"Implementation of a Cooperative MAC protocol using a Software Defined Radio Platform"

Thanasis Korakis

Joint work with Ankit Sharma, Vikas Gelara, Shashi Raj Singh,
Pei Liu, Shivendra Panwar

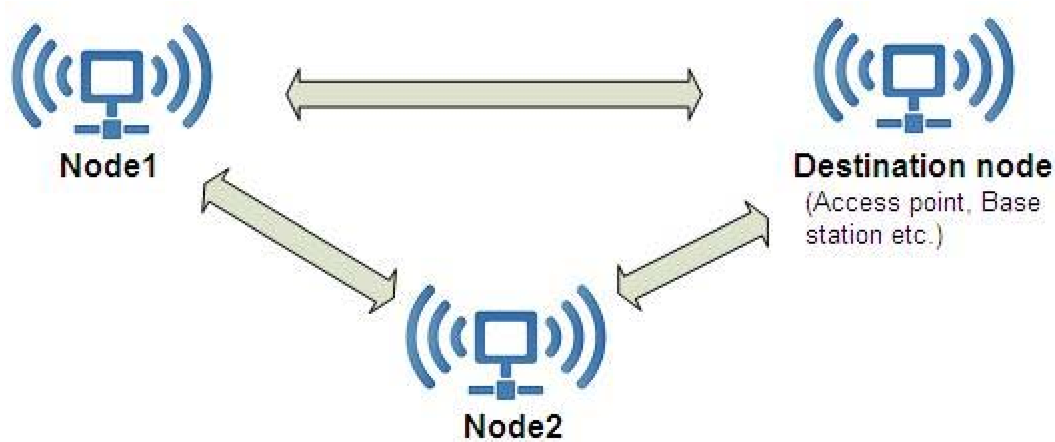
Center for Advanced Technology in Telecommunications
Polytechnic Institute of NYU
Brooklyn, NY 11201

Outline

- Motivation
- Cooperative MAC (CoopMAC) Protocol
- Implementation efforts
- Experiment results
- Conclusions – Future Work

Motivation for Cooperative Communications

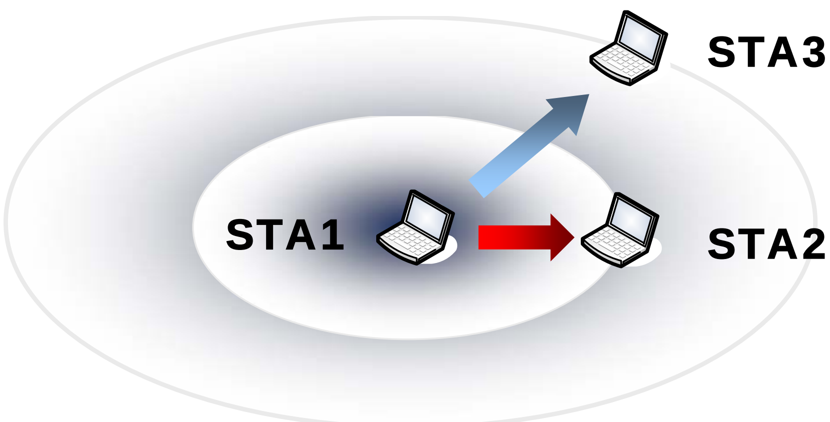
- Use other wireless nodes
  - To relay information
  - For robustness to channel losses and variations



- Results in
  - Spatial diversity, higher data rates, power saving, extended coverage, higher aggregate network performance.

