MTP First Stage

To Design, Implement and Evaluate Multi-Hop TDMA System

By Nirav Uchat

Guide: Prof. Kameswari Chebrolu

And

Prof. Bhaskaran Raman

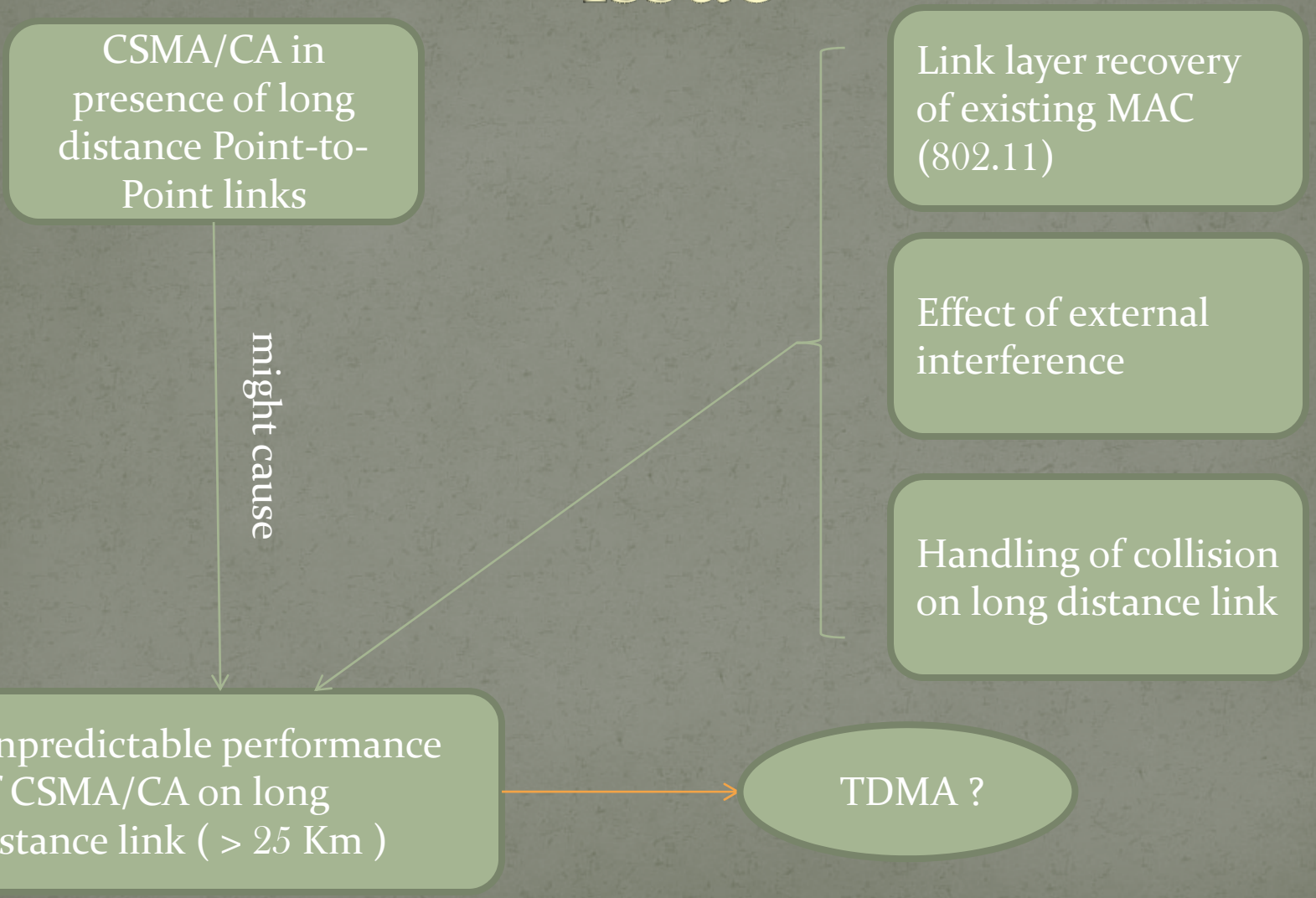
Motivation

- Digital inclusion of remote villages
- Giving access to WWW, to the people leaving in wide geographically distributed area
- Use of open source driver and off the shelf Wi-Fi hardware
- Vested interest of telecom giants

