Implementation of WiFiRe MAC protocol

M.Tech Project defense

Ranjith Kumar Madalapu
(05329R08)

Under the guidance of
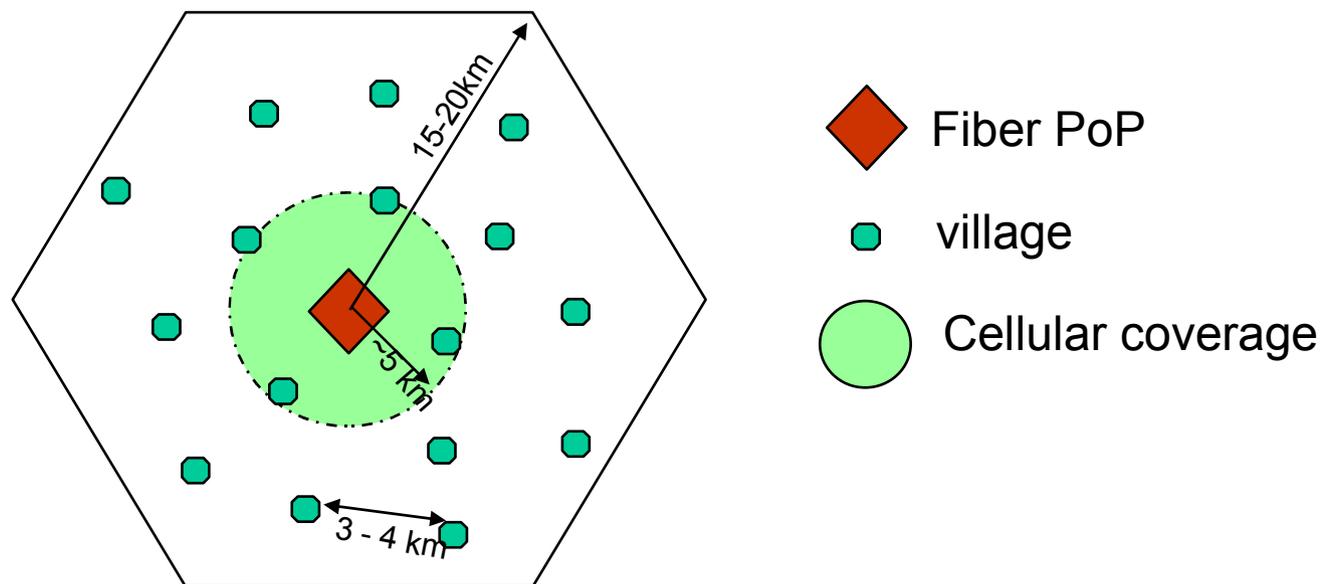## Prof. Sridhar Iyer and Prof. Anirudha Sahoo

Department of Computer Science
IIT Bombay

July 11th

Outline

- Introduction and Motivation

- WiFiRe Protocol

- Problem statement

- LAN Emulation

- Implementation Modules

- Experiments - Demo

- Learnings

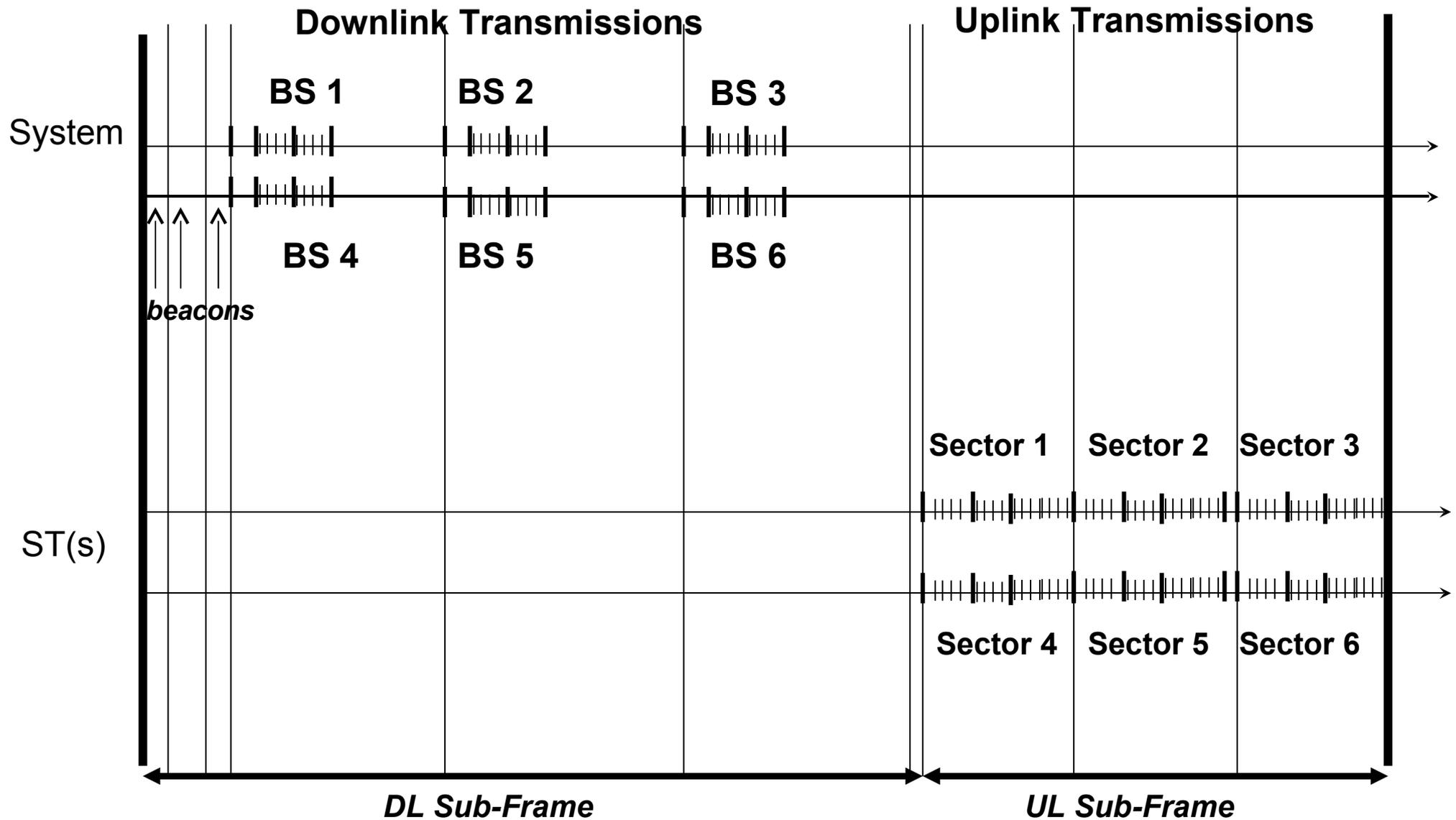- Conclusions

Introduction and Motivation

- 250-300 villages per PoP
- Wired connectivity not possible in many cases
- Installing more BS, costlier
- Technology requirements
  - Long range, Low cost, Broadband Wireless Access (BWA)
  - Guaranteed QoS for VoIP traffic
- Aim: Data and voice connectivity in remote villages

Fig. Ref: WiFiRe draft [1]

Introduction and Motivation (Cont.)