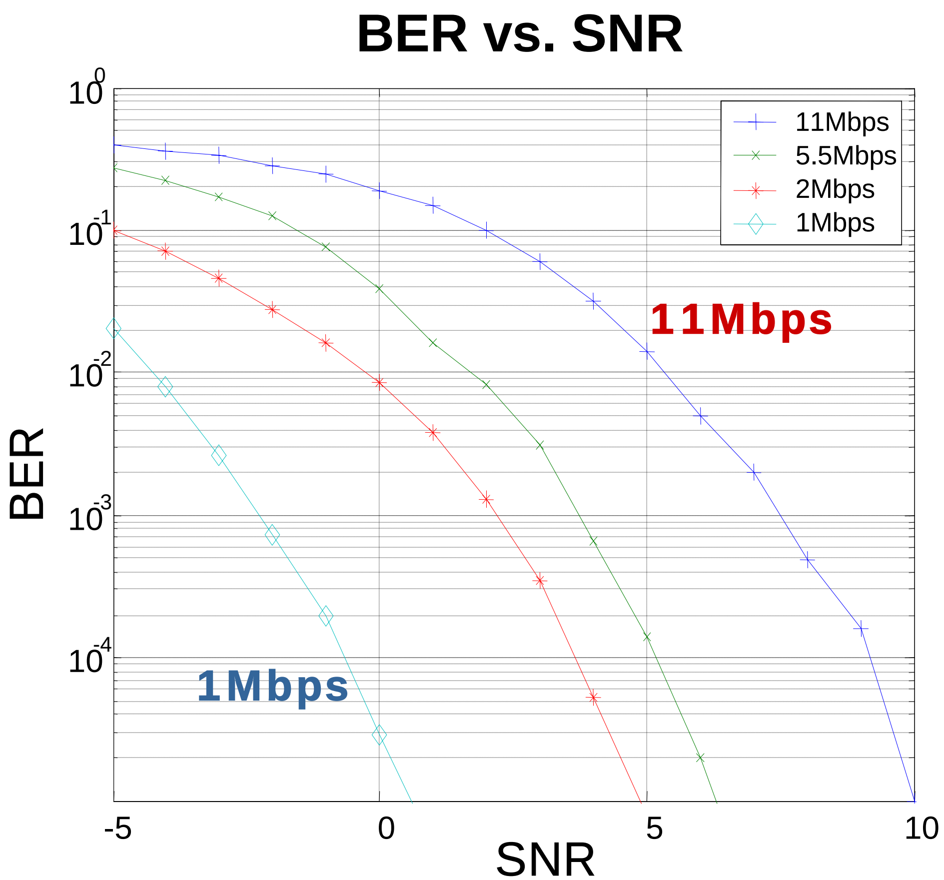
Motivation for Cooperative MAC protocols

- Wireless channel by nature is a broadcast one.
  - The broadcast channel can be fully exploited for broadcast traffic.
  - But it is considered more as a foe than a friend, when it comes to unicast.



Motivation for Cooperative MAC protocols

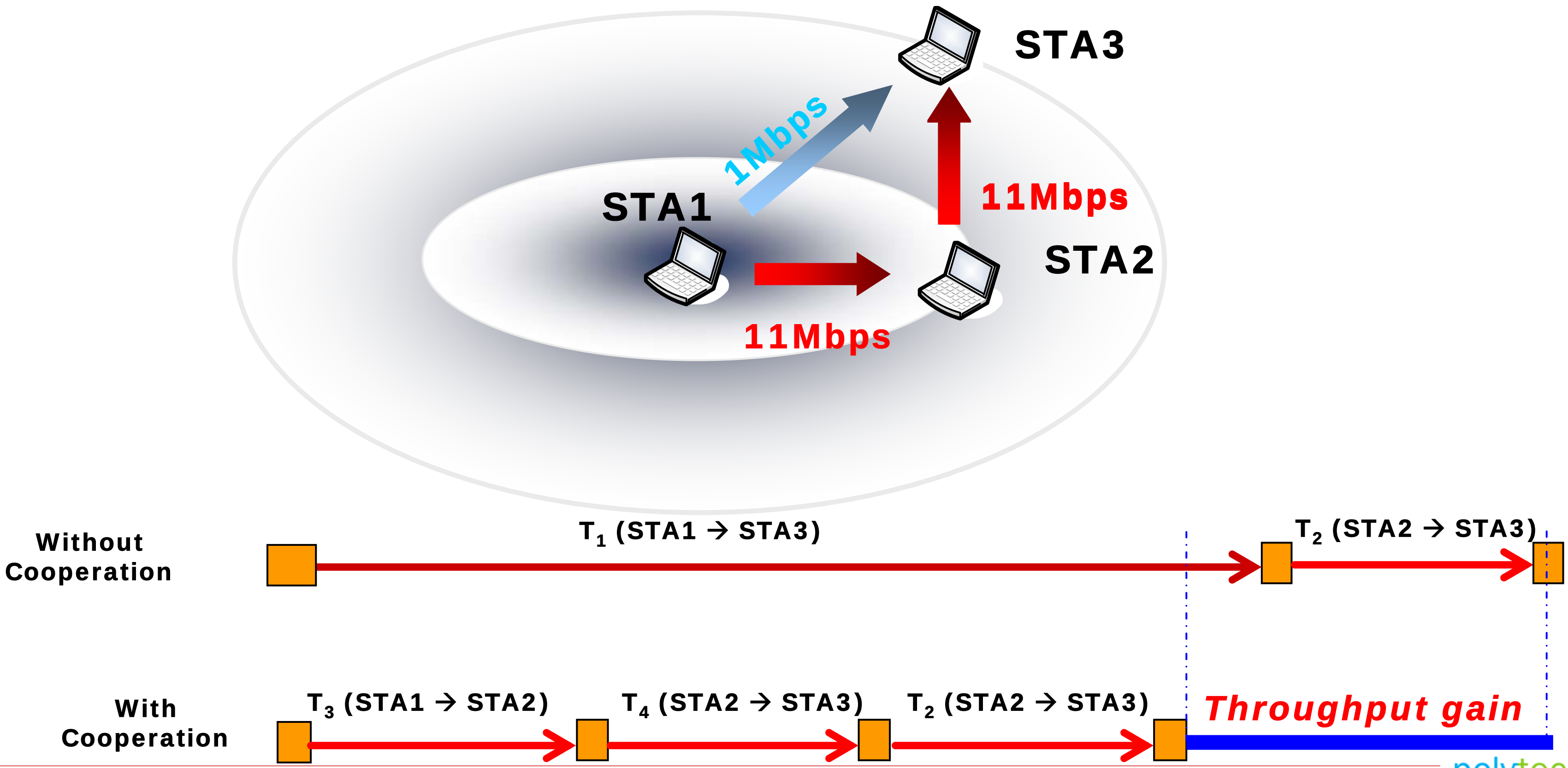
Multi-rate Capability of Wireless Protocols



