Link Virtualization: A Framework for Simultaneous Evaluation of Multiple Protocols

Wei Quan, Guevara Noubir, Xin Liu, Prathap Ramachandra
Email: {qwlli, noubir, liux, pramach@ccs.neu.edu}

Motivation

- Need for accurate estimation of channel and medium status
- Inaccurate modeling of Wireless Physical Layer in Simulation Environments
- Unavailability of tools for experimental comparison with same channel conditions
- Difficulties with Experimental Evaluation
- Unavailability of tools for developing and testing cross-layer protocols

Issues with Simulation Environments

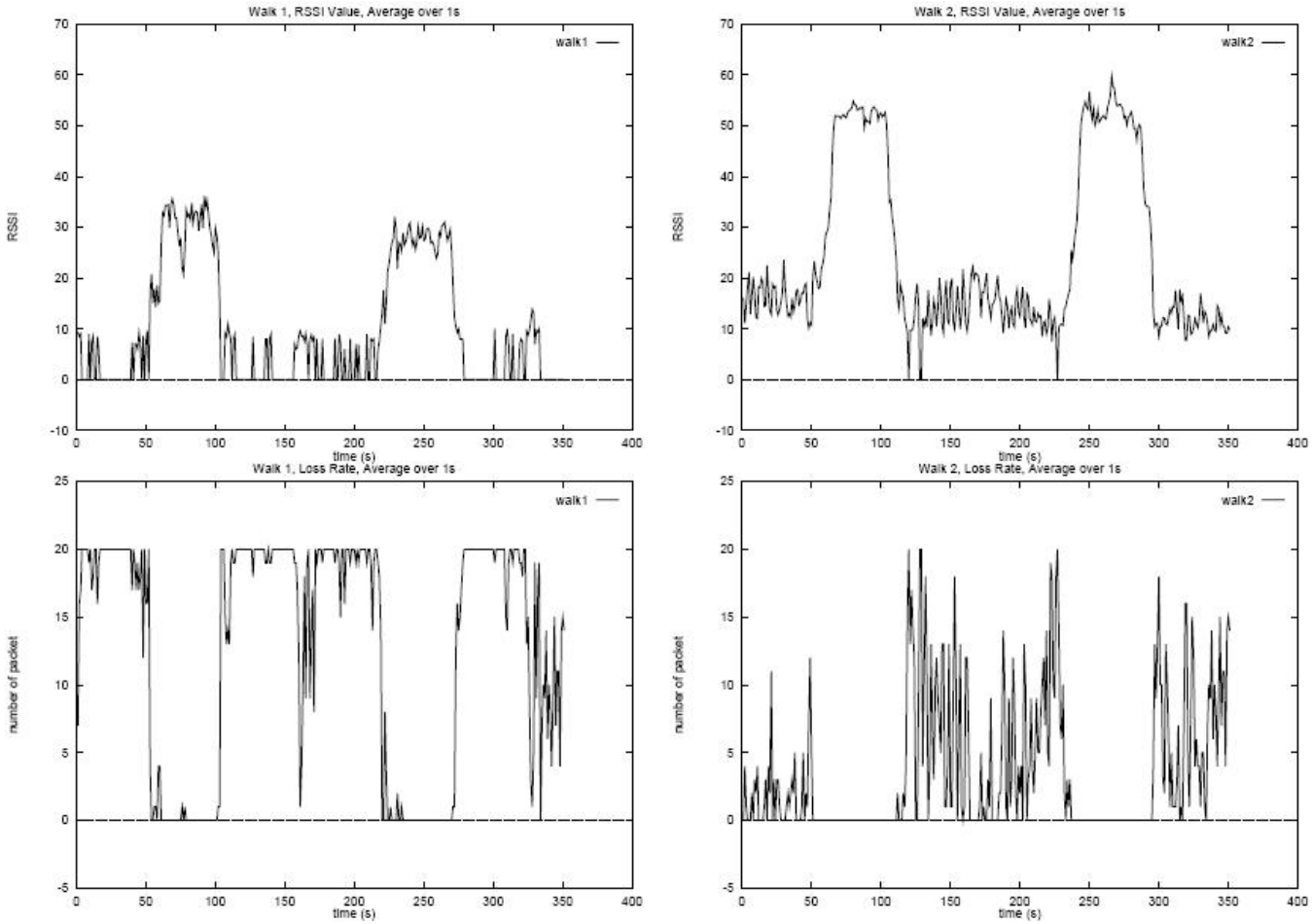
- Speed Decay Problem with Random Waypoint Model
- Inconsistency of Simulation Environments
- Inaccurate account of interference in NS-2 to compute BER and FER
- Only a single rate used for an entire simulation period
- Only a single sensitivity threshold for all rates

Issues with Experimental Evaluation

- Difficulty with reproducing the same channel conditions
- Alternatives like usage of channel emulators to connect the RF front ends are very expensive and inadequate with large number of links

