FRACTEL – Design, Implementation And Evaluation of Multi-hop Wireless TDMA System

Nirav Uchat

Faculty Mentors

Prof. Kameswari Chebrolu and Prof. Bhaskaran Raman

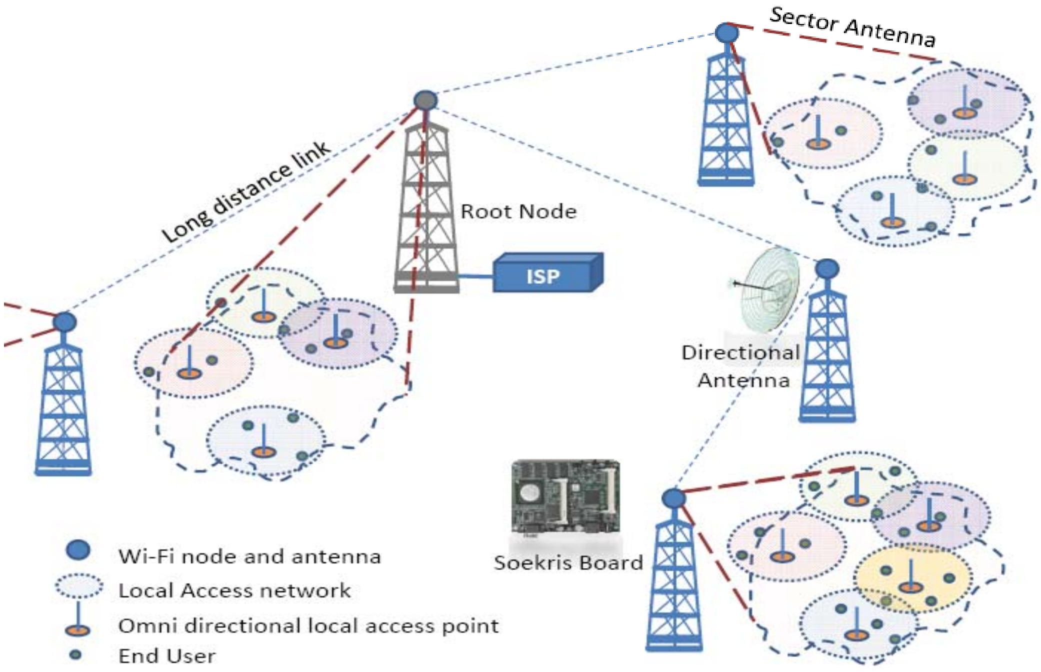
SYNERG MTP-2 Workshop

January 17, 2009

Department of Computer Science and Engineering
IIT Bombay

Project Vision

- Digital inclusion of remote villages
- Providing data, voice and video connectivity with QoS guarantees
- Cost effective solution by using off the shelf hardware, open source driver and license free band



Challenges in Wireless

- Issues in using 802.11 Wi-Fi protocol
  - Long distance carrier sensing
  - Difficult to assure QoS guarantees
  - Poor performance on long distance link
- How about using TDMA?
  - Communication with precise slot boundaries; no CSMA
  - Minimum collision due to synchronous operation
  - Guaranteed fulfillment of QoS requirements due to centralized scheduling
- TDMA more suitable than CSMA for our requirements

Problem Statement

Design, Implementation and Evaluation of multi-hop wireless TDMA system

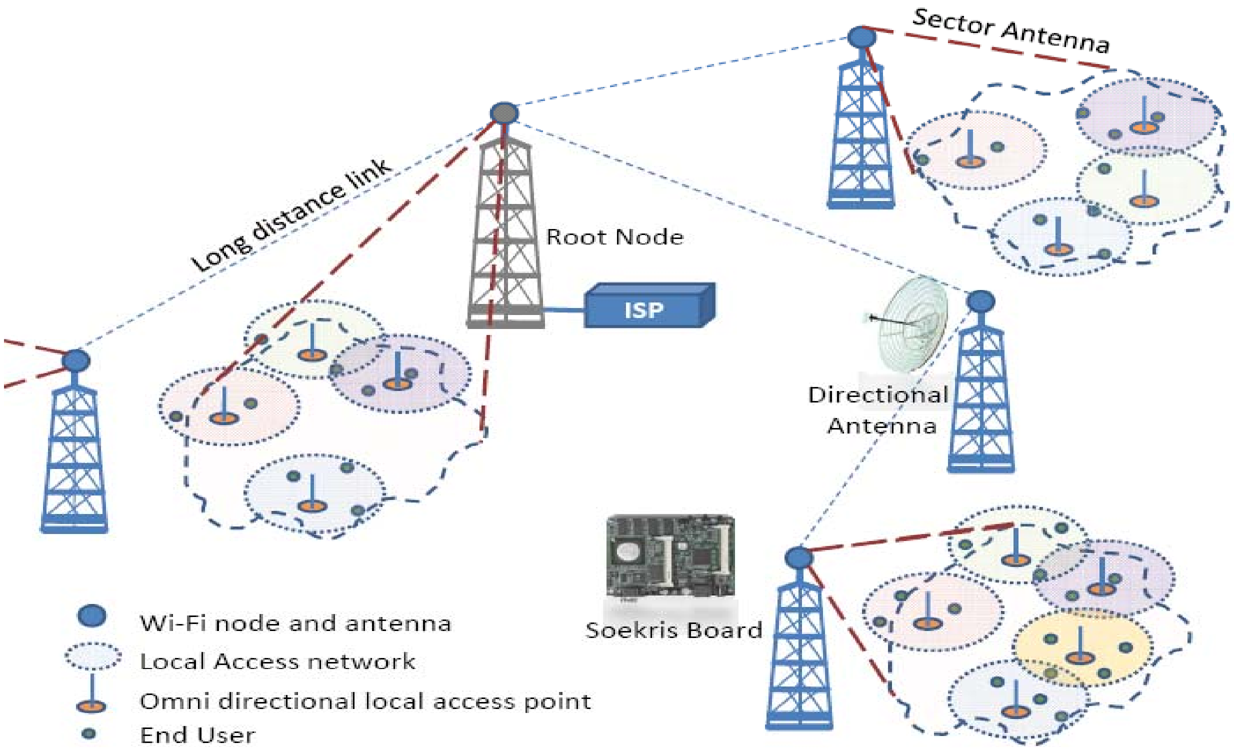
- – Dynamically adapting the schedule in response to change in network load and topology
- – Should support both best effort (HTTP, FTP) and real-time (voice, video) traffic