- IEEE 802.11b: 1Mbps, 2Mbps, 5.5Mbps and 11Mbps.
- Primary objective: Combat adversary wireless channel conditions and deliver packets with acceptable BER/PER.
- **Basic principle**: Adjust the modulation scheme and transmission rate, based upon the perceived channel condition

Turn a Foe into a Friend?

- Inefficiency in 802.11 when different rates are used from different stations
  - "Slow" stations lower their throughput as well as the aggregate throughput of the network.
- A cooperative MAC protocol would leverage both the broadcasting and multirate capabilities of the existing MAC protocol



Outline

- Motivations
- Cooperative MAC (CoopMAC) Protocol
- Implementation efforts
- Experiment results
- Conclusions – Future Work

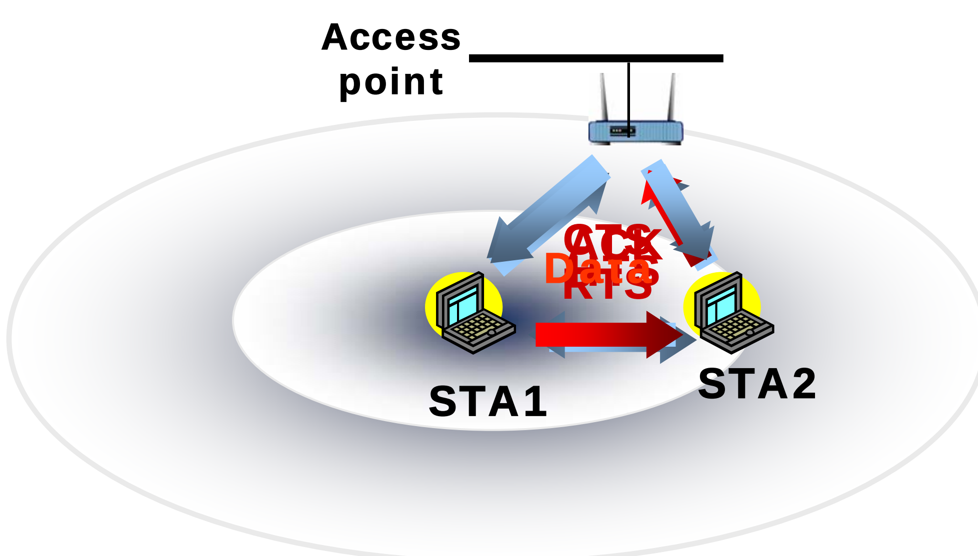
Protocol Definition

<u>Procedure for selecting a helper</u>

- Each station maintains a table (CoopTable) with information about neighboring stations and their ability to help
- Based on CoopTable, find a helper that satisfies the following condition
  - $\text{Time}_{[\text{Direct Tx}]} > \text{Time}_{[\text{Two-hop Relaying}]}$

<u>Data transmission procedure</u>

- Handshake
  - RTS (Ready To Send)
  - HTS (Helper ready To Send)
  - CTS (Clear To Send)
- Data transmission
- Acknowledgement



