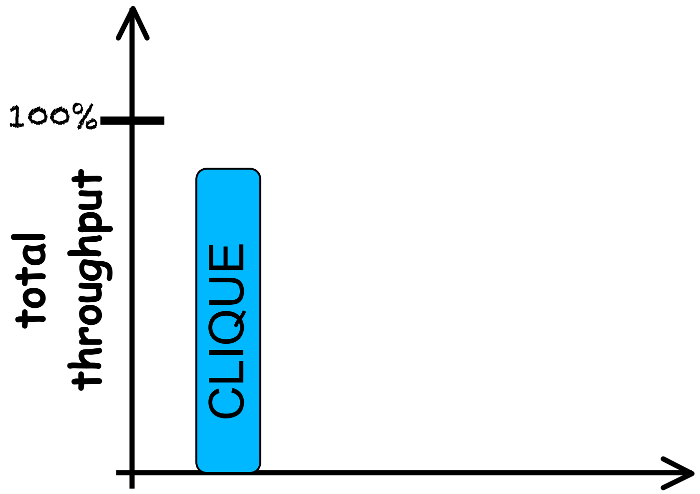
802.11*ec*: Collision Avoidance without Control Messages

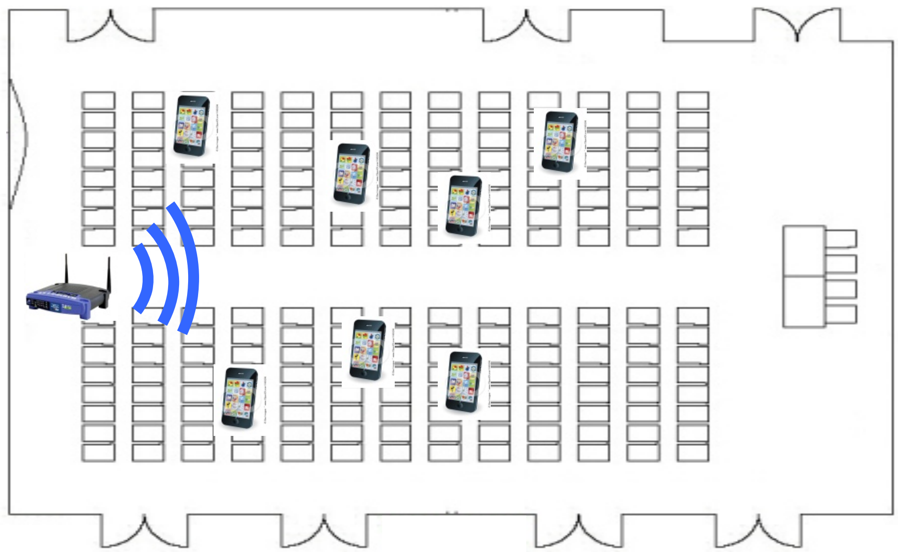
802.11*ec* *stands for 802.11 with Encoded Control and does not represent an IEEE standard*

Eugenio Magistretti, Omer Gurewitz,
and Edward Knightly

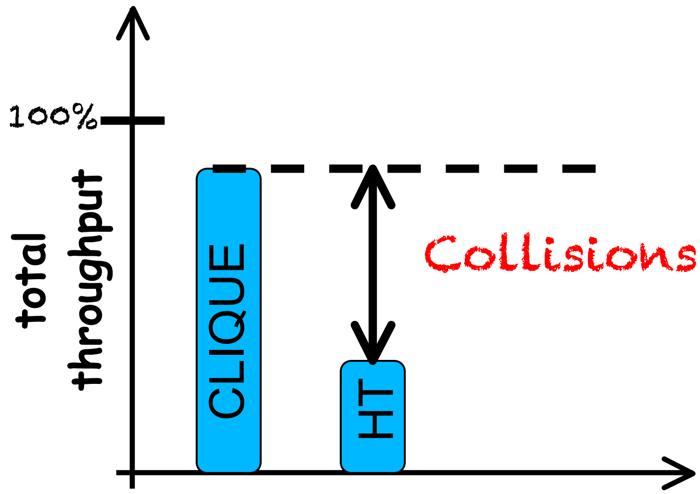
Rice University
Ben Gurion University

Ideal Wi-Fi

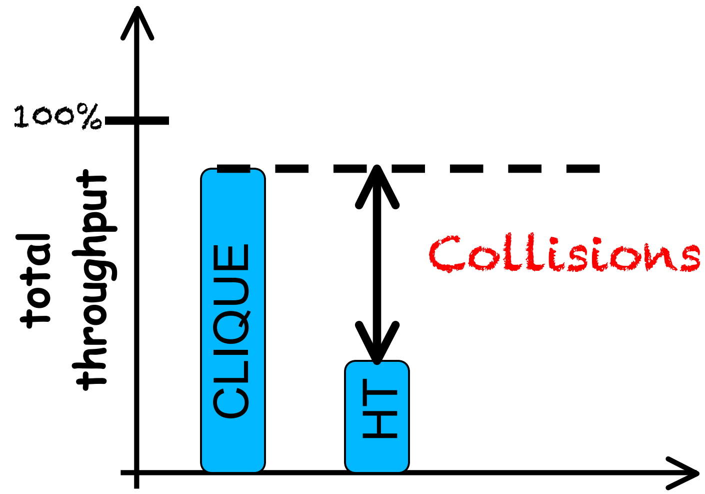




Low Power & Obstructions



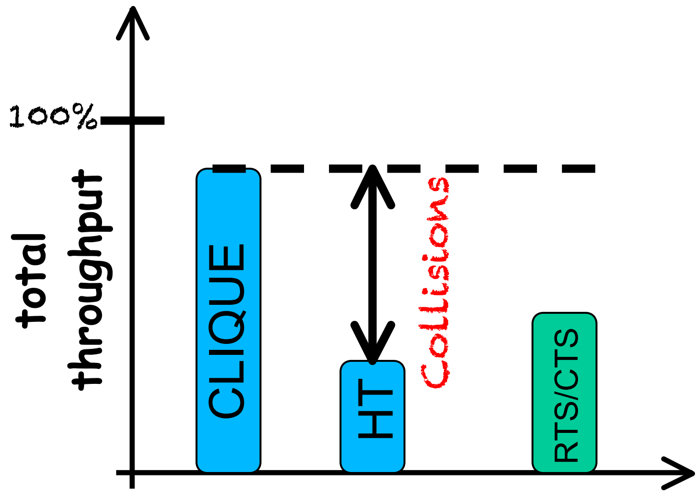
Low Power & Obstructions



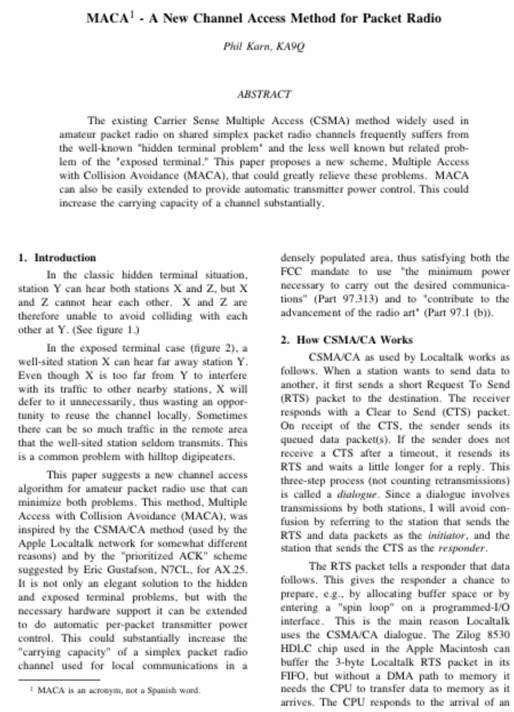
Ex.

Two hidden terminals
70% throughput loss

Low Power & Obstructions



RTS/CTS



MACA[1] - A New Channel Access Method for Packet Radio

Phil Karn, KA9Q

ABSTRACT

The existing Carrier Sense Multiple Access (CSMA) method widely used in amateur packet radio on shared simplex packet radio channels frequently suffers from the well-known "hidden terminal problem" and the less well known but related problem of the "exposed terminal." This paper proposes a new scheme, Multiple Access with Collision Avoidance (MACA), that could greatly relieve these problems. MACA can also be easily extended to provide automatic transmitter power control. This could increase the carrying capacity of a channel substantially.

1. Introduction

In the classic hidden terminal situation, station Y can hear both stations X and Z, but X and Z cannot hear each other. X and Z are therefore unable to avoid colliding with each other at Y. (See figure 1.)

In the exposed terminal case (figure 2), a well-sited station X can hear far away station Y. Even though X is too far from Y to interfere with its traffic to other nearby stations, X will defer to it unnecessarily, thus wasting an opportunity to reuse the channel locally. Sometimes there can be so much traffic in the remote area that the well-sited station seldom transmits. This is a common problem with hilltop digipeaters.

This paper suggests a new channel access algorithm for amateur packet radio use that can minimize both problems. This method, Multiple Access with Collision Avoidance (MACA), was inspired by the CSMA/CA method (used by the Apple Localtalk network for somewhat different reasons) and by the "prioritized ACK" scheme suggested by Eric Gustafson, N7CL, for AX.25. It is not only an elegant solution to the hidden and exposed terminal problems, but with the necessary hardware support it can be extended to do automatic per-packet transmitter power control. This could substantially increase the "carrying capacity" of a simplex packet radio channel used for local communications in a densely populated area, thus satisfying both the FCC mandate to use "the minimum power necessary to carry out the desired communications" (Part 97.313) and to "contribute to the advancement of the radio art" (Part 97.1 (b)).

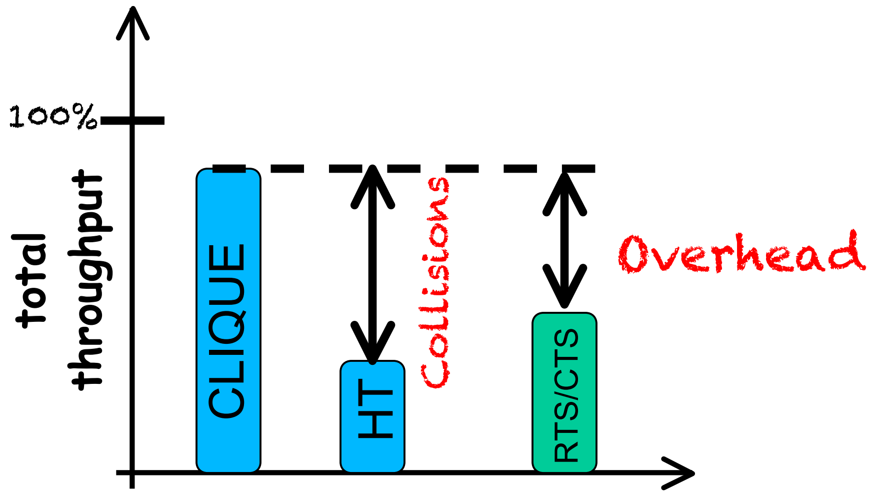
2. How CSMA/CA Works

CSMA/CA as used by Localtalk works as follows. When a station wants to send data to another, it first sends a short Request To Send (RTS) packet to the destination. The receiver responds with a Clear to Send (CTS) packet. On receipt of the CTS, the sender sends its queued data packet(s). If the sender does not receive a CTS after a timeout, it resends its RTS and waits a little longer for a reply. This three-step process (not counting retransmissions) is called a *dialogue*. Since a dialogue involves transmissions by both stations, I will avoid confusion by referring to the station that sends the RTS and data packets as the *initiator*, and the station that sends the CTS as the *responder*.

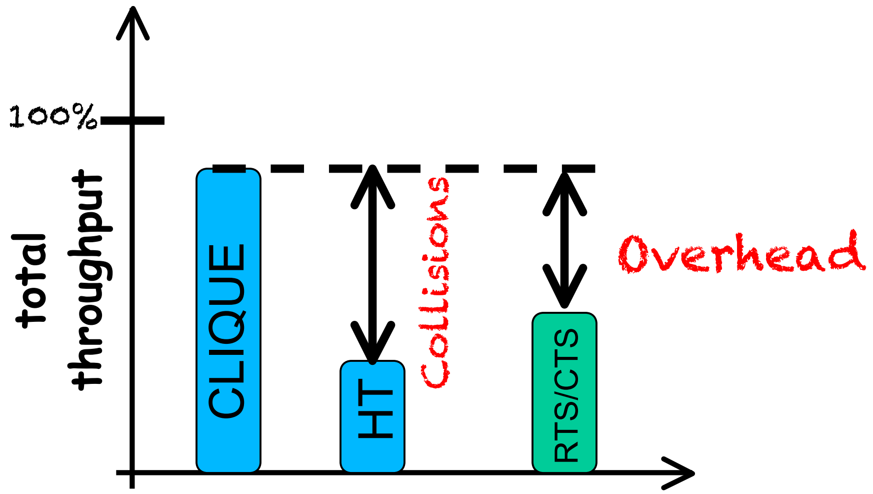
The RTS packet tells a responder that data follows. This gives the responder a chance to prepare, e.g., by allocating buffer space or by entering a "spin loop" on a programmed-I/O interface. This is the main reason Localtalk uses the CSMA/CA dialogue. The Zilog 8530 HDLC chip used in the Apple Macintosh can buffer the 3-byte Localtalk RTS packet in its FIFO, but without a DMA path to memory it needs the CPU to transfer data to memory as it arrives. The CPU responds to the arrival of an

[1] MACA is an acronym, not a Spanish word.

Collision Penalty → Overhead Penalty



Collision Penalty → Overhead Penalty



Ex.