Mobile/Outdoor

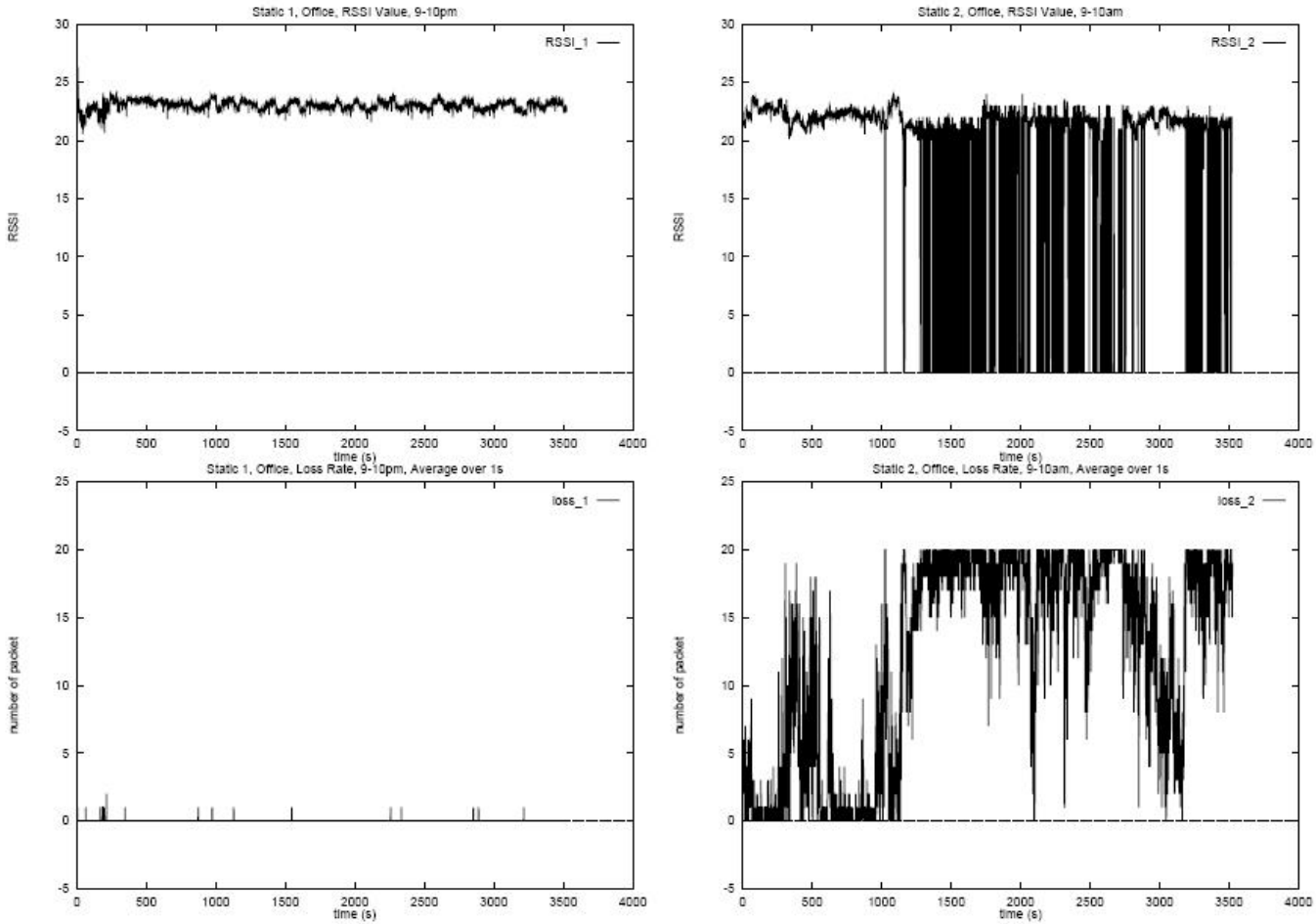
RSSI and Frame Loss Rate For Two walks across MFA in Boston



Issues with Experimental Evaluation

Static/Indoor

RSSI and Frame Loss Rate For Two experiments inside VWH in Northeastern University, Boston



Issues with Experimental Evaluation

Contributions

- Platform for simplifying the development of cross-layer MANET protocols
- Platform allows:
  - The control of the physical layer and MAC layer parameters on a per-packet basis
  - Fine grain channel and medium status assessment
- Framework for the performance evaluation of MANET protocol stacks under almost identical propagation environments

Platform Architecture and Framework

Functionality - Virtual Environment

- Protocol Scheduler interleaves multiple instances of protocols in TDMA-like fashion
- Each instance of Physical Protocol makes a Virtual Protocol
- Virtual Protocol has access to Virtual Link Layer and Virtual Time
- Virtual Link Scheduler is synchronized with the Protocol Scheduler

Functionality – MAC/Physical Layer Control

- Enhanced API for Physical Layer (driver/firmware) to specify, for each packet:
  - Transmission Power Level
  - Modulation/Scheme (i.e., rate)
  - Fragmentation
  - Maximum number of retransmissions
- Number of Retransmissions (Failure) information for Transmitter
- RSSI information for Receiver

Functionality – Peripheral Tools

- Driver-level Light-Weight Protocol for Synchronization below 1ms
- Monitoring Service for gathering information about Packets Transmitted, Retransmissions, Rate, Power Level, Timing, etc.

Functionality – Assumptions

- Channel coherence time is in the order of the interleaving time of protocols under evaluation
- The macro-characteristics like loss rate etc. are stable