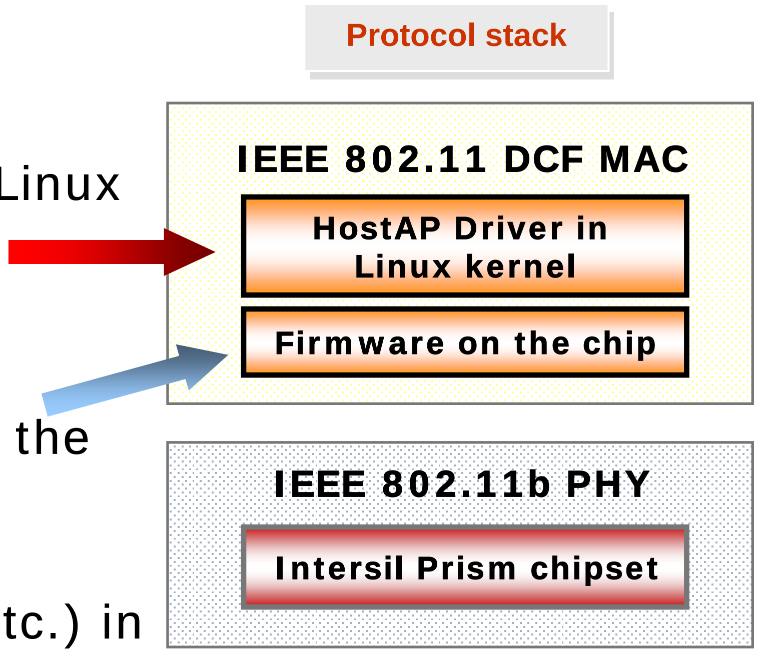
Outline

- Motivations
- Cooperative MAC (CoopMAC) Protocol
- Implementation efforts
- Experiment results
- Conclusions – Future Work

Implementation approaches

Previous Work

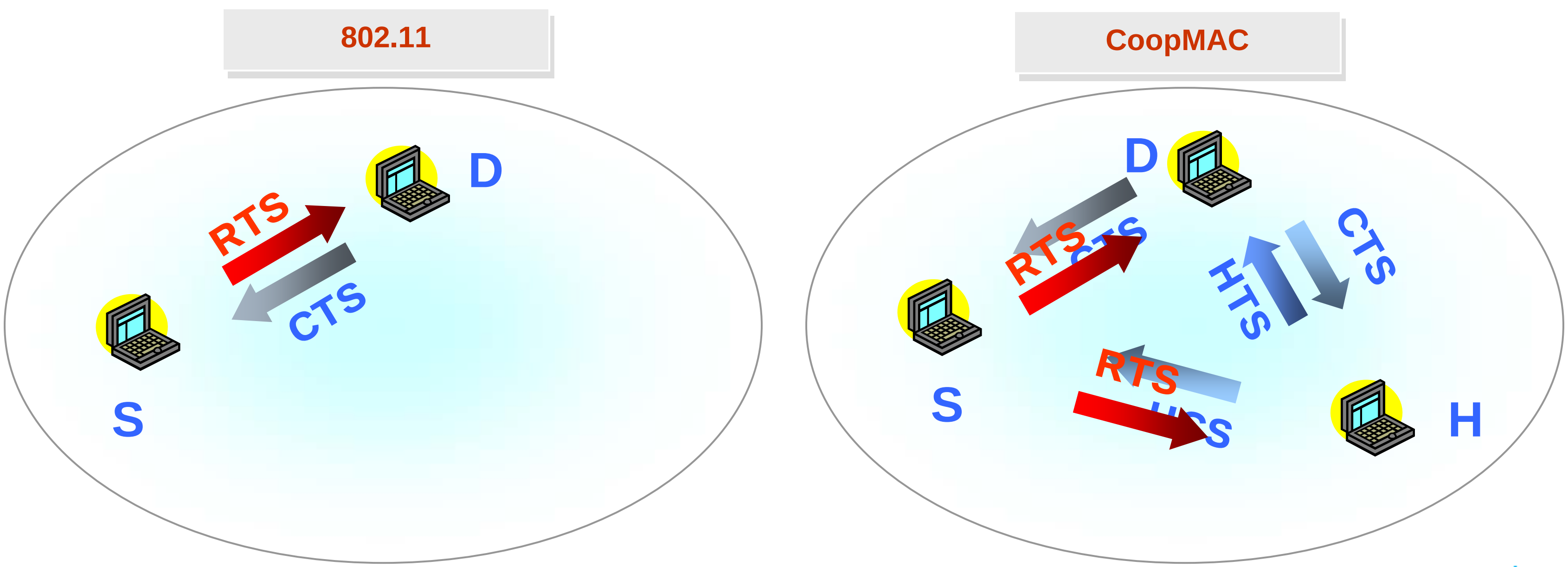
- Platform
  - Based upon HostAP driver on a Linux platform.
  - Modified driver for CoopMAC implementation.
- Limitations Faced: Inaccessibility of the firmware for modification.
  - Time sensitive functions (e.g. transmission of a Packet (ACK etc.) in SIFS interval) are controlled by the firmware.
  - RTS-CTS functionality is controlled by the firmware.