- WiFi (802.11)
    - Free licensed spectrum (2.4GHz),
    - Off-the-self chipset
    - MAC access, CSMA/CA doesn't support for long distances
- WiMAX (802.16)
    - Support Long Ranges (30 to 40km)
    - Efficient MAC, PHY is too costly
- WiFiRe : WiMAX like MAC on top of WiFi PHY

WiFiRe - Architecture

- Star topology(15km radius)

- BS at fiber PoP

- 40m tower at BS with sectorized antenna

- 10m pole at ST with directional antenna

- Single WiFi channel shared among all the sectors

- 25Mbps data rate (UL and DL)

Ref: WiFiRe draft [1]

Frame format - (TDD-MSTDM)
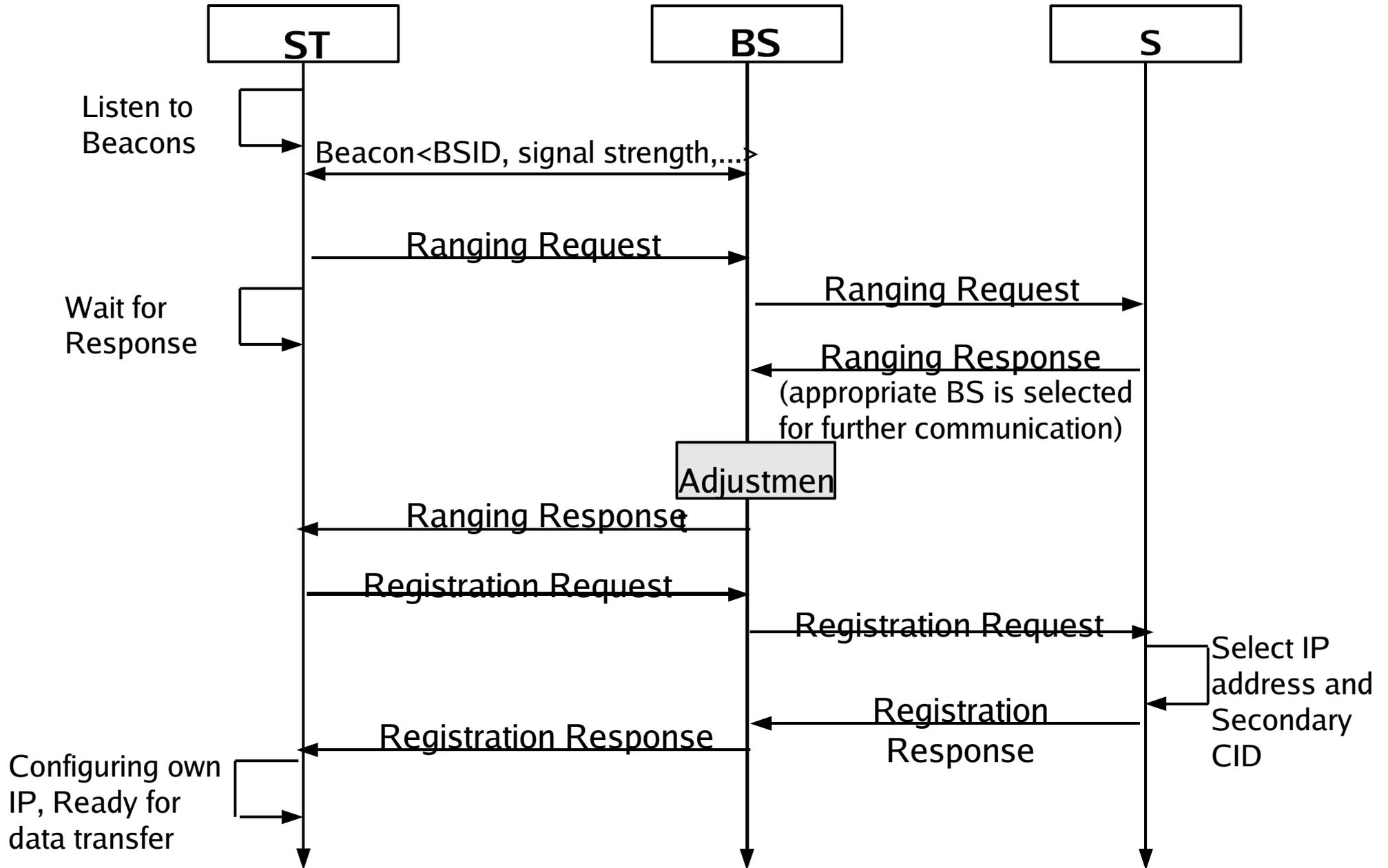

Ref: WiFiRe draft [1]

Network Initialization

Fig. Ref: WiFiRe draft [1]

Problem Statement

- Implementation of single sector WiFiRe MAC

  – Single BS, multiple STs and Clients under each ST

- Previous work done

  – Design of WiFiRe

  – Testbed prototype (single ST)

  – Suggested improvements in draft

- Present work done

  – Testbed with multiple STs in a single sector

  – Implementing all required modules

  – End-to-End connectivity

  – Provide different services ex: HTTP, VoIP, FTP etc.

WiFiRe LAN Emulation

- MAC and PHY are two separate entities
- WiFiRe PHY is under development (IITM)
- Simulation of MAC using OPNET is done already
- To test MAC using existing infrastructure
- MAC to be ported on PHY