Components

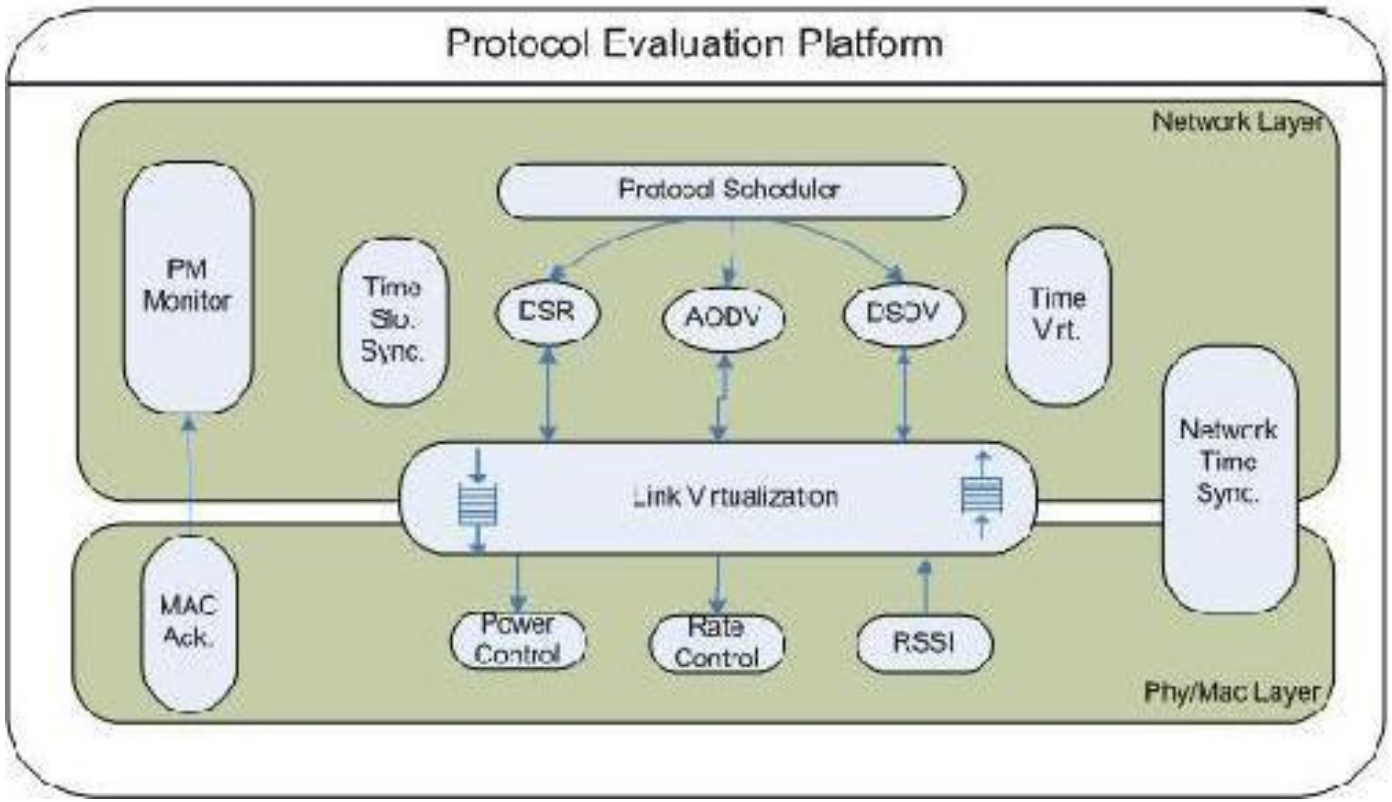
- Nodes
  - Laptops equipped with IEEE802.11a/b/g wireless adapters, running an instance of the Platform each
- Operating System
  - Linux Red Hat 9.0 with 2.4.24 Kernel or above
  - Kernel time interval configured to 1ms for accurate scheduling
- Wireless Interface
  - Dynamic power/modulation/coding control on a per-packet basis
  - Uses wireless adapters supporting dynamic control

Architecture

- Platform is implemented as a Click-Modular Router Element

- Click-Modular Router
  - Intercepts packets from Applications, process them and pass them on to the appropriate Network Interface
  - Retrieves incoming packets from the specified network interface, process them and hand them over to either Kernel to be delivered to the applications or to the Network Interface to be forwarded

Protocol Evaluation Platform

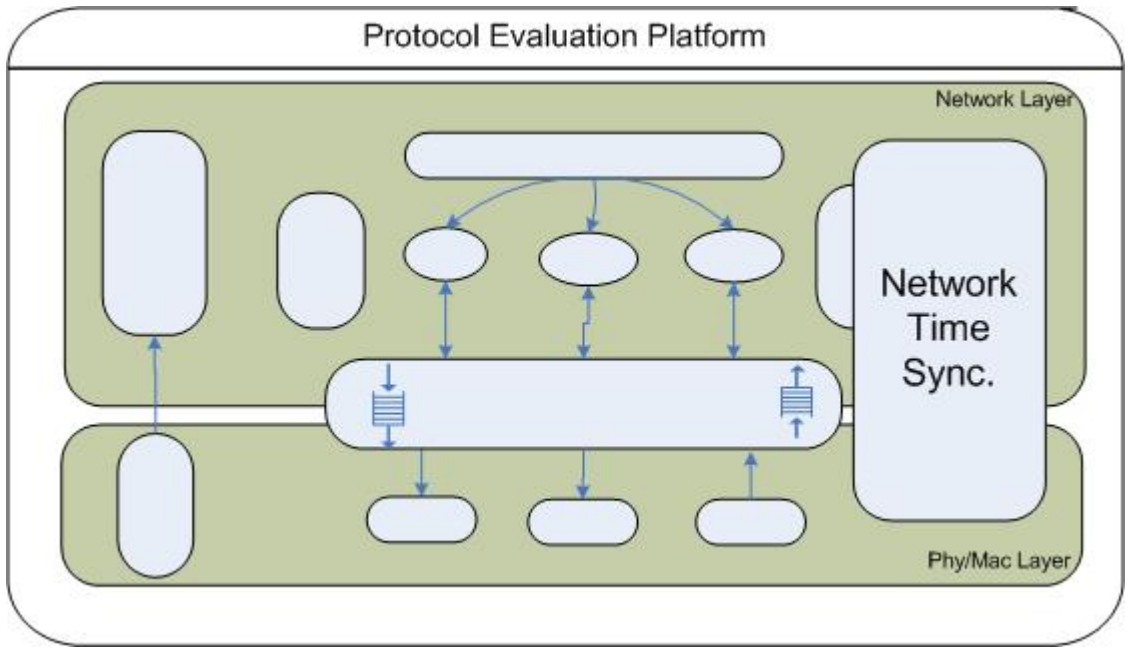


Platform Architecture and Framework

Architecture - Internal Structure

- Demultiplexing and Multiplexing for multiple protocols
- Makes use of:
  - Network Time Synchronizer
  - Time Virtualization Module
  - Time Slot Synchronization Module
  - Protocol Scheduler
  - Link Virtualization Module
  - MAC Acknowledgement
- Extends into Linux Drivers

