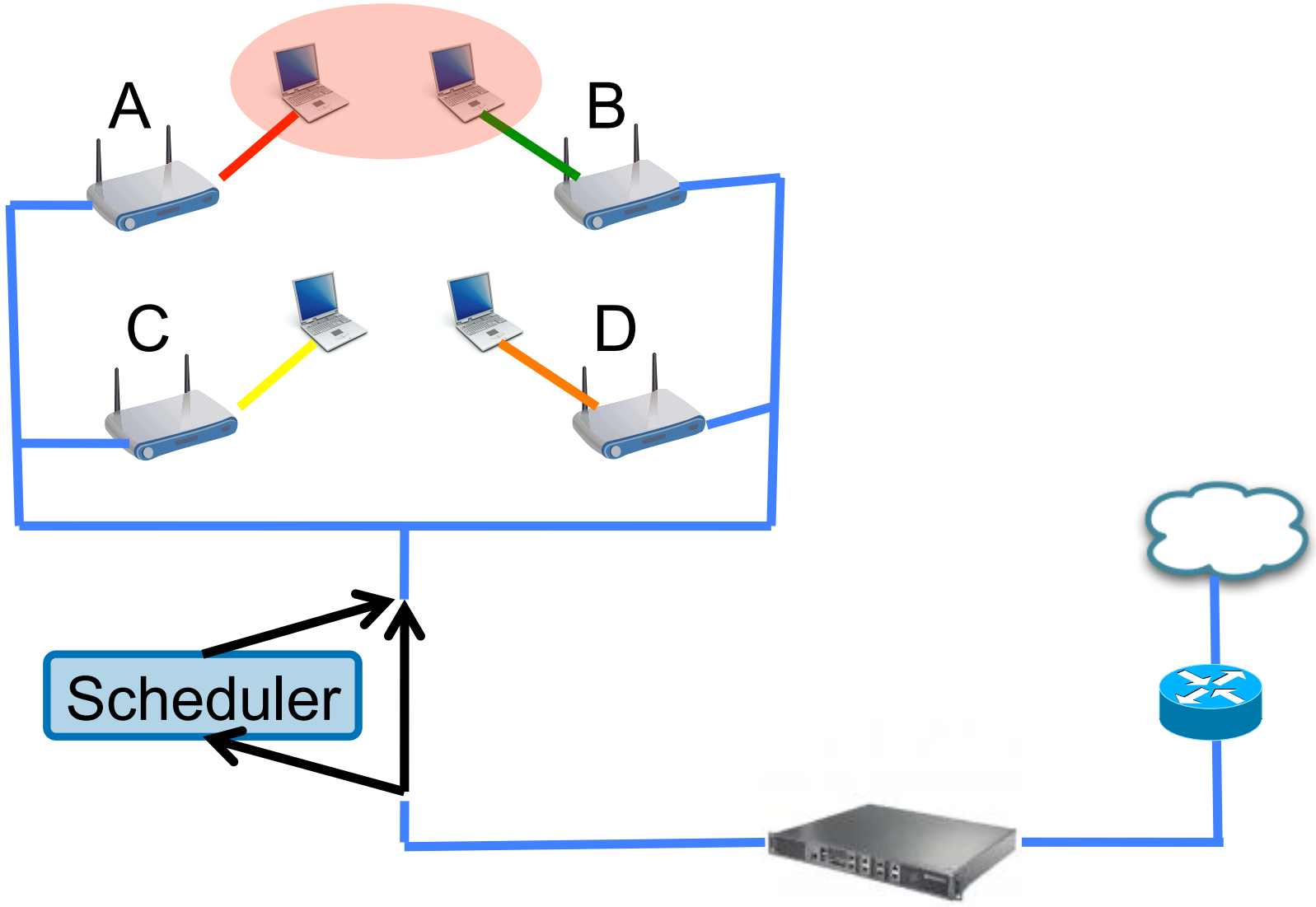
# (3) Avoid Degrading Normal Links



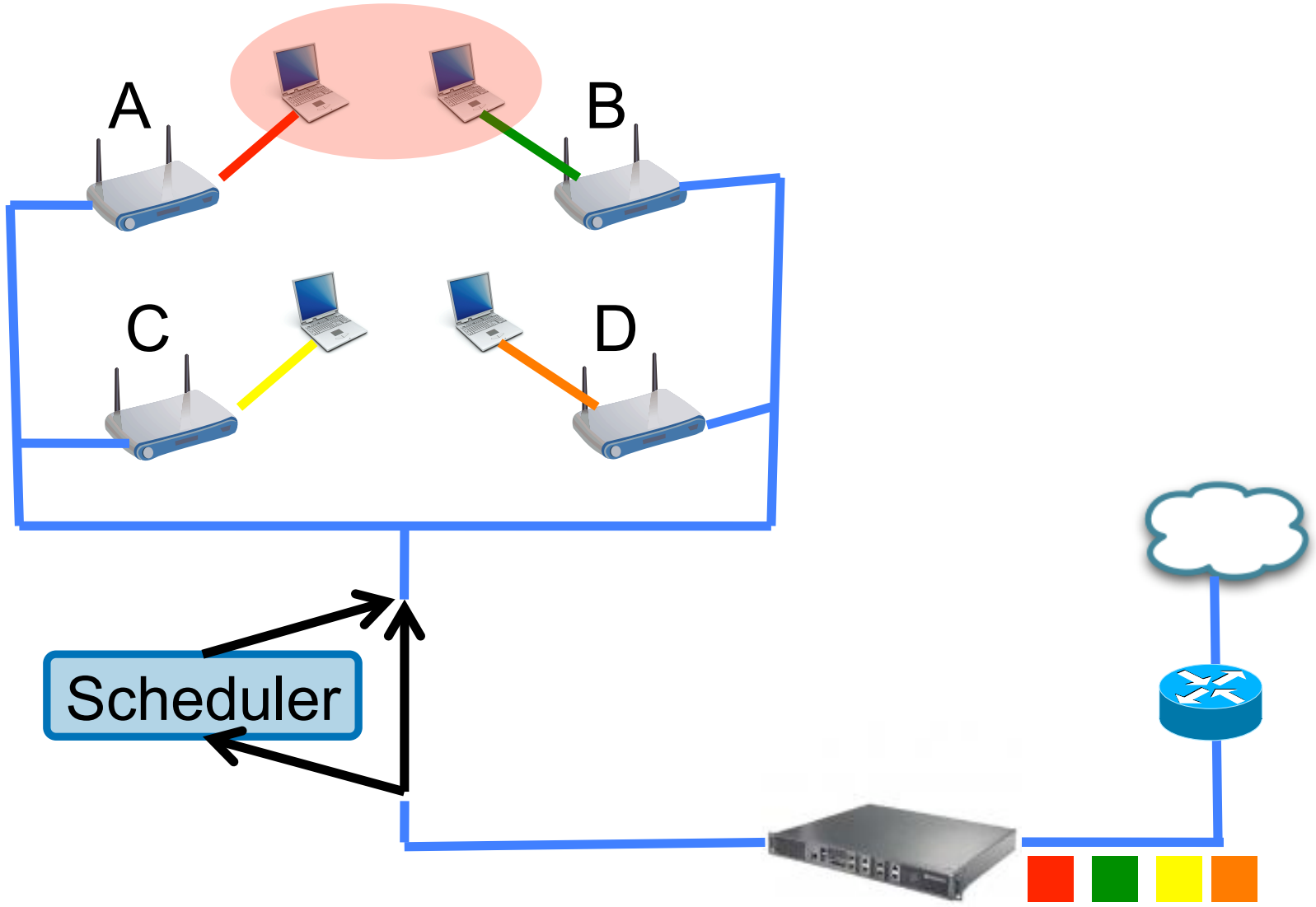
# (3) Avoid Degrading Normal Links



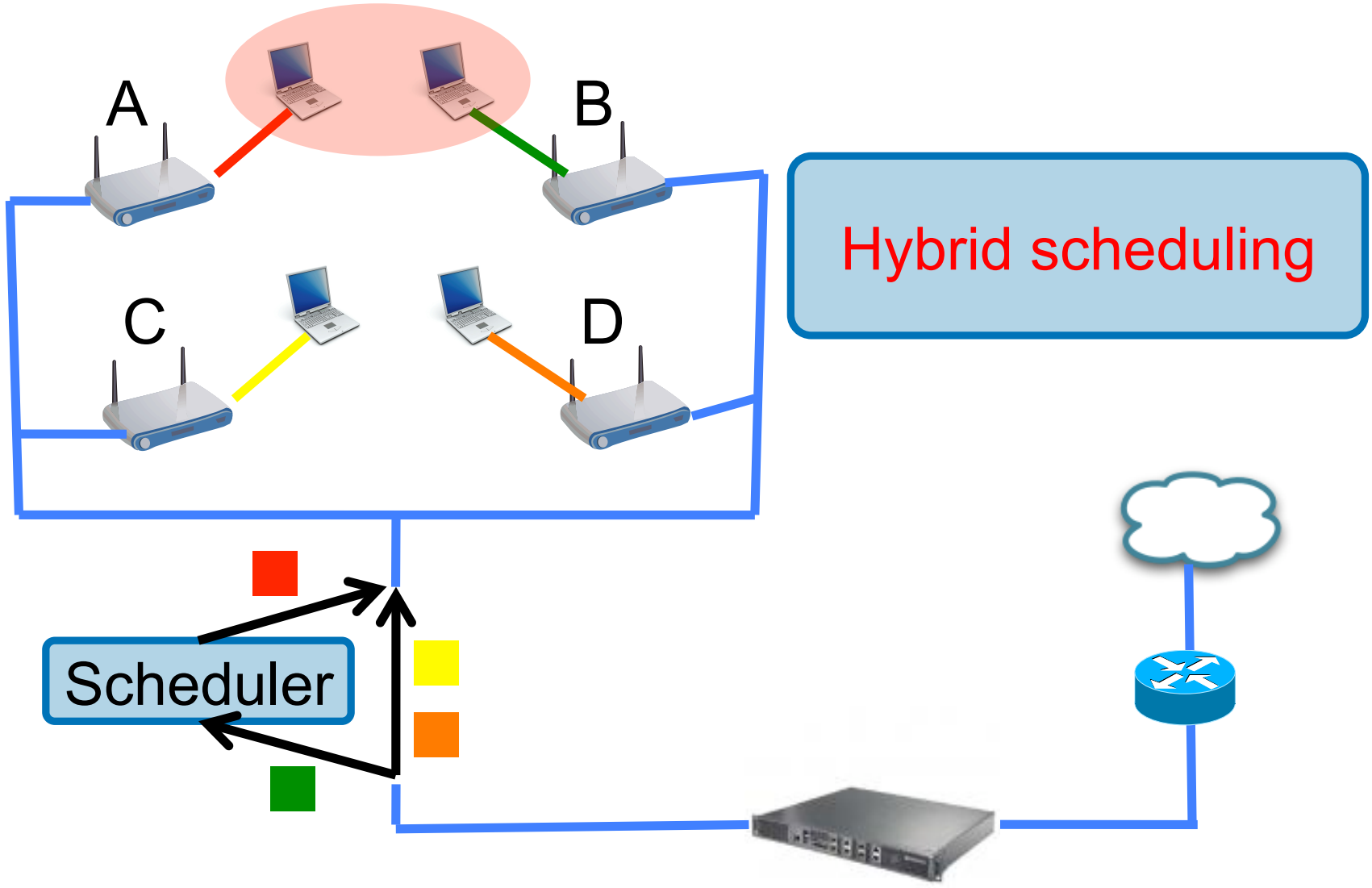
# (3) Avoid Degrading Normal Links



# (3) Avoid Degrading Normal Links



# (3) Avoid Degrading Normal Links



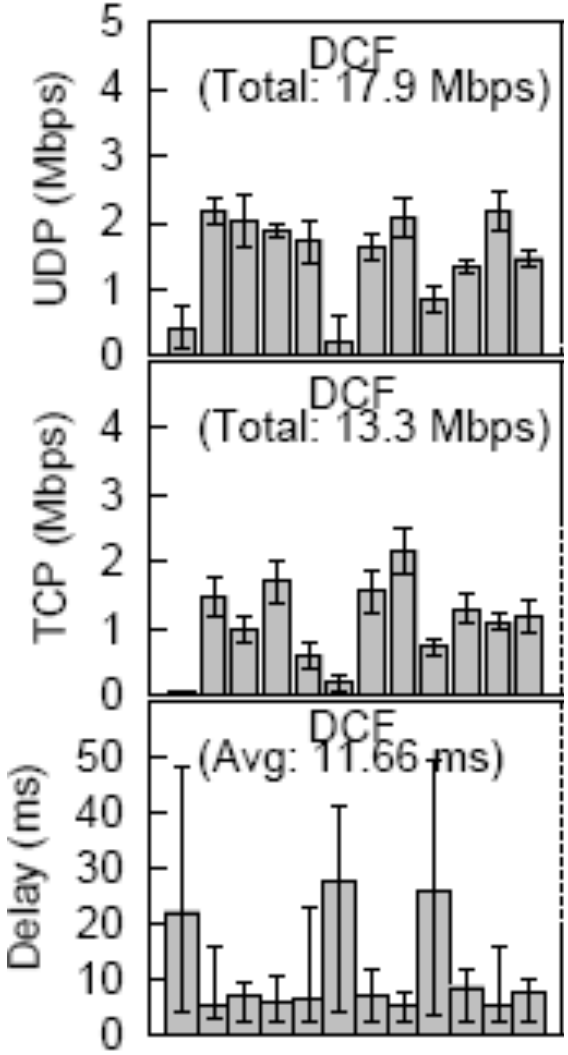
# Outline

- A naïve attempt at centralized scheduling (DET)