Related Work

- Existing protocols provide hooks into madwifi drivers for
  - stripping off CSMA mechanism (SoftMAC – NOV, 2005)
  - using different MAC protocols based on network conditions (MultiMAC – NOV, 2005)
  - precise time synchronization (MadMAC – SEP, 2006)
  - control over radio configuration and time critical functions (FreeMAC – AUG, 2008)
- Different Approach
  - Overlay MAC approach - works above MAC layer (JUN,2005)
  - 2P Protocol on bipartite graph with marker packet - HostAP driver on prism chipset – (AUG, 2005)
  - SRAWAN - IIT Kanpur (May, 2006)
- To our knowledge, there is no implementation of multi-hop TDMA system yet

Our Approach

- Centralized TDMA scheduler
  - Root node creates a global schedule and disseminates it across the network
  - Adapting schedule based on bandwidth requests
- Synchronization mechanism
- Multi-hop TDMA implementation at MAC layer



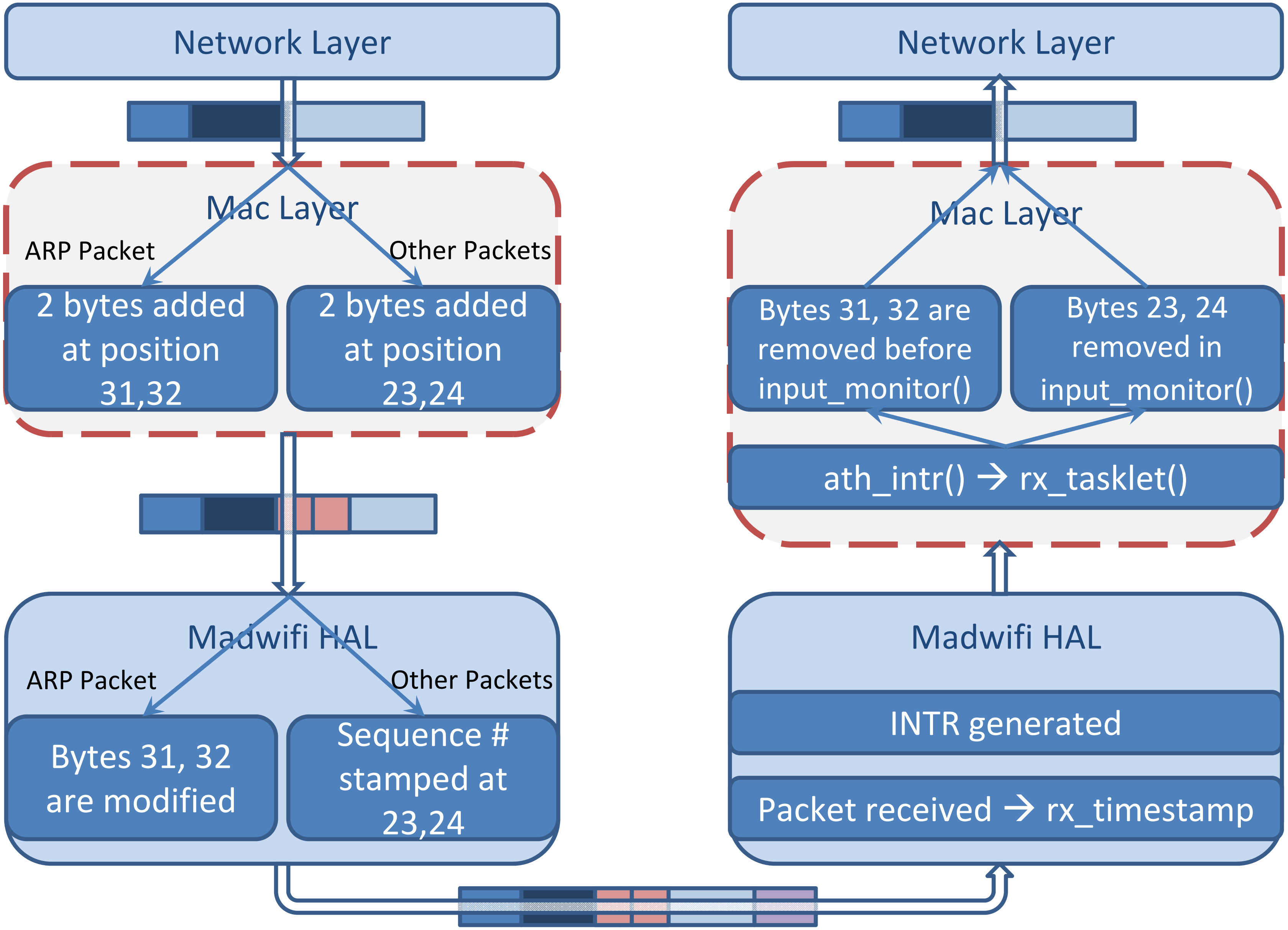
Modifications to Madwifi 1 of 2

- Disabled MAC level acknowledgments - Tested
- No RTS/CTS – Tested
- Raw packet transmission; no 802.11 frame - Tested
- Disabled random/post back-off mechanism - Tested
- Tweaked CCA mechanism to always sense channel clear - Not Tested
- Generating hardware time stamped packets - Tested