Existing Live system

- In mountain region of Venezuela, they claim to have achieved Wi-Fi link more than 300 km long between two device.
- They have used WiLD framework
- Arvind Eye Clinic in India uses same framework for video conferencing between doctor and patient
- It has more than 6 node

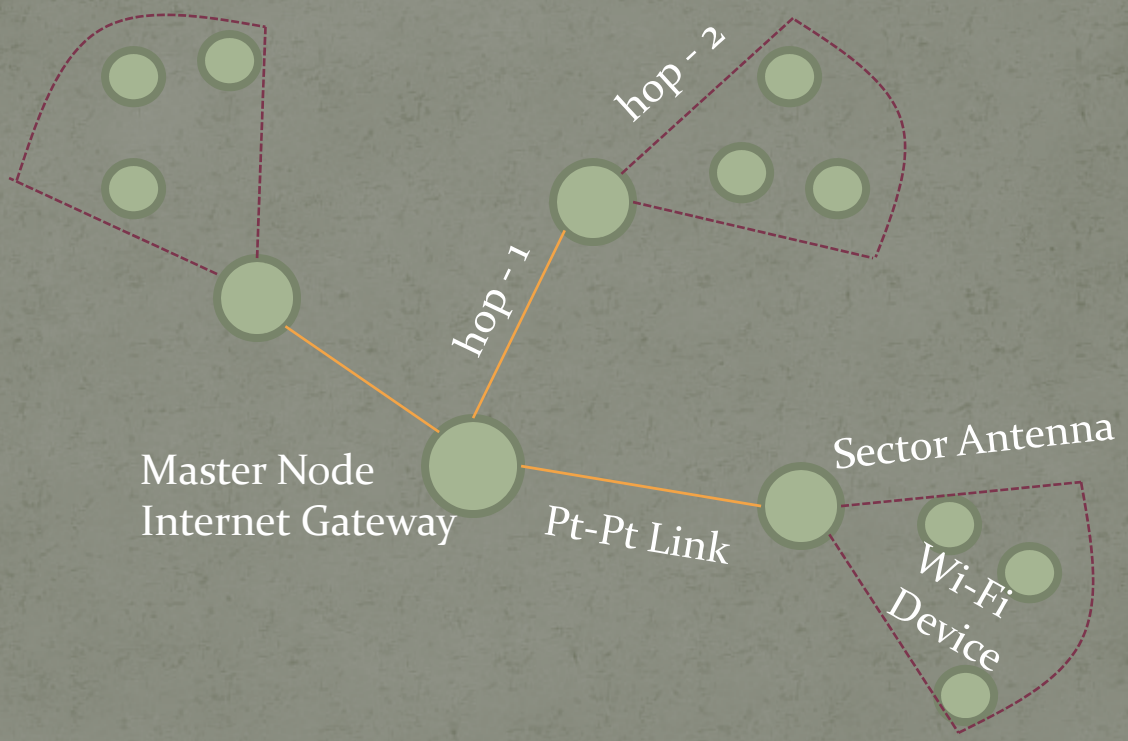
Issue


CSMA/CA in presence of long distance Point-to-Point links
might cause
Link layer recovery of existing MAC (802.11)
Effect of external interference
Handling of collision on long distance link
Unpredictable performance of CSMA/CA on long distance link (> 25 Km)
TDMA ?

Problem Statement

- Design Multi-Hop TDMA system
- It should work for both short and long distance network
- It should support
  - HTTP
  - FTP
  - Audio & Video
- Modified TDMA code should run on any wired or Wi-Fi device

