Future Internet Winter Camp 2009

프로그래머블 무선 메쉬 네트워크 테스트베드 및 가상화

이준희, 이경한, 이진성, **Sachin Lal. Shrestha,** 정재성, **and** 정송

School of EECS, KAIST

{junhee,khan,ljs,sachinls,yicdk}@netsys.kaist.ac.kr,
song@ee.kaist.ac.kr

2009, Feb, 24 (Tue)

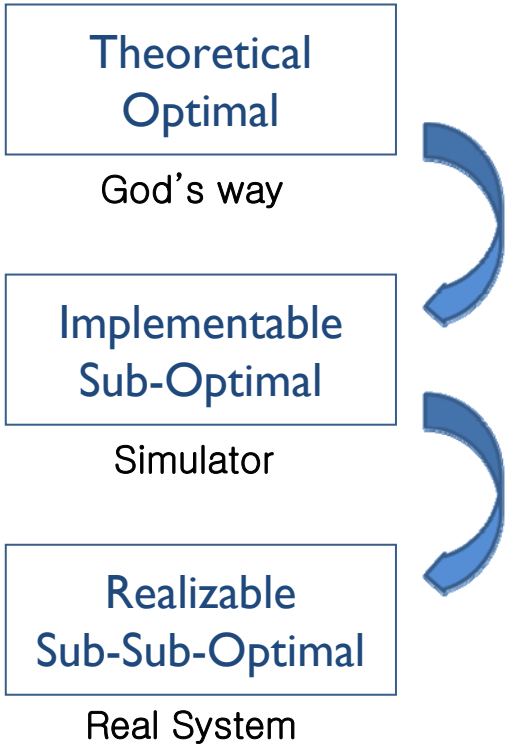
Korea Advanced Institute of Science and Technology
Network Systems Lab.

Introduction [1/2]

- Internet Clean-Slate Design [Feldmann07][Dovrolis08]
  - Redesign from scratch
    - ✓ Clean-slate thinking
    - ✓ Evaluation of a clean-slate attempt
    - ✓ **Experimentation using testbed**
    - ✓ Deployment of clean-slate ideas
- Clean-Slate Design for Wireless Mesh Networks
  - Known theoretical optimal solution [Tiss92]
    - ✓ Source/Link algorithm
  - Cross-layer design
    - ✓ Far from Internet layered architecture
    - ✓ Implementation is not easy

Introduction [2/2]

- Programmable Wireless Mesh Network Testbed



Distributed
Overhead
Complexity
……

It follows basic intentions but
simulator cannot represent
real world wireless environment

System Program
Device Driver
Protocol Stack
……

Another implementation

Programmable ???
Real system that has easy program environment such as the simulator

Previous Work

- Protocol Design and Evaluation [Korpff06]

Simulator

(Serialized processing, using simulation clock)

NS-2/NS-3 [NS2][NS3], JiST/SWANS [Barr04], GloMoSim [GloMoSim], QualNet [QualNet], OPNET [OPNET]

Emulator

(Real application, simultaneous events, real system clock)

Judd(DSP/FPGA) [Judd04], JiST/MobNet [Krop07], ORBIT(802.11Radio) [Raych05], <Click Modular> [Kohler00]

system)

Real World Testbed

(Real wireless channel without modification)

WiMesh [WiMesh], RoofNet [RoofNet], TrueMobile [Johnson05], <PlanetLab> [Peter02]

projects)

Concerns of Previous Work

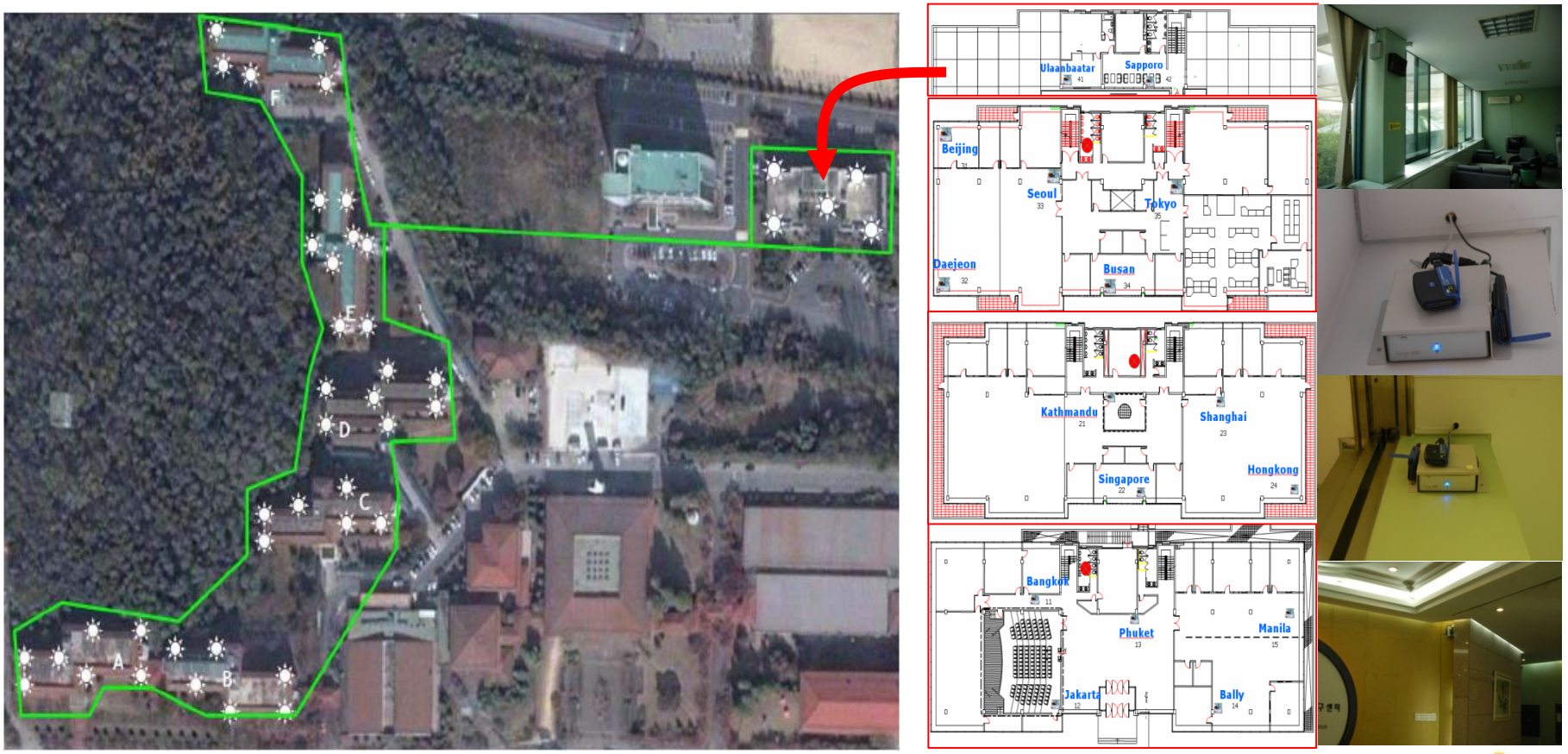
- Do they have concerns about implementation of clean-slate ideas ?
  - Simulator
    - ✓ Probably "Yes"
  - What about emulator & testbed ?