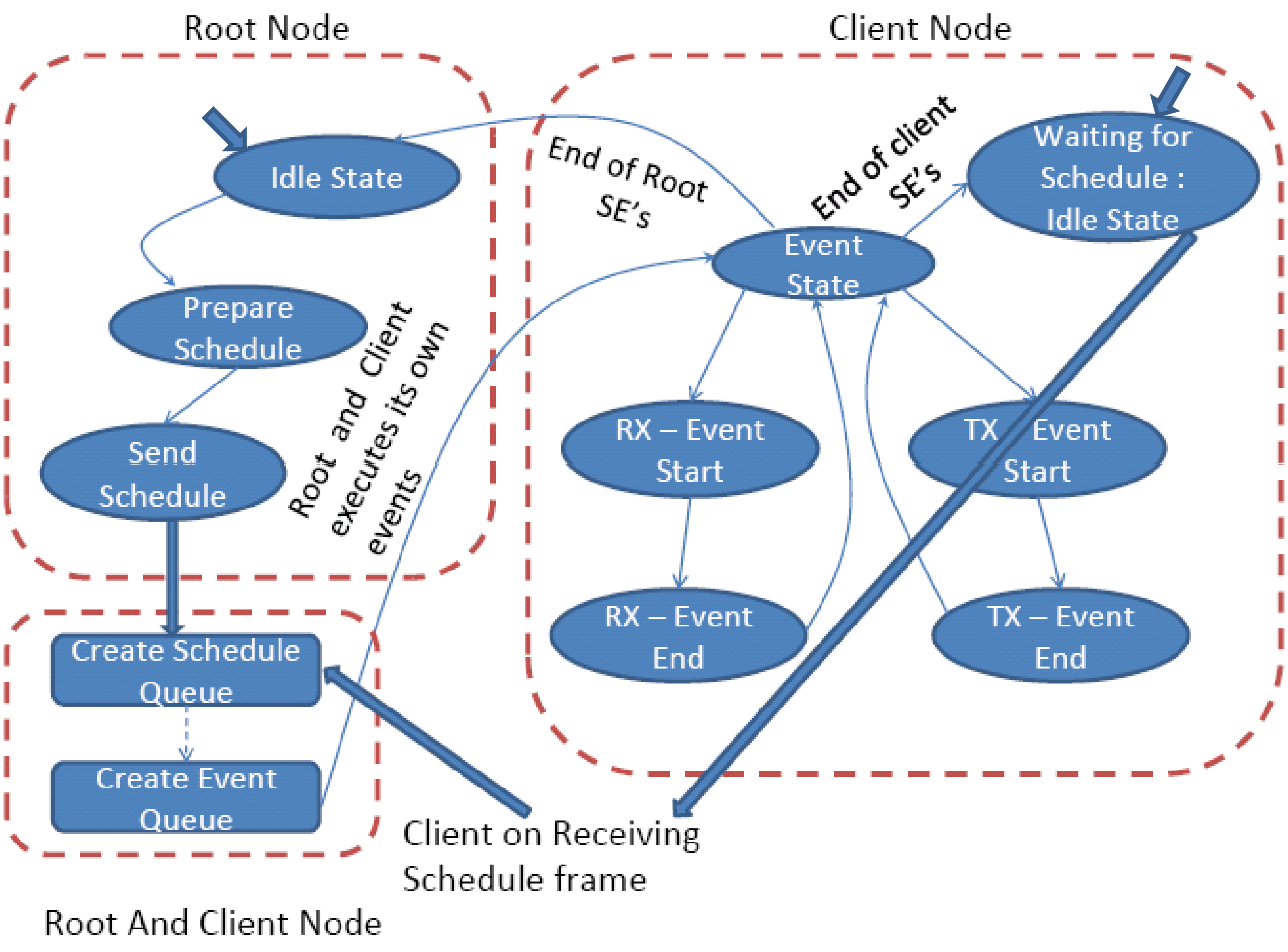
Modifications to Madwifi 2 of 2

- Packet send/receive in monitor mode - Tested
- Generation of control packets at MAC layer (in monitor mode) - Tested
- Enabled channel switching from driver code - Tested
- Packet Filtering based on destination MAC address and discarding packet with CRC and PHY error - Tested
- Plugged in TDMA schedule header, data header and scheduling elements - Implemented and Tested