Basic Testbed setup

Client

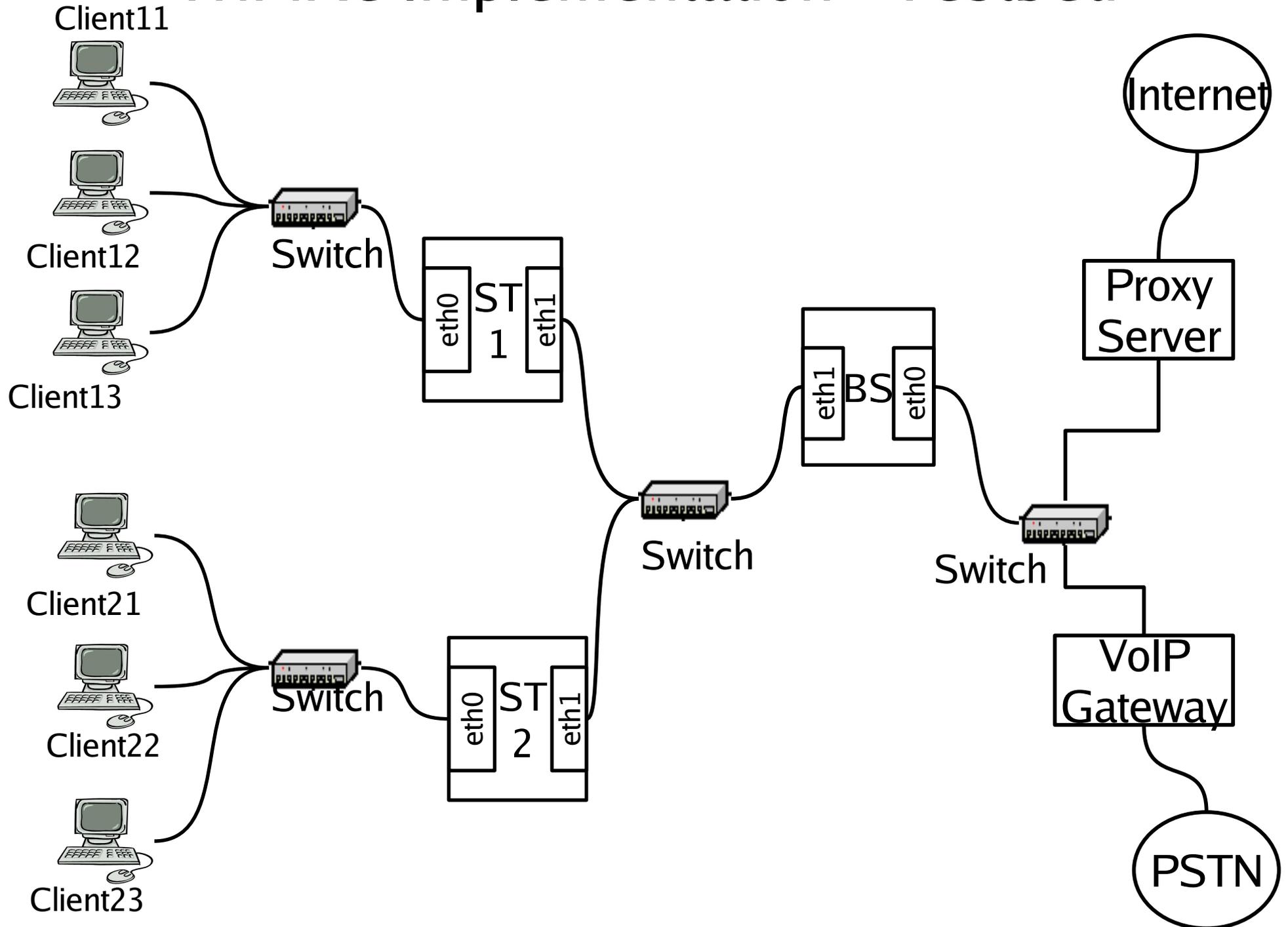Client

Client

Switch

eth0 ST eth1
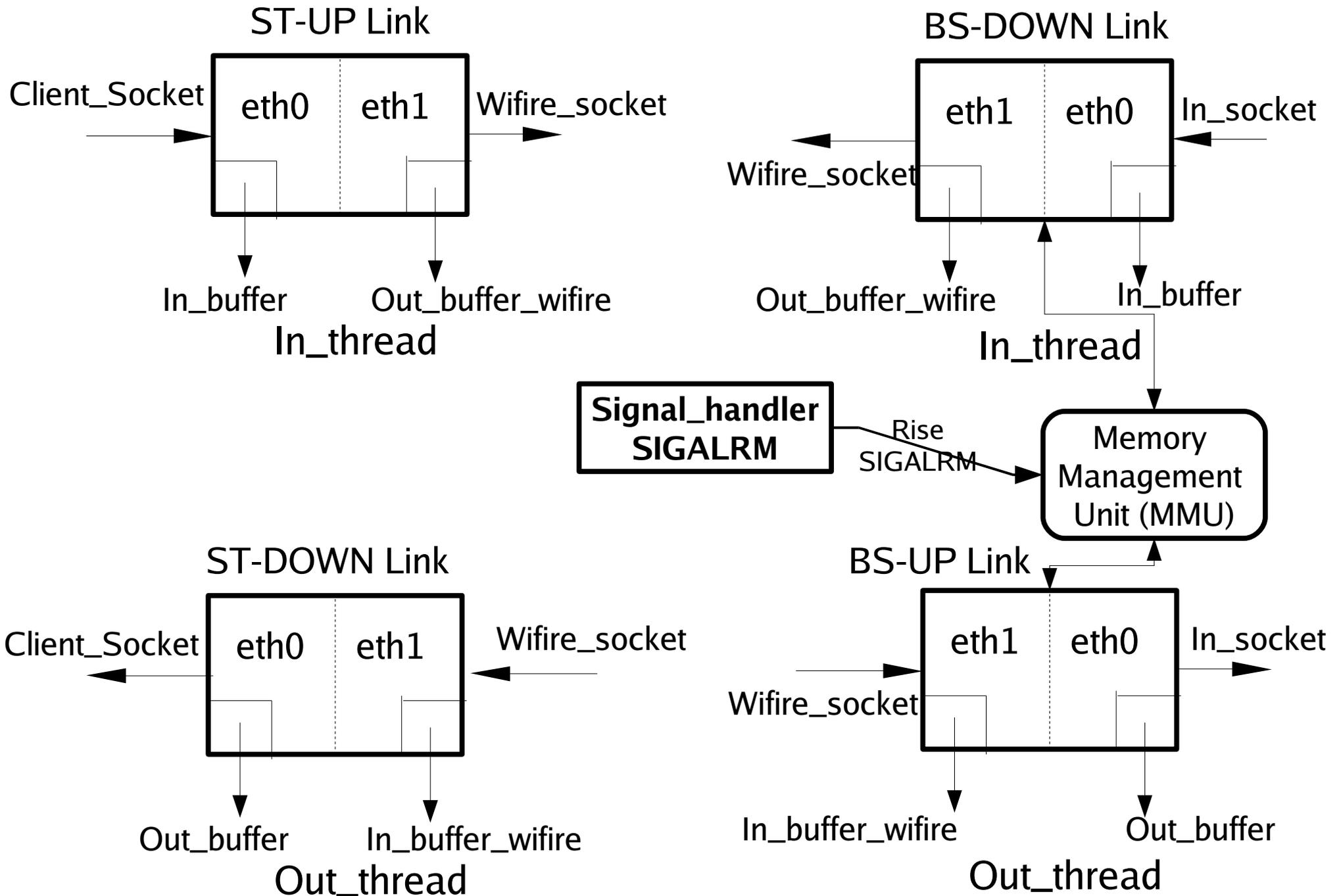
Cross Cable

eth1 BS eth0
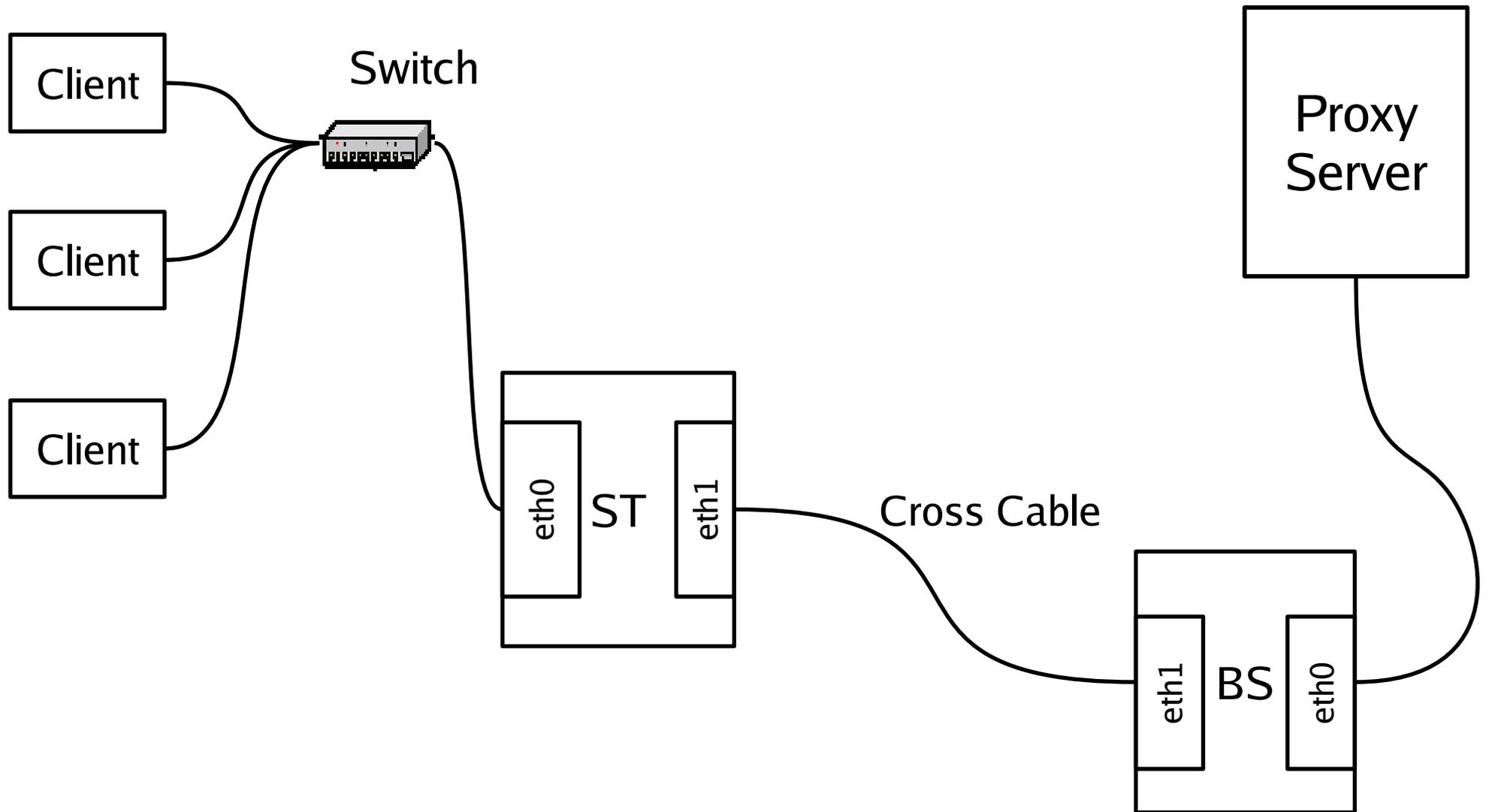
Proxy
Server

LAN emulation over Ethernet

- Single BS, multiple STs and Clients
- BS and ST are layer-2 device
- Clients are connected to ST using Ethernet switch
- BS is connected to Proxy using 802.3
- We emulate WiFiRe link between BS and ST
- Propagation delay between BS and ST is negligible

WiFiRe Implementation - Testbed

Threads, Sockets and Buffers

ST-UP Link

Client_Socket → | eth0 | eth1 | → Wifire_socket

In_buffer   Out_buffer_wifire

In_thread

BS-DOWN Link