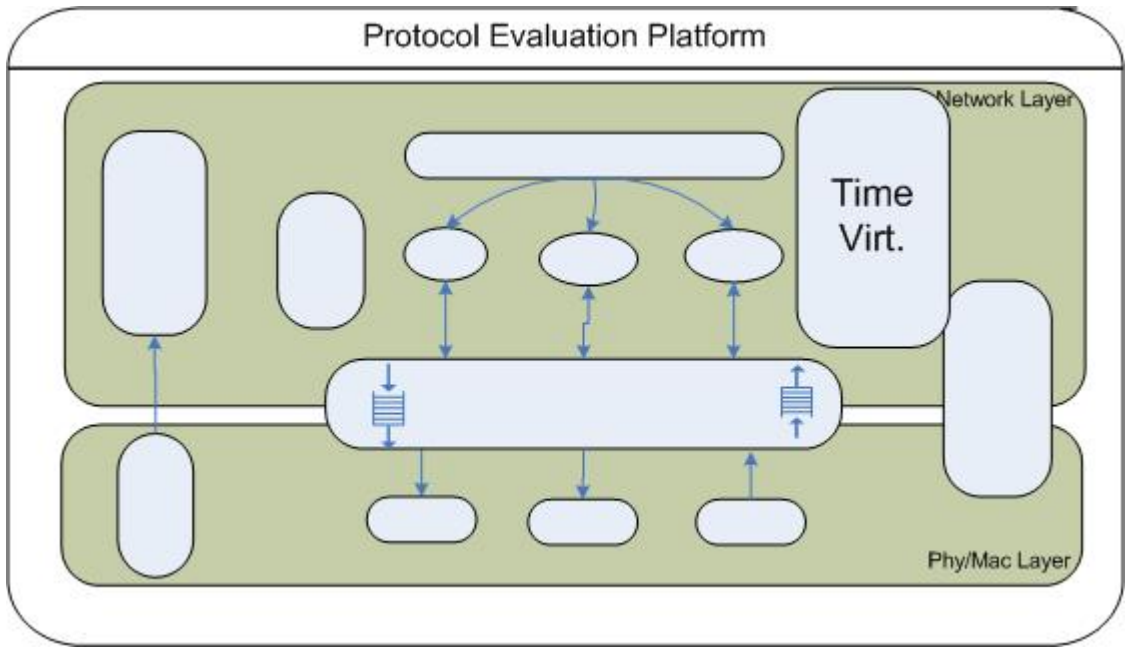
Protocol Evaluation Platform – Network Time Sync



Architecture – Network Time Synchronizer

- Multi-hop network time synchronizer using broadcast references
- All nodes synchronize to one pre-decided server node before the experiment
- Subsequent Synchronization messages sent at random within a window based on node density
- System time difference between nodes are maintained below 1ms

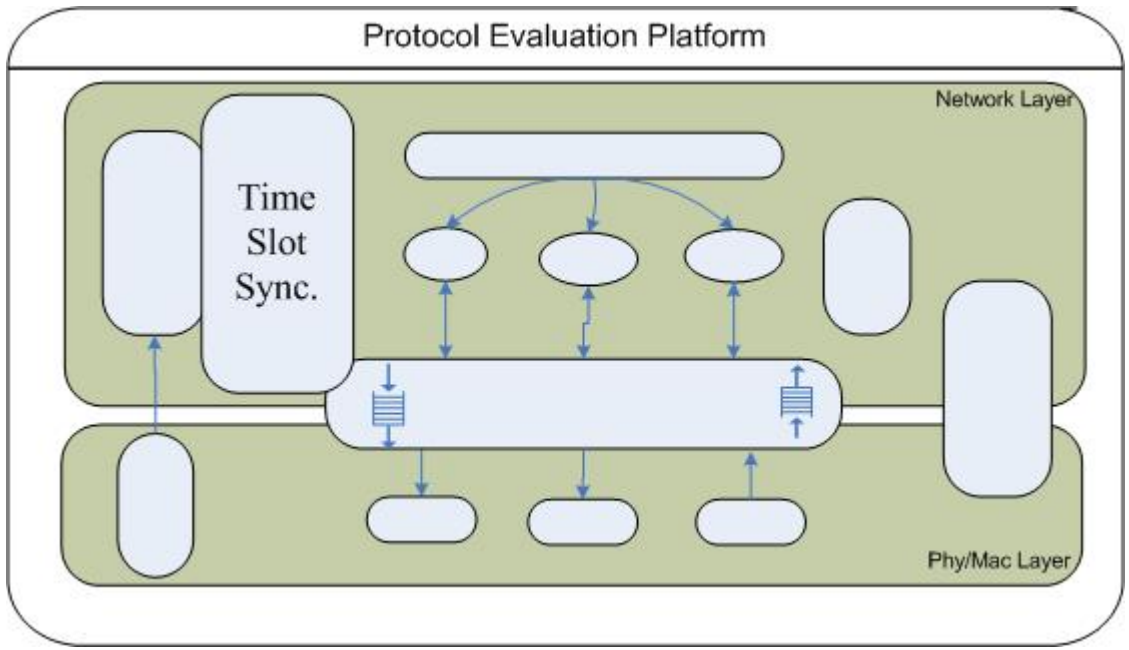
Protocol Evaluation Platform – Time Virtualization