Modifications to Monitor mode

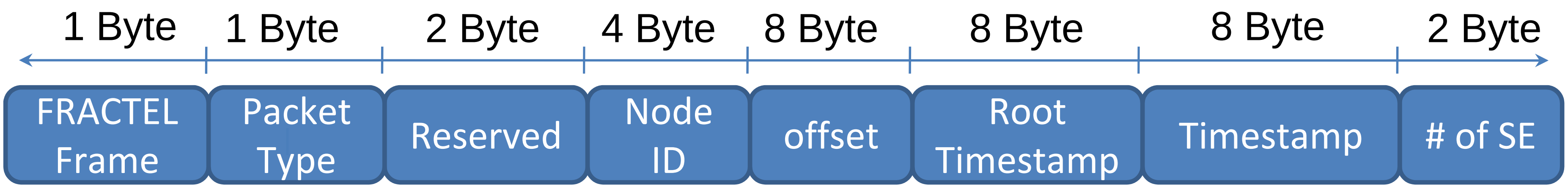

Network Layer
Mac Layer
ARP Packet
Other Packets
2 bytes added at position 31,32
2 bytes added at position 23,24
Madwifi HAL
ARP Packet
Other Packets
Bytes 31, 32 are modified
Sequence # stamped at 23,24
Network Layer
Mac Layer
Bytes 31, 32 are removed before input_monitor()
Bytes 23, 24 removed in input_monitor()
ath_intr() → rx_tasklet()
Madwifi HAL
INTR generated
Packet received → rx_timestamp

TDMA State Diagram



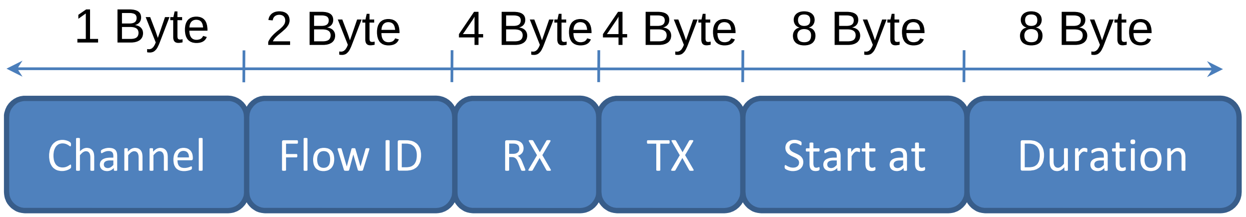
TDMA Frame Format

Fractel - Schedule Header



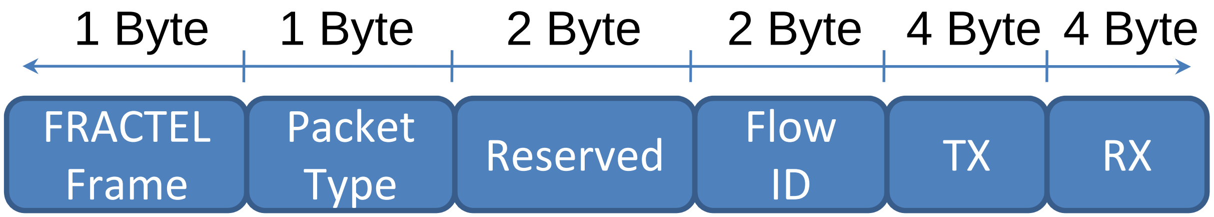
1 Byte	1 Byte	2 Byte	4 Byte	8 Byte	8 Byte	8 Byte	2 Byte
FRACTEL Frame	Packet Type	Reserved	Node ID	offset	Root Timestamp	Timestamp	# of SE

Fractel - Scheduling Element



1 Byte	2 Byte	4 Byte	4 Byte	8 Byte	8 Byte
Channel	Flow ID	RX	TX	Start at	Duration

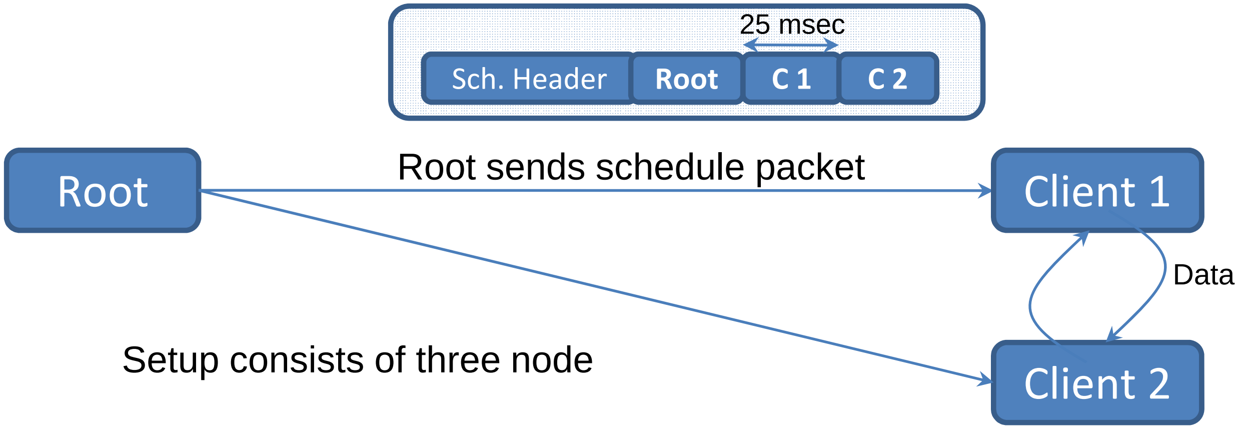
Fractel - Data Header



1 Byte	1 Byte	2 Byte	2 Byte	4 Byte	4 Byte
FRACTEL Frame	Packet Type	Reserved	Flow ID	TX	RX

FRACTEL Frame	0xFF	
Packet Type	0x08	Schedule
	0x0C	Data
Reserved	To suppress NAV	
RX/TX	IP Address	
Offset	Used in SYNC	
Root TS	Used in SYNC	
Time Stamp	Reserved for Muti-Hop TDMA	
# of SE	# of Scheduling Elements	
Start at	Start time for SE	
Duration	Duration of SE	
Flow ID	Reserved	
Channel	Reserved	