- Our hybrid centralized scheduler (CENTAUR)
- **Evaluation**
- Related Work
- Summary

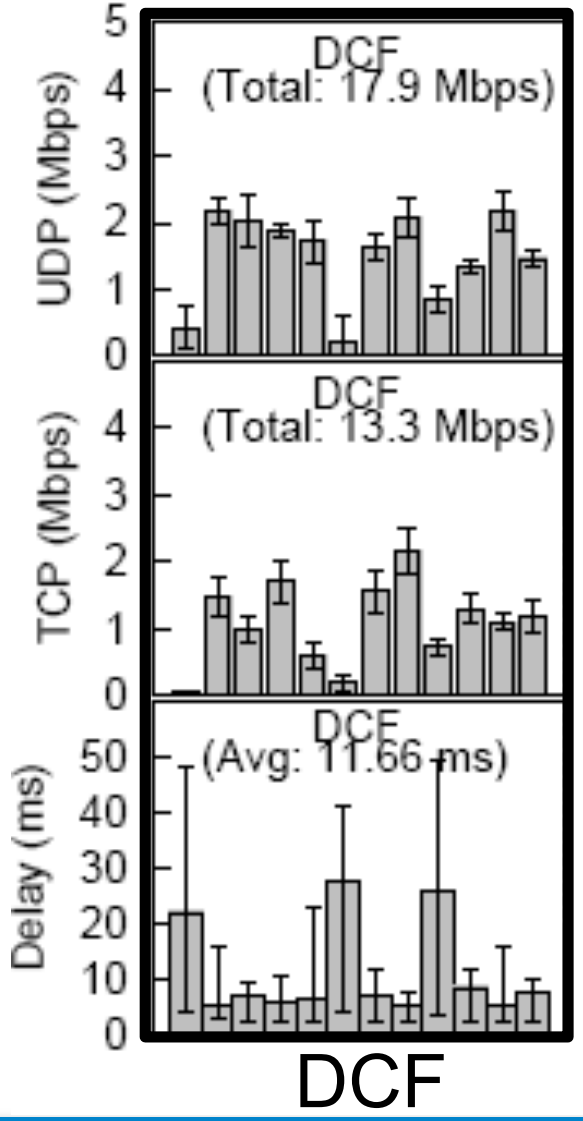
# Large-Scale Experiments

- Platform:
  - Two WLAN testbeds in separate buildings
- Topology
  - Representative*:* 7 APs, 12 clients
- Traffic and metrics
  - UDP, TCP, VoIP, HTTP (real traces)
  - Throughput, delay, MOS, web transaction delay