Architecture – Time Virtualization Module

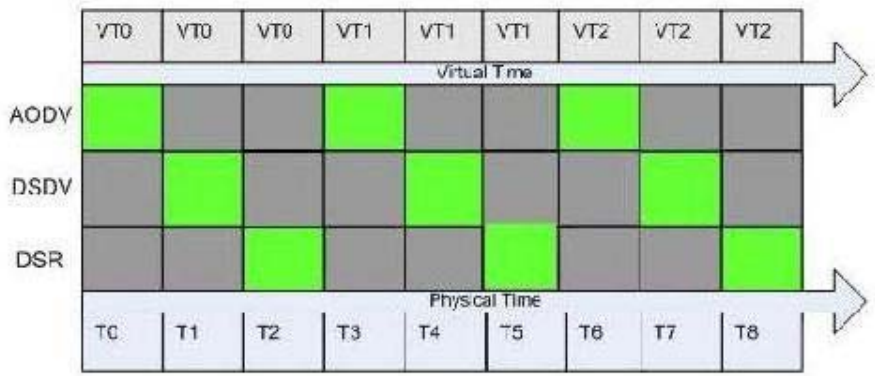
- Platform schedules Protocols in TDMA-like fashion
- Each Protocol Instance has independent virtual time space that it sees as continuous

Protocol Evaluation Platform - Time Slot Sync

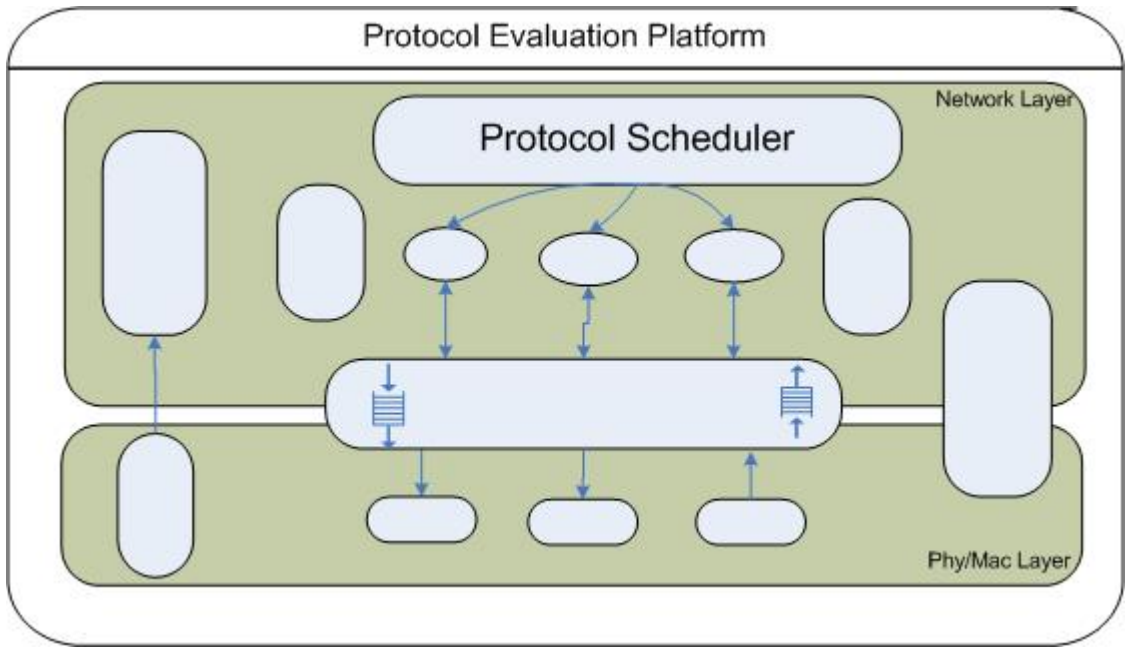


Architecture – Time Slot Synchronization Module

- Influencing and Configurable Parameters:
  - Number of Protocols
  - Time-slot duration



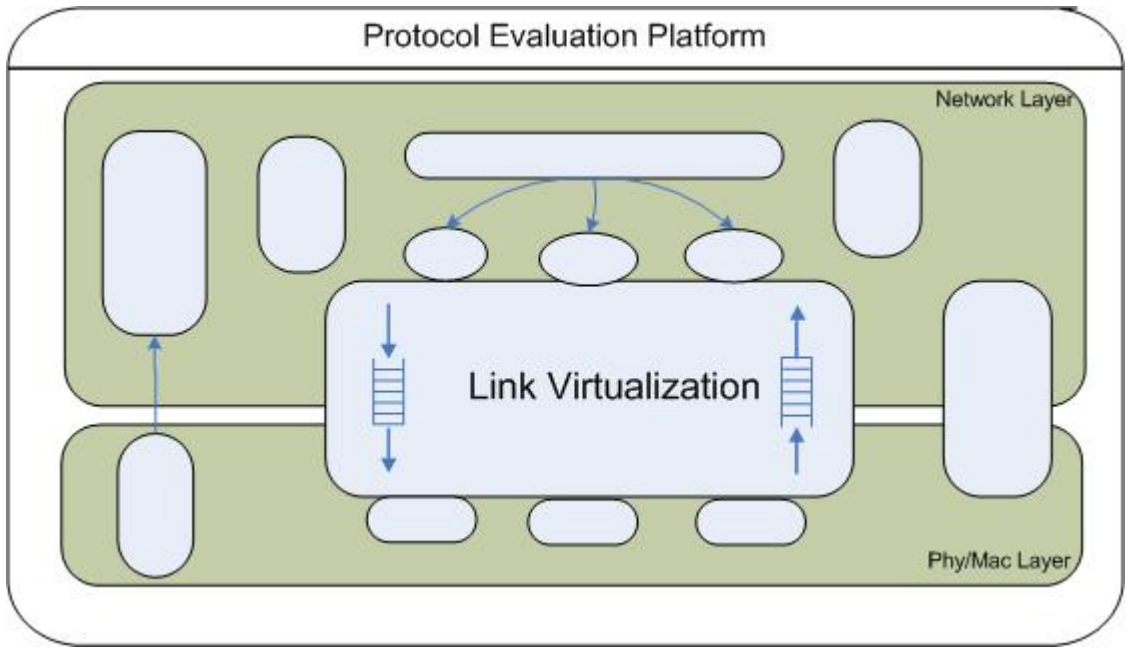
Protocol Evaluation Platform – Protocol Scheduler