Experiment Setup

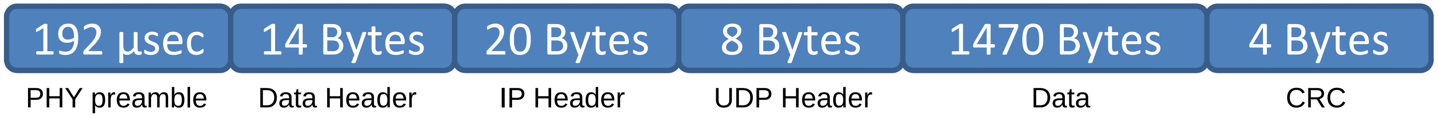


- Setup
  - Root sends schedule with three scheduling elements
  - One transmission slot for each node
  - Root node sends only schedule packet
  - Client communicate as per TDMA slot structure
  - Once schedule is over, clients will wait for next schedule
  - Root node will send schedule upon completion of current schedule

Results

Rate	Slot Size	SE	UDP (Mbps)	
(Mbps)			One-Dir	Bi-Dir
11	20 msec	3	7.52	7.53
11	10 msec	3	7.35	7.59
11	5 msec	3	7.20	7.36

- Observation
  - Theoretical throughput calculation

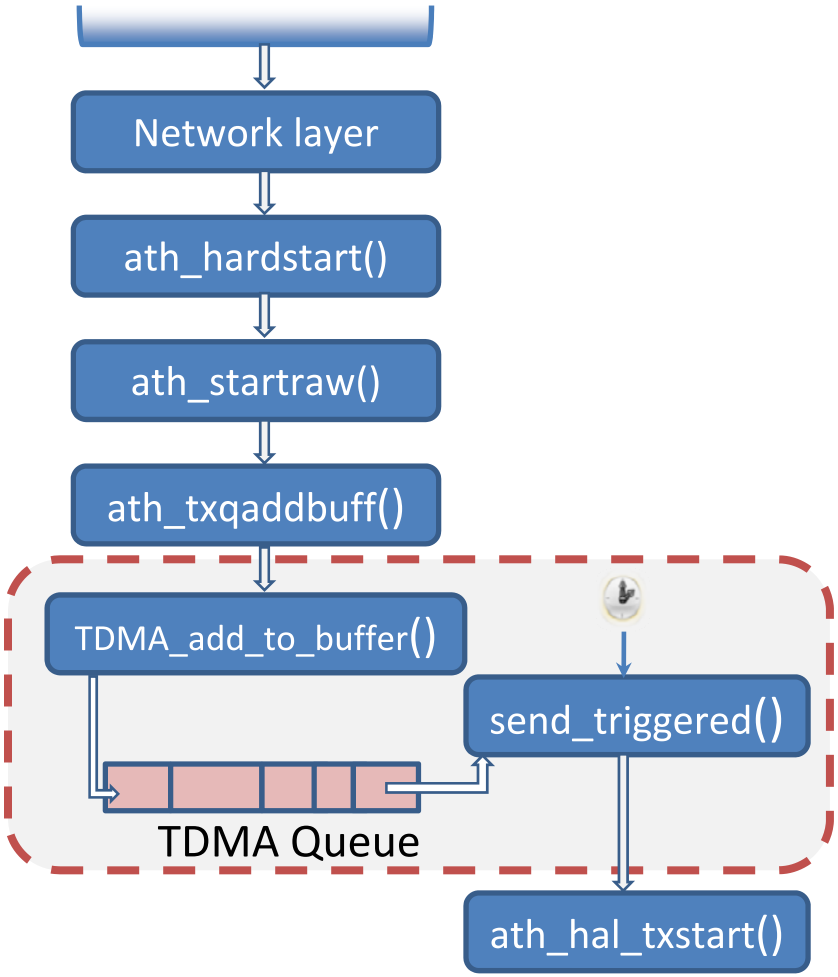


192 µsec	14 Bytes	20 Bytes	8 Bytes	1470 Bytes	4 Bytes
PHY preamble	Data Header	IP Header	UDP Header	Data	CRC

  - Sending 1470 Bytes required (1516*8/11) + 192 µsec + 20 µsec = 1314 µsec
  - Average Throughput = 1470*8/1314.55 µsec = 8.95 Mbps
  - Given alternate slot for transmission + one slot for root node, expected throughput should be 1/3 of 8.95 Mbps = 2.98 Mbps

- Conclusion
  - Either nodes are not obeying slot timing or problem with TDMA queuing mechanism

TDMA Queuing Mechanism


Network layer

ath_hardstart()

ath_startraw()

ath_txqaddbuff()