Emulator	Testbed
Accuracy (Fidelity) Scalability	Measurement Evaluation
Management, Monitoring, Friendly User Interface, Capacity	

    - ✓ Limited programmability (because substrate is real thing)
- Programmable Platform in Wireless Network
  - OpenWRT, WARP, NS-3(Emulation), Jist/MobNet

KAIST WiSEMesh Testbed Overview

- An Open Research Testbed (http://wisemesh.kaist.ac.kr)
  - 56 Fixed Wireless Mesh Routers (Campus-scale testbed)



KAIST WiSEMesh Platform

- Architecture for Open Research
  - Hardware platform
    - ✓ Small form-factor PC
      - Pentium CM 1.4GHz
      - 512MB RAM, 60G HDD
      - Ralink x 3, Atheros x 1 (802.11b/g)

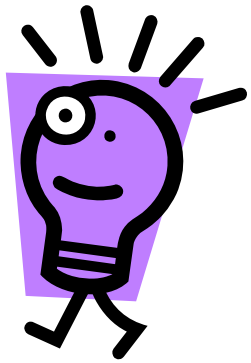
  - Sofrware platform
    - ✓ Ubuntu Linux (Debian)
    - ✓ All open source codes for protocols and drivers
  - Management/Measurement
    - ✓ Dedicated wired management channel
    - ✓ Dedicated management server
    - ✓ Visualization tool (WiVi) (http://143.248.238.6/~sachin/realization/)

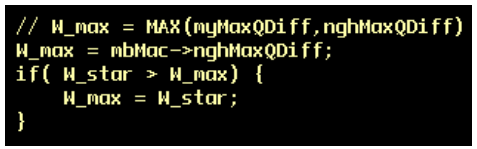
WiSEMesh Testbed Pros & Cons

- Characteristics for Open Research
  - Open source architecture for open research
  - Remotely accessible, light management overhead
  - Automatic experiment configuration
  - Automatic measurement collection
  - Visualization of current network status
- Limitation
  - **Very difficult to implement one's idea on it**
    - ✓ Transport: TCP/UDP in protocol stack
    - ✓ Routing protocol: monolithic, closed, static and inflexible
    - ✓ Datalink layer: proprietary hardware
    - ✓ Cross-layer: Impossible

Common Code Architecture

- Common Code Base
  - A set of protocols or algorithms to avoid re-implementation





(Implementation)

(Reuse without Modification)