- This implementation missed several critical functional characteristics of the original protocol due to above limitations.

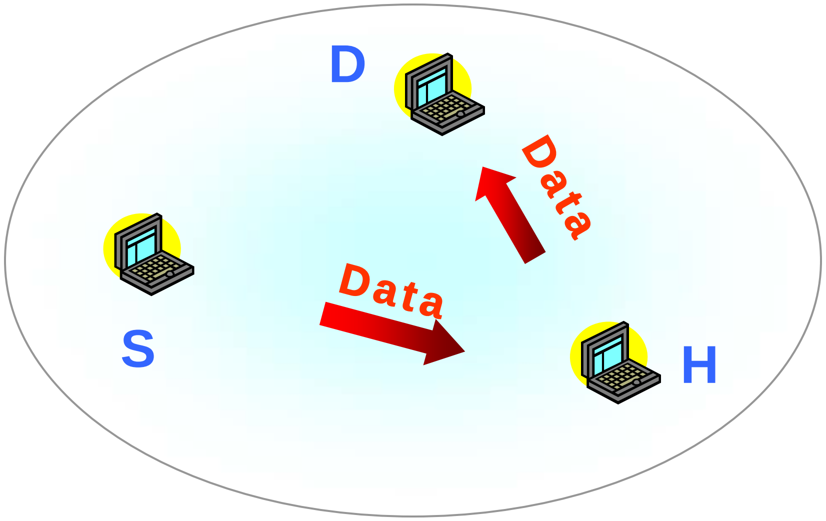
Drivers Implementation

- **Challenge 1**
  - To replace the initial control packets (RTS-CTS) with the RTS-HTS-CTS packets
  - SIFS (e.g., 10ms for IEEE 802.11b) time as the interval between two consecutive control messages.
- **Approach:** We suspend the initial control frames, as we can't control the timing of the transmission of those frames.



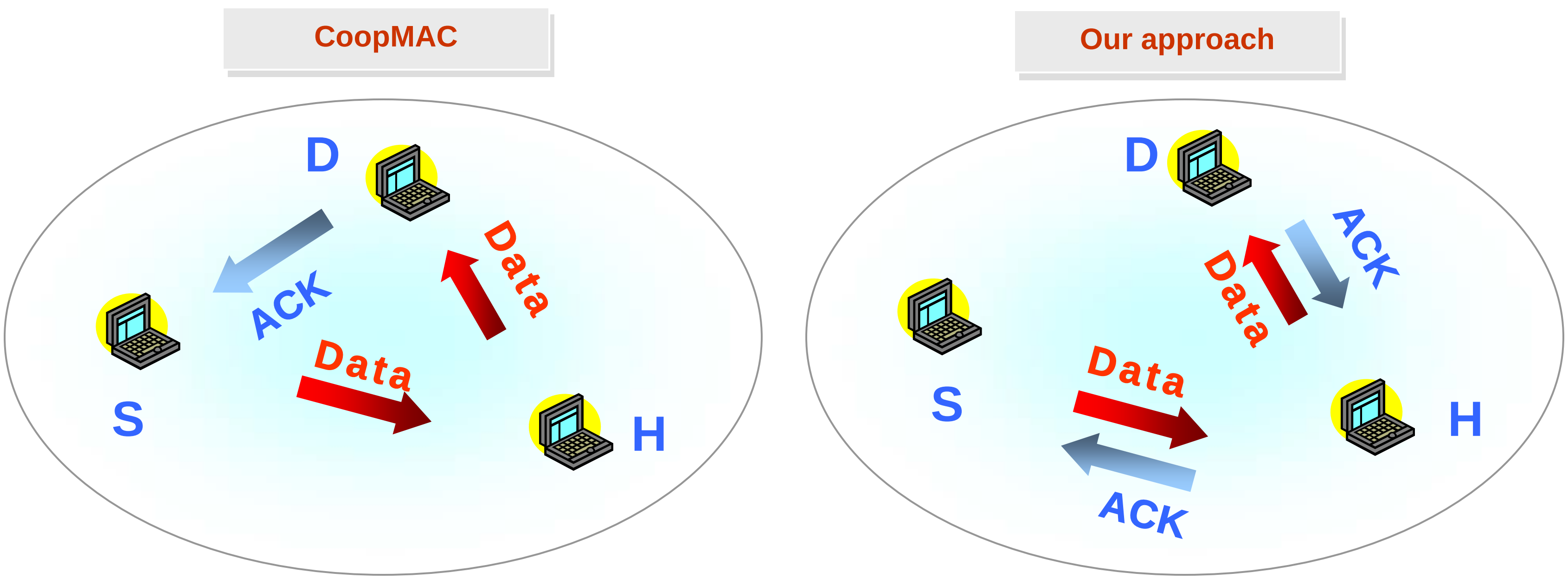
Drivers Implementation

- **Challenge 2:** To transmit the data packet from the helper to receiver without competing for the medium (SIFS period after its reception).
- **Approach:** As we can't control time sensitive functions of MAC (they are controlled by the firmware), we go with the approach of medium contention during the second hop transmission, which increases the overhead.