RTS/CTS/ACK overhead
~ max size packet
@ 54 Mbps

Objective

- Fundamentally **re-think the way control information is conveyed** in order to guarantee **low overhead** and **robustness**

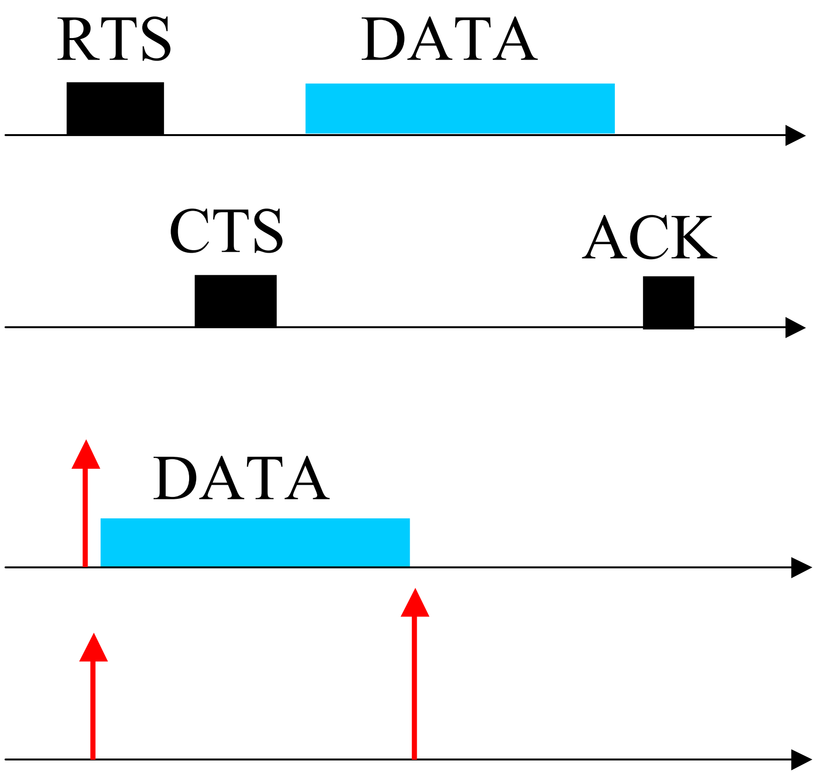
Control Messages → CSS

Messages

- Preamble
- Header
- Base Rate Data

✗ Long Duration

Control Messages → CSS



Messages

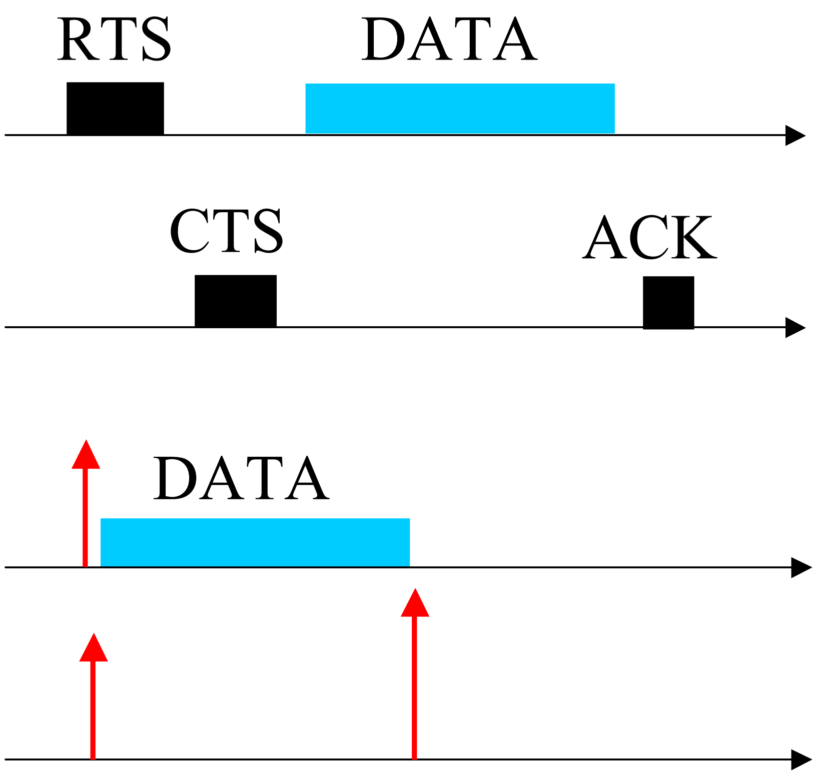
- Preamble
- Header
- Base Rate Data

✗ Long Duration

Correlatable Symbol Sequences

- Correlatable Symbol Sequences (CSS) are **pre-defined** pseudo-noise (PN) bit sequences detected via **cross-correlation**

Control Messages → CSS



Messages

- Preamble
- Header
- Base Rate Data

✗ Long Duration

Correlatable Symbol Sequences

- ~~Preamble~~
- ~~Header~~

✔ Shorter

✔ More Robust

Outline

- Correlatable Symbol Sequences (CSS)
- Control Information via CSS
- **11***ec* Protocol Illustration
- Experimental Results

Why Correlatable Symbol Sequences?

CSS's key idea:

Signal detection is more robust than decoding

Why Correlatable Symbol Sequences?

CSS's key idea:
Signal detection is more robust than decoding

- Sender and receiver agree upon a set of pre-defined CSS
- The receiver can detect a specific CSS u(k) by **correlating** it with the incoming samples y(k)

Why Correlatable Symbol Sequences?

CSS's key idea:
Signal detection is more robust than decoding

- Sender and receiver agree upon a set of pre-defined CSS
- The receiver can detect a specific CSS u(k) by correlating it with the incoming samples y(k)

$$C(\Delta) = \sum_{0}^{L-1} u^*(k) y(k+\Delta)$$

CSS MATCHING

$$C(\Delta) \approx L$$

CSS NON MATCHING

$$C(\Delta) \approx 0$$

Why Correlatable Symbol Sequences?

CSS's key idea:
Signal detection is more robust than decoding

- Sender and receiver agree upon a set of pre-defined CSS
  - CSS length L determines the max size of the CSS set
- The receiver can detect a specific CSS u(k) by correlating it with the incoming samples y(k)

$$C(\Delta) = \sum_{0}^{L-1} u^*(k)\, y(k+\Delta)$$

CSS MATCHING

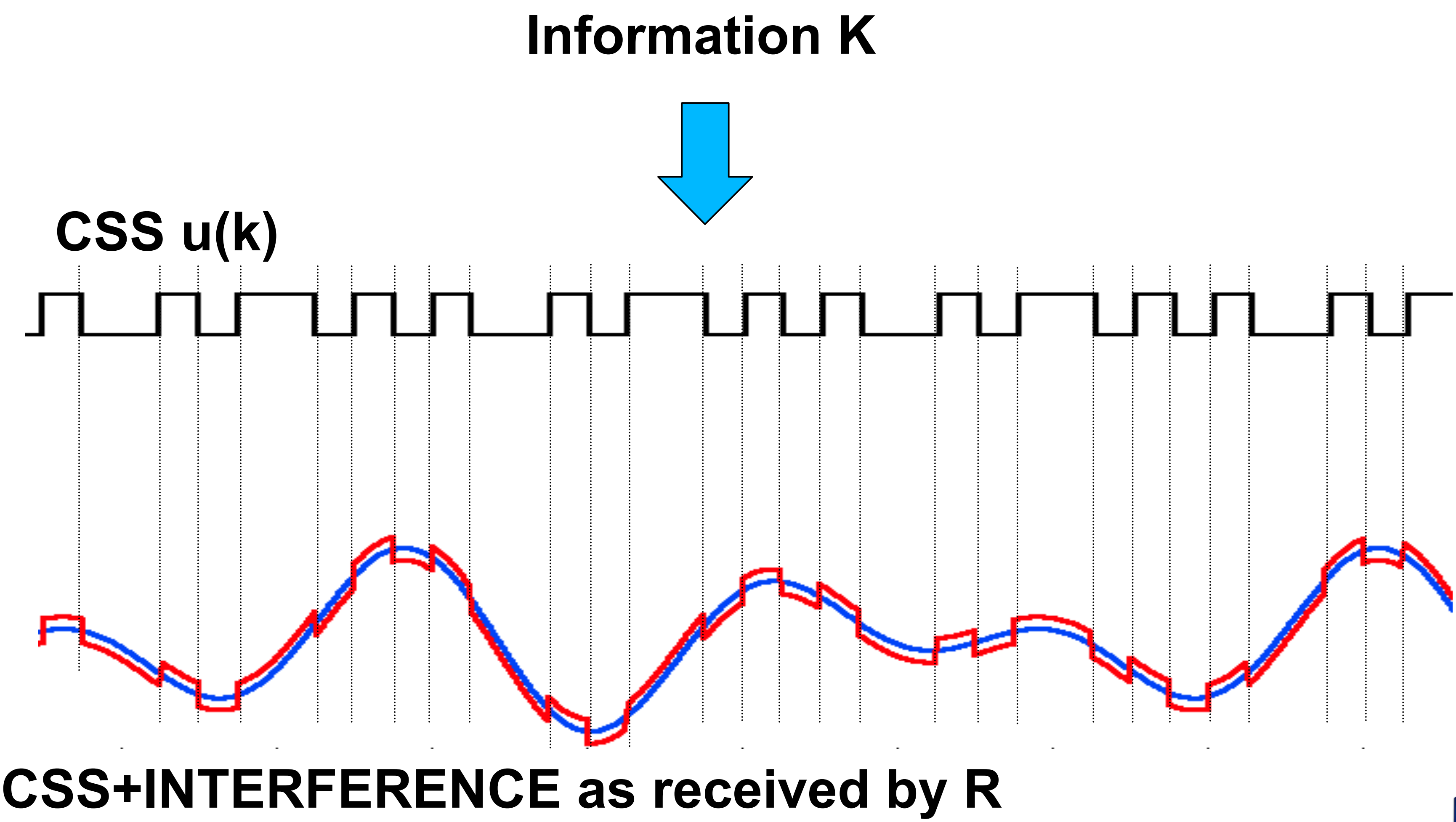
$$C(\Delta) \approx L$$

CSS NON MATCHING

$$C(\Delta) \approx 0$$

CSS Intuition

- Node S wants to convey control information (“Information K”) to node R via a pre-defined CSS u(k)



CSS Intuition

- Node S wants to convey control information ("Information K") to node R via a pre-defined CSS u(k)

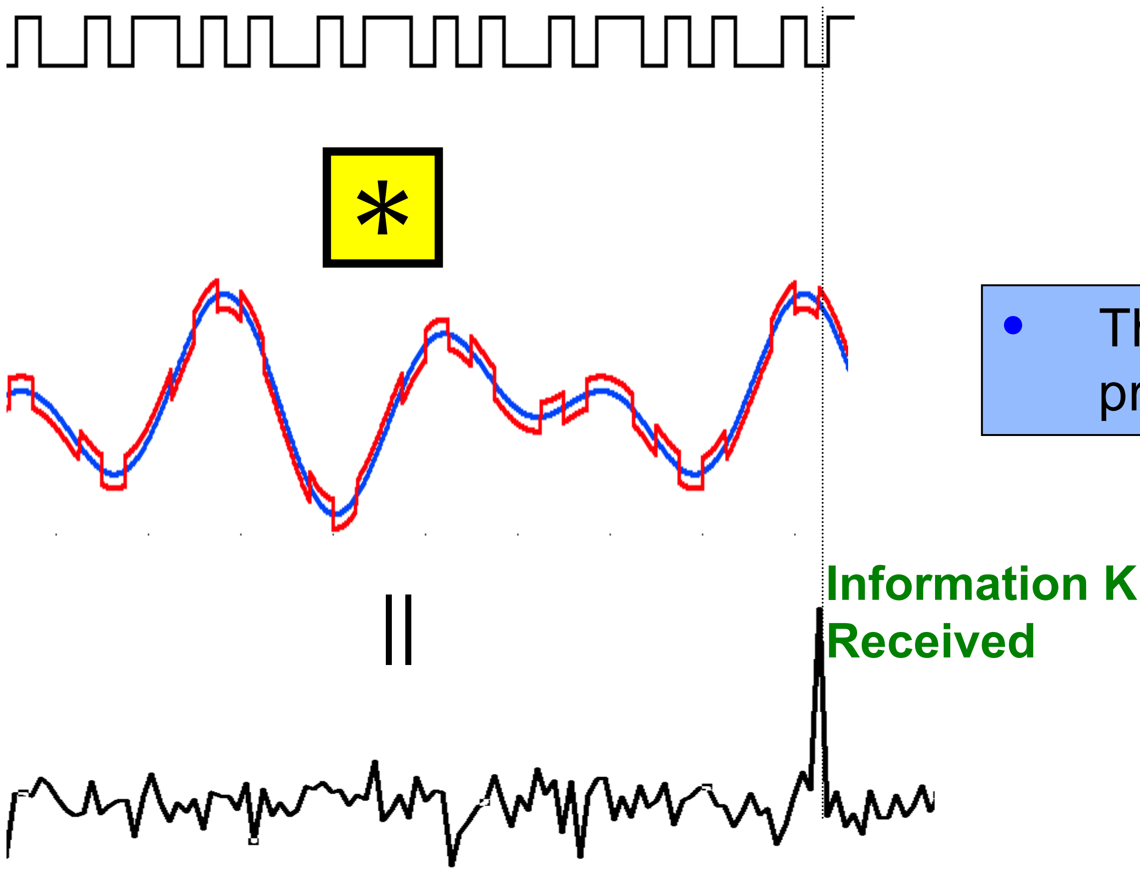
CSS u(k)

CSS+INTERFERENCE as received by R

CSS Intuition

- Node S wants to convey control information (“Information K”) to node R via a pre-defined CSS u(k)

MATCHING at R



- The correlator spikes when the pre-defined CSS is received

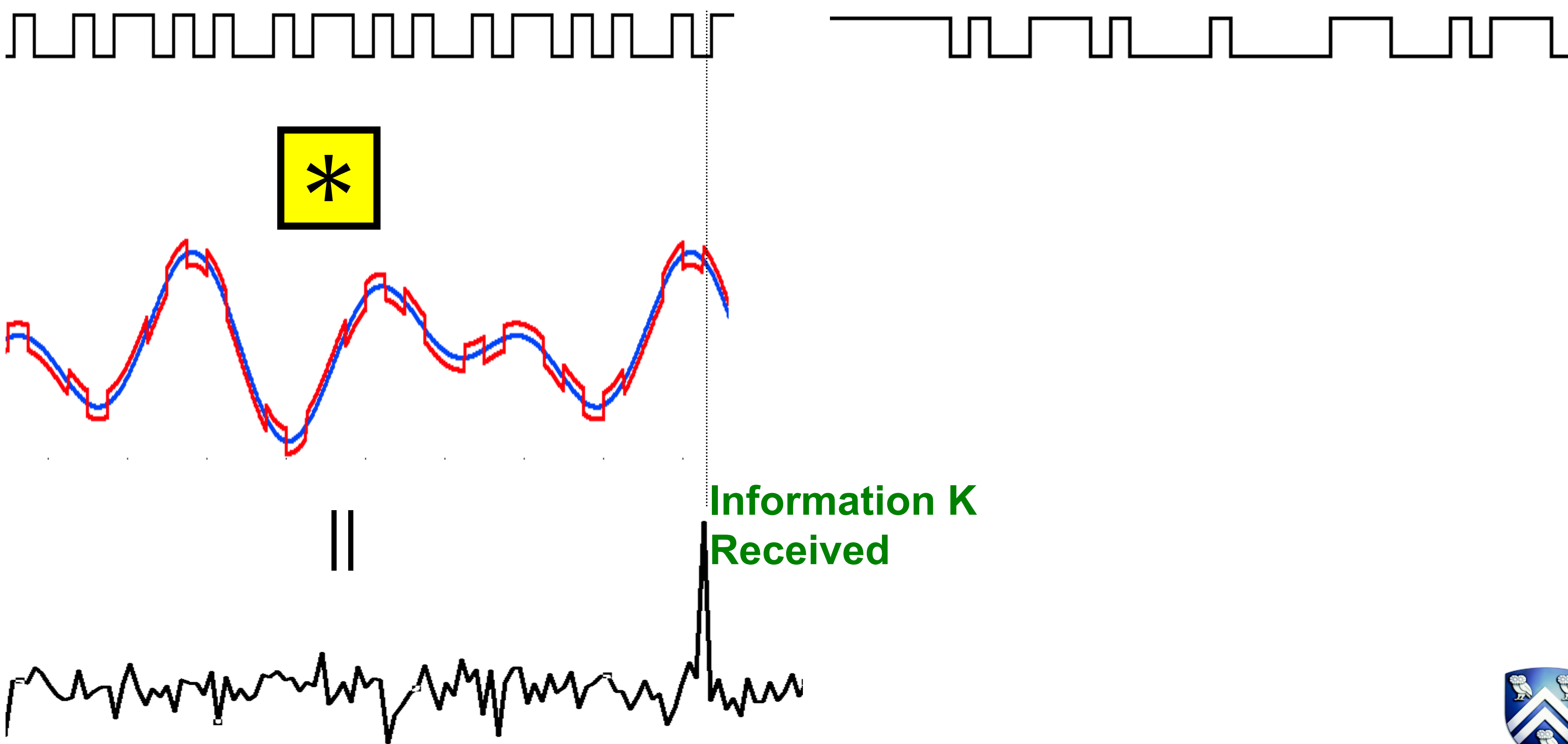
CSS Intuition

- Node S wants to convey control information ("Information K") to node R via a pre-defined CSS u(k)