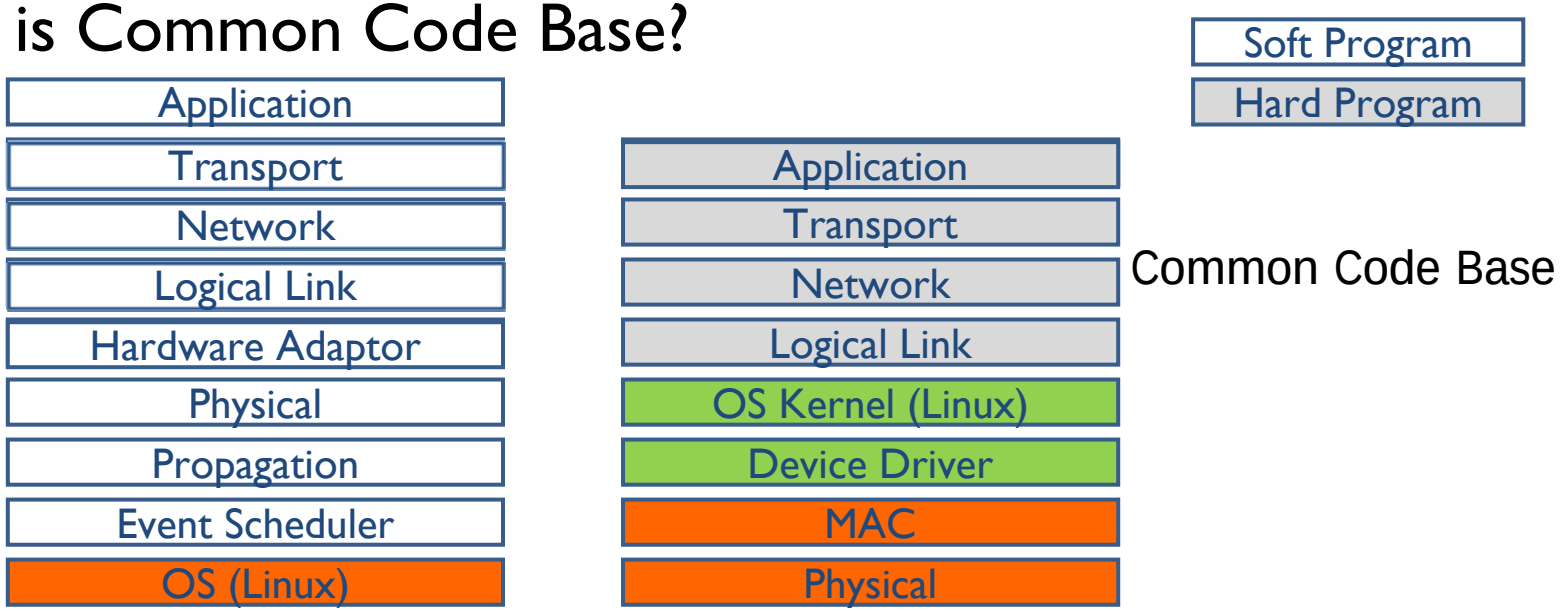
Simulator

Testbed

- Common Code Architecture
  - Architecture that utilizes a Common Code Base
  - A common set of metrics and tools for monitoring and analysis

Simulator *vs.* Testbed Node

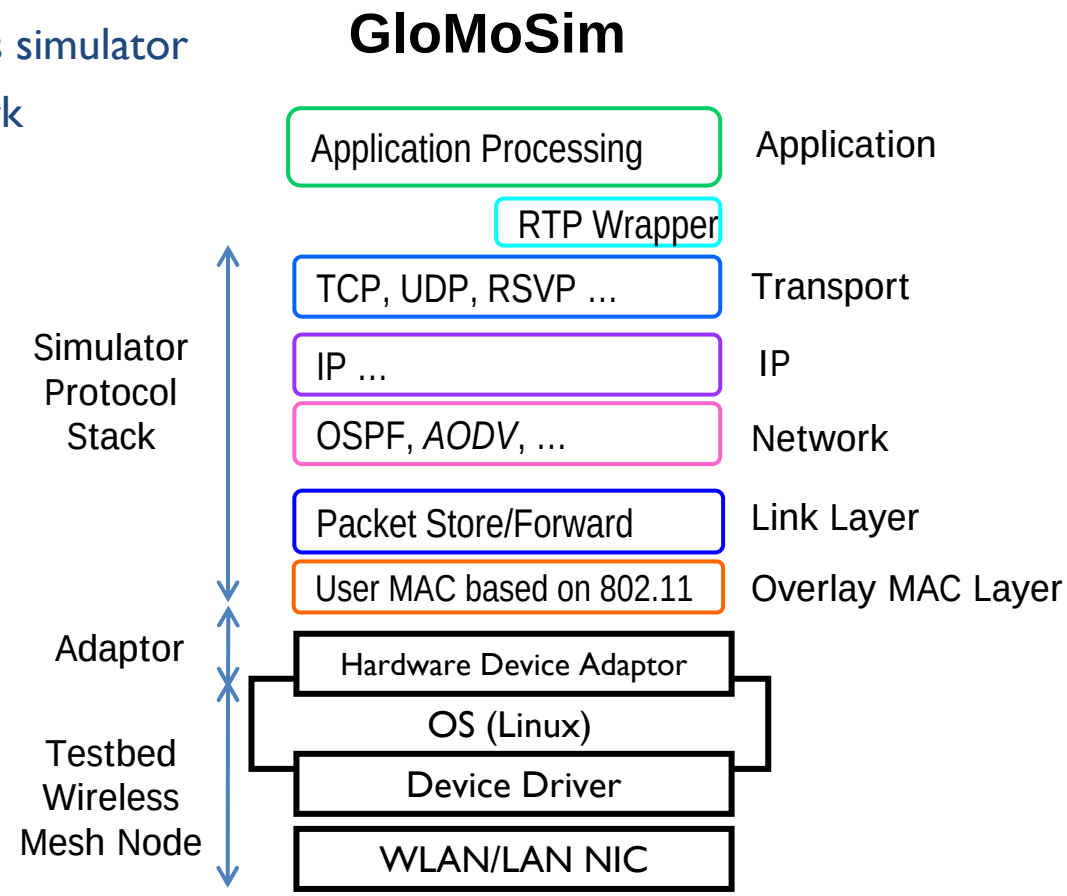
- What is Common Code Base?



- Solution
  - With practiced simulator protocol stack
  - Replace protocol stack of wireless mesh router with that of simulator
  - Hardware adaptor for a real hardware
  - Application adaptor for a real application

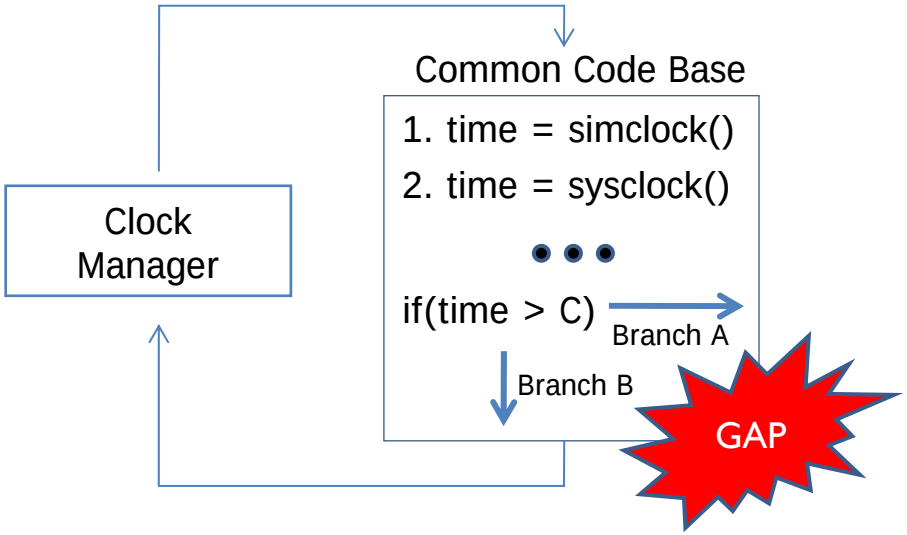
Programmable Architecture

- GloMoSim Simulator
  - Well-known multi-hop wireless simulator
  - Designed for a wireless network
  - Custom C-language
  - Modular protocol component
- Why Not NS-2
  - Designed for wired network
  - Dual mode (C++/Otcl)
- Adaptors for Emulation
  - Hardware device adaptor
    - ✓ Raw socket program
      - Libnet driver
      - Libpcap driver