Wifire_socket ← | eth1 | eth0 | ← In_socket

Out_buffer_wifire   In_buffer

In_thread

Signal_handler SIGALRM — Rise SIGALRM → Memory Management Unit (MMU)

ST-DOWN Link

Client_Socket ← | eth0 | eth1 | ← Wifire_socket

Out_buffer   In_buffer_wifire

Out_thread

BS-UP Link

Wifire_socket → | eth1 | eth0 | → In_socket

In_buffer_wifire   Out_buffer

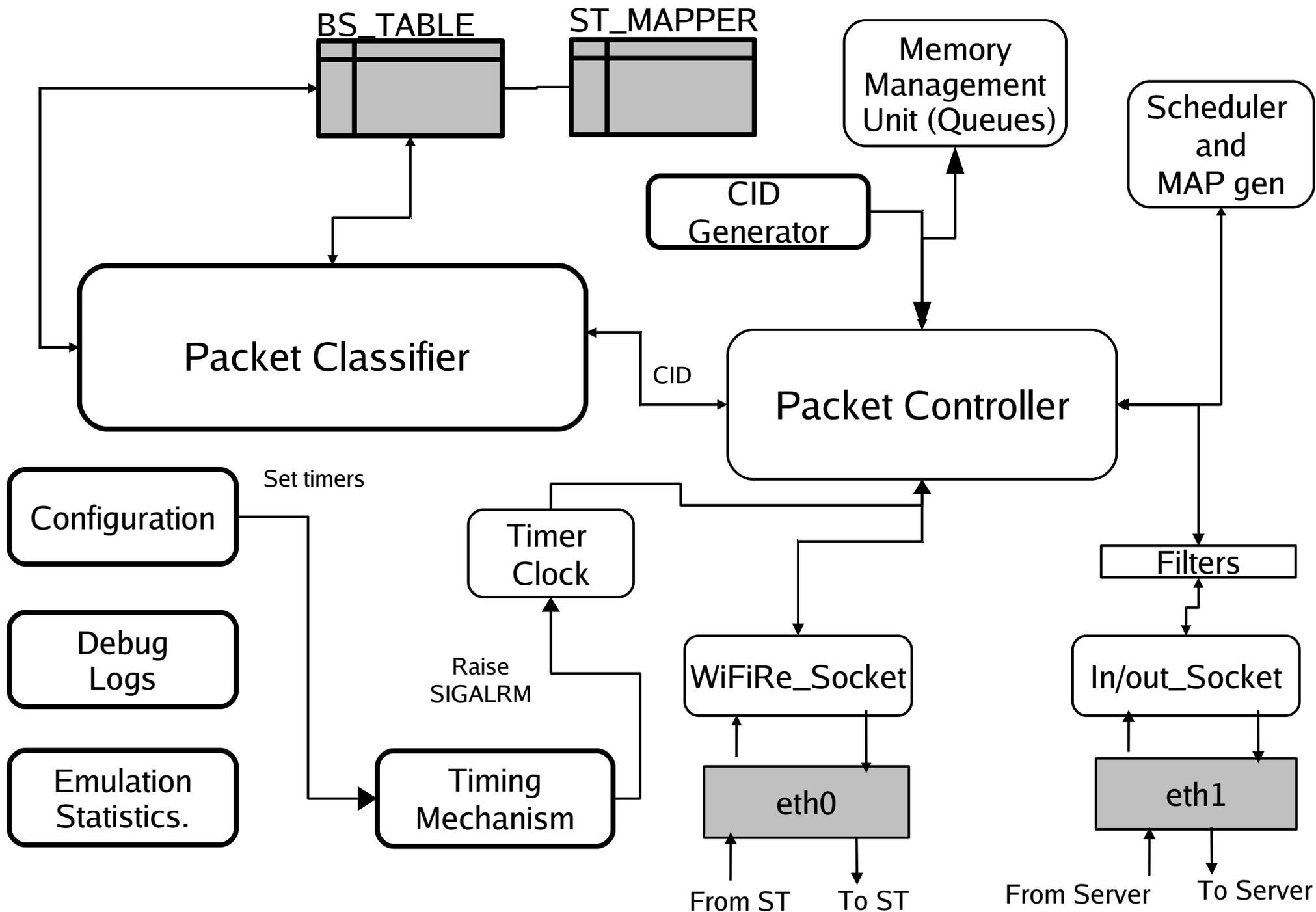Out_thread

Ethernet sockets

- Using PF_SOCK in C

- Binding with particular NIC,Byte level access

- Send/receive data using sockets on MAC layer

- Allows non-Ethernet packets (like WiFiRe frame); Ethernet switch broadcast those packets, Ethernet MAC header not mandatory

- Why Socket and not kernel?