Architecture



All Wi-Fi node will run modified TDMA code

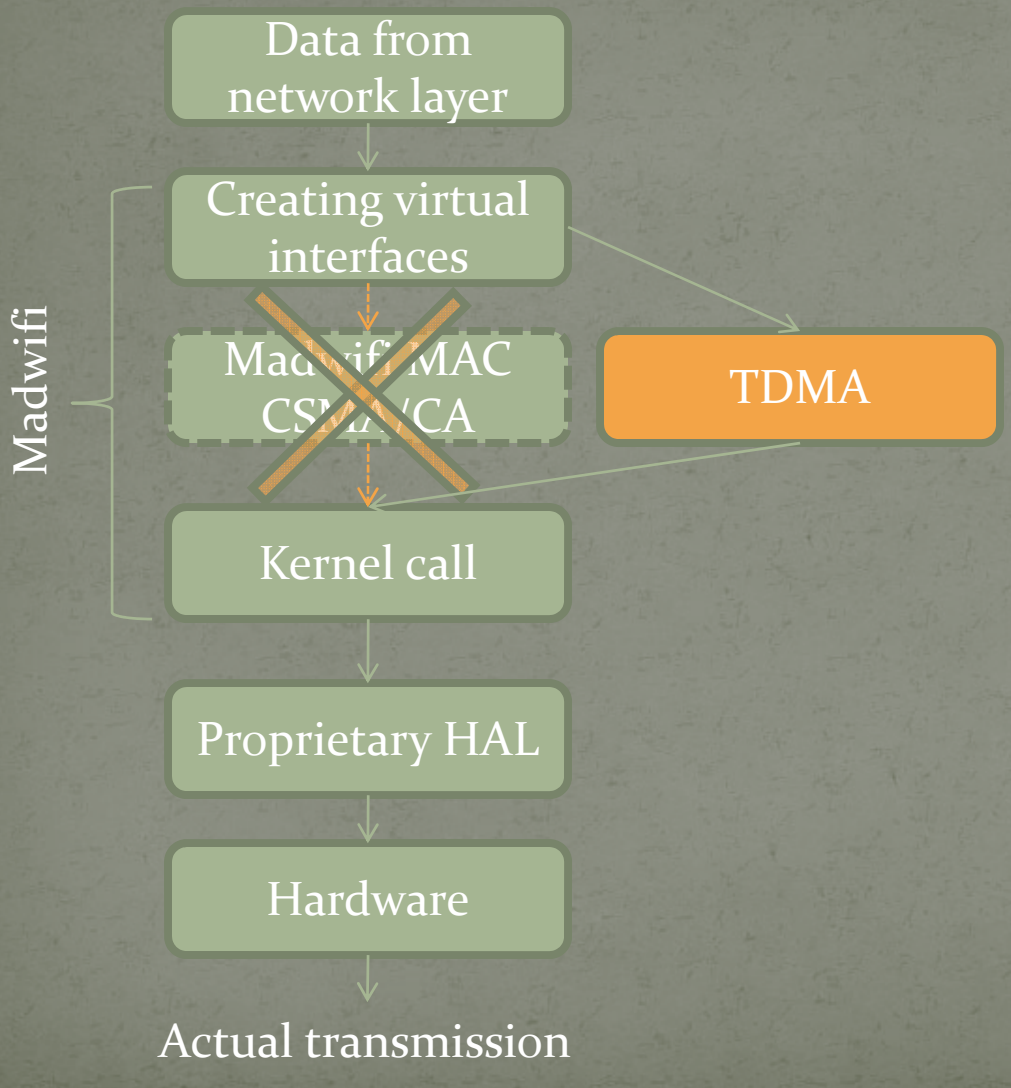
&

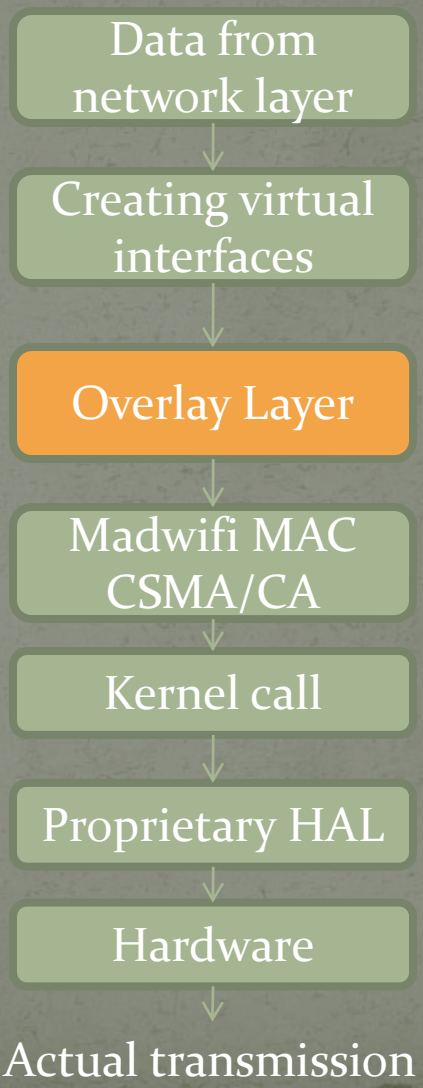
All wired device will send data to Wi-Fi device