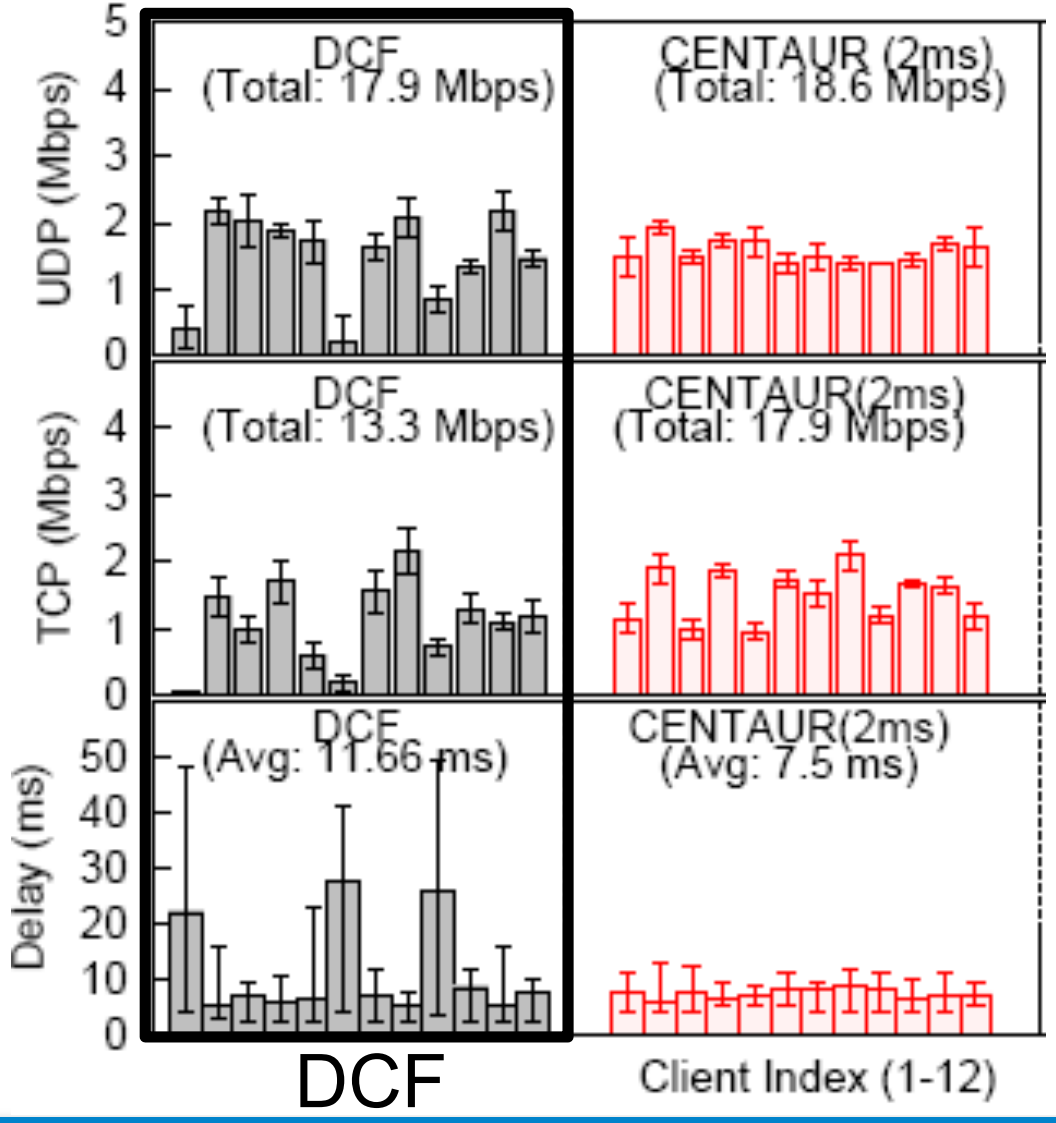
# Result 1: UDP/TCP Performance



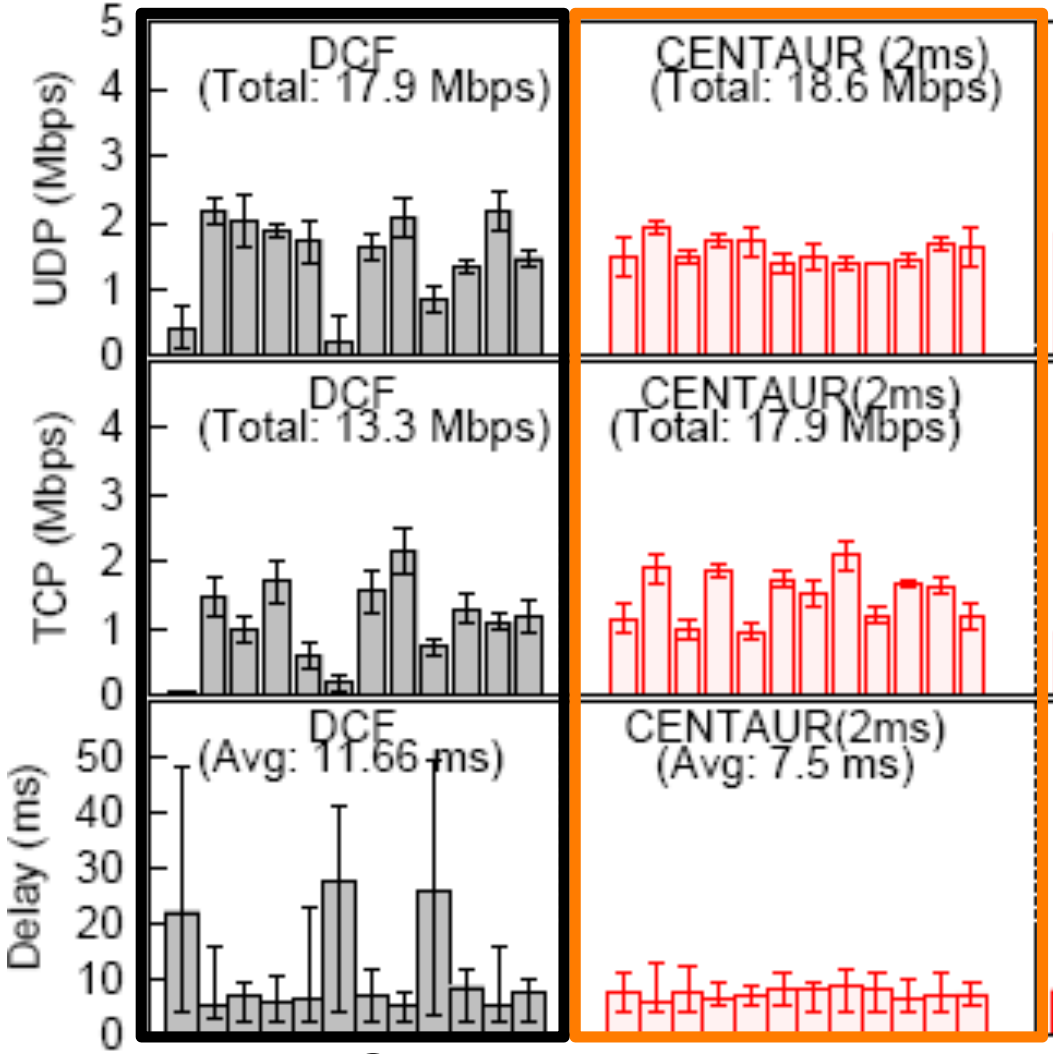
# Result 1: UDP/TCP Performance



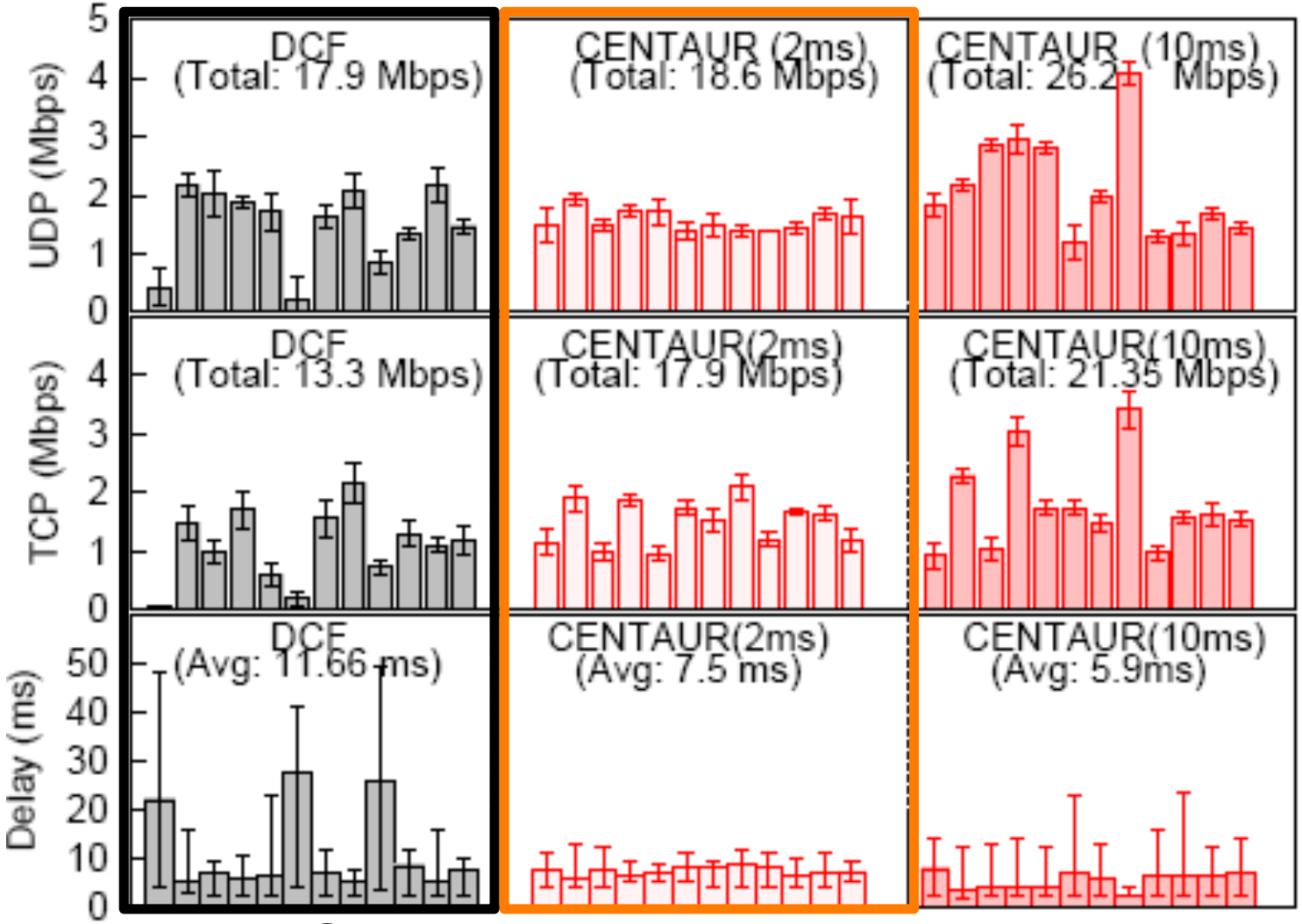
# Result 1: UDP/TCP Performance



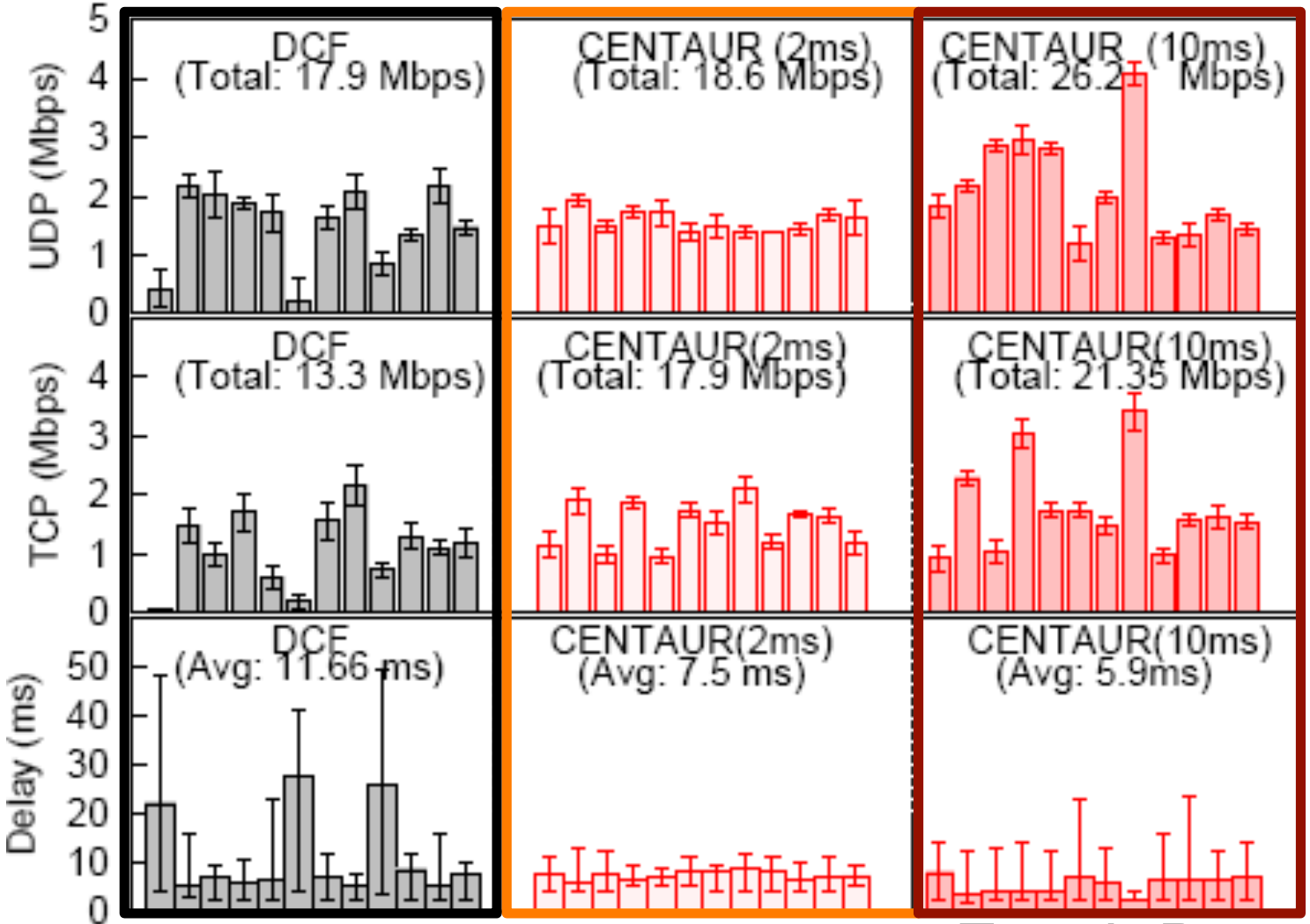
# Result 1: UDP/TCP Performance



# Result 1: UDP/TCP Performance

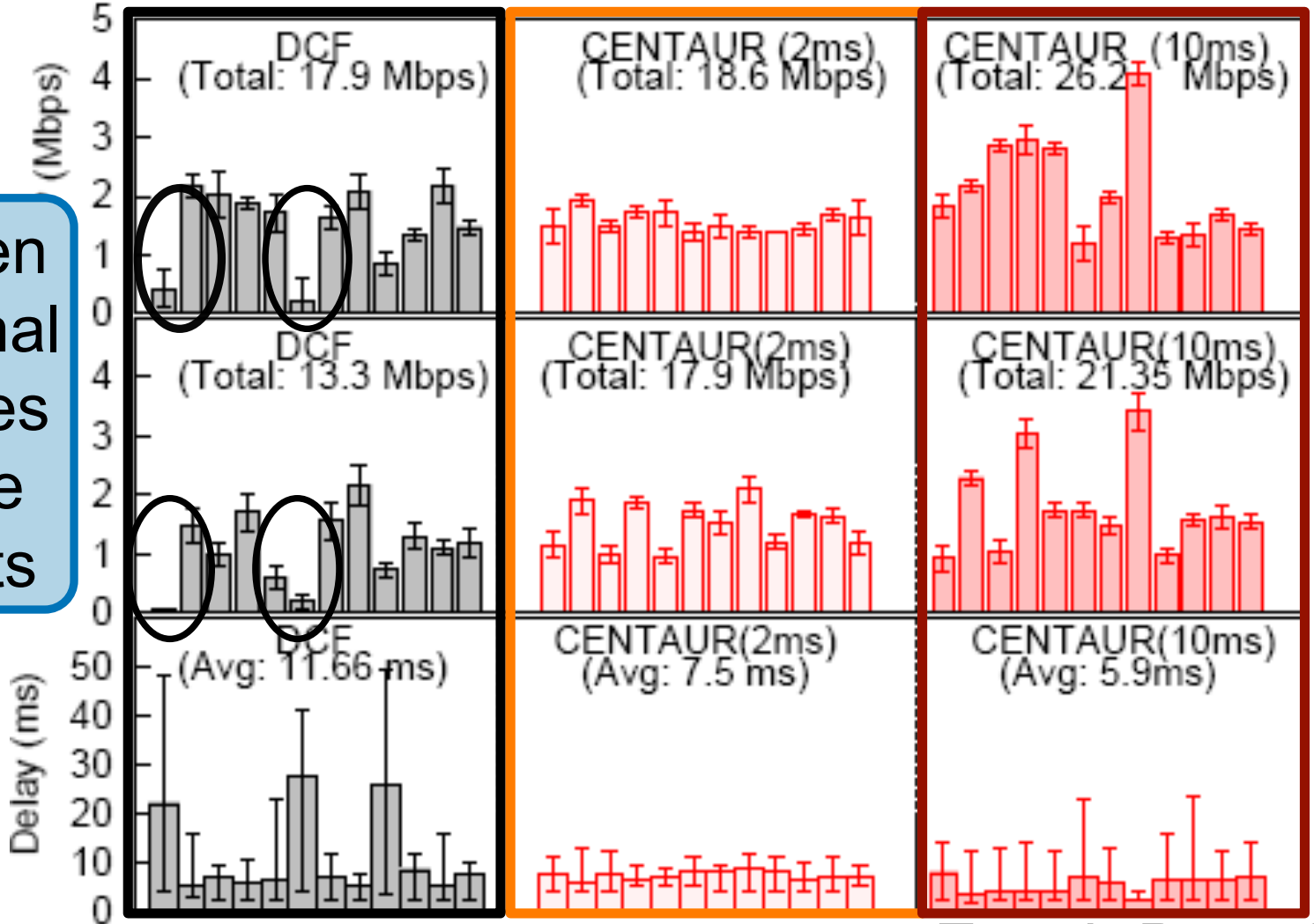


# Result 1: UDP/TCP Performance



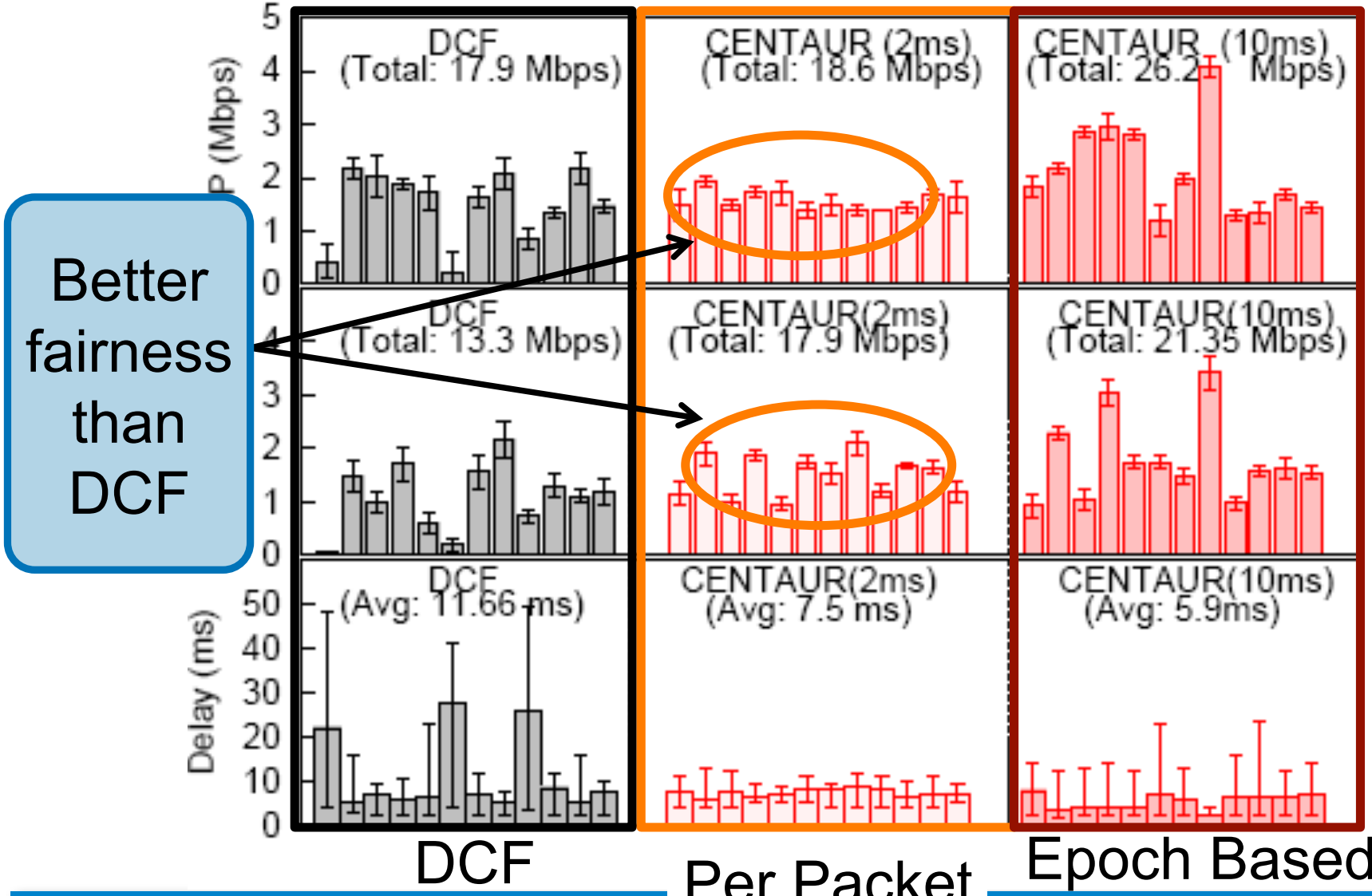