Redesign of GloMoSim

- Step I: Parsec Compiler ➔ C Compiler
  - Parsec is a C-based simulation language for parallel computing
  - Parsec library is not open source
- Step II: Simulation Clock ➔ System Clock
  - Simulation clock can be stopped
  - System clock never be stopped



Computing Resource Problem

$$\sum_{i \in E} p_i + \sum_{j \in B} p_j \leq \mathbf{P}$$

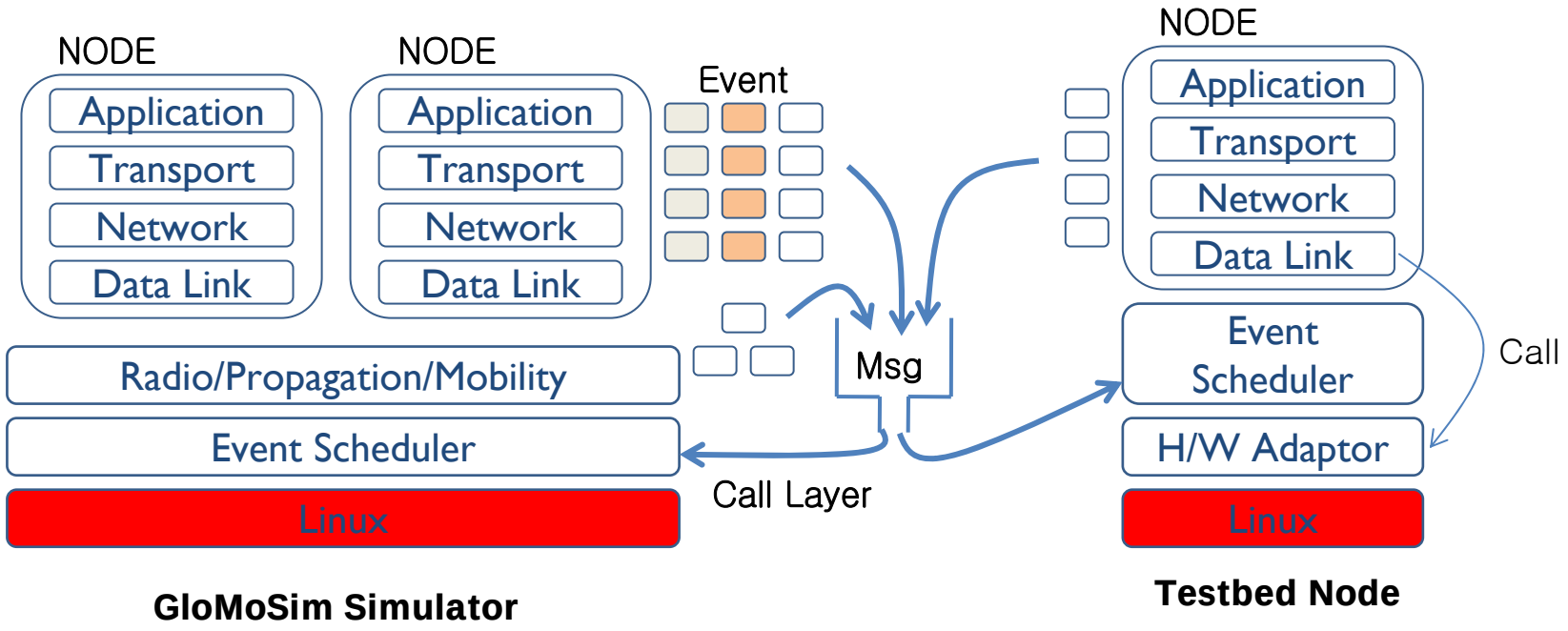
E: event set
B: event set (bandwith)
p: computing power (cycle/sec)

Causality Problem

if(time < $Clock_{target}$) wait();
else schedule();

Protocol Stack Transplantation

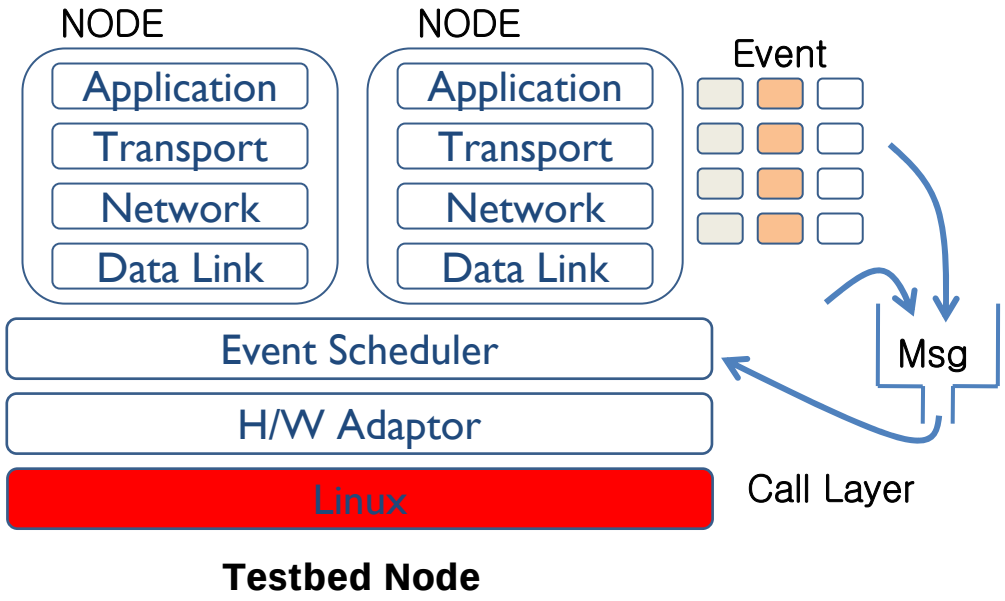
- Step III: Protocol Stack Transplantation