Drivers Implementation

- **Challenge 3:** to suppress the ACK at the helper
- **Approach:** Keep the ACK from the destination, at the expense of living with the ACK from the helper.

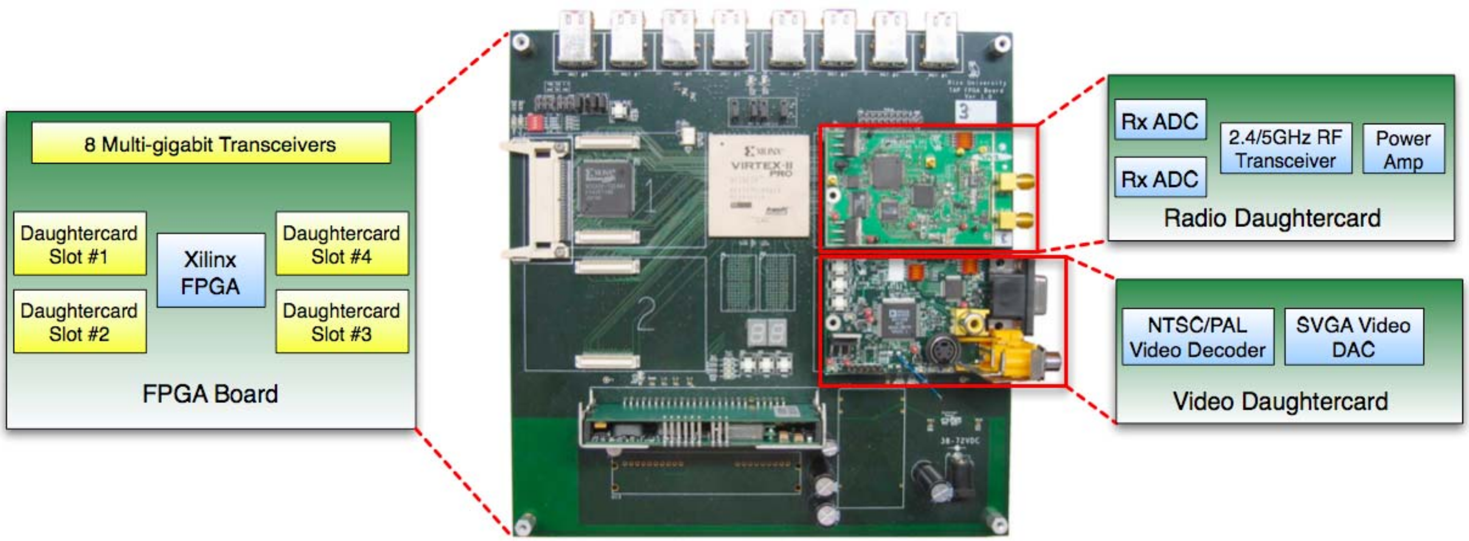


CoopMAC implementation

- In order to overcome those limitations, we had to go from the drivers implementation to an **all software radio implementation**:
- Two options:
  - WARP
  - GNU Radio
- We chose WARP:
  - More powerful
  - Realistic transmission rates (up to 54Mbps)
  - Convenient to build cooperation in the PHY layer

WARP Platform

- Software defined radio platform developed by Rice University.
- Xilinx Virtex II Pro FPGA board with embedded power PC Processors.
- OFDM based PHY
  - Loosely Based on 802.11a standard.
  - 2.4GHz/5 ISM/UNII bands for transmission.
- Provides a MAC framework called WARPMAC.
- Both the PHY and the MAC layers can be changed to any extent.