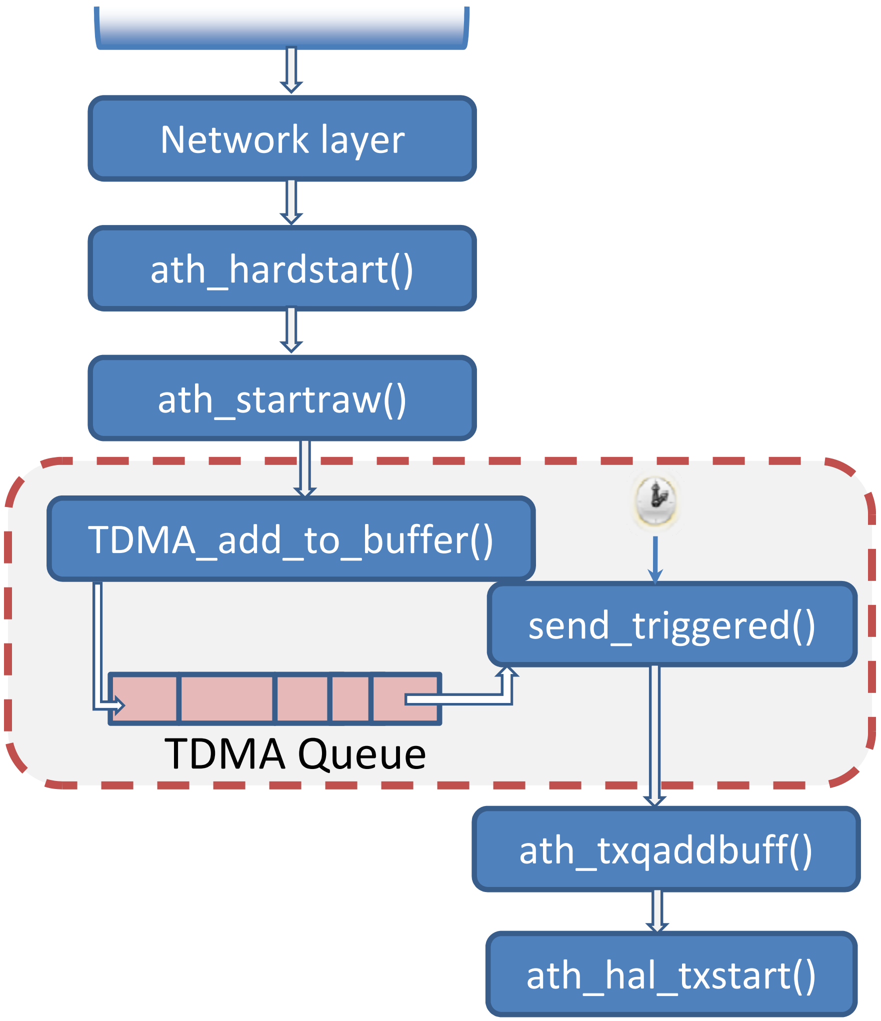
TDMA_add_to_buffer()

send_triggered()

TDMA Queue

ath_hal_txstart()

Current Implementation


Network layer

ath_hardstart()

ath_startraw()

TDMA_add_to_buffer()

send_triggered()

TDMA Queue

ath_txqaddbuff()

ath_hal_txstart()

Correct Implementation

Work Done

- Stage 1
  - Understanding madwifi driver
  - Understanding Transmit and Receive path in monitor and Ad-hoc mode
  - Prototype TDMA implementation in Ad-hoc mode
- Stage 2
  - Monitor mode communication
  - Disabling CSMA mechanism
    - Disabling MAC-ACK
    - Disabling RTS/CTS
    - Raw packet transmission
    - Disabling random back off
    - Disabling virtual carrier sensing
    - Disabling CCA
  - Generating packets at MAC layer
  - Enabling channel switching
  - Implementing TDMA frame structure
  - Raw packet transmission; no 802.11 frames

Timeline For Stage 3

- Fixing TDMA queuing mechanism
- Design and Implementation of
  - Multi-hop packet forwarding
  - Schedule dissemination across network
  - Node join, flow request and bandwidth allocation
- Testing
  - Indoor and outdoor benchmarking

References

- Kameswari Chebrolu and Bhaskaran Raman. FRACTEL: A Fresh Perspective on (Rural) Mesh Networks, ACM SIGCOMM Workshop on Networked Systems for Developing Regions (NSDR'07), A Workshop in SIGCOMM 2007, Aug 2007, Kyoto, Japan.
- Bhaskaran Raman and Kameswari Chebrolu. Design and Evaluation of a new MAC Protocol for Long-Distance 802.11 Mesh Networks, MOBICOM, Aug/Sep 2005, Cologne, Germany
- S. Pavan Kumar. Design, Implementation, and Evaluation of new MAC Protocols for Long Distance 802.11 Networks. IIT Kanpur, May 2006, Master Thesis.
- Michale Neufeld, Jeff Field, Christian Doerr, Anmol Sheth and Drik Grunwald. softMAC-Flexible Wireless Research Platform, HotNets-IV, Nov 2005.
- Ashish Sharma, Mohit Tiwari, Haitao Zheng. MadMAC: Building a Reconfigurable Radio Testbed Using Commodity 802.11 Hardware, WSDR 2006.
- Christian Doerr, Michael Neufeld, Jeff Field, Troy Weingart, DC Sicker, Dirk Grunwald. MultiMAC – An Adaptive MAC Framework for Dynamic Radio Net working, DySPAN 2005.
- Ashish Sharma, Elizabeth M. Belding FreeMAC:Framework for Multi-Channel MAC Development on 802.11 Hardware, PRESTO 2008
- Ananth Rao, Ion Stoica. An Overlay MAC Layer for 802.11 Networks, MobiSys 2005.
- Rabin Patra, Sergiu Nedevschi, Sonesh Surana, Anmol Sheth, Lakshminarayanan Subramanian, Eric Brewer. WiLDNet: Design and Implementation of High PerformanceWiFi Based Long Distance Networks, NSDI 2007
- E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The Click Modular Router. TOCS 2000.
- http://www.madwifi.org
- http://www.soekris.com/

Thank You!