- Hardware device adaptor
  - ✓ Interface with data link layer in protocol stack
  - ✓ Interface with linux kernel (device driver)

Virtualization

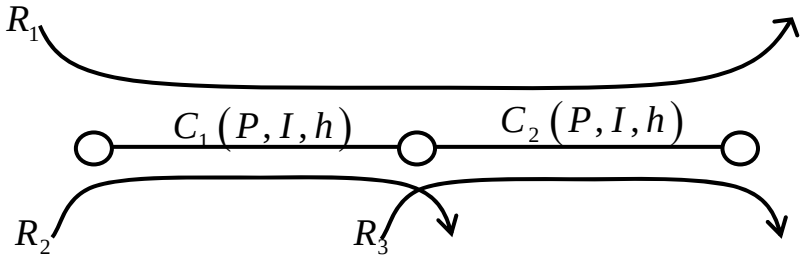
- User-level Virtualization



- Node virtualization by event scheduler
- Network slicing by H/W adaptor and deamon

Theoretical Optimal Solution

- Multi-hop Wireless Networks



P : power allocation
I : link scheduling
h : channel state

  - ✓ Link capacity is time-varying and a function of resource control
  - ✓ Joint rate, power allocation and link scheduling

- Long-term Network Utility Maximization [Tiss92][Chen06]

$$\max_{R,P,I} \sum_s U(R_s)$$

s.t.

$$R \in F(P,I)$$

Flow Control at Source

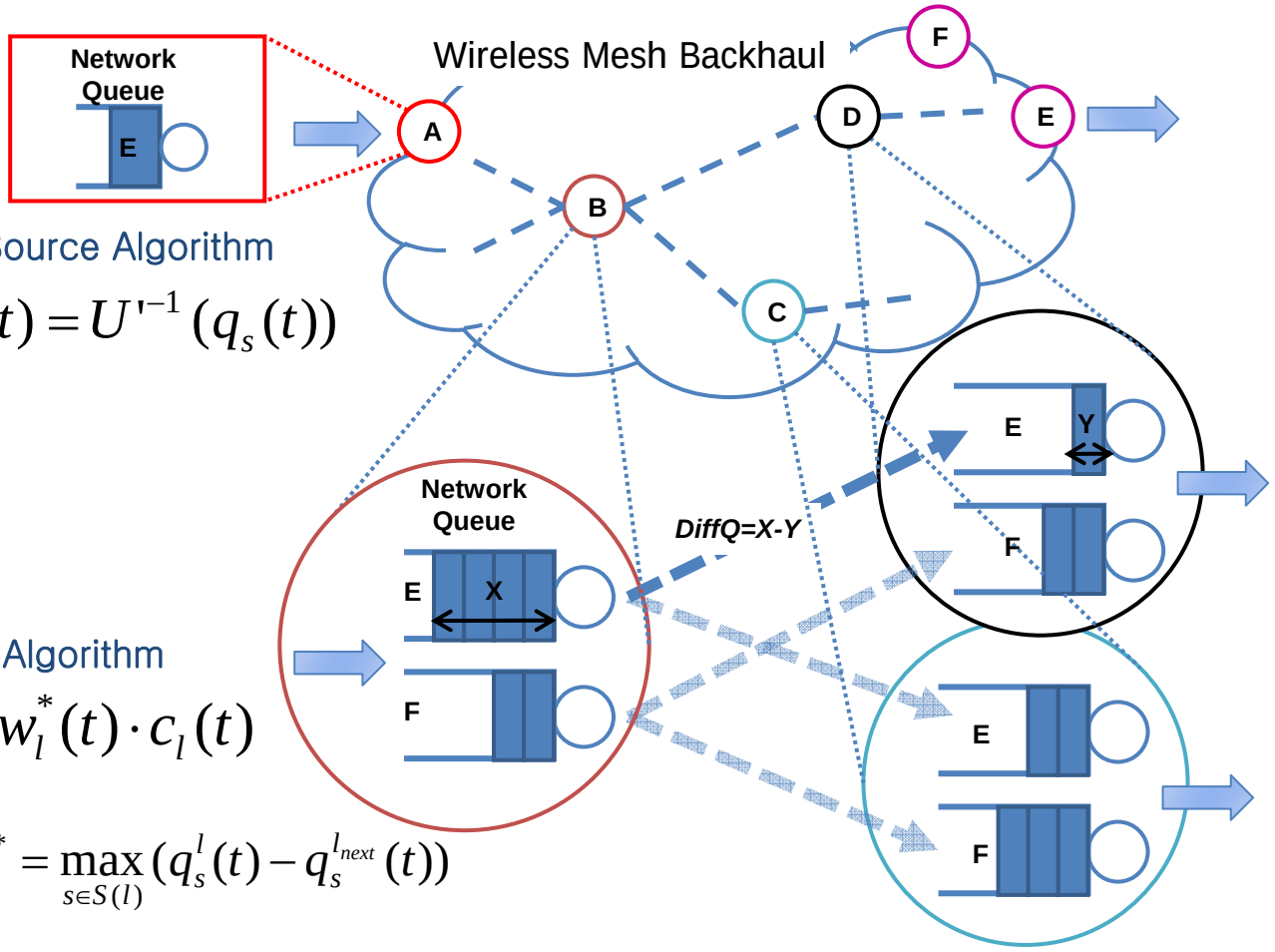
$$\max_r \sum_s \left(U(r_s) - r_s \sum_{l \in L(s)} \lambda_l \right) \quad \Rightarrow \quad r_s = U'^{-1}\left(\sum_{l \in L(s)} \lambda_l \right)$$