  - Socket itself supports lower level access
  - Kernel module handles various issues, ex: traps, memory etc. which are not important/relevant
  - Debugging is easier
- PCAP?

BS Modules

BS_TABLE

ST_MAPPER

Memory Management Unit (Queues)

Scheduler and MAP gen

CID Generator

Packet Classifier

CID

Packet Controller

Set timers

Configuration

Timer Clock

Filters

Debug Logs

Raise SIGALRM

WiFiRe_Socket

In/out_Socket

Emulation Statistics.

Timing Mechanism

eth0

eth1

From ST   To ST

From Server   To Server

ST Modules

ST_TABLE

Memory Management Unit (Queues)

Timing Mechanism

MAP Parser

Set timers

Timer Clock

Packet Classifier

Packet Controller

Raise SIGALRM

Emulation Stats.

Local Scheduler (FCFS)

Debug Logs

Configuration

Filters

Client_Socket

WiFiRe_Socket

eth0

eth1

From Client

To Client

From BS
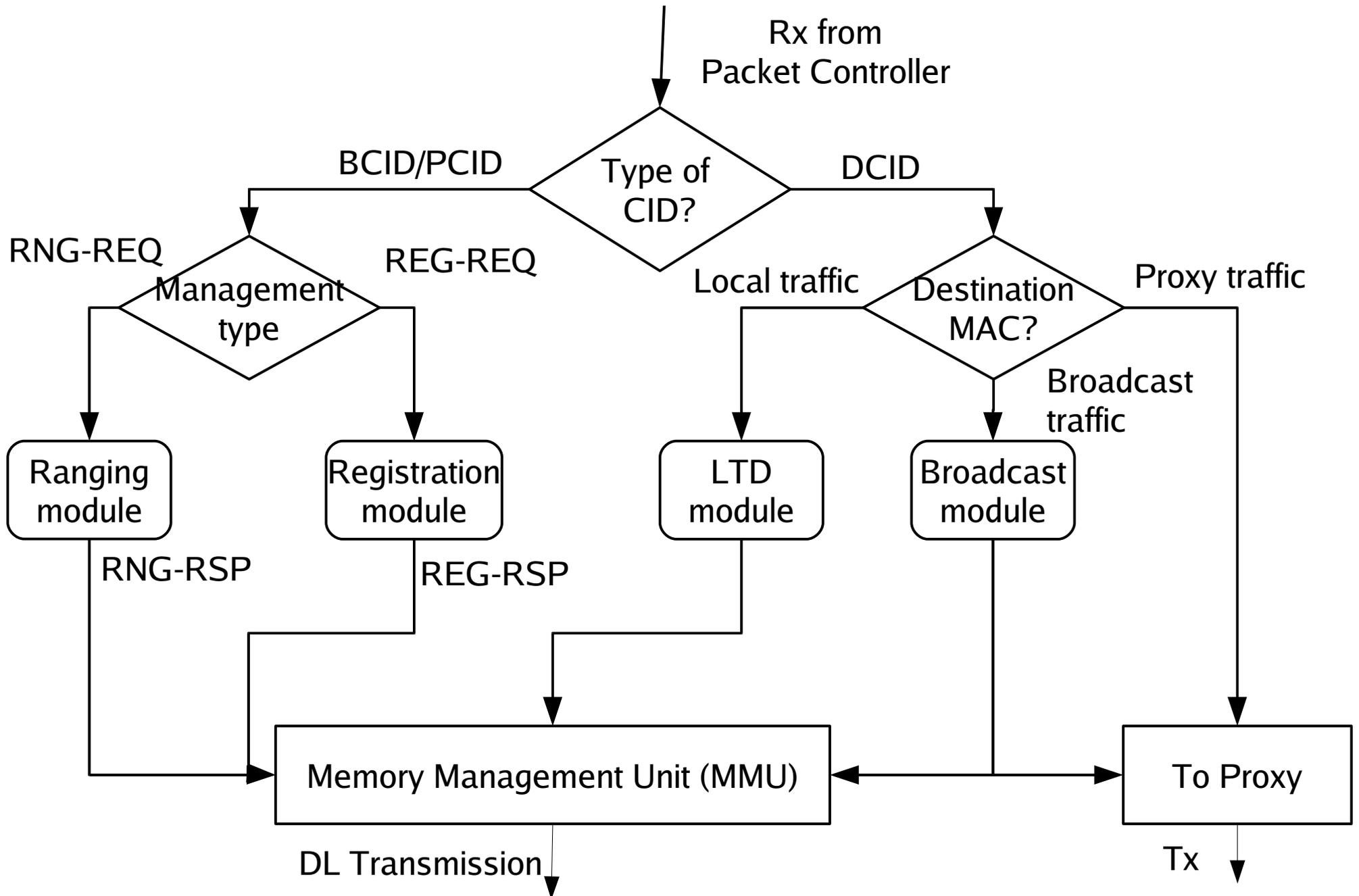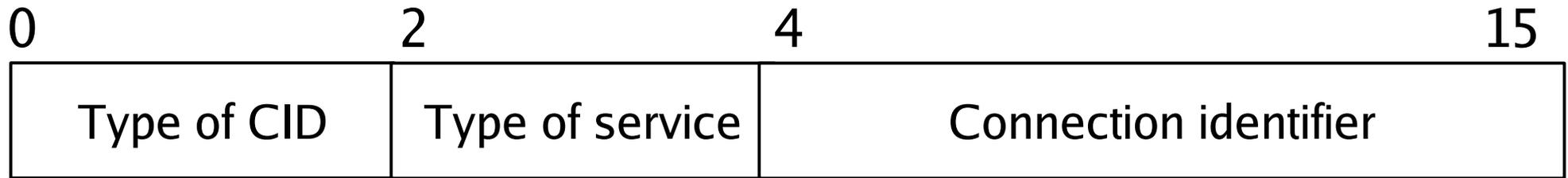
To BS

Packet Classifier at BS

Beacon, Ranging and Registration

- BS sends periodic beacons
  - When SIGALRM caught by *signal_handler*
  - Operational parameters are transmitted
  - [Operation-ID, System-ID, BSID, DLMAP, ULMAP]
- ST's ranging request to BS
  - ST receive beacon frame
  - ST send request
  - BS generate new BCID and PCID
  - Update ST_MAPPER table
- ST sends registration request
  - Client can not start communication before registration
  - BS/ST shutdown and restart conditions are handled

Beacon, Ranging, Registration etc.