SDR Implementation

- Two operational modes for transmission
  - *Direct* mode: legacy direct mode under the CSMA protocol.
  - *Cooperative mode*: that enables CoopMAC. Packet is forwarded to the destination through the helper using two fast hops.
- Enhanced packet structure called *CoopFrame*
  - *CoopDestinationID*: new subfield in header. Used in cooperative mode and indicates the *final destination* of the packet.
  - Two new packet types (PktType subfield):
    - *COOPPACKET*: used in CoopMAC for the first hop transmission (source to helper).
    - *COOPFINAL*: A packet that is used in CoopMAC for the second hop transmission

Functionality modifications

- **Transmitter**
  - Based on the CoopMAC table, selects the cooperative mode or Direct mode.
  - Based on the mode, the header of Coopframe is constructed.
- **Receiver**

  Based on the type of the packet it receives:
  - *DATAPACKET*: an ACK is transmitted back to the source node.
  - *COOPPACKET*: the receiver realizes that it should react as a helper. It replaces the *Destination Address* field with that of the final destination address based on the *CoopDestinationID* field, and forwards the packet immediately (defining it now as a COOPFINAL packet), without contending for the channel.
  - *COOPFINAL*: sends back an ACK, directly

    to the source node.

CoopMAC Table

- Under the CoopMAC protocol, a station updates its CoopMAC Table passively, based upon the overheard packets exchanged between its neighbors.
- In addition to the passive mechanism, an active approach has also been taken in the implementation.
  - Every station broadcasts a Hello packet, which explicitly indicates the rate between itself and the neighbors.
  - This is used to create a more controllable experiment environment.
- Every time a station receives a Hello Packet, it updates the corresponding record in the CoopMAC Table or it adds a new entry.

CoopMAC Table

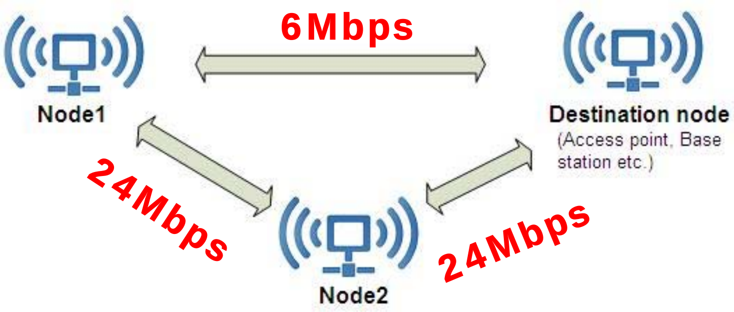
Destination MAC	*DirectR (Mbps)*	*Helper MAC*	R_{sh} $(Mbps)$	R_{hd} $(Mbps)$
16.24. ...e2.c3	6	16.24. ...e2.c4	24	24
16.24. ...e2.c7	6	16.24. ...e2.c8	24	12
...	...	....	....	....

Outline

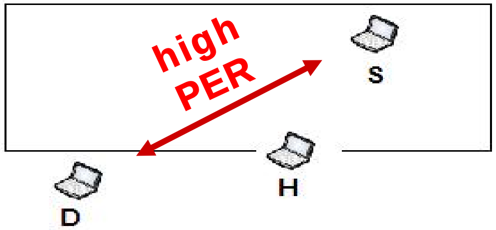
- Motivations
- Cooperative MAC (CoopMAC) Protocol
- Implementation efforts
- Experiment results
- Conclusions – Future Work

Experimental Study

- **Scenario 1**
  - Shows the gains obtained by boosting the transmission rates for fixed PER.



- **Scenario 2**
  - Shows the gains obtained by decreasing the PER while fixing the transmission rates.