MATCHING at R

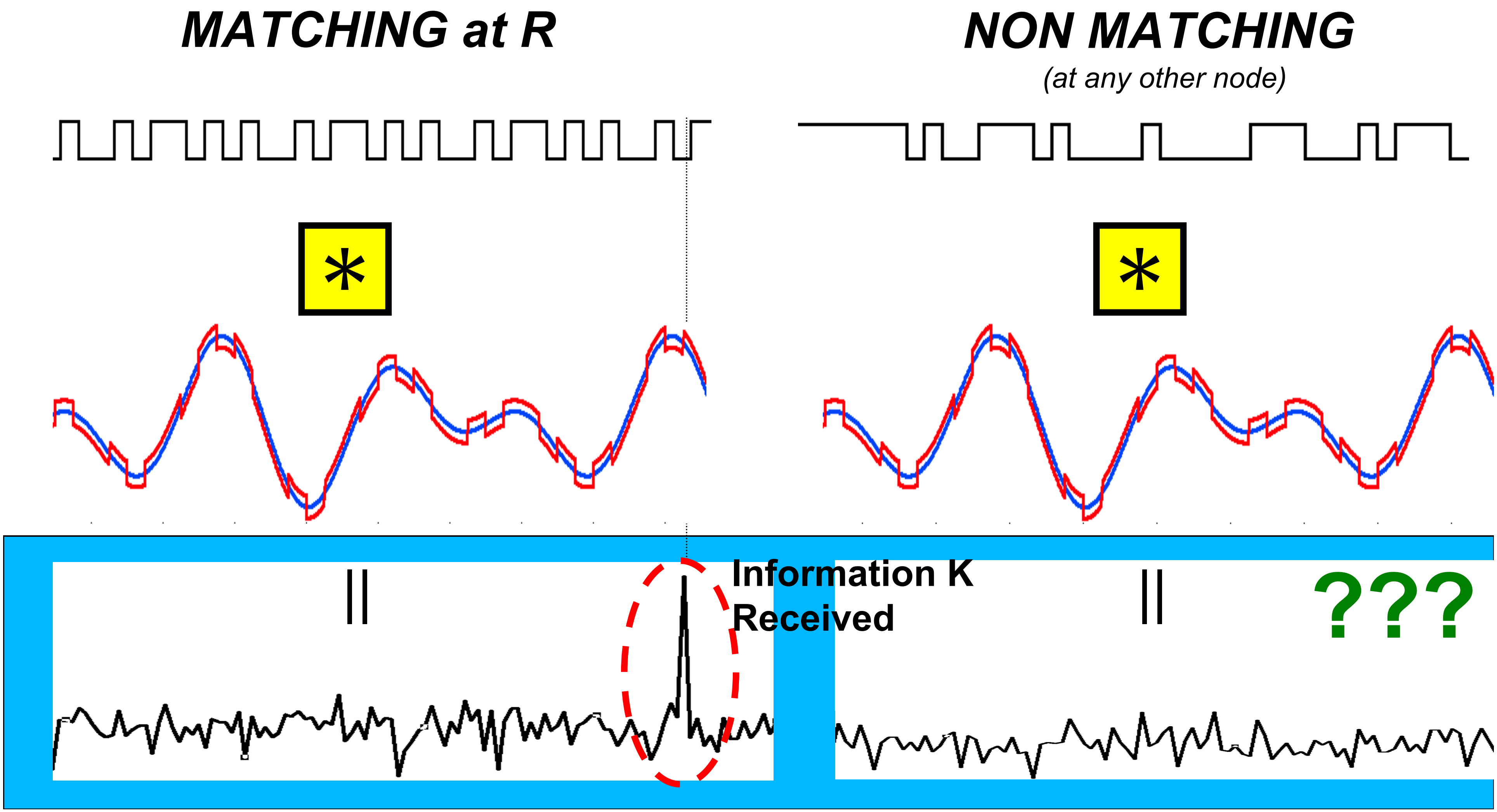
NON MATCHING

(at any other node)



CSS Intuition

- Node S wants to convey control information (“Information K”) to node R via a pre-defined CSS u(k)



CSS Summary

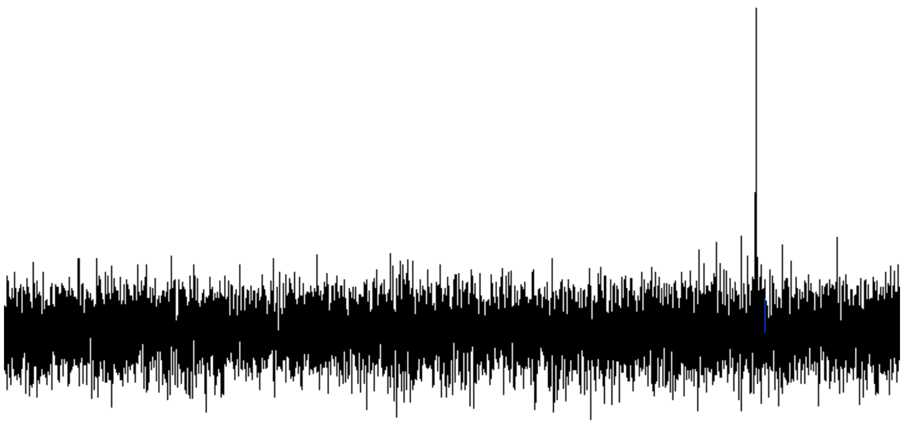
- Correlatable Symbol Sequences (CSS) are pre-defined PN bit sequences detected via **cross-correlation**
- **CSS do not need any preamble/header**
- **<u>Advantages</u>**

Low Overhead

- No preamble
- No encoding

High Robustness

- To low SINR
- To collisions

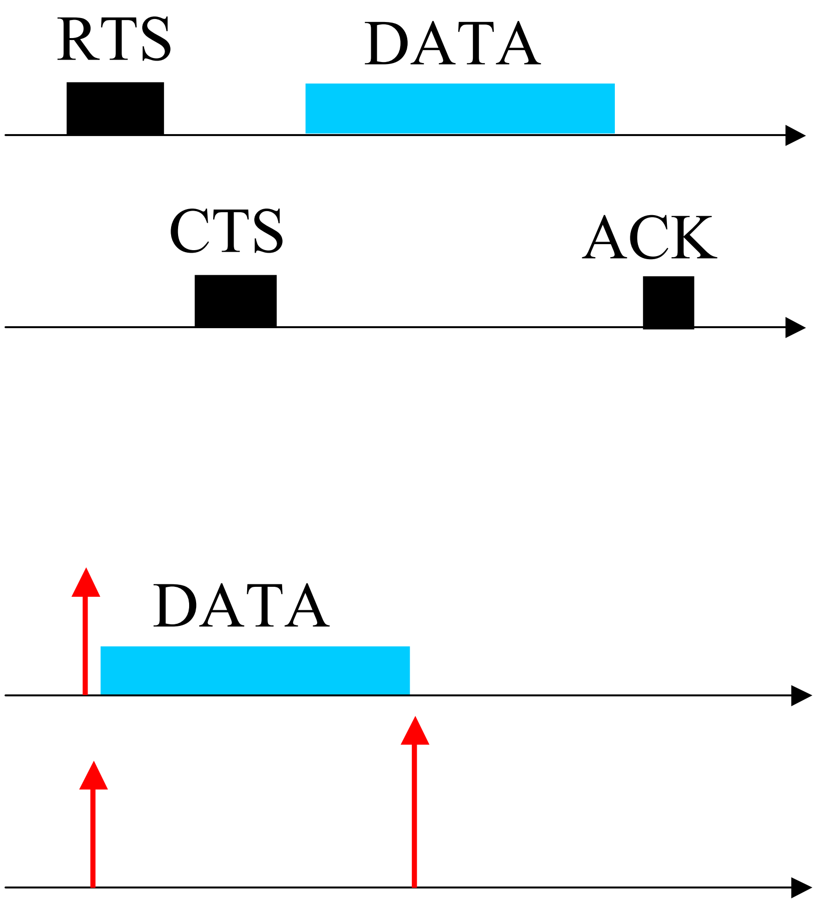


Length L limits number of pre-defined CSS

Outline

- Correlatable Symbol Sequences (CSS)
- Control Information via CSS
- **11***ec* Protocol Illustration
- Experimental Results

Control Messages → CSS



Messages

✗ Long Duration

Correlatable Symbol Sequences

✔ Shorter

✔ More Robust

✗ Limited Number

Control Information via CSS

- 802.11 control message structure

Preamble	Fixed Control Field	Type	Addresses	Duration	

Control Information via CSS - *Dictionary*

- 802.11 control message structure

Preamble	Fixed Control Field	Type	Addresses	Duration

- Define **a small dictionary of CSS** to represent the message information content

Control Information via CSS - *Dictionary*

- 802.11 control message structure

Preamble	Fixed Control Field	**Type**	Addresses	Duration

- Define a **small dictionary of CSS** to represent the message information content
  - Type → Limited Set

Map to different CSS
(e.g., RTS CTS ACK)

Control Information via CSS - *Dictionary*

- 802.11 control message structure

Preamble	Fixed Control Field	**Type**	**Addresses**	Duration

- Define a **small dictionary of CSS** to represent the message information content

  - Type → Limited Set — **Map to different CSS** ***(e.g., RTS CTS ACK)***
  - Address → Large Set but Locally Limited — **One CSS per node**

Control Information via CSS - *Dictionary*

- 802.11 control message structure

Preamble	Fixed Control Field	Type	Addresses	Duration

- Define a **small dictionary of CSS** to represent the message information content

  - Type → Limited Set — **Map to different CSS** ***(e.g., RTS CTS ACK)***
  - Address → Large Set but Locally Limited — **One CSS per node**
  - Duration → Large Set — **New CSS types + Timing Code** ***(e.g., channel free)***

Control Information via CSS - *Scope*

- 802.11 control message structure

Preamble	Fixed Control Field	Type	Addresses	Duration	

- **CSS Information Scope Control**

Control Information via CSS - *Scope*

- 802.11 control message structure

Preamble	Fixed Control Field	**Type**	Addresses	Duration

- **CSS Information Scope Control**

  – Public

To be received by all nodes
(e.g., Channel Reservation/Release)

All nodes detect the CSS
(i.e., possess the correlator)

Control Information via CSS - *Scope*

- 802.11 control message structure

Preamble	Fixed Control Field	**Type**	Addresses	Duration

- **CSS Information Scope Control**

  – Public

To be received by all nodes
(e.g., Channel Reservation/Release)
→
All nodes detect the CSS
(i.e., possess the correlator)

  – Unique Feature: **Private**

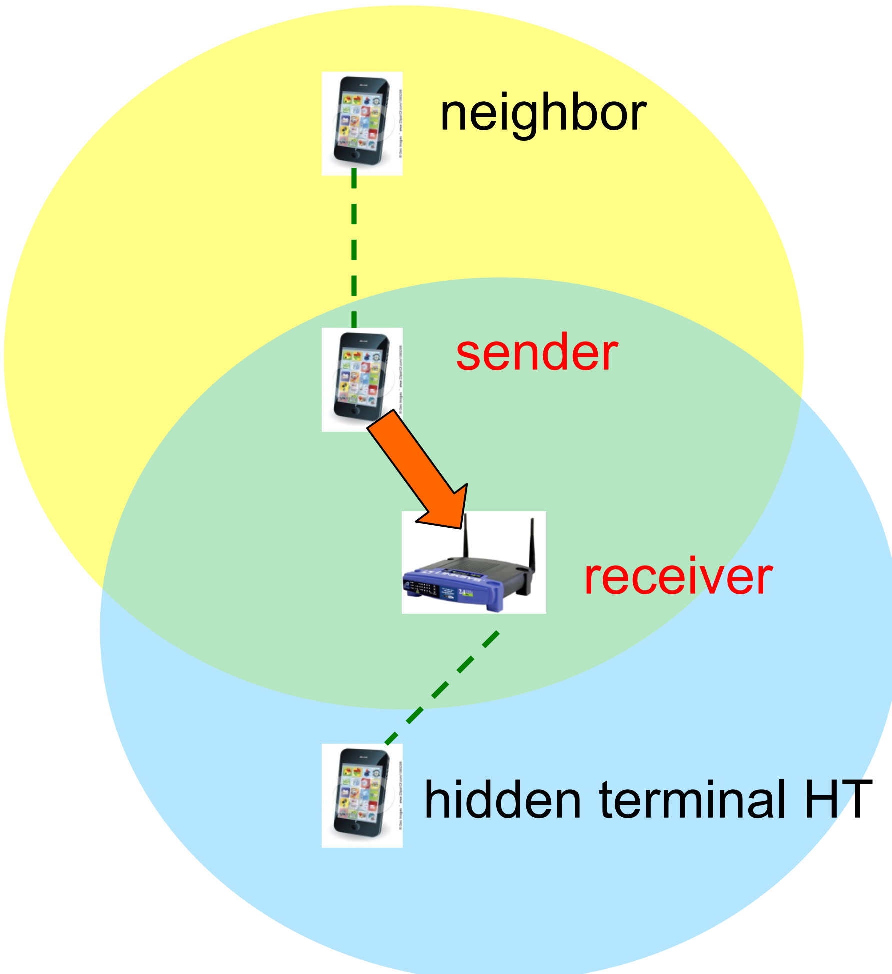
To be received by the other endpoint (e.g., ACK, source/destination address)
→
Selected nodes detect the CSS

Outline

- Correlatable Symbol Sequences (CSS)
- Control Information via CSS
- **11***ec* Protocol Illustration
- Experimental Results