Architecture – Protocol Scheduler

- Wakes up at each time slot
- Puts running Protocol into sleep mode
- Runs the next scheduled Protocol
- Driven by System Timer

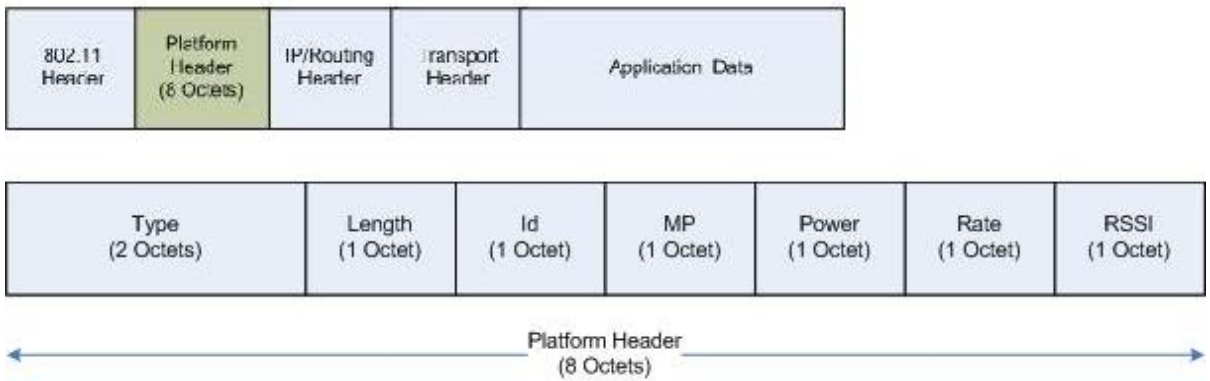
Protocol Evaluation Platform - Link Virtualization



Architecture – Link Virtualization Module

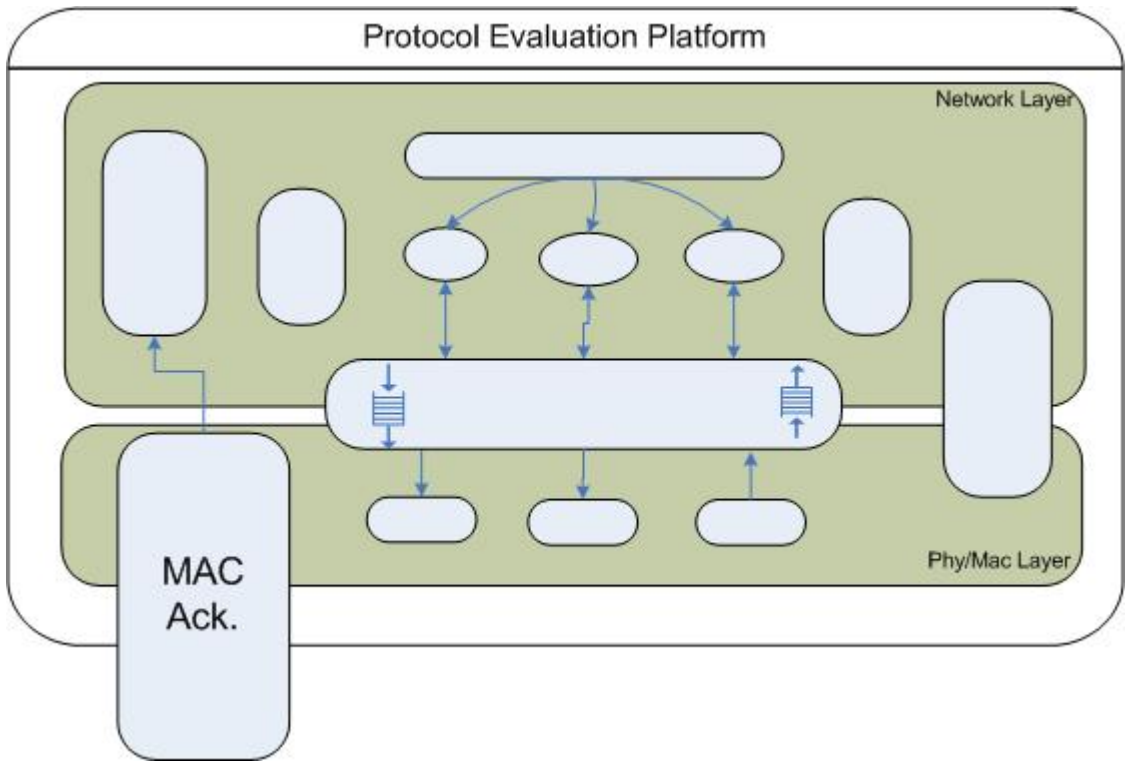
- Dynamic power/modulation/rate control per packet provided by introducing additional Virtualization Layer in TCP/IP stack
- Driver enforces firmware to use parameters specified in Virtualization Layer header
- RSSI information passed on to Platform
- The layer is transparent to the Protocols running in the user-space

Architecture – Link Virtualization Header



- Type - Next frame type
- Length - Size of the platform header
- ID - Protocol identifier for packet (e.g., DSR instance 1)
- MP - MAC Parameters control information (RTS/CTS, Fragment, number of retransmissions)
- Power - Transmission Power
- Rate - Transmission Rate
- RSSI - Received Signal Strength