ST sends request

CID generator

0	2	4	15
Type of CID	Type of service	Connection identifier	

- Type of CID
  - BCID (00), PCID(01), DCID(10 or 11)
- Type of Service
  - UGS(00), rtPS(01), nrtPS(10) and BE(11)
- New CID(type of CID, Type of srvice=11);
- Special CIDs
  - 0xffff, to Tx broadcast frames
- Single Data CID for ST

Tables for BS and ST

Filed Type	Size
STID	6
BSID	2
BCID	2
PCID	2

BS_TABLE (at BS)

Filed Type	Size
STID	6
Client ID	**6**
DCID	2

ST_MAPPER (at BS)

Filed Type	Size
Client ID	**6**
DCID	2

ST_TABLE (at ST)

- ST_TABLE updates when new client frame Rx at ST
- ST_MAPPER updates when new client frame Rx at BS
  - Read when data packet received
- BS_TABLE updates when new ST comes up
  - Read when management packet received

- Soft timer is used
  - Generate SIGALRM periodically
  - Activate Packet controller to prepare frame
- Debug logs levels
  - Keep track of execution of program
- Stats Module
  - Console interaction
  - To measure performance
- Configuration module
  - Parse the *config.wre* file
  - Set all configuration parameters
  - Set default values if not found

Configuration module

```
Extracting Values from Config file

                List of Config Values(config.wre)
proxy_mac                               :   0   8 a1 84 f7 9e
voip_gateway_mac                        :   0 11 95 87  e d2
bs2st_interface             : eth2
bs2proxy_interface                  : eth1
frame_length_milli                  : 10
mtu_threshold                       : 1400
max_buffer_length_per_st_in_pkts    : 100
max_st                      : 50
st_table_flush_time_in_sec          : 120
OPR_ID                      : 35
SYS_ID                      : 10
max_socket_buff             : 2048
debug_level                         : 1
                        WiFiRe System Emulation
-------------------------------------------------------------
Press 'S/s' to Display current Stats
Press 'R/r' to Re-Initialize Stats(ReSet)
Press 'A/a' to Display Advanced Stats
Press 'M/m' to Display current Frame MAP
Press 'X/x' to Exit from Emulation
Press 'h/H' for getting Help
-------------------------------------------------------------

NOTE: For detailed output follow the log file: output.txt

WiFiRe>
```

Configuration values

Set of commands

WiFiRe console & stats display

```
WiFiRe>s
                ST_TABLE entries from System side
                ---------------------------------
ST_ID(STMAC)            BSID    BCID    PCID
---------------------------------------------------------------------
 0 50 bf 63 94 1b          1       1    4001
---------------------------------------------------------------------

                BS_TABLE entries(List of Clients) from System side
                --------------------------------------------------
Client MAC                      STID
---------------------------------------------------------------------
 0  8 a1 85  2 5b               4001
---------------------------------------------------------------------

                        WiFiRe SYSTEM stats
---------------------------------------------------------------------
Current Time:(hh:mm:ss) = 15 : 47 : 11  Emulation Started at:(s):1215398964    Emulation Duration(s):26867
DPR_ID  : 35
SYS_ID  : 10
Bytes Tx ( DL ) in B    : 291075938
Bytes Rx ( UL ) in B    : 162564806
Pkts  Tx ( DL )         : 9010
Pkts  Rx ( UL )         : 11207
Data Bytes  Tx ( DL )           : 881822
Data Bytes  Rx ( UL )           : 881660
Frames Tx from System   : 2686648
Packets Dropped at System: 2206
ST Count        : 1
BS Count        : 1
Client Count    : 1
---------------------------------------------------------------------

NOTE: For detailed output follow the log file: output.txt

WiFiRe>█
```
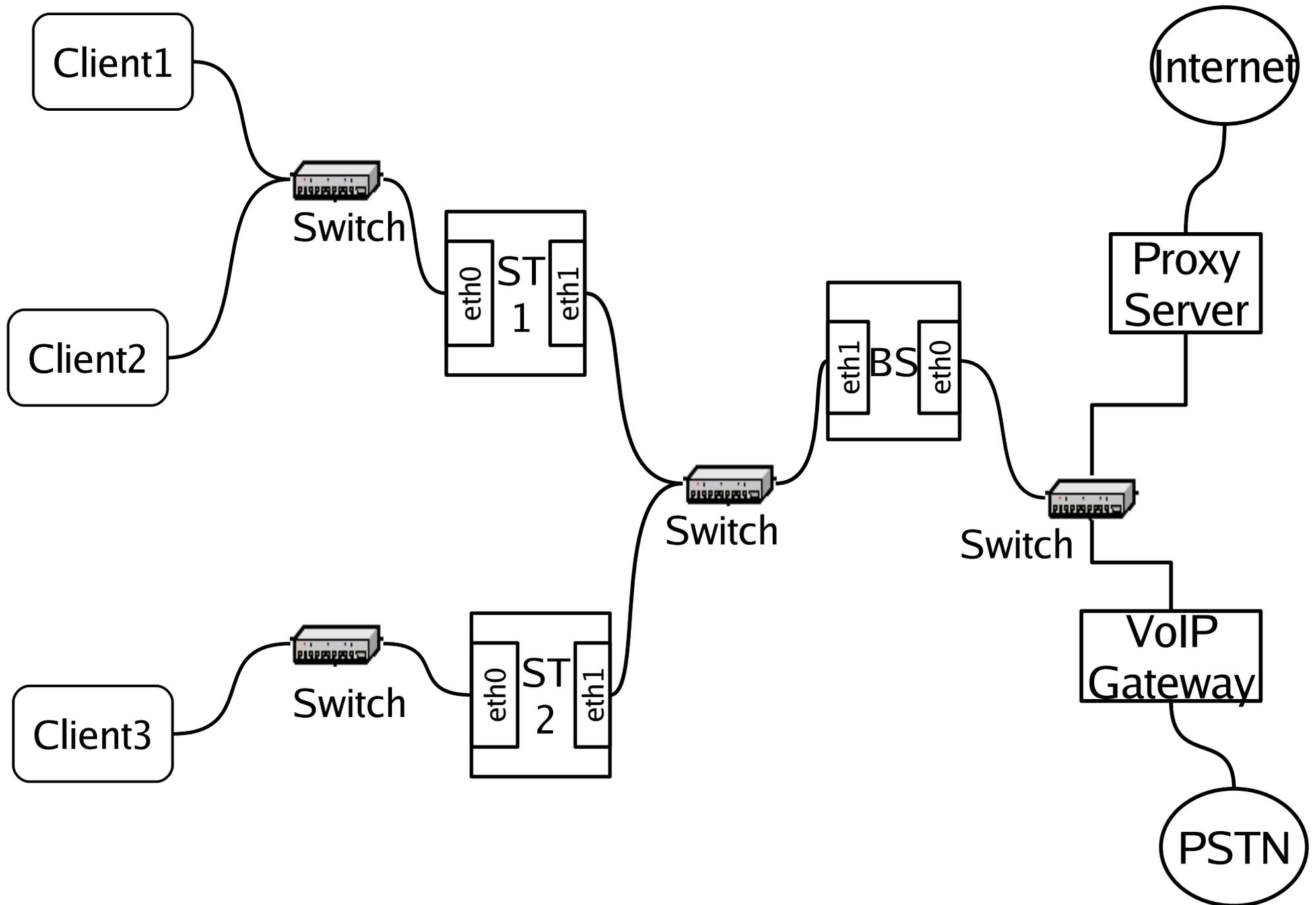
List of STs