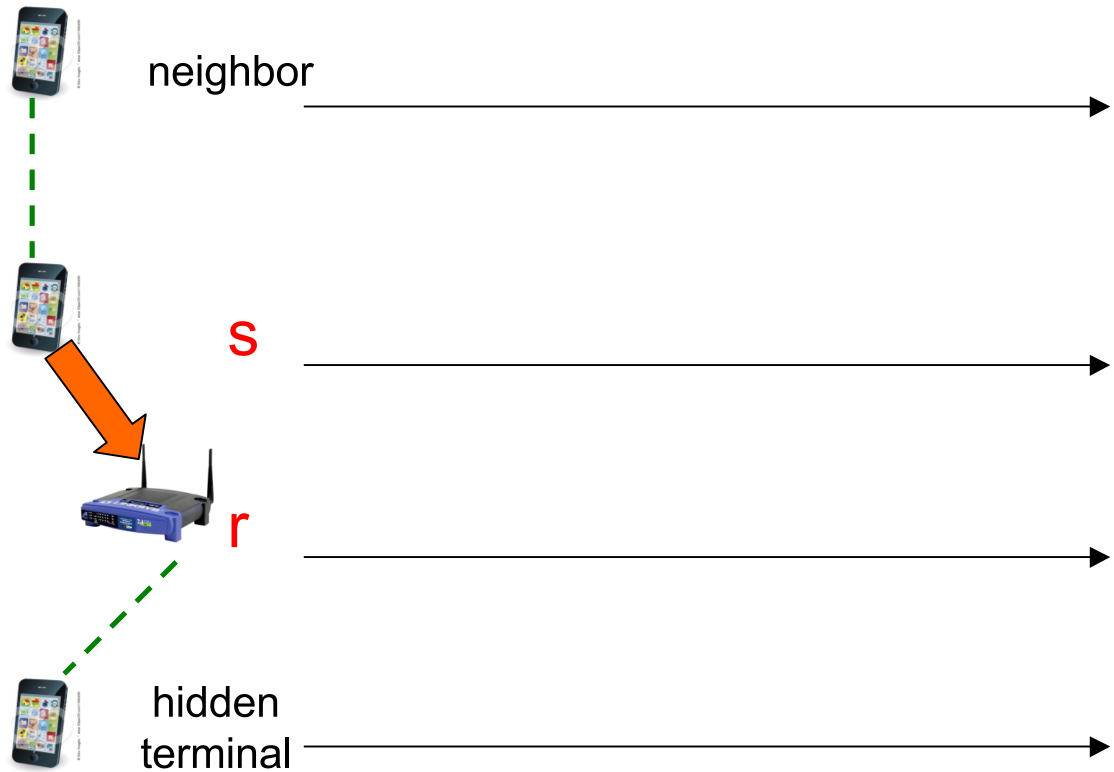
11*ec* Protocol Illustration

- 11*ec* follows the fundamental concepts of 802.11



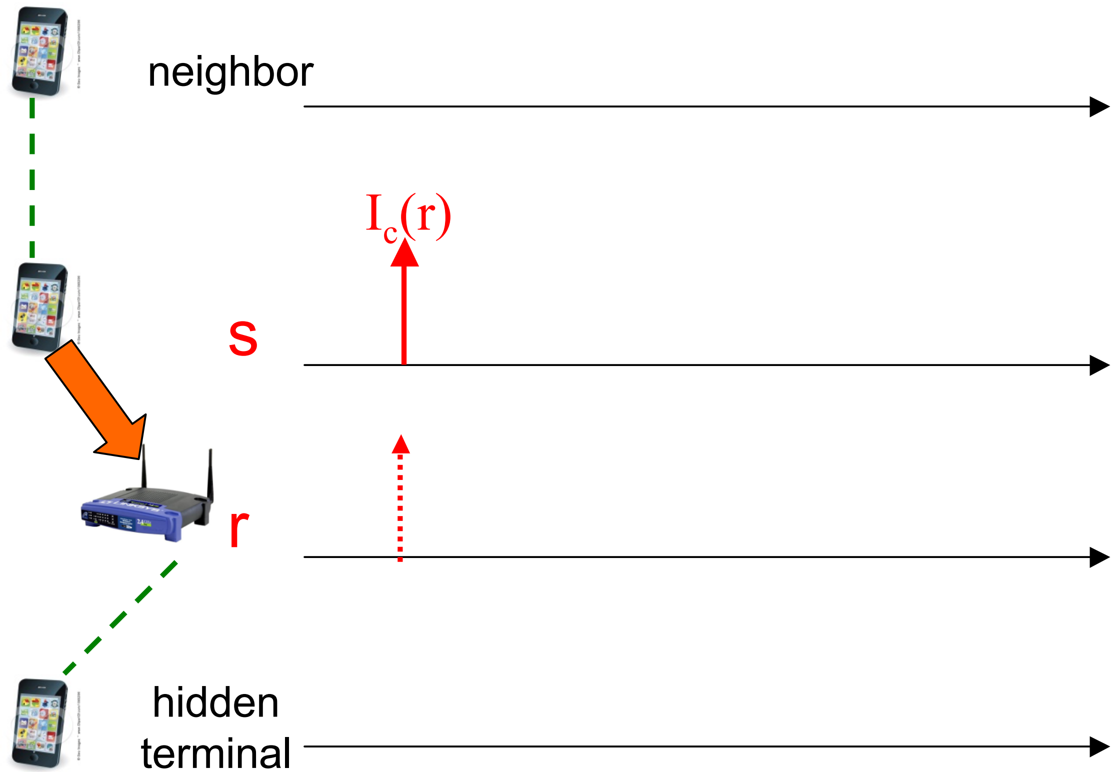
11*ec* Protocol Illustration

- 11*ec* follows the fundamental concepts of 802.11



11*ec* Protocol Illustration

- 11*ec* follows the fundamental concepts of 802.11

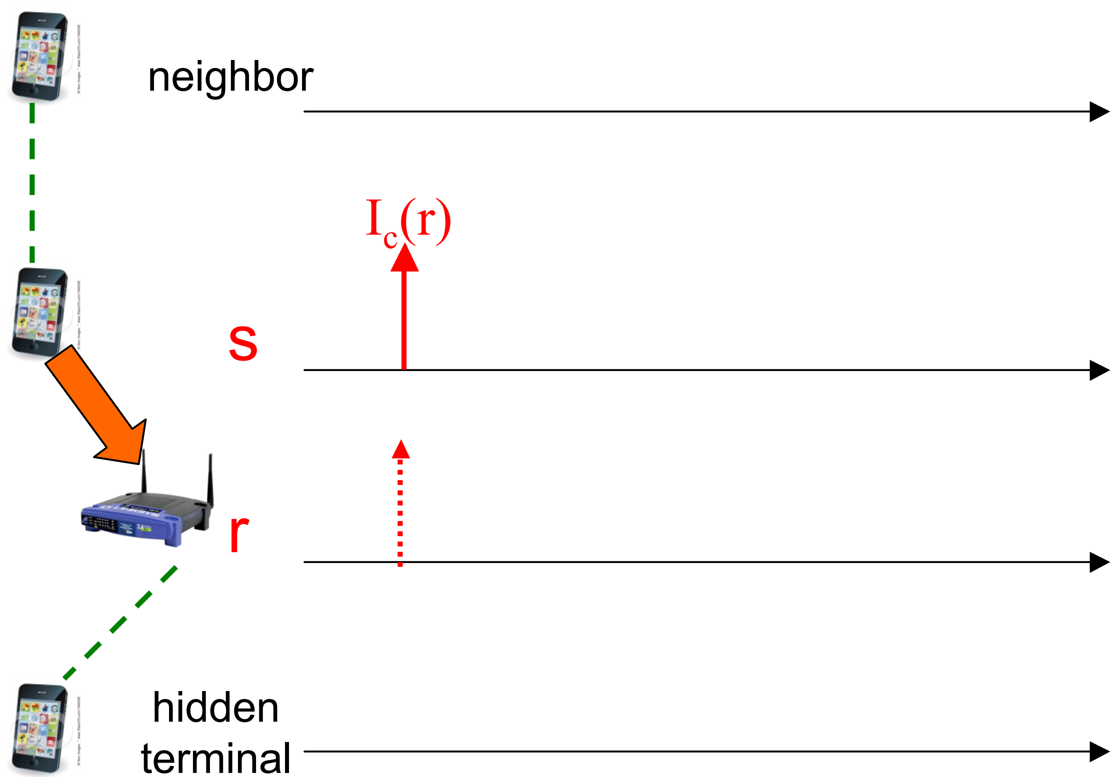


$I_c(r)$ Initiation

Private CSS conveying the receiver address

11*ec* Protocol Illustration

- 11*ec* follows the fundamental concepts of 802.11



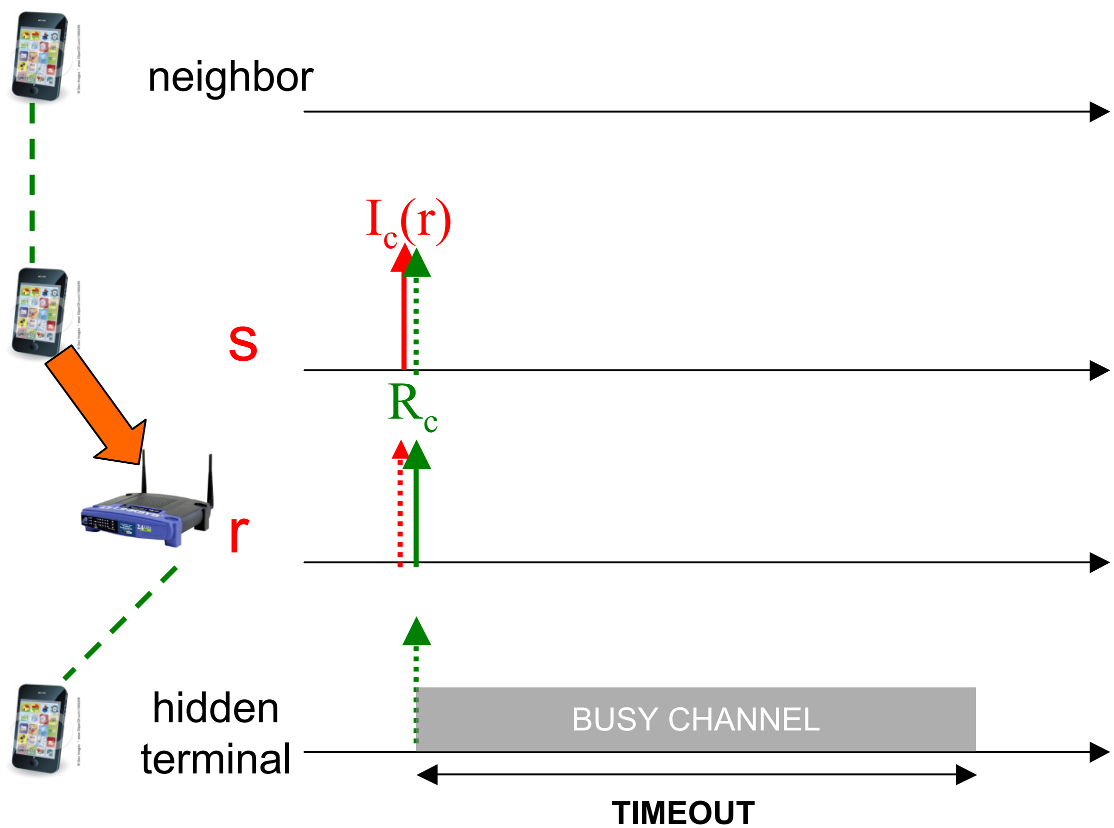
$I_c(r)$ Initiation

Private CSS conveying the receiver address

Every Receiver is associated with an $I_c(r)$

11*ec* Protocol Illustration

- 11*ec* follows the fundamental concepts of 802.11



$I_c(r)$ **Initiation**

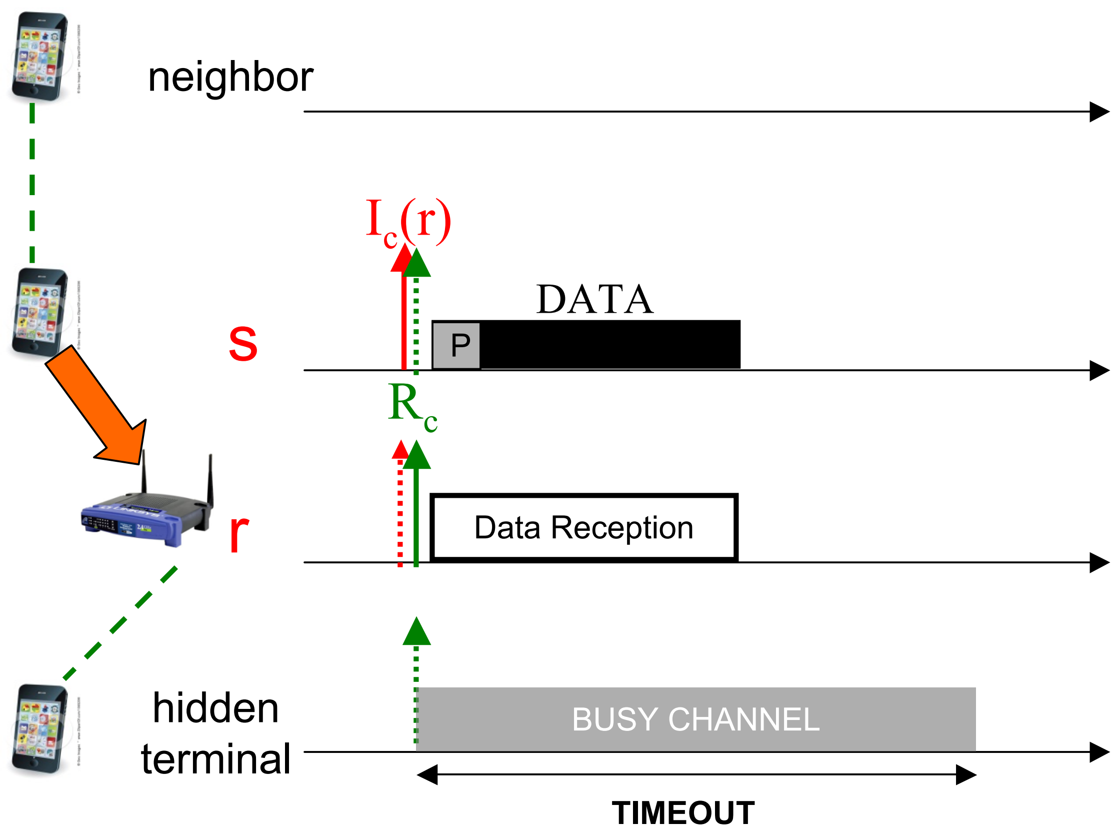
R_c **Channel Reservation**

Public CSS