Scheduling/Power Control at Link

$$\max_{P,I} \sum_l \lambda_l C_l(P,I,h)$$

Per-Destination Queueing Modeling

- Optimal Algorithm



Solution Development for Implementation

- Highly Complicated Scheduling Scheme
  - Applied to a simple network model
    - ✓ One-hop interference model
    - ✓ Symmetric, constant link capacity
  - Relaxation of the optimality requirements
    - ✓ Greedy Maximal Scheduling [Lin06][Leo01]
      - Feasible sub-optimal solution
      - Much lower complexity than that of optimal solution
      - Lower-bound is known as ½ of the optimal solution
        - Node-exclusive interference model
- Development of Decentralized Algorithm
  - Queue length exchanging method (piggyback, overhear)
  - The longest differential backlog serves the first
    - ✓ Mapping function from differential backlog to priorities

Approximated Cross-layer Control

- Source Algorithm

$$x_s(t) = U'^{-1}(q_s(t))$$

  - Rate control based on queue length

- Link Algorithm

$$\max \sum_{l \in L} w_l^*(t) \cdot c_l(t) \qquad where \quad w_l^* = \max_{s \in S(l)} (q_s^l(t) - q_s^{l_{next}}(t))$$

$$\hat{w}_l = \frac{w_l^*}{w_{l\max}}$$

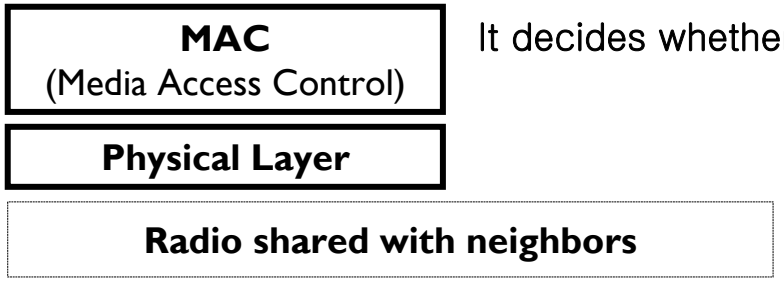
$w_{l\max}$: Maximum weight value maintained link l

CW(Contention Window): $rand((1-\hat{w}_l)K, (1-\hat{w}_l)K + C) \qquad K = CW_{\max} - C$

Contention Window decides independently

Overlay MAC Implementation

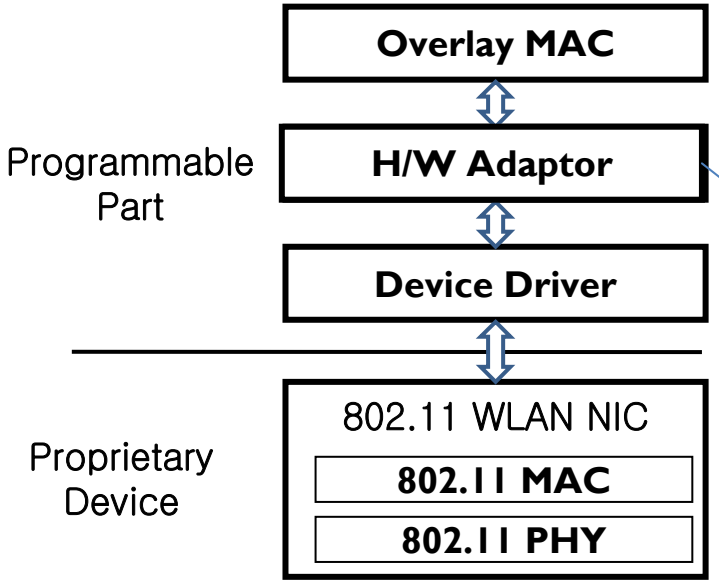
- Radio Status Requirement for MAC Operation



It decides whether transmitting or not

RADIO_TRANSMITTING
RADIO_IDLE
RADIO_RECEIVING
RADIO_SENSING

- Overlay MAC Design



Act as a MAC layer

RADIO_TRANSMITTING
RADIO_IDLE
RADIO_RECEIVING

Buffers must be nullified

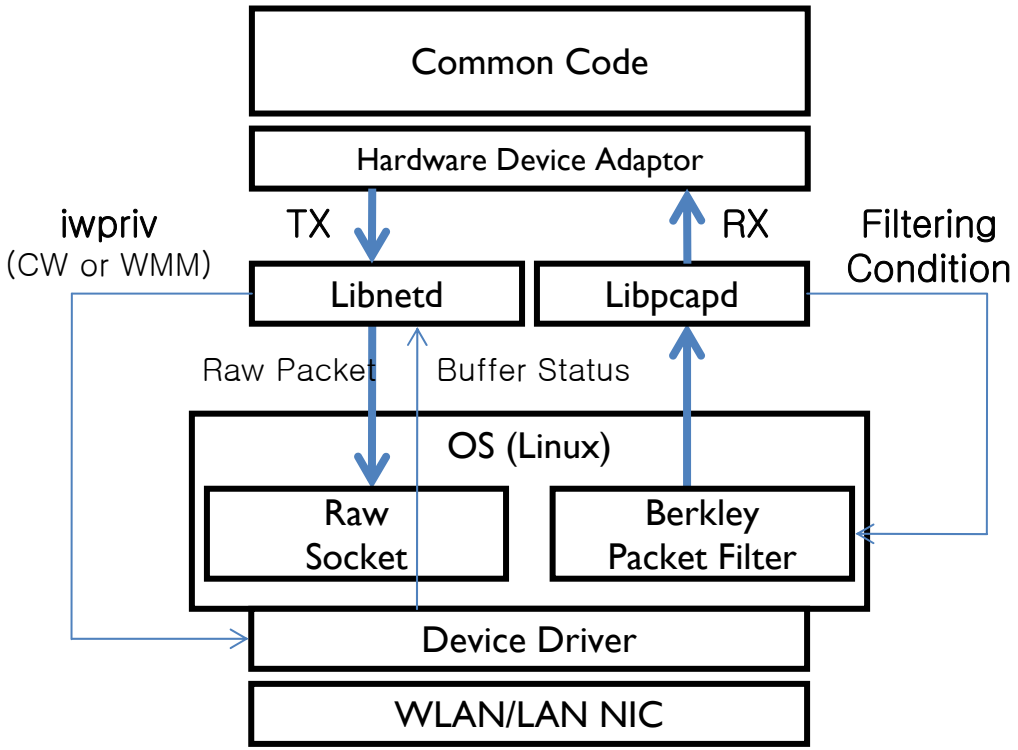
$$CW(\hat{w}_l = \frac{w_l^*}{w_{l\max}})$$