Protocol Evaluation Platform - MAC Ack



Architecture – MAC Acknowledgement

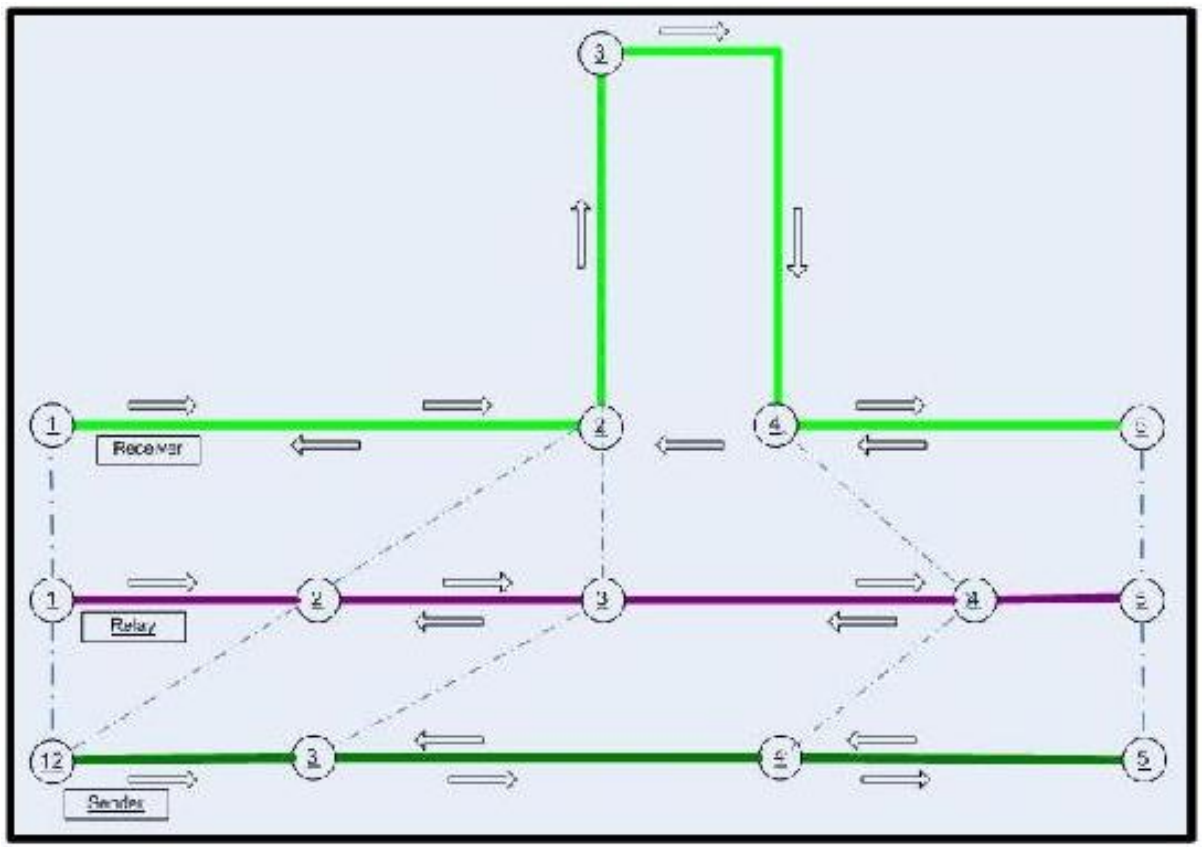
- Used for estimation of Medium and Channel State
- MAC ACK generated in Kernel for Protocol with:
  - Retransmission Count
  - Power Level
  - Rate
  - Owner Protocol
- Data used for Protocol Performance Evaluation

Architecture – Developing Protocols

- Platform written as an Element in Click-Modular Router in C++
- Easy to insert New Protocols
- Easy to plug the new Protocol into evaluation environment

Platform Evaluation

- Mobility Pattern:



Platform Evaluation

- Common Parameters:
  - Mobility Area: 500 meters x 300 meters
  - Transmission Rate: 512 bytes every 20 ms
- Interleaved Runs:
  - Protocols and Number of nodes: 2 DSR on 3 nodes
  - Number and Duration of Runs: 9 runs of 10 minutes each
- Non-interleaved Individual Protocol Runs:
  - Protocol and Number of nodes: 1 DSR on 3 nodes
  - Number and Duration of Runs: 6 runs of 5 minutes each

Platform Evaluation Results

- Two instances of DSR interleaved. The left side corresponds to the log on a sender node and the right side corresponds to the log of the DSR destination node. The log file shows the nodes synchronization and packets exchanged

```
---------- Switch to DSR0 ----------
The current scheduling time is at 1120659544000.387939 in microseconds
                    Switch to DSR1
The current scheduling time is at 1120659545000.489014 in microseconds
---------- Switch to DSR0 ----------
The current scheduling time is at 1120659546000.585938 in microseconds
DSR0:Broadcast host route request for 192.168.0.110
DSR0:Receive Route Reply from 192.168.0.110 -> US
DSR0:Send Packet to 192.168.0.110
DSR0:Send Packet to 192.168.0.110
DSR0:Send Packet to 192.168.0.110
DSR0:Send Packet to 192.168.0.110
---------- Switch to DSR1 ----------
The current scheduling time is at 1120659547000.300049 in microseconds
DSR1:Broadcast host route request for 192.168.0.110
DSR1:Receive Route Reply from 192.168.0.110 -> US
DSR1:Send Packet to 192.168.0.110
DSR1:Send Packet to 192.168.0.110
DSR1:Send Packet to 192.168.0.110
DSR1:Send Packet to 192.168.0.110
```

```
---------- Switch to DSR0 ----------
The current scheduling time is at 1120659544000.930908 in microseconds
                    Switch to DSR1
The current scheduling time is at 1120659545000.574951 in microseconds
---------- Switch to DSR0 ----------
The current scheduling time is at 1120659546000.191895 in microseconds
DSR0:Receive Route Request with accumulated route: 192.168.0.107 ->  US
DSR0:Generate route reply back to 192.168.0.107
DSR0:Receive Data packet: 192.168.0.107 -> 192.168.0.110
DSR0:Receive Data packet: 192.168.0.107 -> 192.168.0.110
DSR0:Receive Data packet: 192.168.0.107 -> 192.168.0.110
DSR0:Receive Data packet: 192.168.0.107 -> 192.168.0.110
---------- Switch to DSR1 ----------
The current scheduling time is at 1120659547001.263916 in microseconds
DSR1:Receive Route Request with accumulated route: 192.168.0.107 ->  US
DSR1:Generate route reply back to 192.168.0.107
DSR1:Receive Data packet: 192.168.0.107 -> 192.168.0.110
DSR1:Receive Data packet: 192.168.0.107 -> 192.168.0.110
DSR1:Receive Data packet: 192.168.0.107 -> 192.168.0.110
DSR1:Receive Data packet: 192.168.0.107 -> 192.168.0.110
```