R_c is a unique CSS

11*ec* Protocol Illustration

- 11*ec* follows the fundamental concepts of 802.11



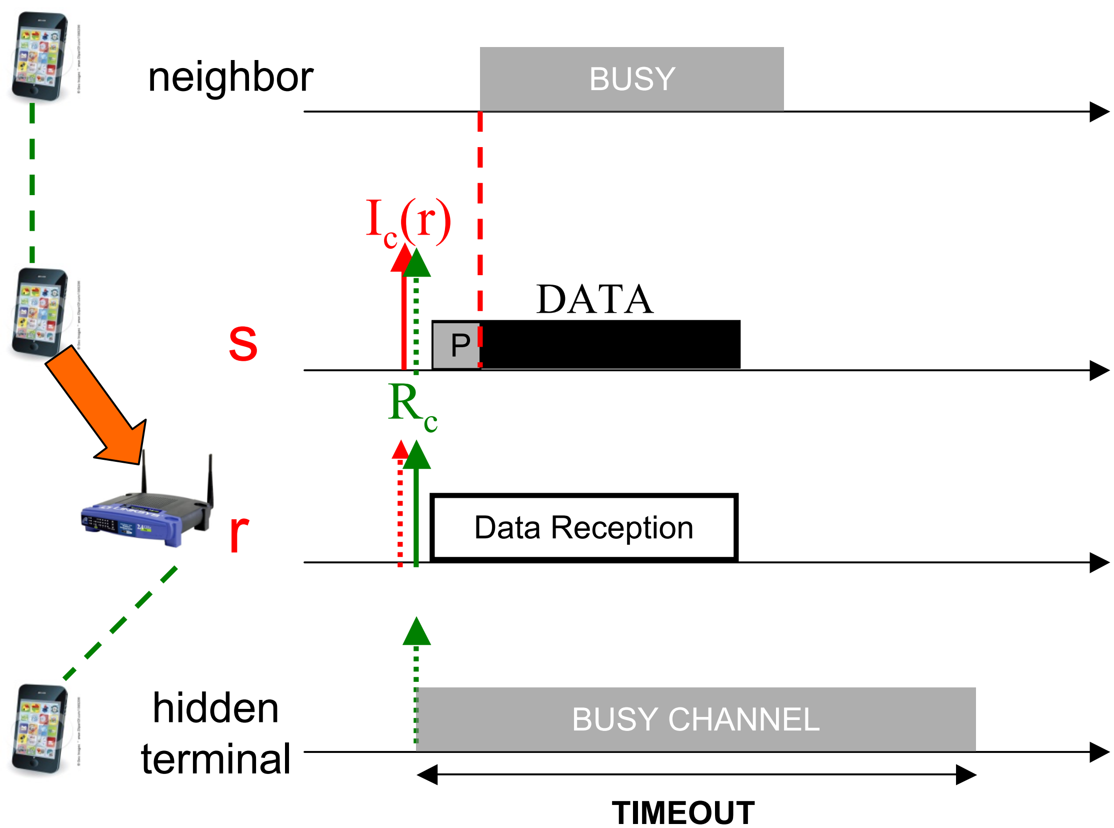
$I_c(r)$ Initiation

R_c Channel Reservation

DATA contains 802.11 preamble P

11*ec* Protocol Illustration

- 11*ec* follows the fundamental concepts of 802.11



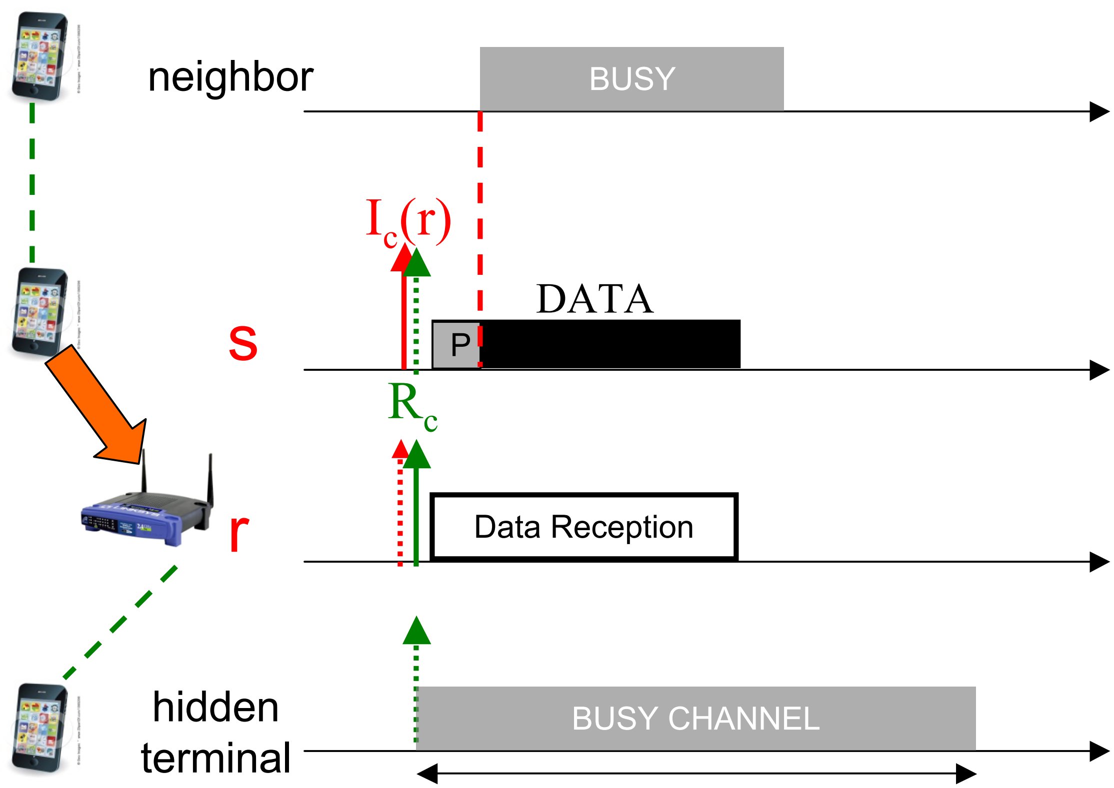
$I_c(r)$ Initiation

R_c Channel Reservation

Neighbor nodes detect P and NAVs

11ec Protocol Illustration

- *11ec* follows the fundamental concepts of 802.11



$I_c(r)$ Initiation

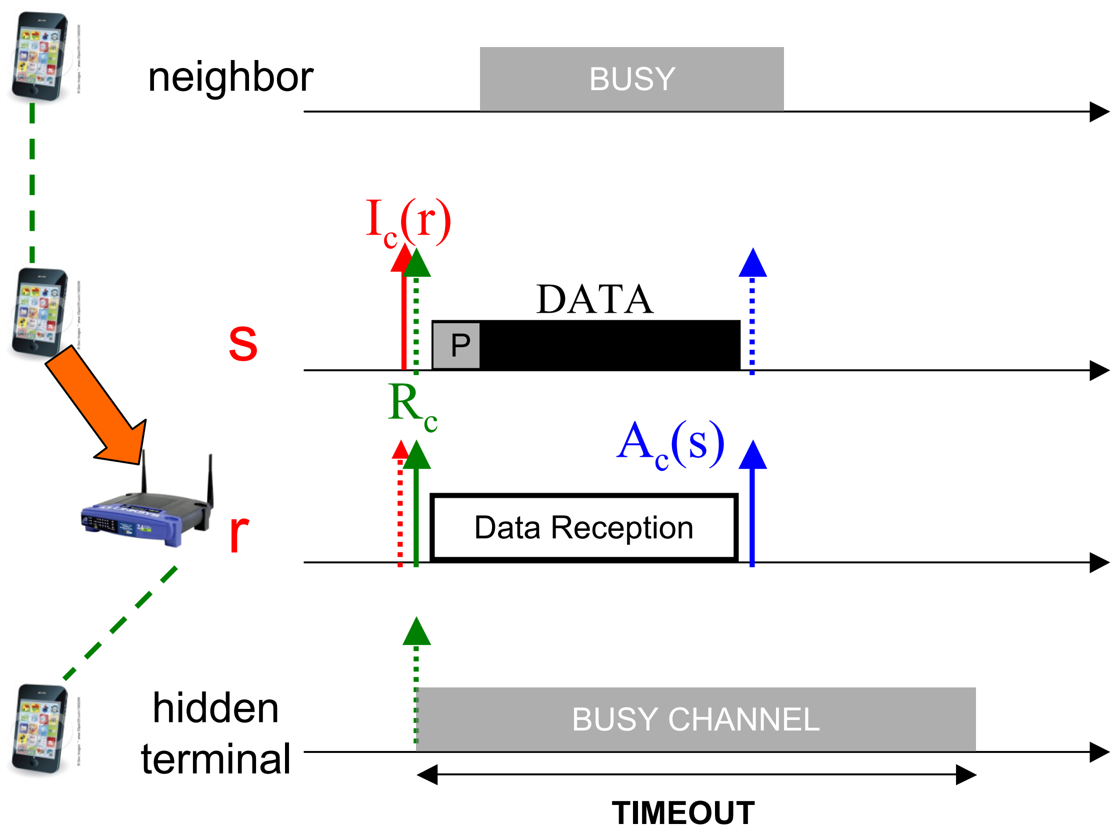
R_c Channel Reservation

Neighbor nodes detect P and NAVs

802.11ec can co-exist with 802.11

11*ec* Protocol Illustration

- 11*ec* follows the fundamental concepts of 802.11



$I_c(r)$ Initiation

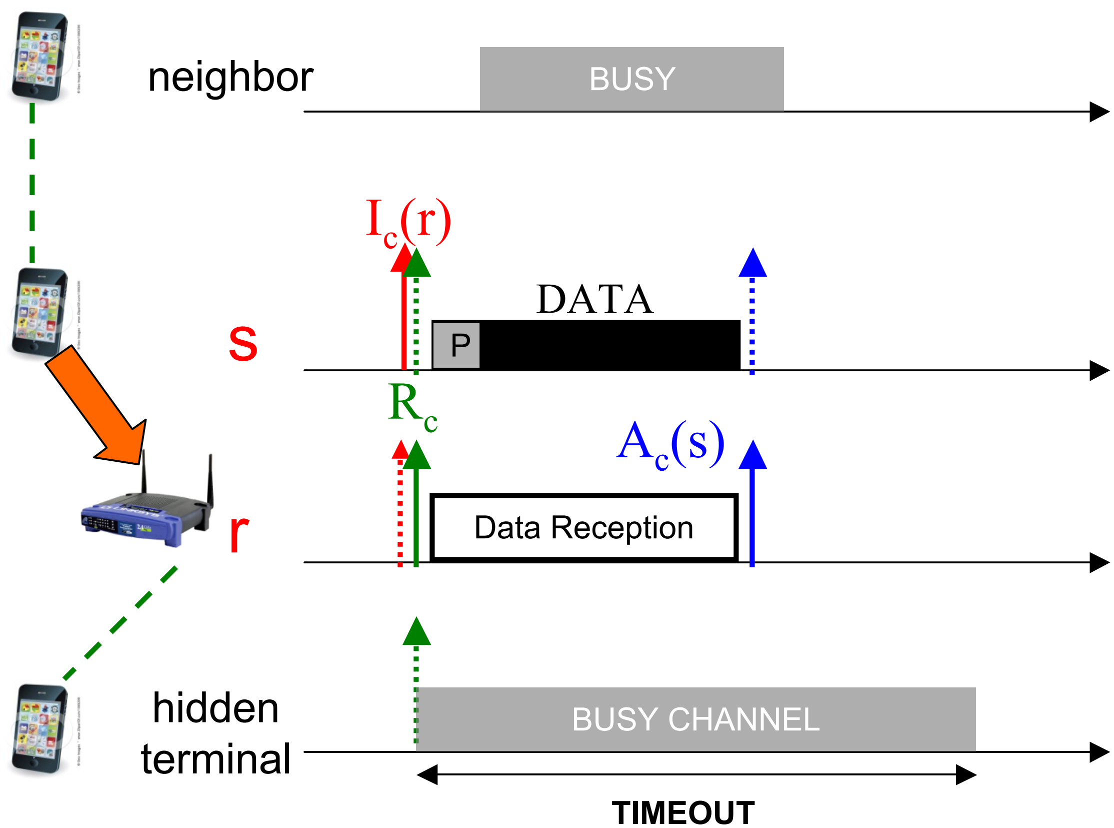
R_c Channel Reservation

$A_c(s)$ **Acknowledgement**