List of Clients

Data Tx/Rx
stats from BS

Experiments – Demo, Testbed setup

Experiments - Demo

- Within Same ST
  - Ping between client and proxy
  - Accessing web pages
  - Bulk data download
- Between different ST
  - Ping between Client11 and Client21
  - VoIP call between Client11 and Client21
- VoIP call between Client and PSTN phone

WiFiRe Demo

Learnings

- ARP Cache flush
- Moving from 32-bit to 64-bit machines
- Problem with Non-WiFiRe packets
- Problem with multiple DHCP servers
- Segmentation fault

Conclusions

- Implemented protocol is working as expected
- Able to Tx/Rx different application protocol frames
    - Ex: HTTP, FTP, VoIP
- Modules implemented can be used directly while integration
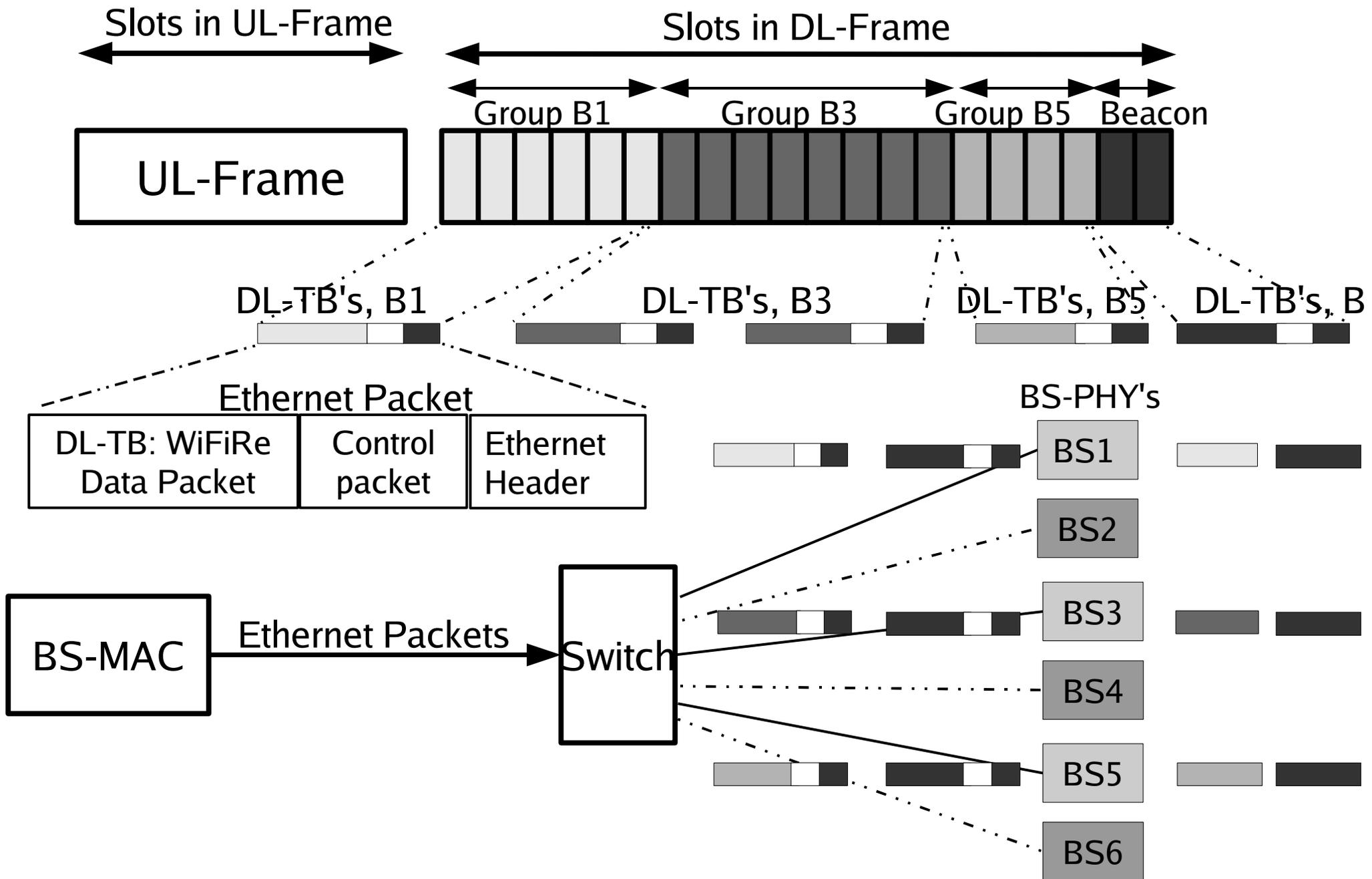
Future work

- Extending to multiple sectors
- Efficient scheduler required at S
- Implementing QoS and CAC
- Slot level implementation has to be done
- Integration with actual hardware
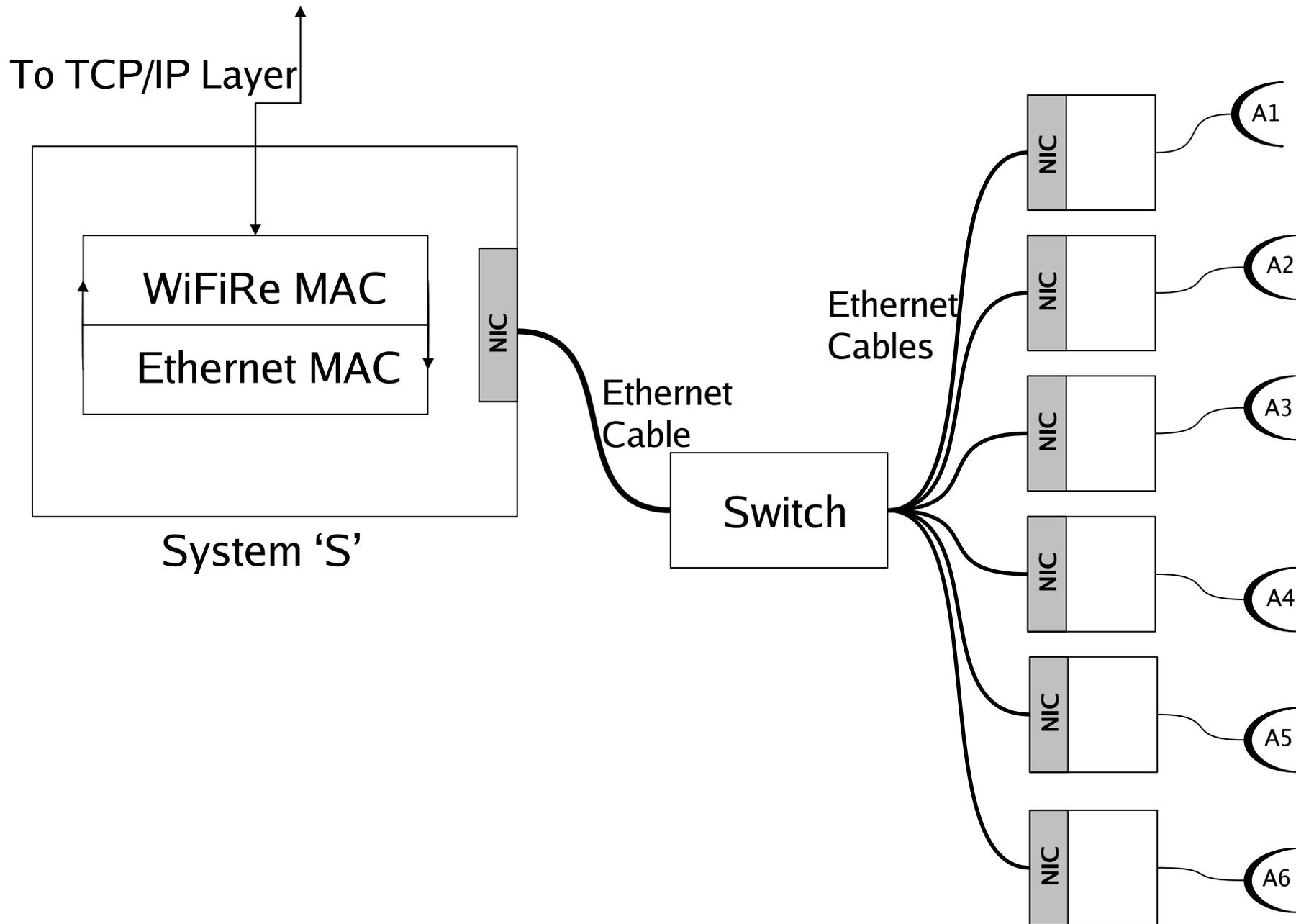
References