# Result 1: UDP/TCP Performance

Hidden terminal starves some clients

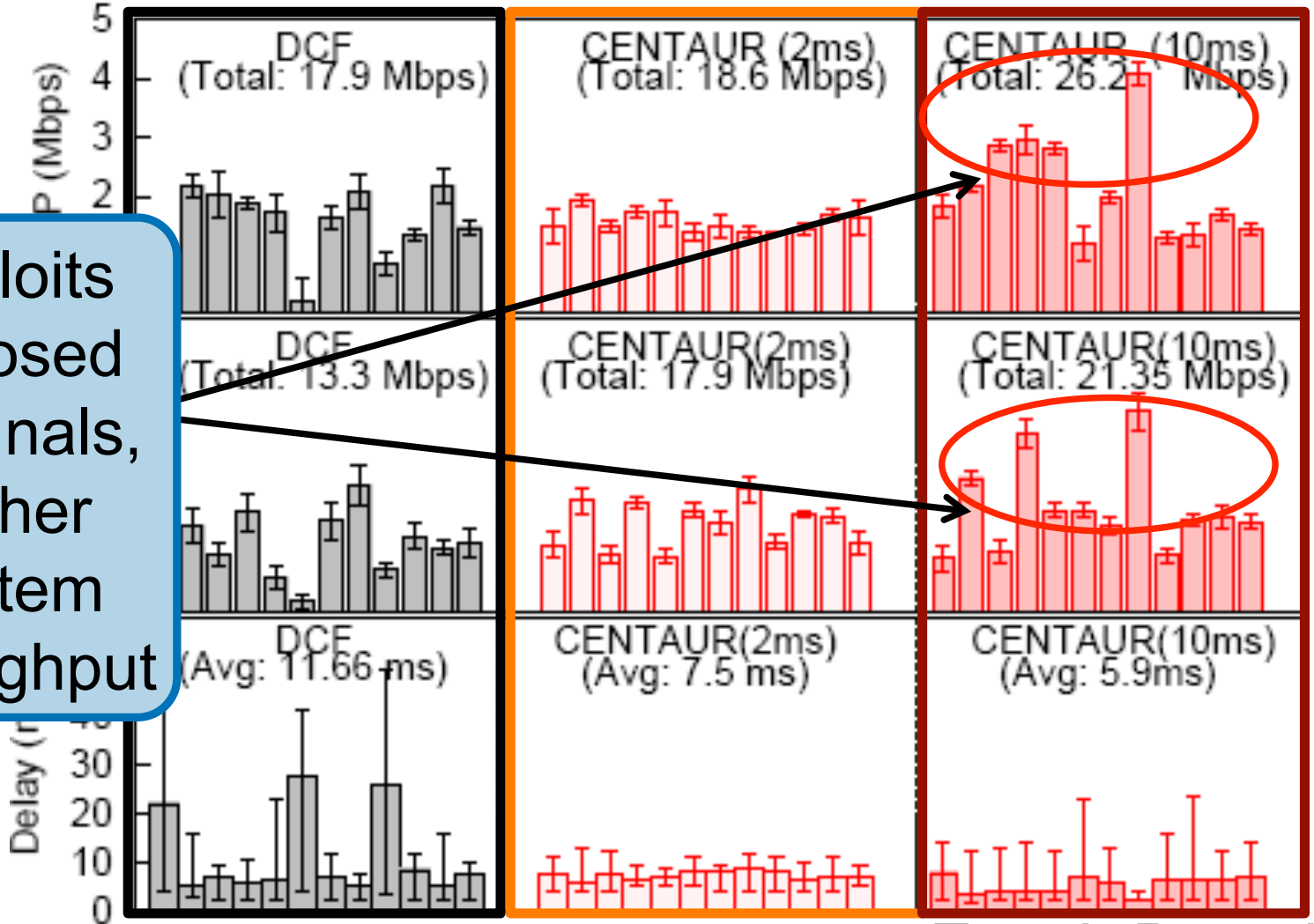



DCF | Per Packet | Epoch Based

# Result 1: UDP/TCP Performance

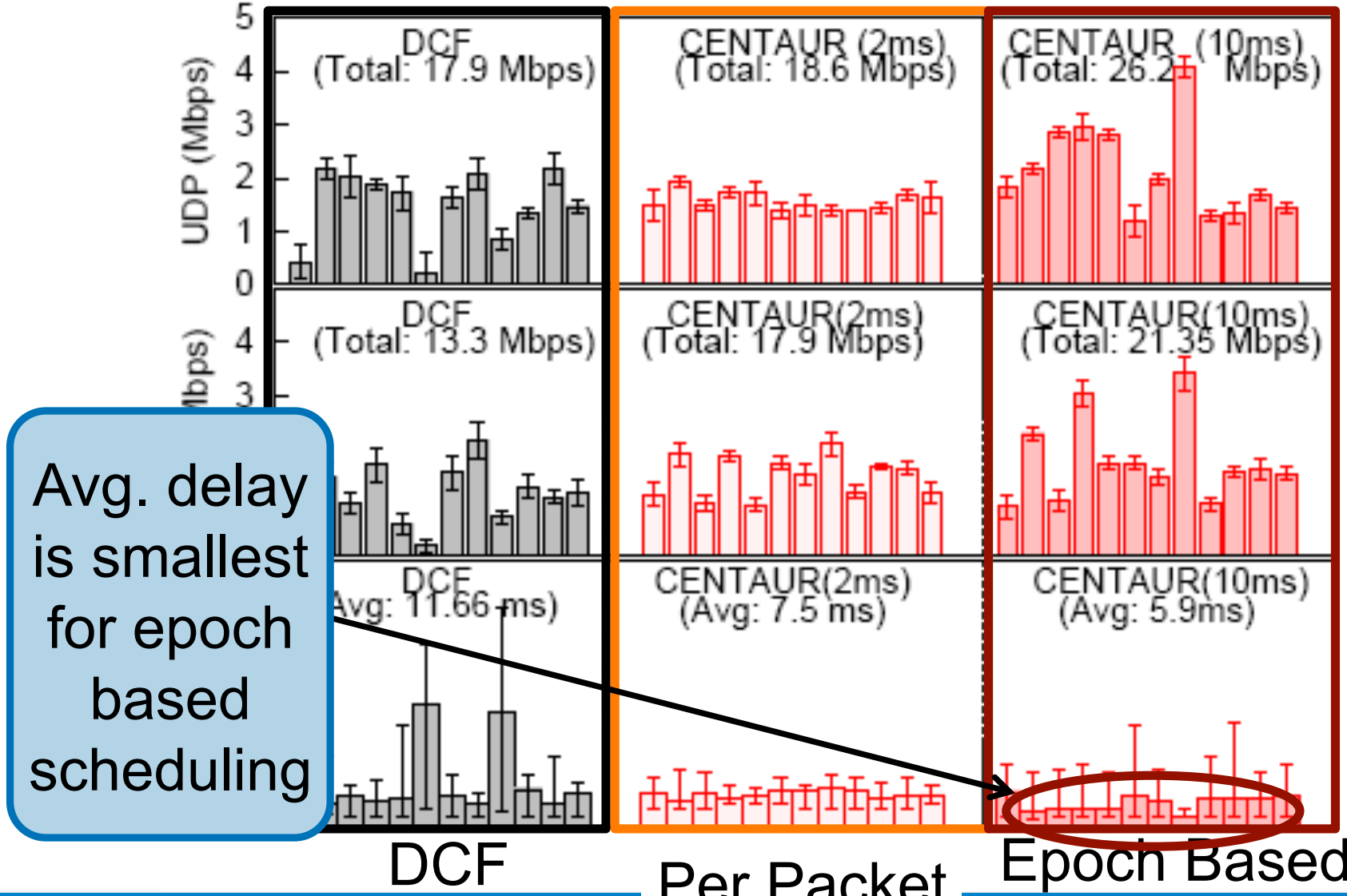


# Result 1: UDP/TCP Performance

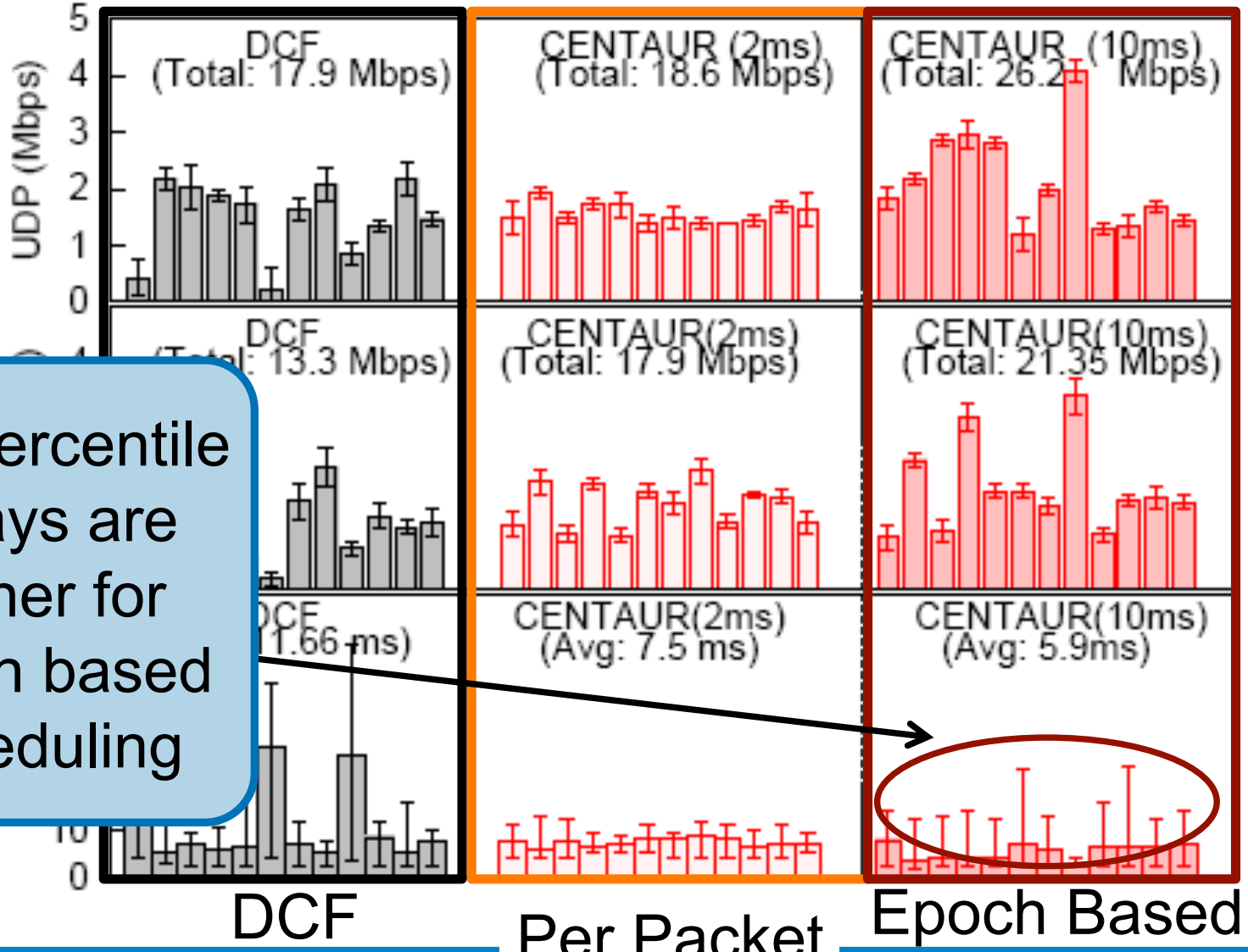



Exploits exposed terminals, higher system throughput

# Result 1: UDP/TCP Performance

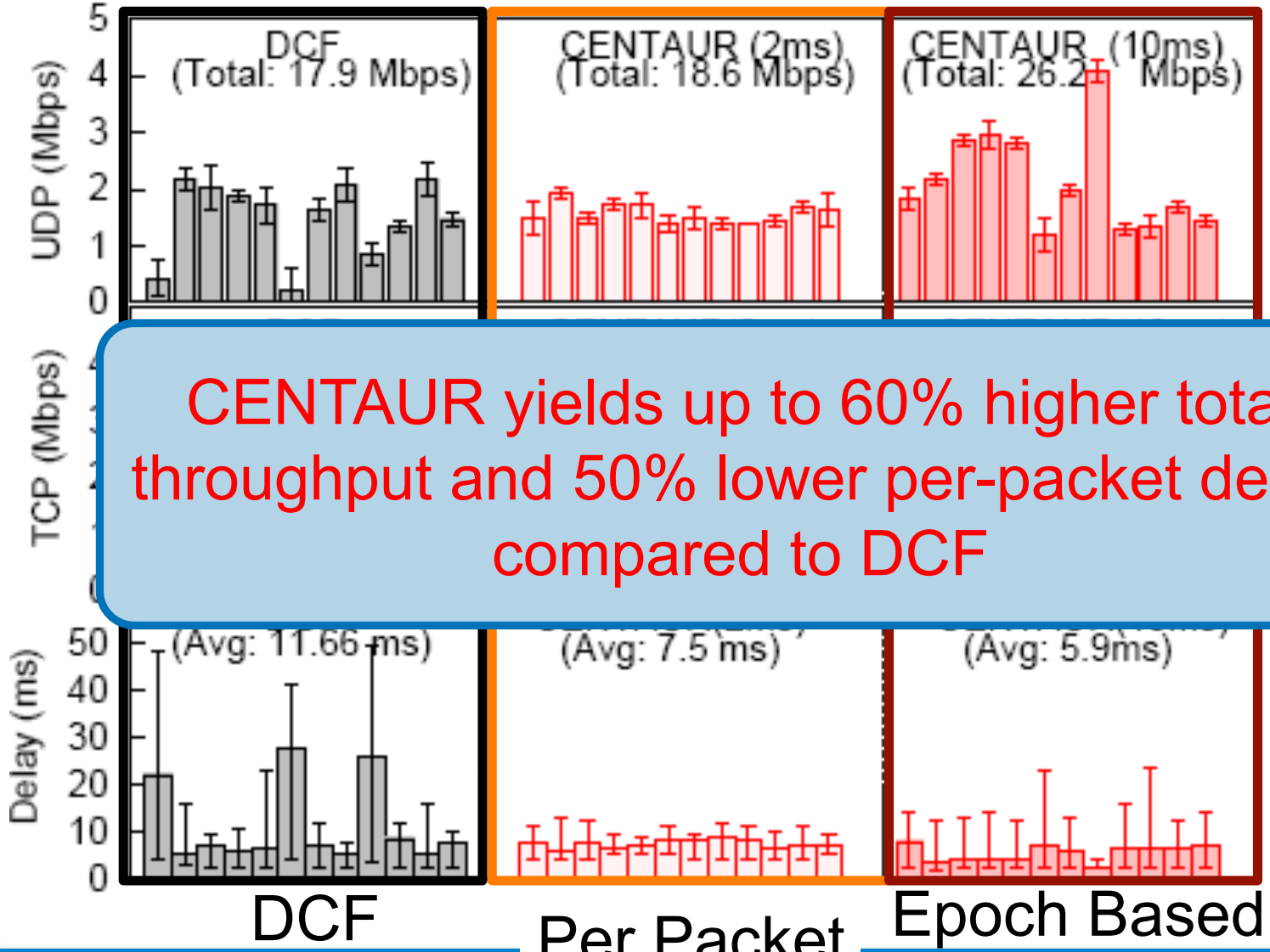