Private CSS conveying the sender address

$11ec$ Protocol Illustration

- $11ec$ follows the fundamental concepts of 802.11



$I_c(r)$ Initiation

R_c Channel Reservation

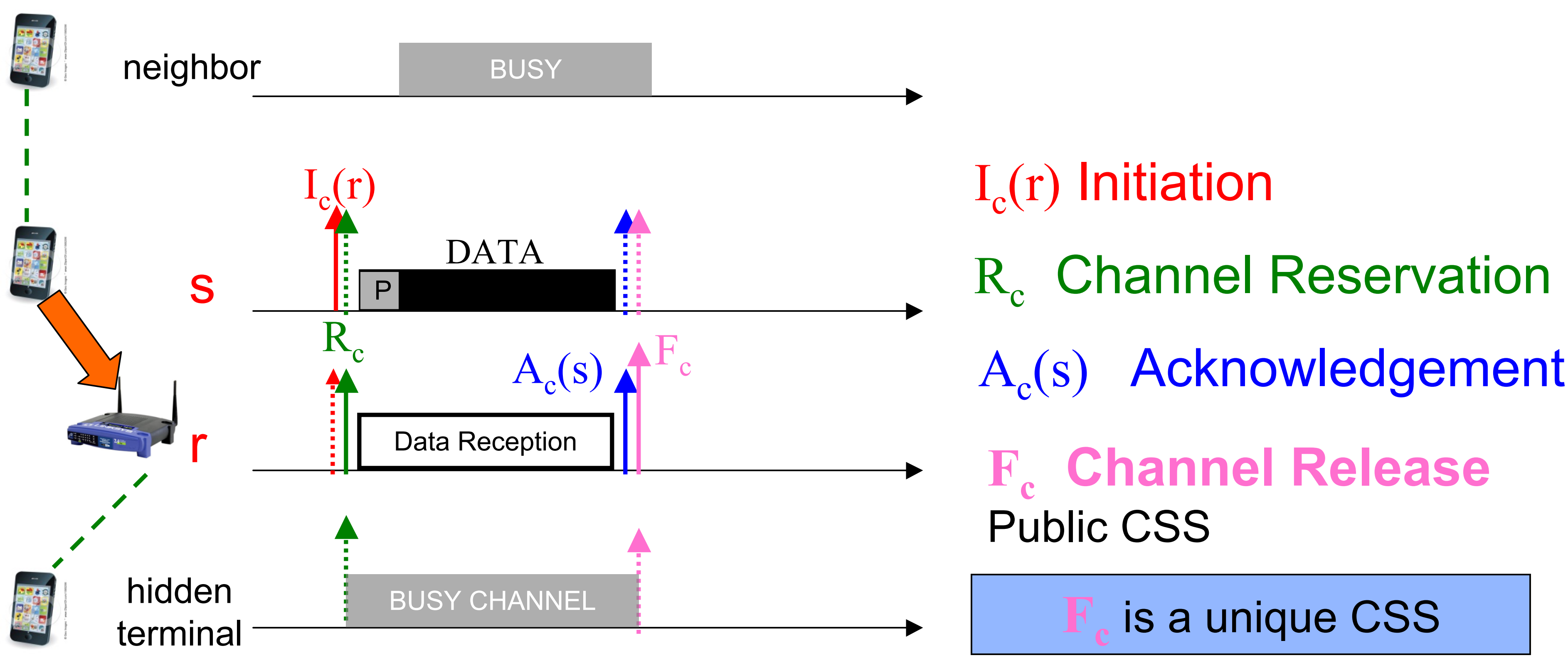
$A_c(s)$ Acknowledgement

Private CSS conveying the sender address

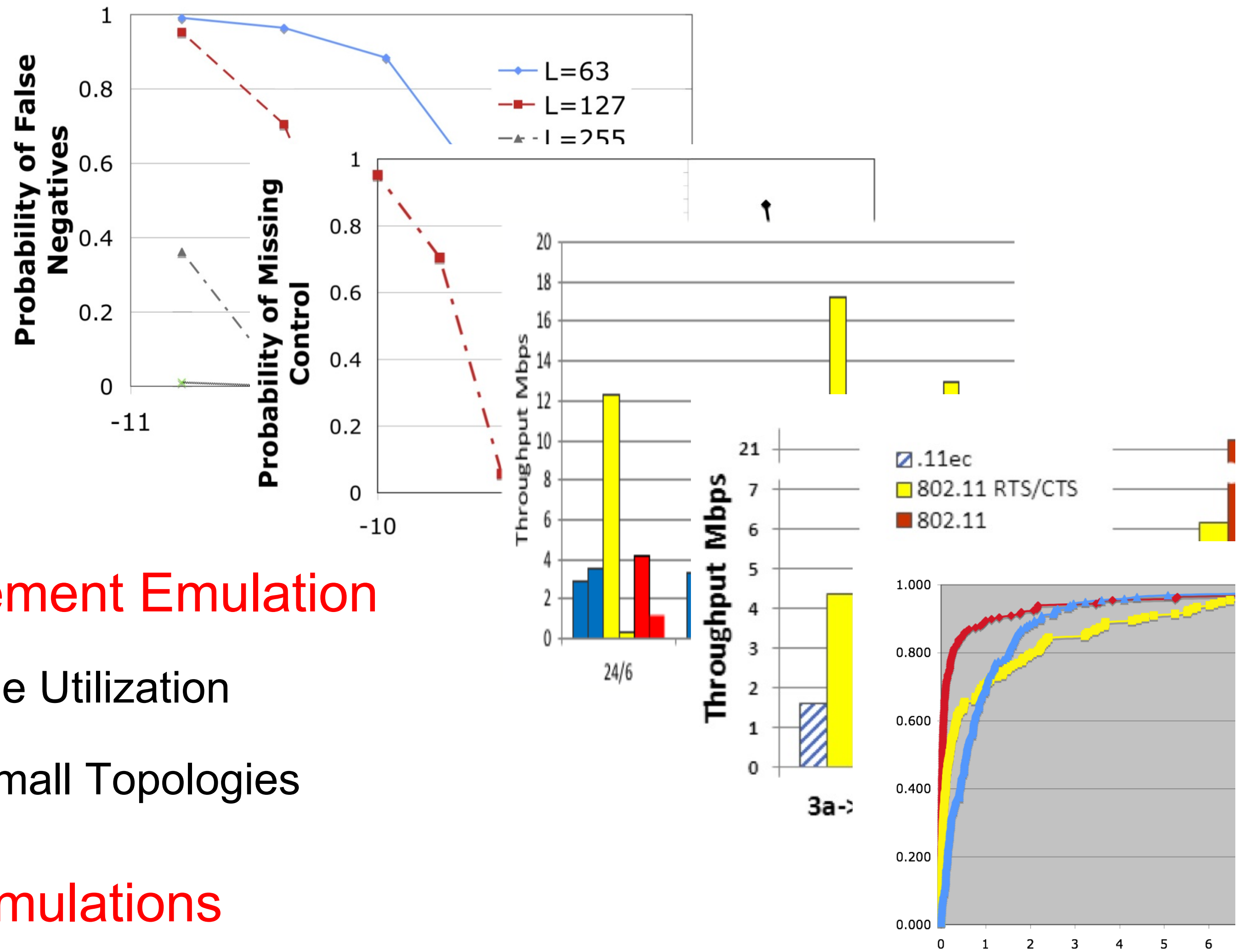
Every Sender is associated with an $A_c(s)$

11*ec* Protocol Illustration

- 11*ec* follows the fundamental concepts of 802.11



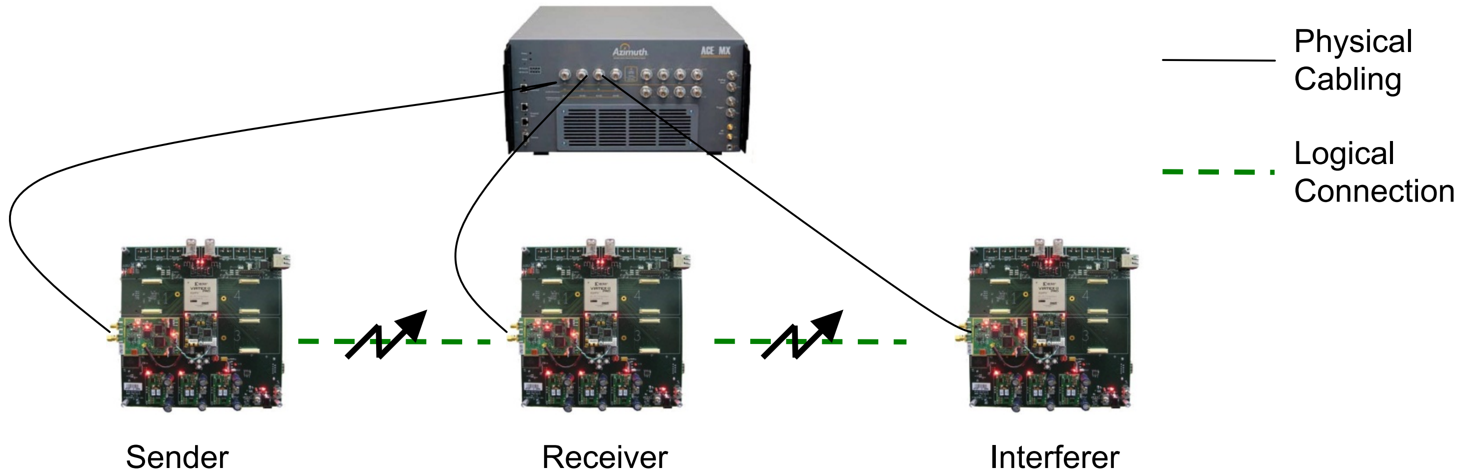
Experimental Results Roadmap



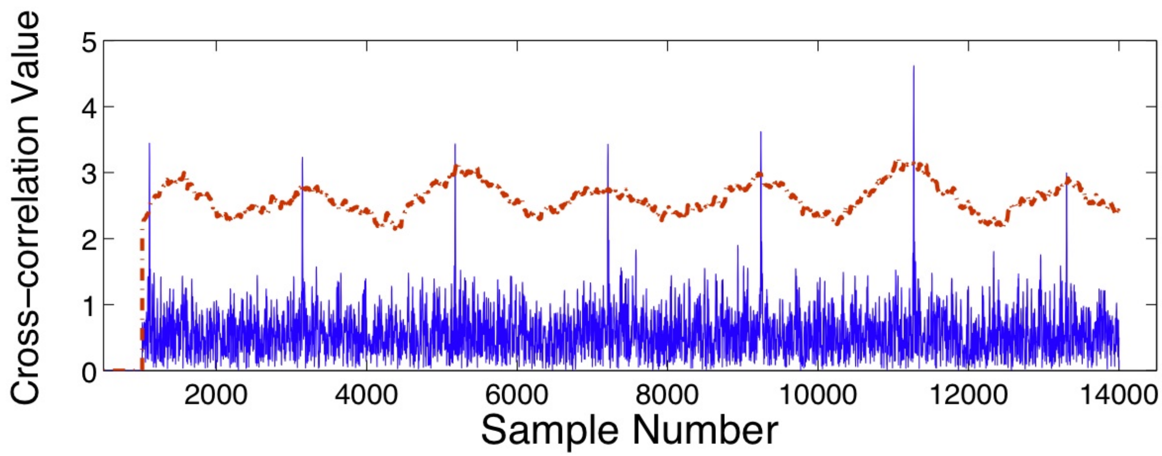
- WARP
  - CSS Robustness
- Channel Measurement Emulation
  - Tput/Fairness/Airtime Utilization
  - Fundamental and Small Topologies
- Large Network Simulations
  - 20-node Random Topology

CSS Implementation

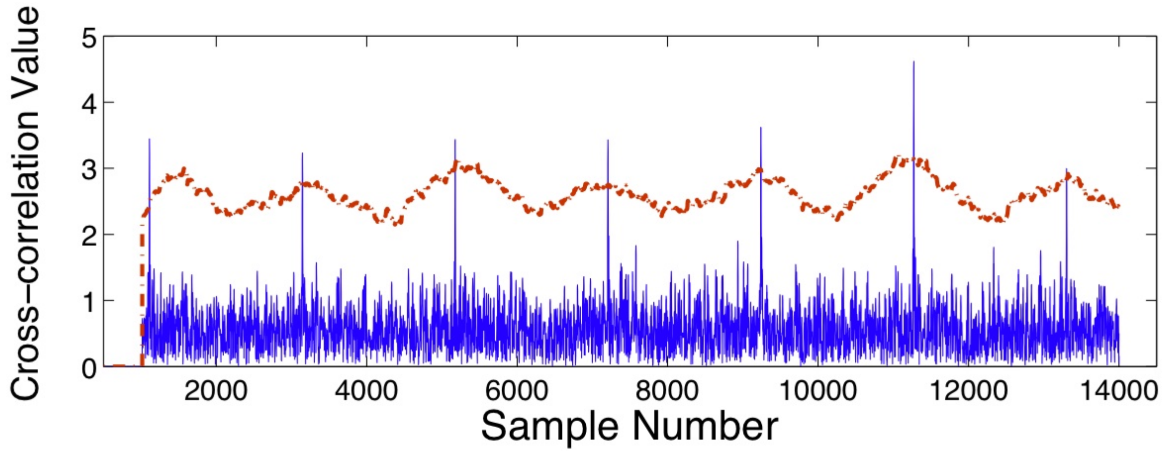
1. **Repeatable** and **controllable** experiments