[1] Sridhar Iyer (IIT Bombay), Krishna Paul (Intel), Anurag Kumar (IISc Bangalore), and Bhaskar Ramamurthi (IIT Madras). Broadband Wireless for Rural Areas–
WiFiRe: Medium Access Control (MAC) and Physical Layer (PHY) Specifications . August 2006.

[2] Shravan Kumar Hullur. Design and Implementation of MAC Layer of WiFiRe Protocol, M.Tech Thesis, WiFiRe team, IIT Bombay, 2007.

[3] Sameer Kurkure. Design and Implementation of WiFiRe MAC Layer Protocol, M.Tech Thesis, WiFiRe team, IIT Bombay, 2007.

[4] Bhaskaran Raman Pravin Bhagavat and Dheeraj Sanghi. Turning 802.11 Inside-Out. In ACM SIGCOMM, pages 33–38, 2004n

[5] Sockets. The Linux Socket Filter: Sniffing Bytes over the Network. http://www.linuxjournal.com/article/4659.

Thank you

Meta Frame Construction

WiFiRe real system component

To TCP/IP Layer

WiFiRe MAC

Ethernet MAC

NIC

System 'S'

Ethernet Cable
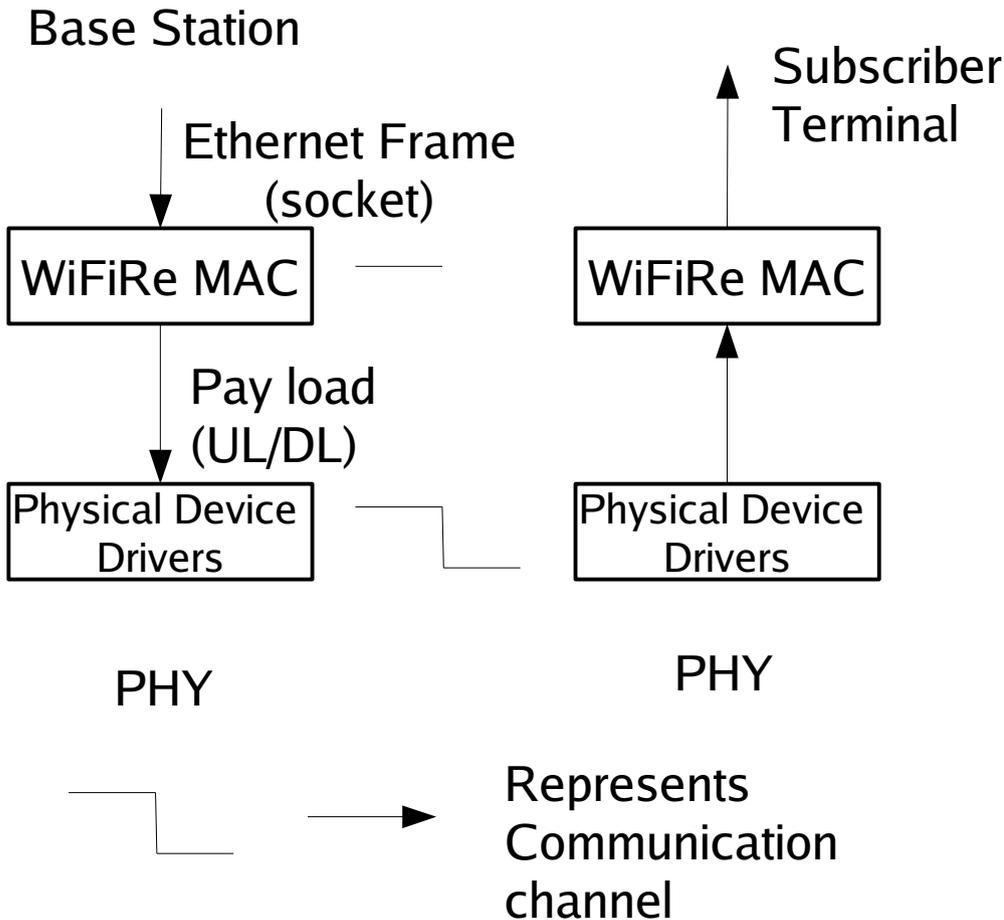
Switch

Ethernet Cables

NIC — A1

NIC — A2

NIC — A3

NIC — A4

NIC — A5

NIC — A6

Assumptions on PHY

- Clock generate one tic per every slot
- Reading control packet
- Handling multiple Ethernet packets
- Min buffer size required (210 slots x 44B)
- FCFS queue at BS

WiFiRe LAN Emulation

Base Station

Ethernet Frame
(socket)

Subscriber
Terminal

WiFiRe MAC

WiFiRe MAC

Pay load
(UL/DL)

Physical Device
Drivers

Physical Device
Drivers

PHY

PHY

Represents
Communication
channel

MAC functionalities
   Network initialization
   Processing frames
   Handling buffers
PHY functionalities
   Tx frames
   Rx frames

What is emulation?
   exact reproduction of external behavior of the system

Fig. Ref: Shravan Thesis[2]

- Threads
  - Main_thread and Signal_thread
  - In_thread and Out_thread
- Sockets
  - Used PF_PACKET socket in C
  - Bind with particular NIC
  - Byte level access
  - Tx/Rx non-Ethernet packets
- Buffers
  - Used for temporary storage purpose at socket