Hardware and Software

- 233 MHz soekris board with 256 MB HDD and 64 MB RAM running voyage Linux
- Atheros Wi-Fi chipset AR5213A
- Open source Madwifi 0.9.4 wireless driver
- Linux kernel 2.6.25.4

MAC or Overlay Layer ?





MAC or Overlay Layer ?

- Implementation at MAC layer gives direct access to hardware through HAL
- At MAC layer we can control packet transmission timing
- Overlay layer works on top of MAC layer and can use functionality exposed by MAC
- Overlay layer has no control over actual packet transmission
- We decided to use MAC layer approach for our TDMA implementation

Previous work

- softMAC suggest way to disable CSMA/CA
- MadMAC uses custom frame format for tight synchronization
- MuliMAC uses multiple MAC depending on working condition (E.g. CSMA for short distance and TDMA for long distance)
- FreeMAC uses beacon hardware timer of Atheros chipset for better control over packet transmission
- But all has static TDMA schedule and only works for single hop network (mostly between two device)

Our approach and Issue

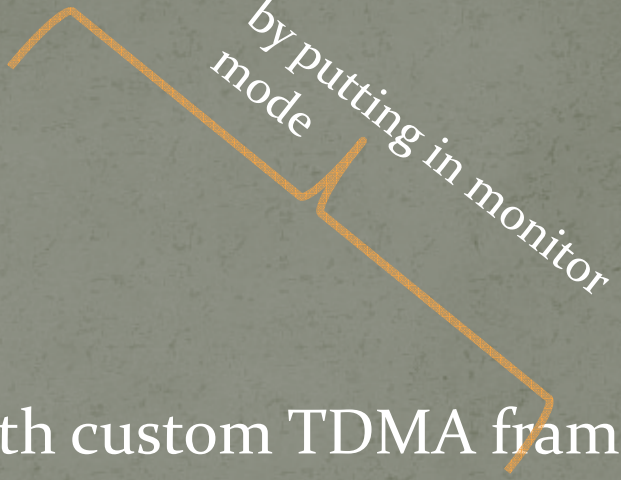
- Disabling CSMA/CA functionality
- Multi-Hop TDMA
- Dynamic TDMA Schedule depending on network load
- Custom frame format
- TDMA schedule dissemination
- As of now, single MAC protocol I.e. TDMA at MAC layer

Madwifi

- Five different mode to work with
  - Adhoc, Monitor, Sta, AP and Ahdemo
- Monitor mode
  - Disables
    - Mac layer ACK's
    - RTS/CTS exchange
    - 802.11 frame format
  - Enables
    - Transmission of custom frame format
- We decided to use monitor mode for our implementation