# Result 1: UDP/TCP Performance




90th percentile delays are higher for epoch based scheduling

# Result 1: UDP/TCP Performance




CENTAUR yields up to 60% higher total throughput and 50% lower per-packet delay compared to DCF

# More Results in Paper

- Centaur micro-benchmarks: performance for exposed and hidden terminals under centaur
- Uplink traffic: coexistence and persistent gains with different fractions of uplink traffic
- Data rate: robust to changes in data rate and ARF
- Realistic HTTP traces: significant reduction in web transaction delay
- VoIP traffic: better performance (*MOS*) for voice traffic with small epoch duration

# Related Work

- Commercial WLAN offerings (Aruba, Meru)
- Theoretical formulations (Vaidya ‘00, Kanodia ‘01)
- Epoch based scheduling (Kompella ‘05, 802.11n/e)
- Interference mitigation (CMAP, SIC, Shuffle)

# Summary

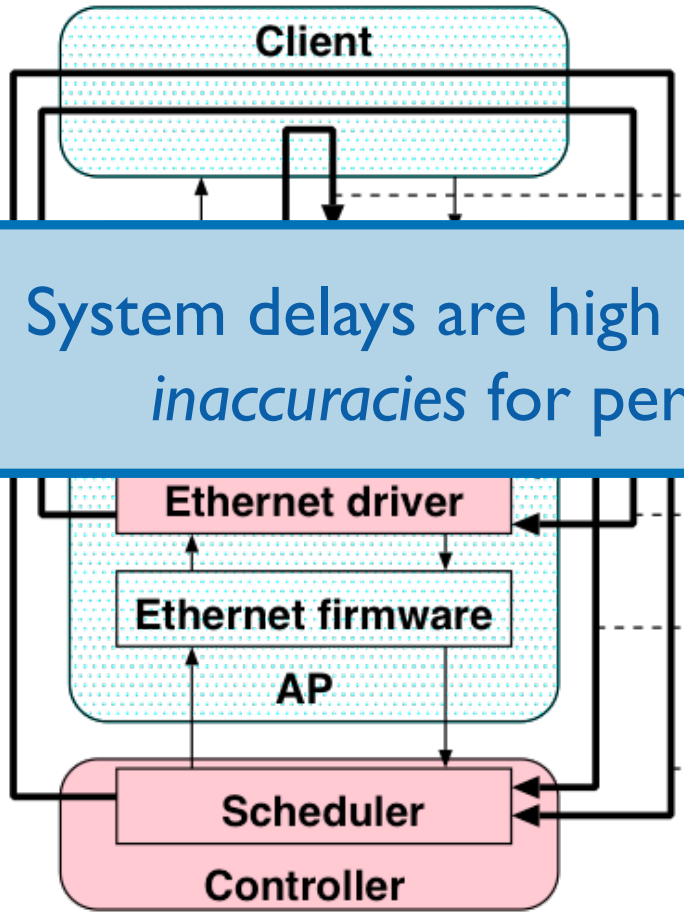
- Interference a growing problem in enterprises
- Careful design of a centralized data plane provides substantial performance gains
- CENTAUR implements a hybrid data path to improve aggregate performance without client modifications
- CENTAUR does not disable carrier sense and co-exists with non-enterprise and uplink traffic

# Future work

- Even more efficient conflict graph generation
- What if we were allowed client modifications ?