Changes in more detail

- Disabling MAC-ACK
- Disabling RTS/CTS
- Raw packet transmission

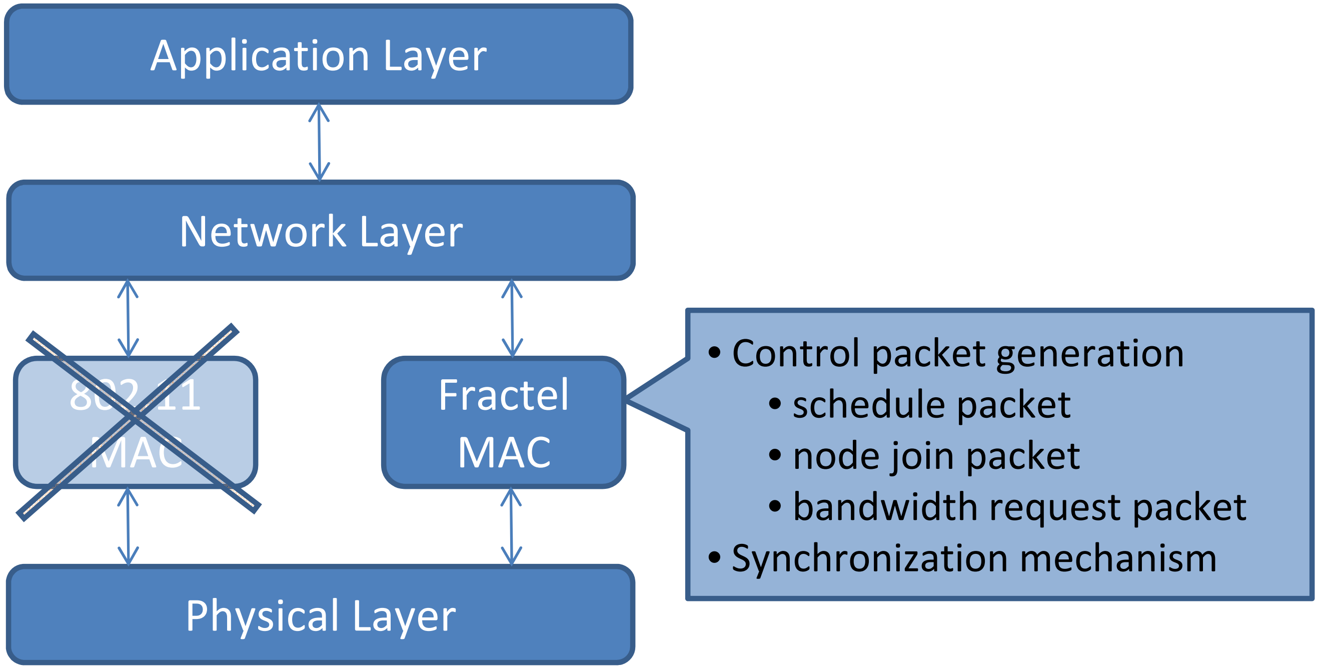
By putting in Monitor mode

- Disabling random back off – Partly By Ashutosh
  - By Setting CWmin and CWmax to 1 and setting HAL_TXQ_BACKOFF_DISABLE flag of hardware queue
- Disabling virtual carrier sensing
  - Setting NAV field to zero
- Disabling CCA – By Ashutosh
  - Setting noise floor to high value such that channel is always sensed free

Changes in more detail

- Packet send/receive in monitor mode
- Generation of control packets at MAC layer (in monitor mode)
- Enabled channel switching from driver code
- Packet Filtering based on destination MAC address and discarding packet with CRC and PHY error
- Plugged in TDMA schedule header, data header and scheduling elements