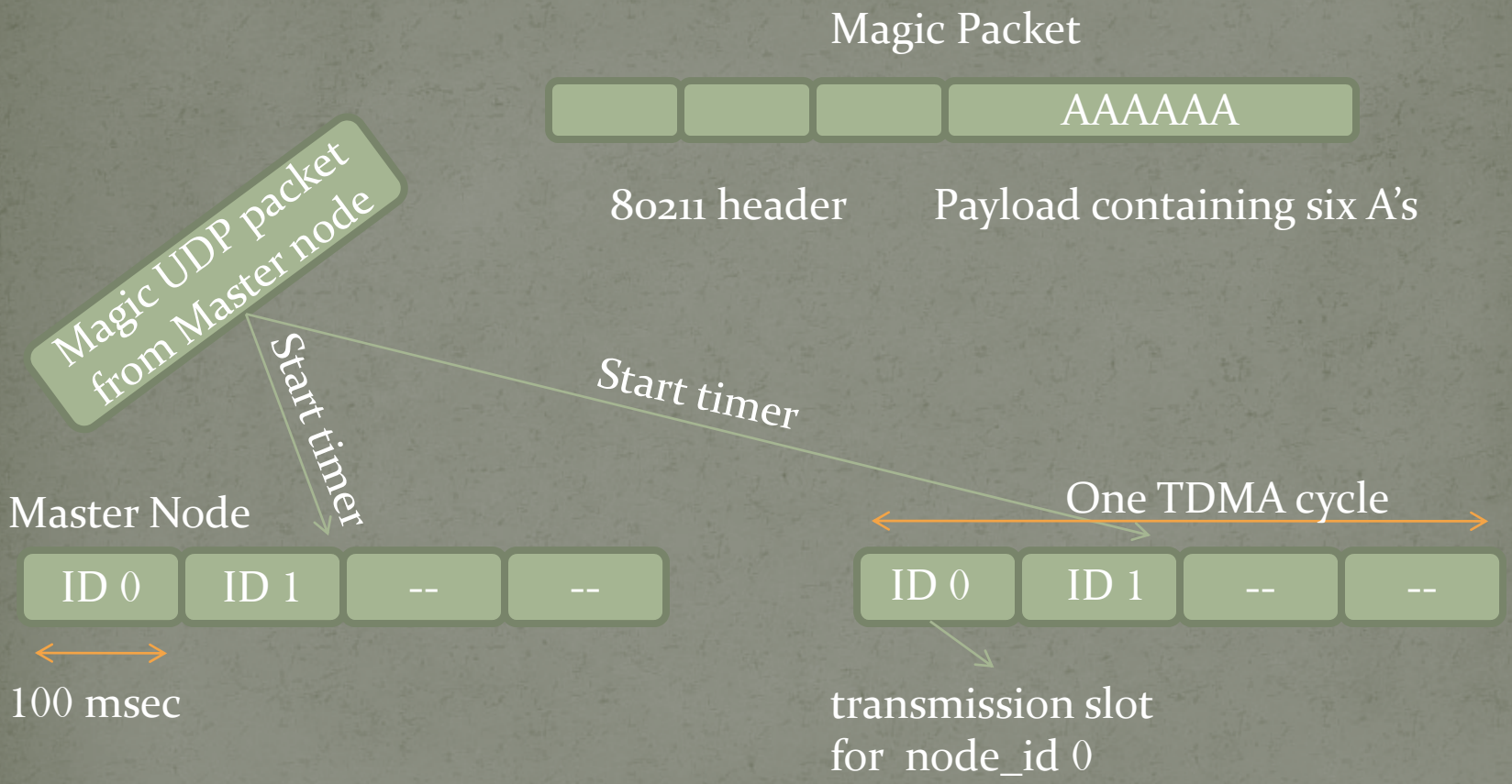
Six steps to disable CSMA/CA

- Disabling MAC level ACK
- Disabling RTS/CTS exchange
- Override 802.11 frame format with custom TDMA frame
- Disable virtual carrier sensing
- Disable Transmission back off
- Disable CCA (clear channel assessment)