## Questions ?

# Characterizing System Latencies




| | Soekris (266MHz) 100 Mbps | VIA (1.2 GHz) GigE |
|---|---|---|
| Wireless RTT | 2190 (50) | 2195 (21) |
| AP RTT | 2550 (80) | 2449 (42) |
| Wired delay | 285 (22) | 95 (18) |
| Controller RTT | 2800 (200) | 2585 (110) |

Mean (Variance) in microseconds

System delays are high and variable, leading to *inaccuracies* for per-packet scheduling

# Evaluation of Micro-Probing

| Topologies | 20 node | 30 node |
|---|---|---|
| Bandwidth Tests | 16.2 mins | 1hr 11 mins |
| Micro-Probing | ~4 secs | ~11 secs |

Can be computed in stages with each instance taking ~ 2.5ms

# Result 2: Impact of Uplink Traffic

- Vary proportion of downlink/uplink traffic
- 6 different configurations
- 80/20 → 40/60 (downlink/uplink)

- Results:
- Downlink: *1.6x* → *6.8x* gain in throughput
- Uplink: *1x* → *1.18x* gain in throughput

CENTAUR provides persistent gains for different proportions of uplink and downlink traffic load

# Result 3: Impact of Topology

- Three topologies
- *Hidden Heavy* topology → *10 links*
- *Exposed Heavy* topology → *6 links*
- *Mixed* Topology → *12 links*

- Results: Up to 50% gain in overall system throughput
- Up to 6x gain for HT; Up to 1.7x gain for ET

Improvements from using CENTAUR can be seen across many different network topologies

# Result 2: Impact of Uplink Traffic

| Downlink load (Mbps) | Uplink load (Mbps) | Downlink Throughput 10% | Downlink Throughput 50% | Downlink Throughput 90% | Uplink Throughput median |
|---|---|---|---|---|---|
| 6 | 1.2 | 6.78× | 1.48× | 1.78× | 1.15× |
| 6 | 2.4 | 3.17× | 1.37× | 1.75× | 1.04× |
| 6 | 6 | 2.24× | 1.21× | 1.53× | 1.01× |
| 2.4 | 1.2 | 1.05× | 1× | 1× | 1× |
| 2.4 | 2.4 | 1.32× | 1.11× | 1.27× | 1.06× |
| 2.4 | 6 | 1.68× | 1.21× | 1.49× | 1.18× |

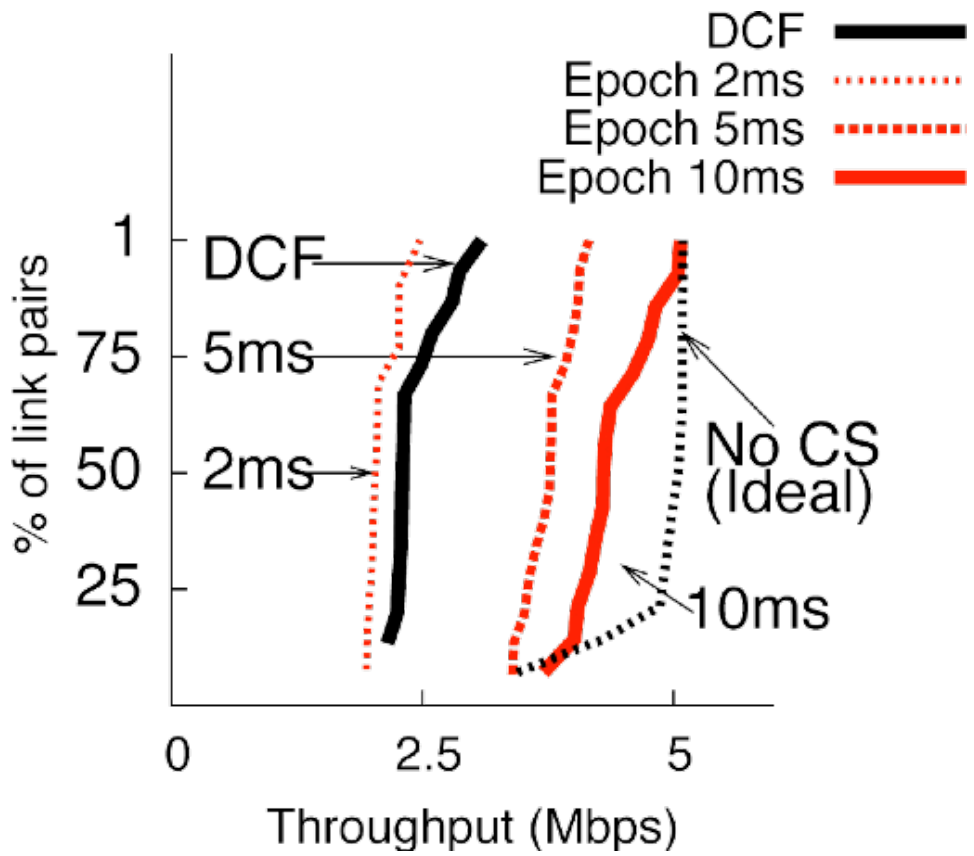
CENTAUR provides persistent gains for different proportions of uplink and downlink traffic load

# Result 3: Impact of Topology

| | Hidden-heavy (Testbed 2) | Exposed-heavy (Testbed 1) | Mixed (Testbed 1) |
|---|---|---|---|
| % HT | 14% | 0% | 6.7% |
| % ET | 0% | 22% | 10.2% |
| Overall Gains (HT/ET gains) | 34.7% (HT: 6×) | 47.2% (ET: 1.7×) | 44% (HT: 3.2×, ET: 1.4×) |

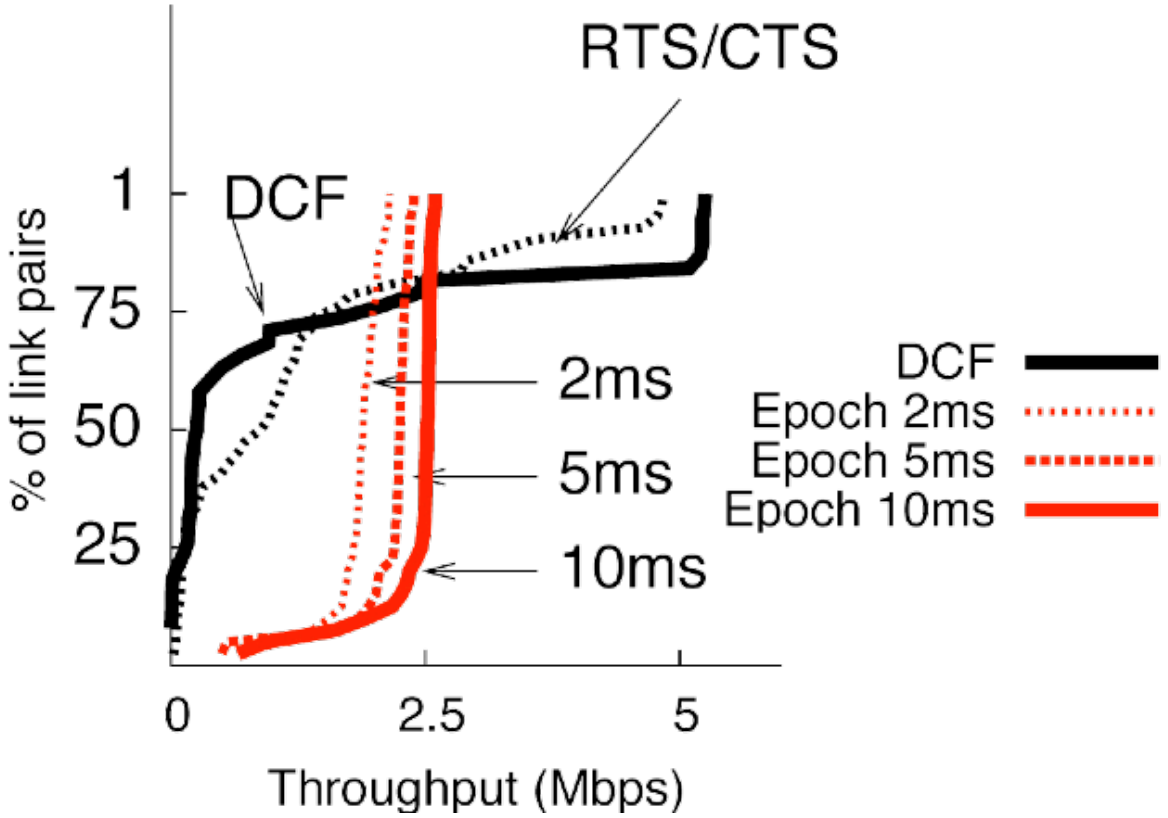
Improvements from using CENTAUR can be seen across many different network topologies

# CENTAUR Micro-Benchmarks (1)




**Exposed Terminals**

# CENTAUR Micro-Benchmarks (II)




**Hidden Terminals**

# Result 1: UDP/TCP Performance

Delay (ms)

CENTAUR yields up to 60% higher total throughput and 50% lower per-packet delay compared to DCF

Client Index (1-12)