Act as a physical layer

RADIO_SENSING → CW of 802.11
WMM of 802.11e

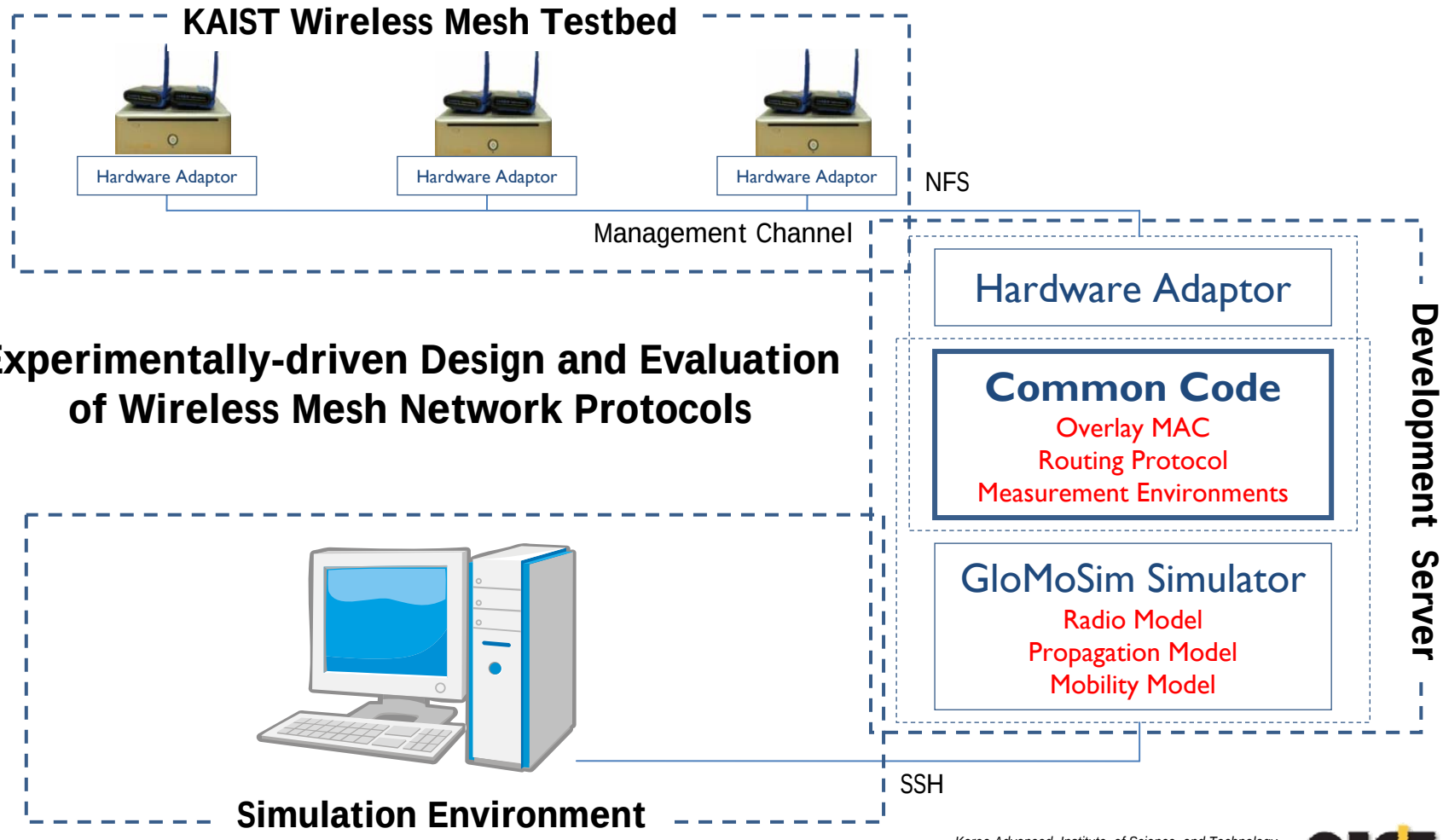
Hardware Device Adaptor

- RT2570(Ralink) / AR5212(Atheros)