Transmission and Reception path in Adhoc mode

First stage work: Prototype TDMA

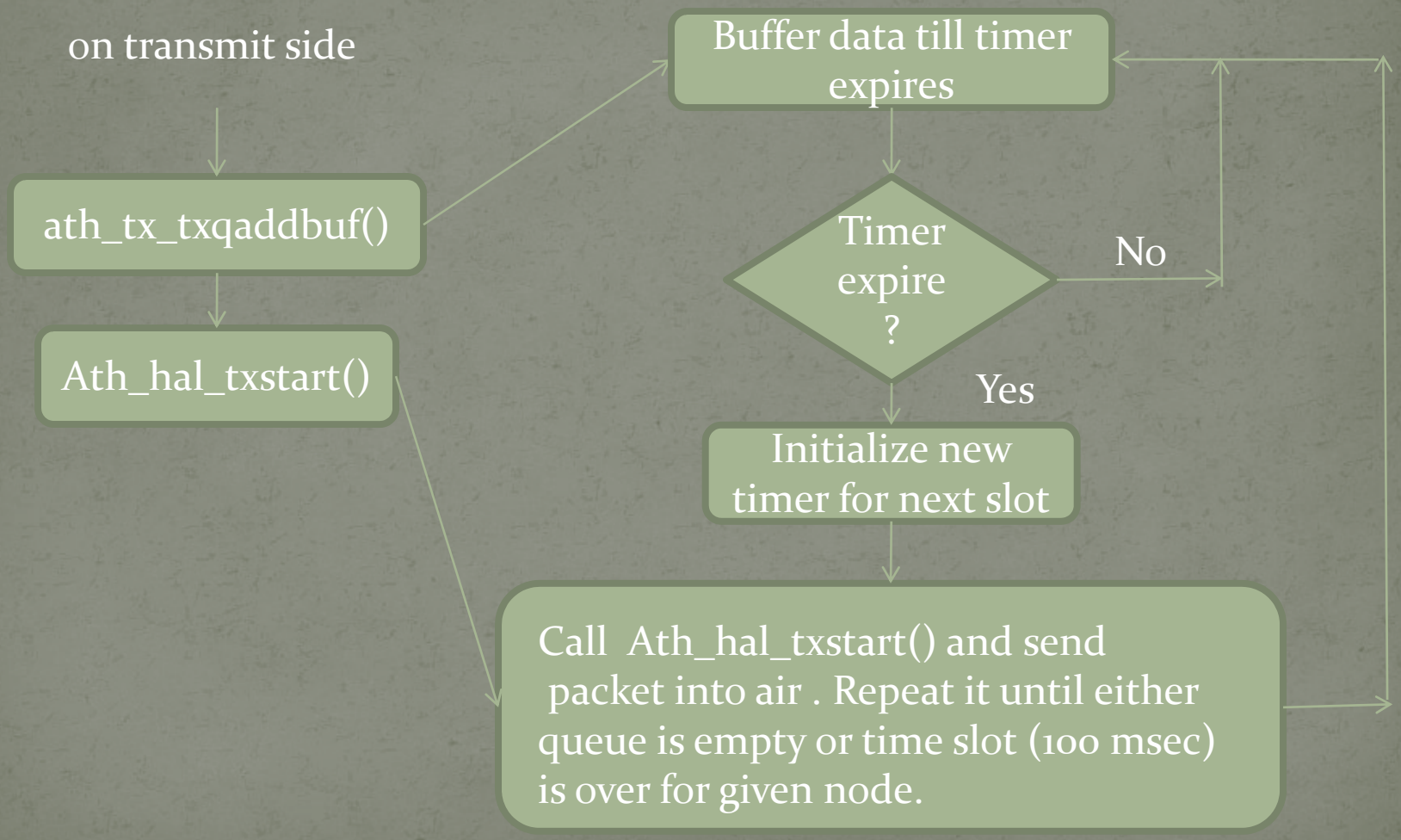


Timer expires at 1000+ node_id * 100

Timer for node_id 0 will expire at 1000 msec after sending magic packet

Timer for node_id 1 will expire at 1100 msec after receving magic packet

Chnages in Madwifi Code



Chnages in Madwifi Code

- On receiver side
  - Start timer on receiving Magic packet
  - Timer will expire at 1000+node_id * 100 msec
- There is not much change in receiving code, on receiving data, Madwifi will strip 802.11 header and will pass data to network layer
- Note : On sending side, we disable receive interrupt just before sending data and enables it before just after end of transmission slot.

Problem faced in First Stage

- We implemented TDMA in presence of CSMA/CA
- In monitor mode we were not able to pass data beyond MAC layer
- Prototype TDMA has very little time synchronization
- Out of Six steps in disabling CSMA/CA , we were able to accomplish first three

Time Line for Second Stage

- Implementing prototype TDMA in monitor mode
- Coming up with TDMA frame structure
- Extending to Multi-hop network
- TDMA schedule dissemination
- We might look at dynamic TDMA schedule

References

- Michale Neufeld, Jeff Fifield, 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 Fifield, Troy Weingart, DC Sicker, Dirk Grunwald.MultiMAC - An Adaptive MAC Framework for Dynamic Radio Networking, DySPAN 2005.
- Ashish Sharma, Elizabeth M. BeldingFreeMAC: 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
- Nelson, R., Kleinrock, L. Spatial TDMA - A collision-free multihop channel access protocol, IEEE Transactions on Communications, vol. COM-33, Sept. 1985, p. 934-944
- 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
- www.eslared.org.ve/articulos/Long Distance WiFi Trial.pdf\bibitem{key:arvind}http://tier.cs.berkeley.edu/wiki/Aravind

Thank You