CSS Performance Metric



CSS Performance Metric

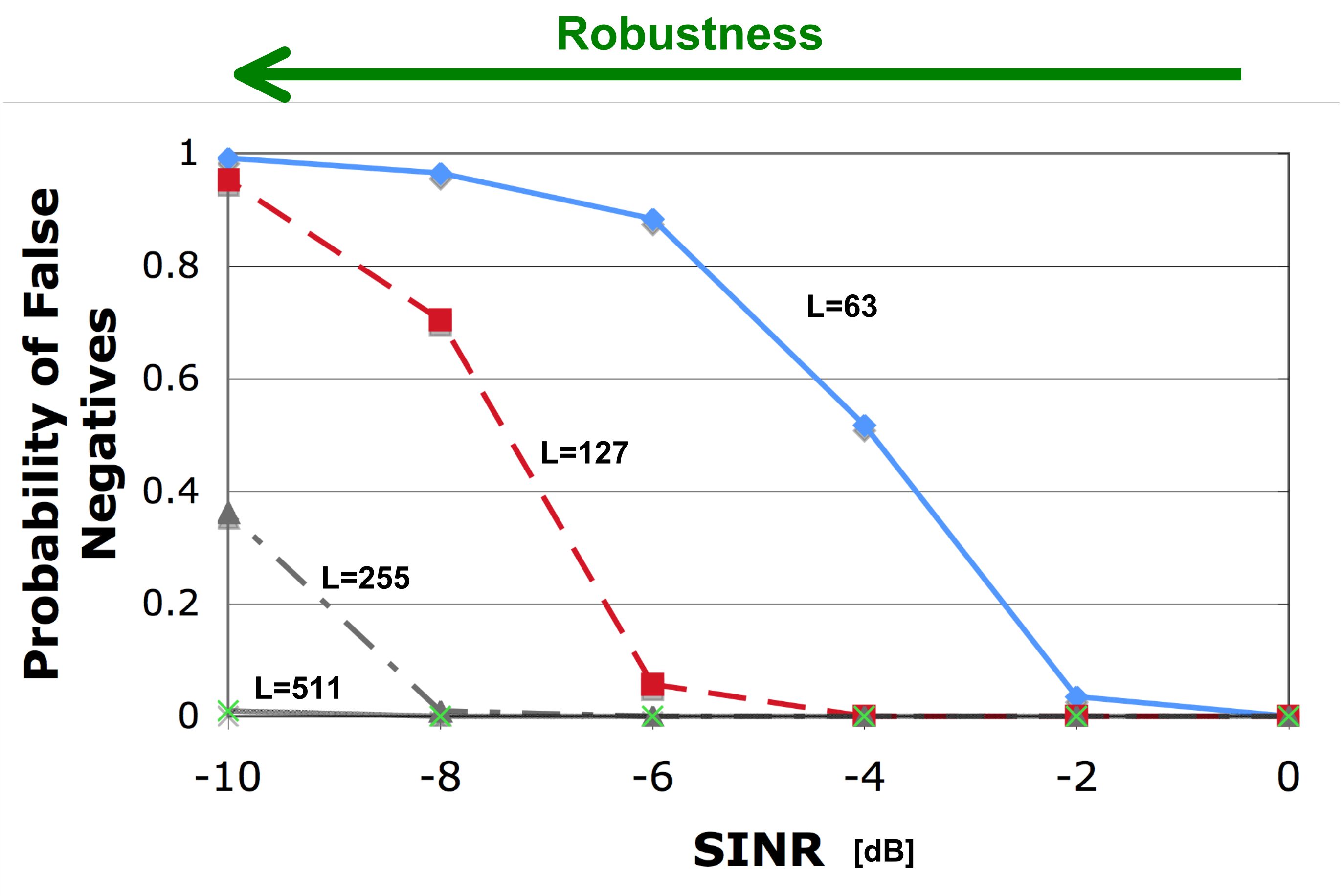


- Threshold Selection → Performance

Probability of false positives ~10^{-8}

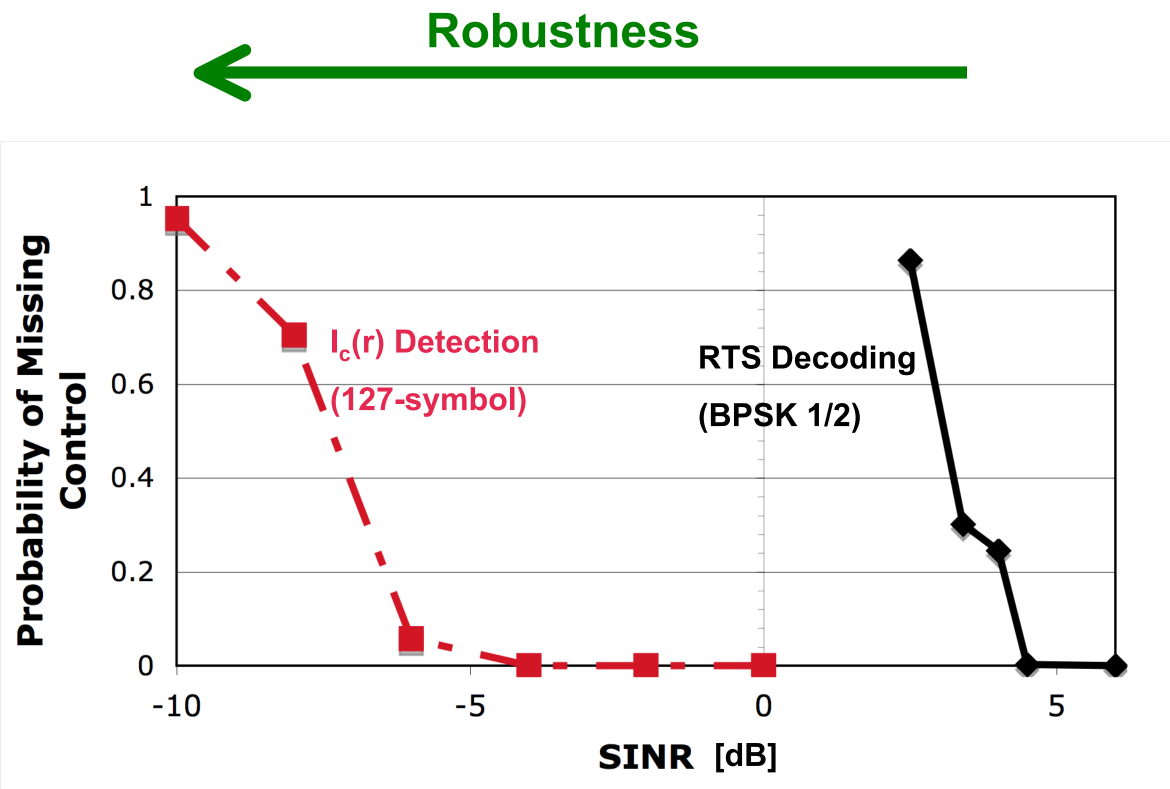
Probability of false negatives

CSS Robustness



- CSS are correlatable even at low SINR

Robustness and Overhead Gains

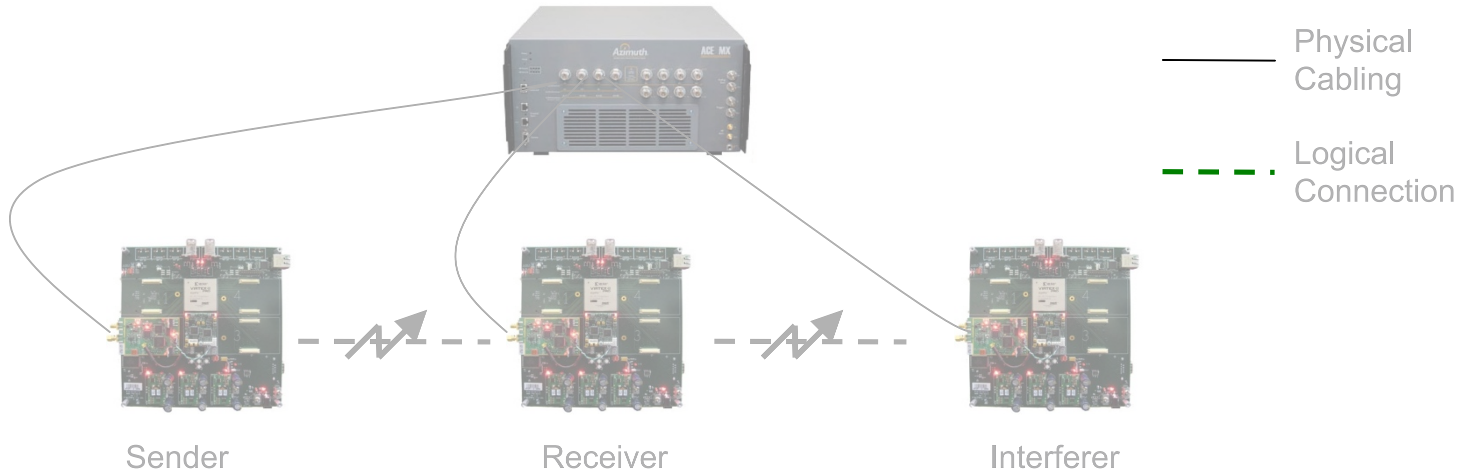


Duration: Message Control vs Coded Control		
11b/g	11a	*11ec*
~1000 μs	~184 μs	**~41 μs**

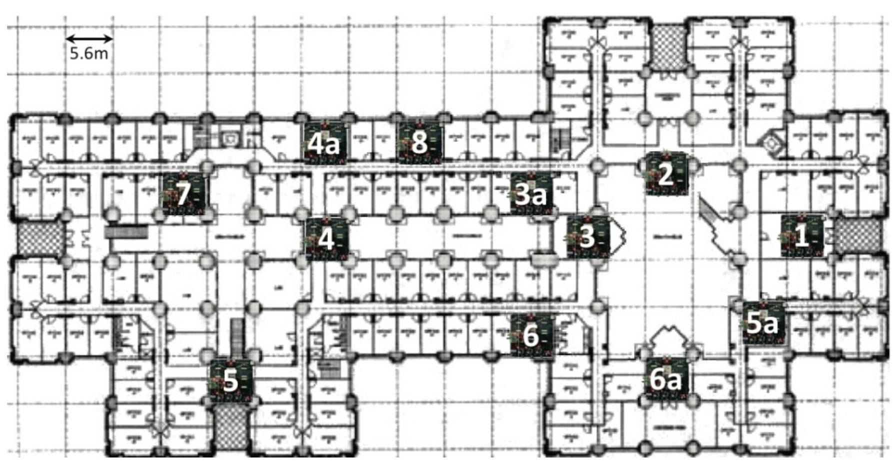
10 dB more robust **78% less overhead**

CSS Implementation

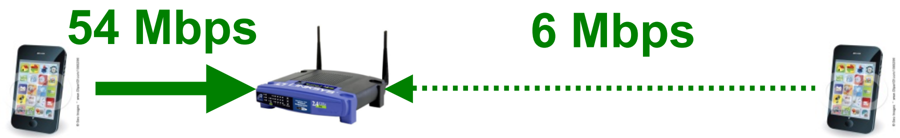
1. **Repeatable** and **controllable** experiments



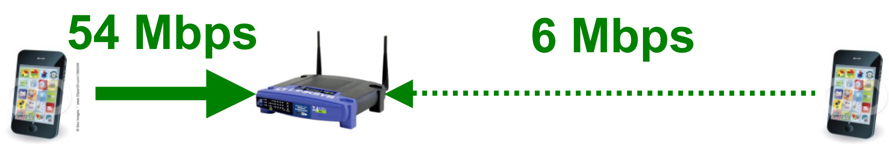
2. Emulation based on Realistic Channel Measurements

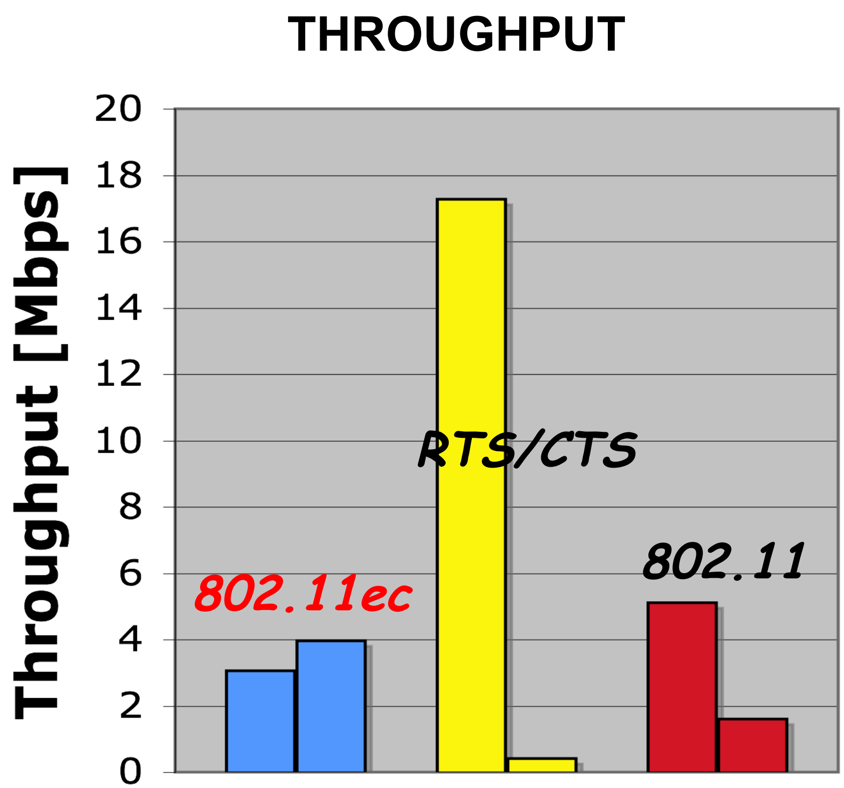


Hidden Terminals with Heterogeneous Rates



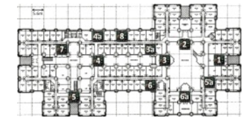
Hidden Terminals with Heterogeneous Rates

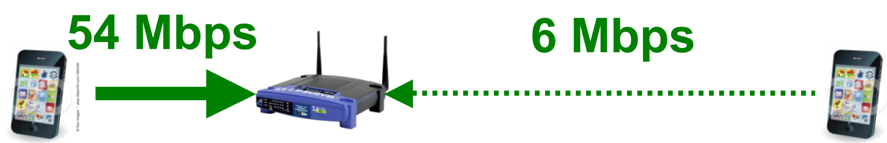


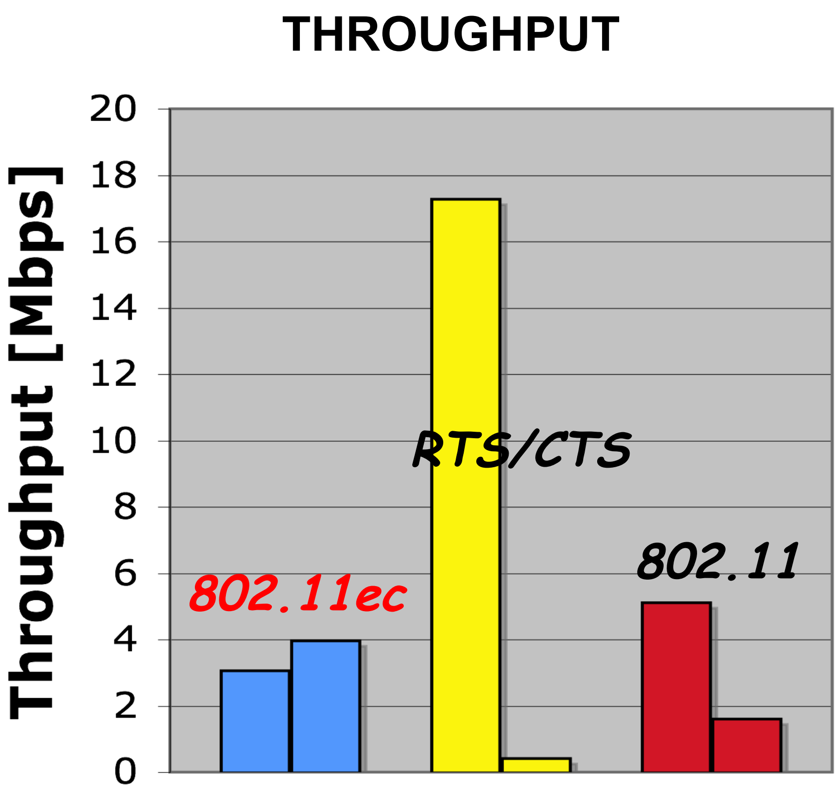


- Strong link RTS captures over weak link’s
- *802.11ec data collisions are rare*

Hidden Terminals with Heterogeneous Rates







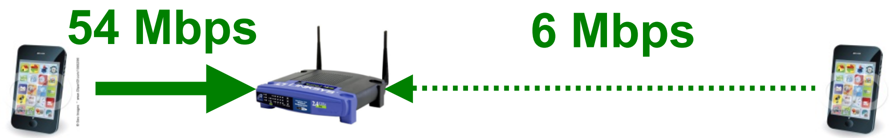
Jain Index	0.98	0.52	0.78
	fairest		

- Strong link RTS captures over weak link's

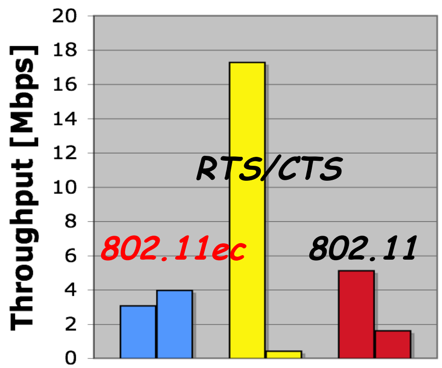
- *802.11ec data collisions are rare*

  FAIRNESS

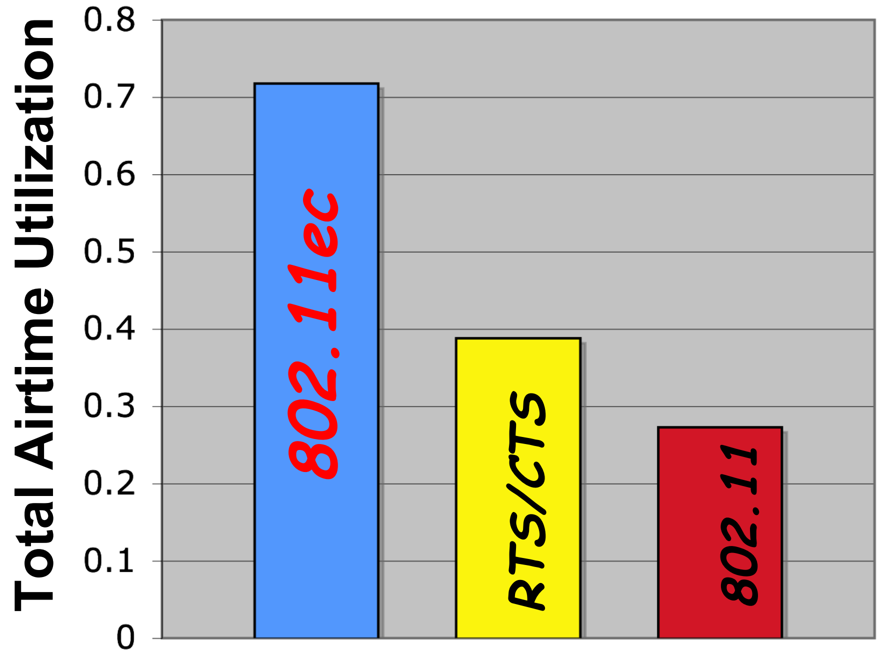
Hidden Terminals with Heterogeneous Rates