Protocol Stack



Application Layer

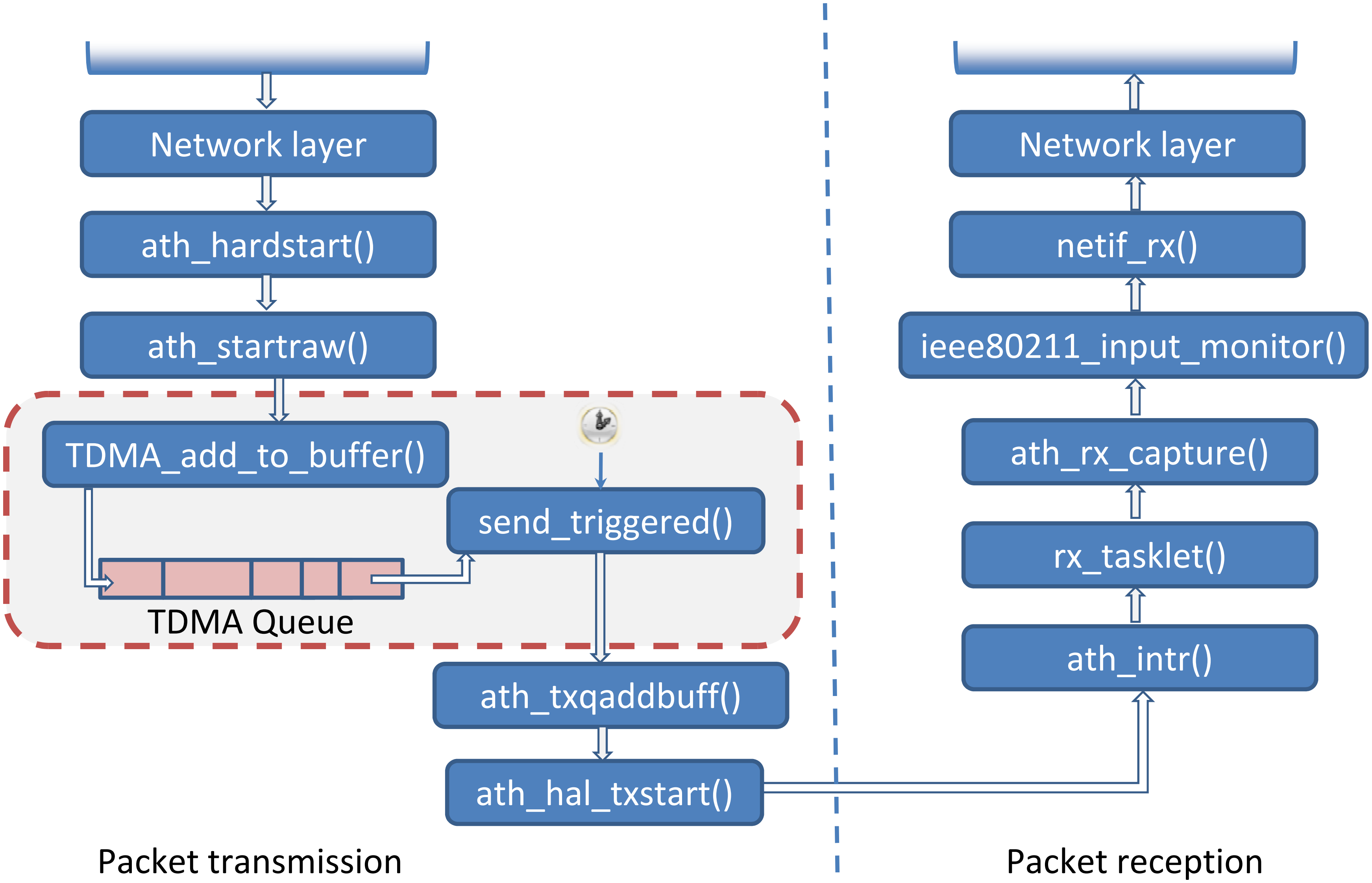
Network Layer

~~802.11 MAC~~

Fractel MAC

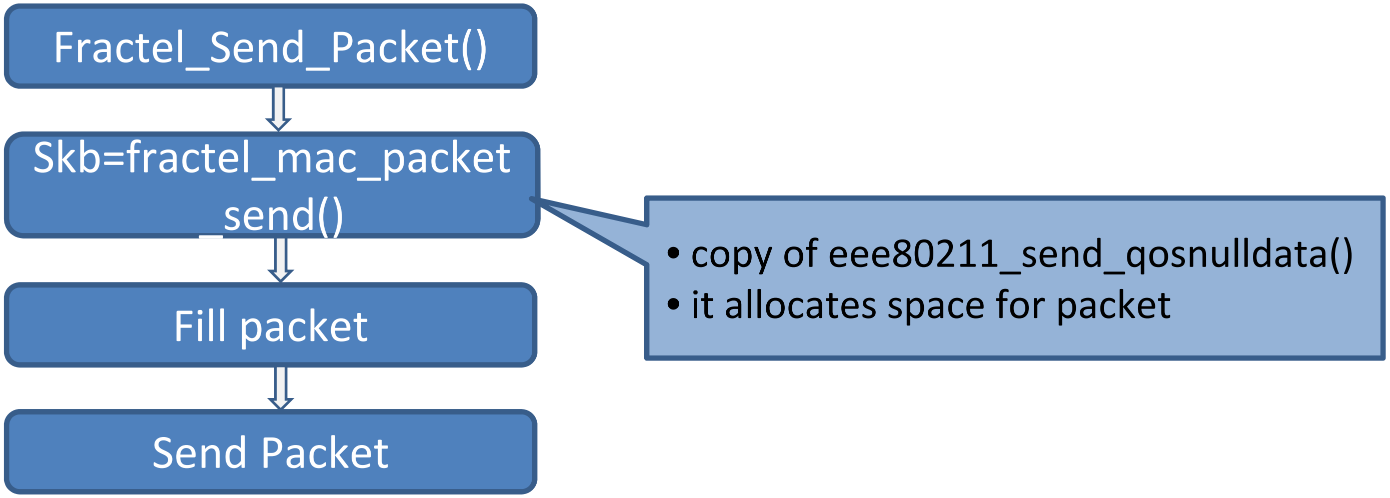
Physical Layer

- Control packet generation
  - schedule packet
  - node join packet
  - bandwidth request packet
- Synchronization mechanism


Network layer
ath_hardstart()
ath_startraw()
TDMA_add_to_buffer()
send_triggered()
TDMA Queue
ath_txqaddbuff()
ath_hal_txstart()
Packet transmission
Network layer
netif_rx()
ieee80211_input_monitor()
ath_rx_capture()
rx_tasklet()
ath_intr()
Packet reception

Generating control packet at MAC layer

- Packet generation at MAC layer to remove additional delays in generation from upper layer



- copy of eee80211_send_qosnulldata()
- it allocates space for packet

Hardware and Software

- 233 MHz soekris board with 256MB HDD and 64MB RAM running voyage Linux
- Atheros Wi-Fi chipset AR5213A
- Open source Madwifi 0.9.4 wireless driver
- Directional Antennas