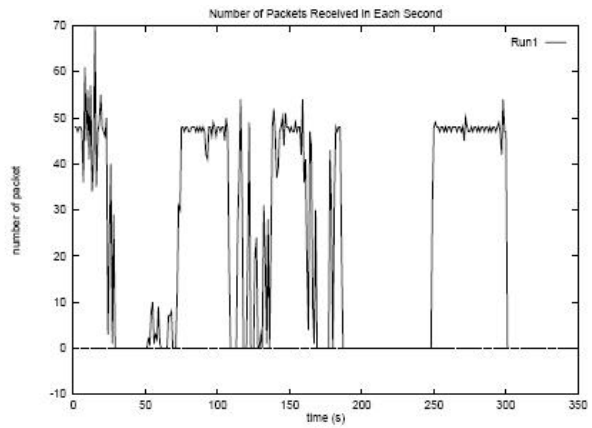
Platform Evaluation Results

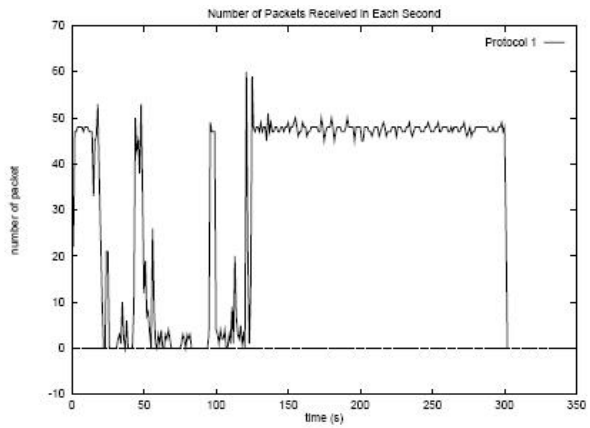
- The average difference in the total delivered packets:
  - Interleaved Protocols: 1.7%
  - Non-Interleaved Individual Protocols: 17% (with high variance)

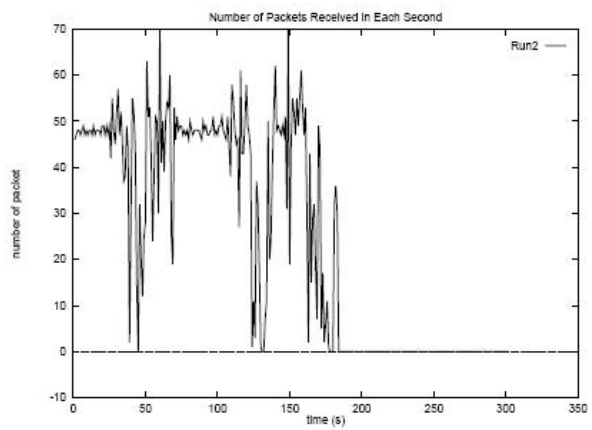
- Correlation between delivered packets per second:
  - Between the two Interleaved Protocols: 0.86
  - Average between multiple runs of Non-Interleaved Individual Protocols: 0.21

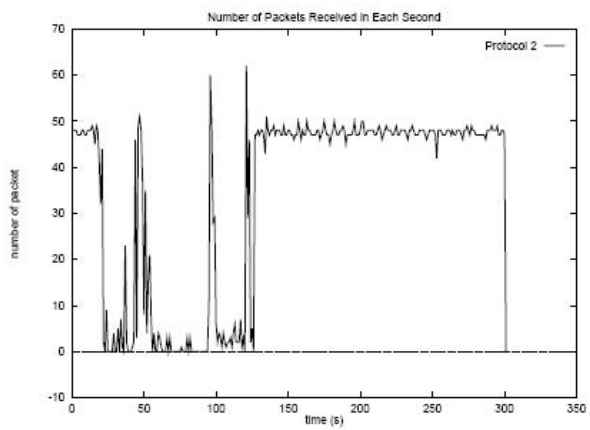
Platform Evaluation Results

- Similarity of the receive pattern when using framework and running the protocol multiple times with the same mobility pattern. The right figures correspond to two instances interleaved. The left figures correspond to two separate runs using the same mobility pattern









Platform Evaluation & Results

V2V

- Common Parameters:
  - Mobility Area: 200 meters x 200 meters
  - Transmission Rate: 512 bytes every 20 ms
- Interleaved Runs:
  - Protocols and Number of nodes: 2 DSR on 3 nodes
  - Number and Duration of Runs: 3 runs of 6 minutes each
- The average difference in the total delivered packets:
  - Interleaved Protocols: 0.61%

Conclusion

- Framework for a fair performance evaluation and comparison of multiple protocol stacks on an almost identical physical environment
- Services to control Physical Layer and MAC layer on a per packet level
- Platform for easy development of Protocols

Future Work

- Carrying an extensive evaluation of Platform with a larger set of nodes, over a larger mobility area
- Plan to improve the time synchronization mechanism in order to be able to carry a finer grain interleaving
- Plan to port other MANET protocols to the Platform to compare them within the Platform and compare the results with those obtained through simulation
- Plan to implement the proposed tools for easy deployment of new protocol stacks over the Platform

Thank You!