THROUGHPUT



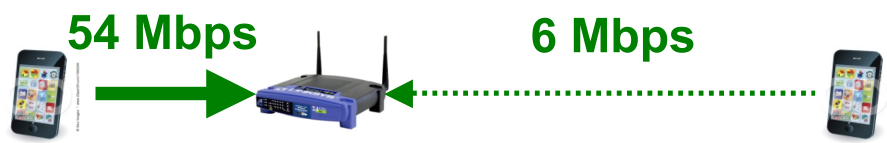
AIRTIME UTILIZATION

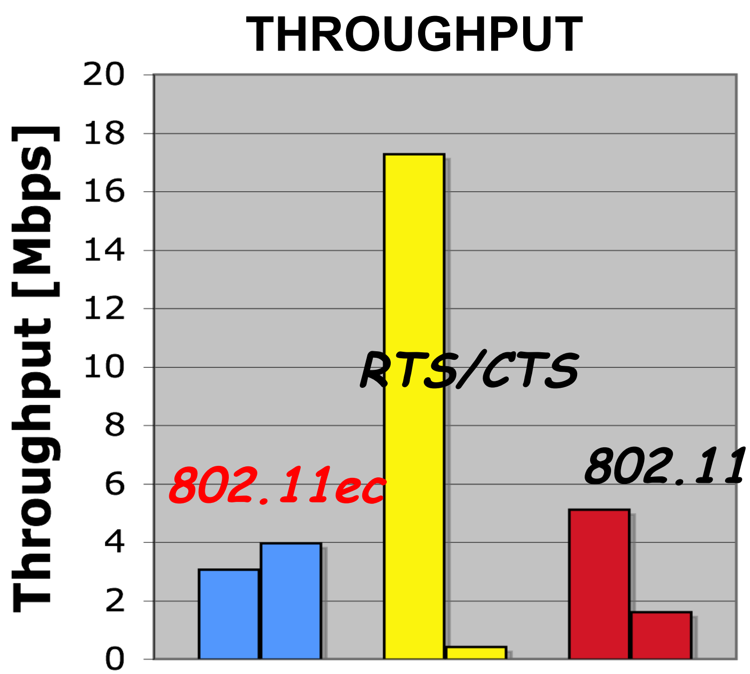


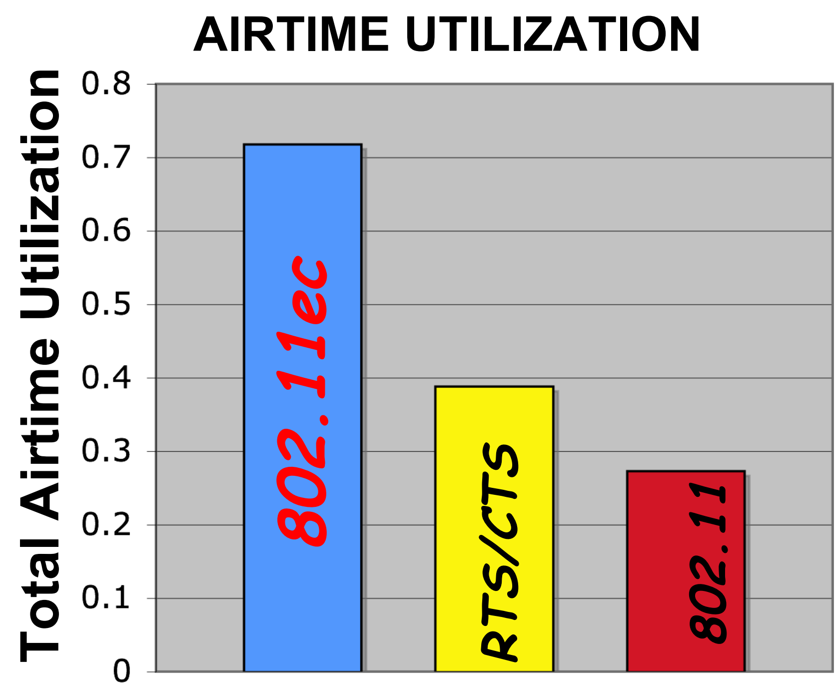
- Differently from 802.11, 802.11*ec* does not penalize weak and low data-rate links

→ ***802.11ec increases airtime utilization***

Hidden Terminals with Heterogeneous Rates





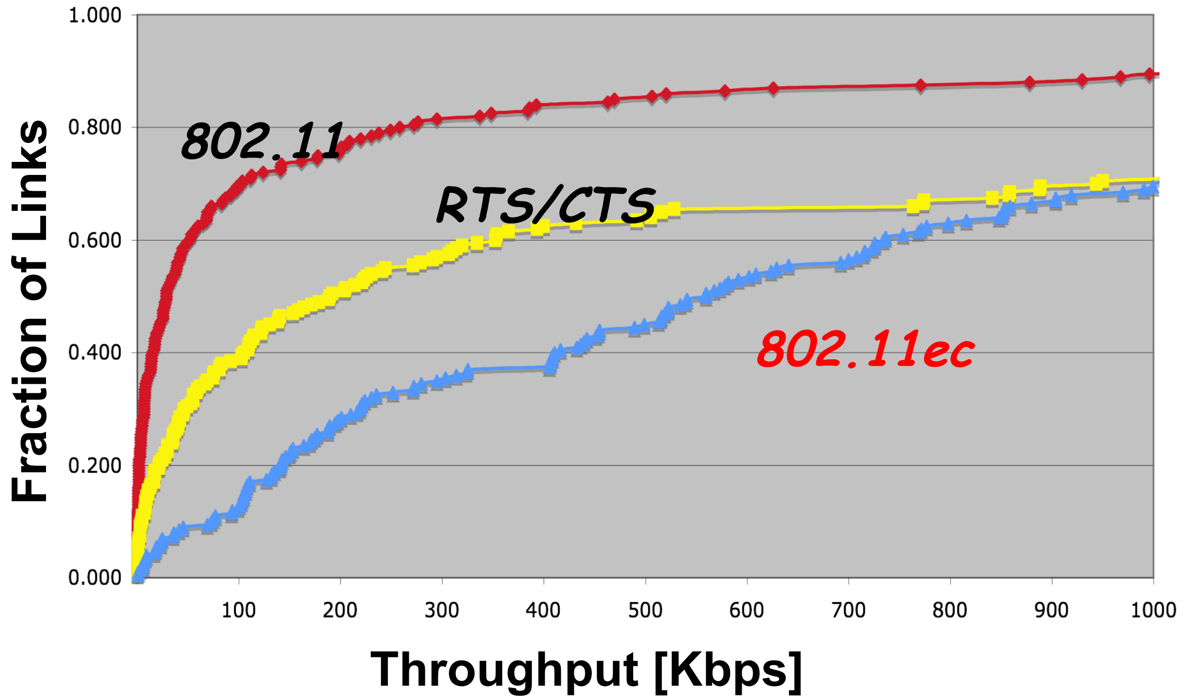


Jain Index 0.98 0.52 0.78

- 802.11*ec* increases **fairness** ***AND*** **channel utilization**

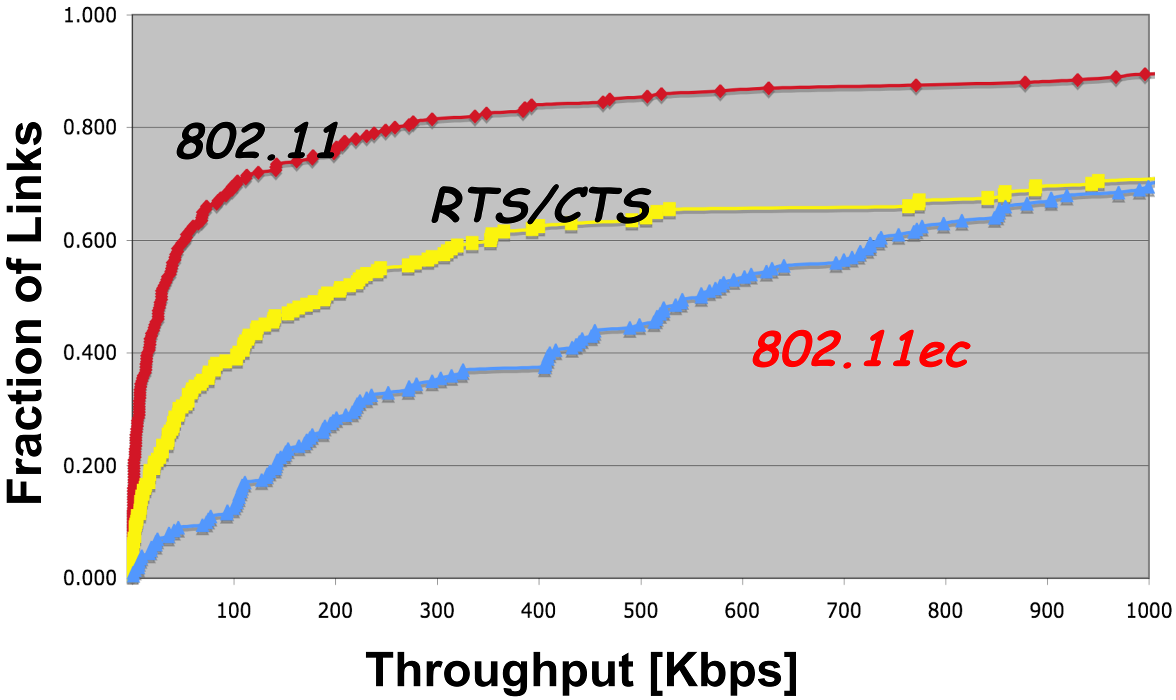
20-Node Random Topologies

- Simulations
- CDF of throughput distribution



20-Node Random Topologies

- Simulations
- CDF of throughput distribution



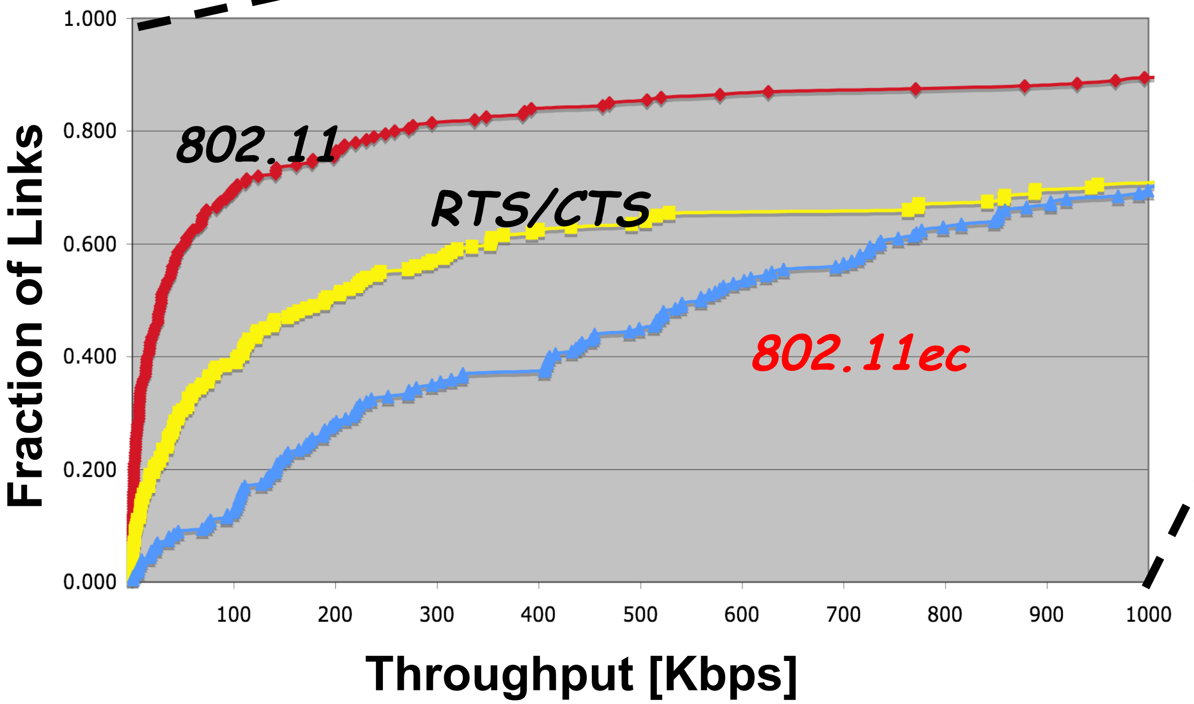
GAINS

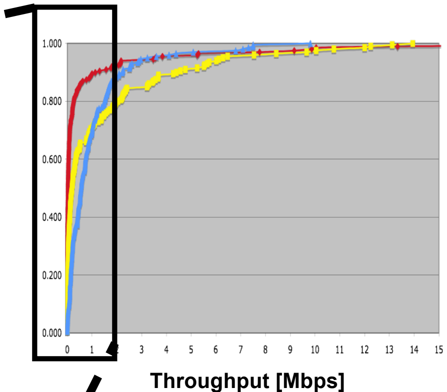
	802.11ec vs RTS/CTS
0.2	**7x**
0.4	**4x**
0.6	**1.9x**

- By reducing collisions, 802.11*ec* improves the weak links

20-Node Random Topologies

- Simulations
- CDF of throughput distribution





GAINS

	802.11ec vs RTS/CTS
0.2	**7x**
0.4	**4x**
0.6	**1.9x**

- By reducing collisions, 802.11*ec* improves the weak links

Summary and Conclusions

- Objective: Fundamentally **re-think the way control information is conveyed** in order to guarantee **low overhead** and **robustness**

- CSS's have short duration and improve robustness
- 802.11*ec* uses CSS's to convey control information
  - Small CSS Dictionary
  - Scope Control

- *802.11ec improves fairness while also increasing channel utilization*
  - Ex. 3x fairness, 1.5x airtime utilization, up to 12x throughput

802.11*ec*: Collision Avoidance without Control Messages

Eugenio Magistretti, Omer Gurewitz,
and Edward Knightly

Rice University
Ben Gurion University