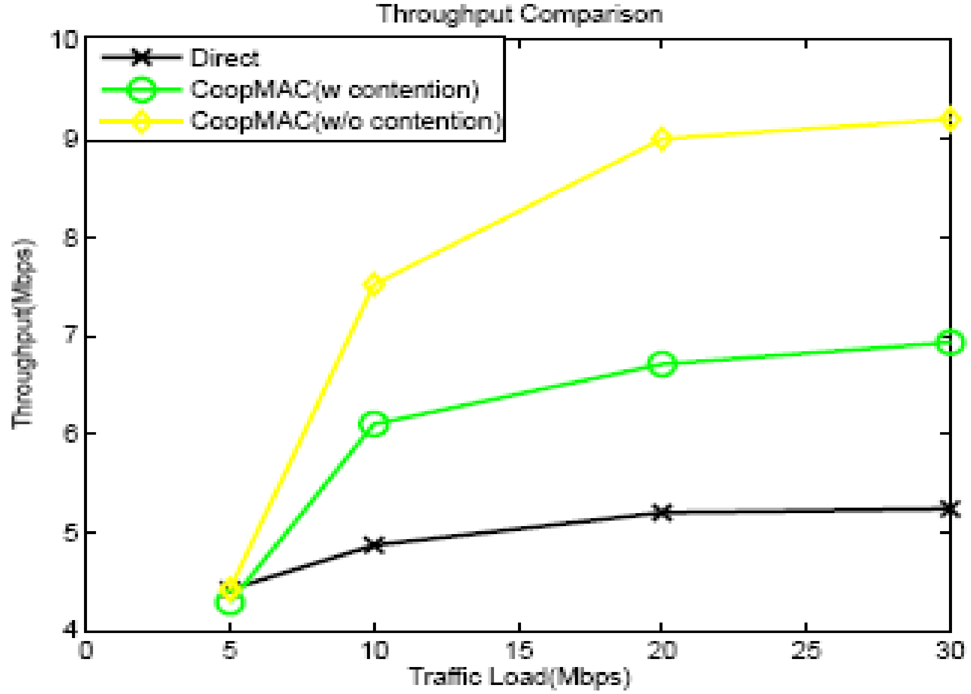


- **Basic experimental settings**
  a. Iperf is used to generate UDP traffic.
  b. Packet size:1470 bytes

Performance Evaluation

Scenario 1

- **Experiment settings:**
  - Direct transmission: 6*Mbps.*
  - Cooperative transmission: both hops 24*Mbps*.



(a) Throughput Performance

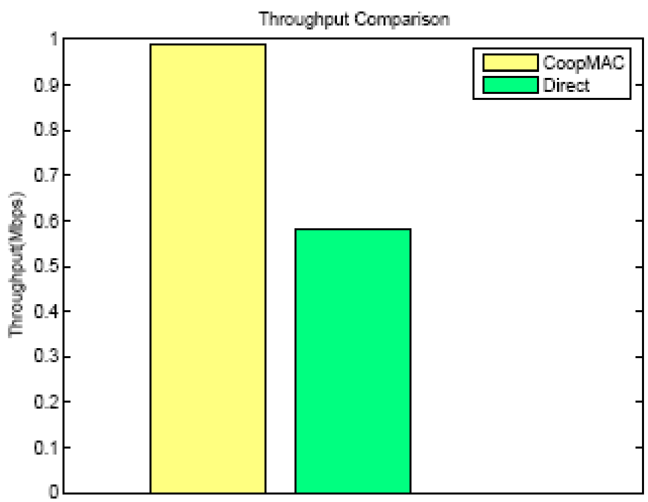
- **Major findings:**
  - CoopMAC (with or without contention) performs better than the direct transmission.
  - New implementation (*CoopMAC without contention*) performs much better than our earlier implementation (*CoopMAC with contention*).

Experimental settings: 1470 bytes payload size

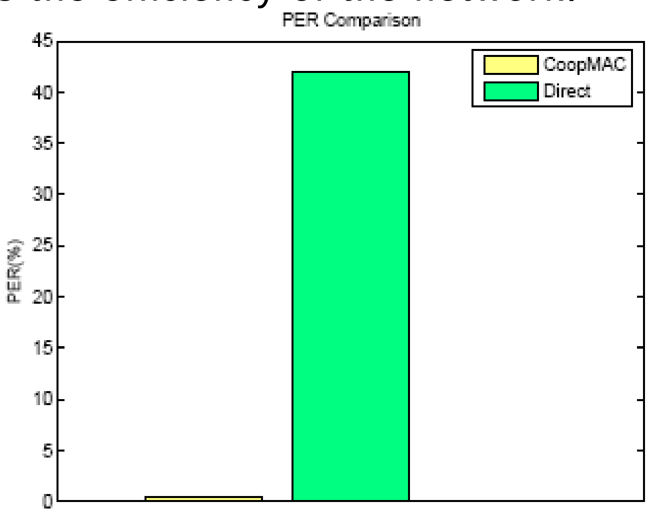
Performance Evaluation

Scenario 2

- **Experiment settings:**
  - Transmission rate for the direct & the two hops of cooperative is 6*Mbps*.
  - Throughput and PER for traffic load of 1*Mbps.*
- **Major findings:**
  - Throughput of CoopMAC is almost double than that this of *direct transmission.*
  - PER for the direct transmission is very high (higher than 40%).
  - Cooperative scheme keeps the PER of the communication at a very low level (less than 2%) and therefore increases the efficiency of the network.



(a) Throughput Comparison



(b) Packet Error Rate Comparison

Conclusions – Future work

- This work explored the full potential of a Cooperative MAC protocol by implementing it on a software defined radio platform.
  - Open source drivers limitations overcame
  - Feasible to be implemented
  - Minimal changes to the 802.11 protocol
  - Significant performance boost

- **Future work**
  - Implementation of cooperative schemes in the PHY layer using the WARP platform, and combine them with the existing cooperative MAC protocol.

Thank you!

Check the project's site: **http://witestlab.poly.edu**