Experimentally-driven Design and Evaluation

- Co-Simulation and Emulation Development Environment



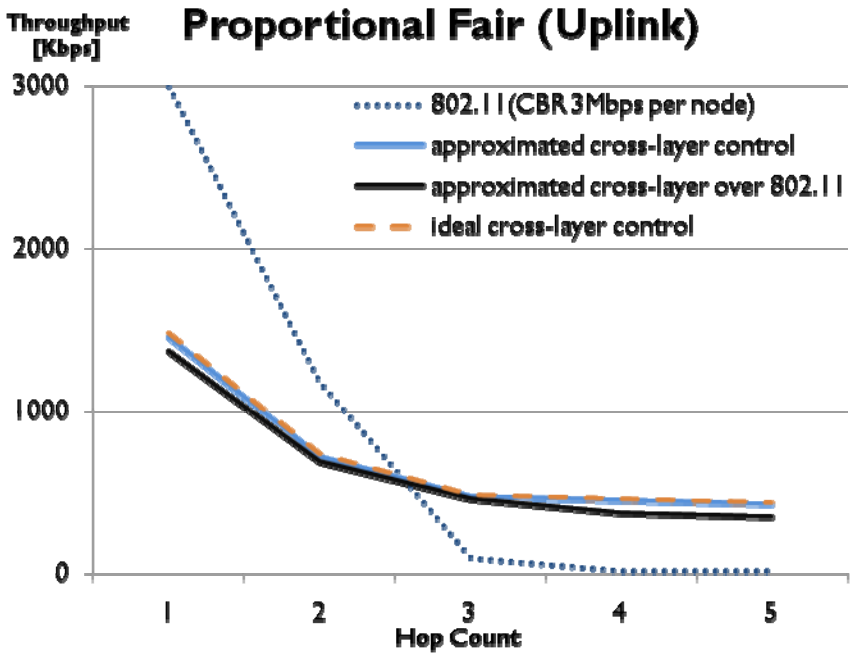
Simulation Results

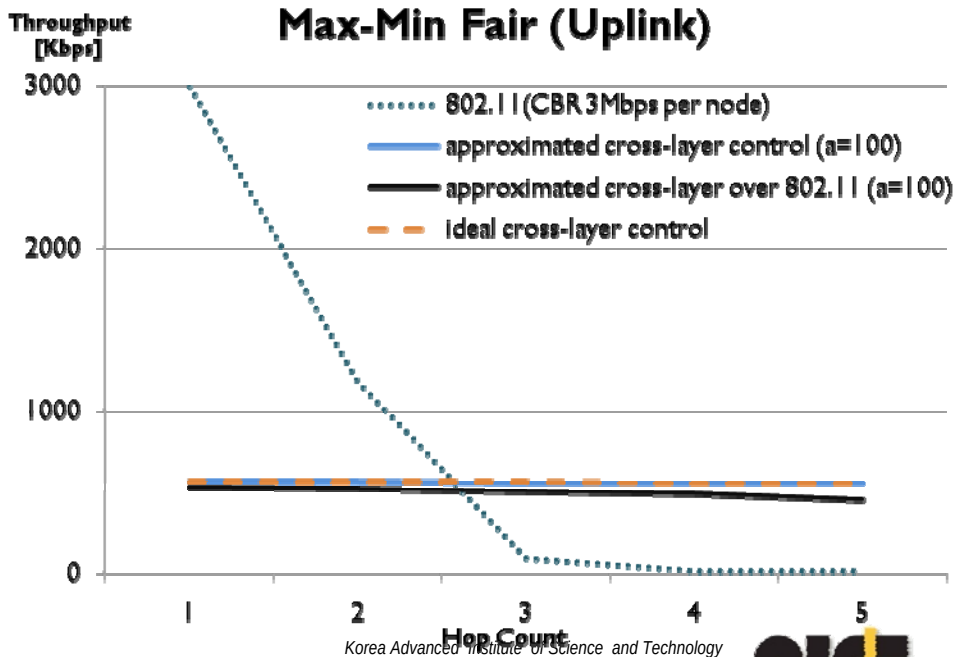
- Network Engineering by Utility Function

$$x = U'^{-1}(q)$$ Intention ⇨ Utility Function → Max-min fair, Proportional fair, Others

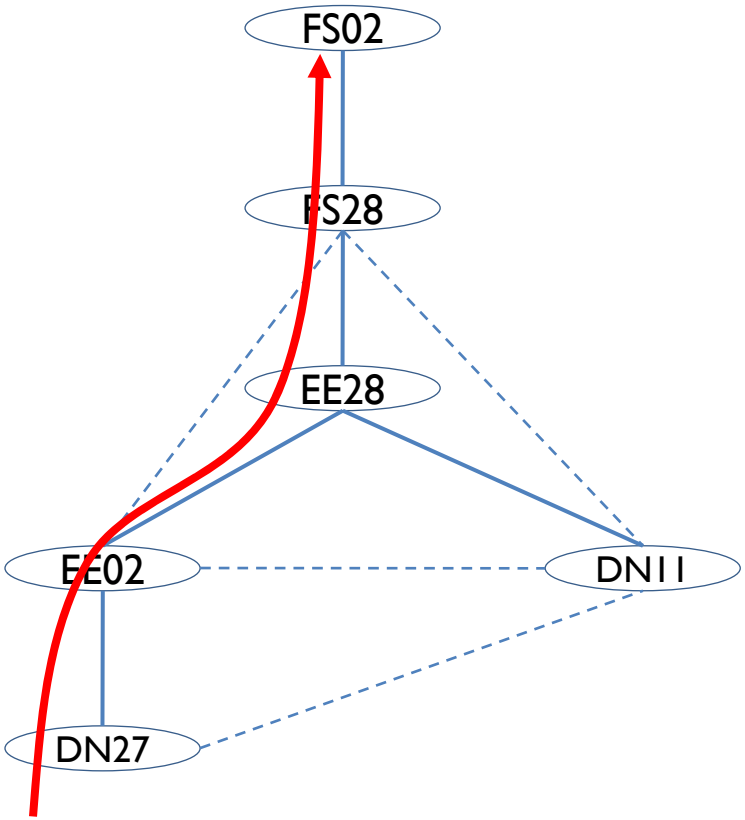
$$U_s(x_s) = \log x_s$$

$$U_s(x_s) = \lim_{a \to \infty} (1-a)^{-1} x_s^{(1-a)}$$



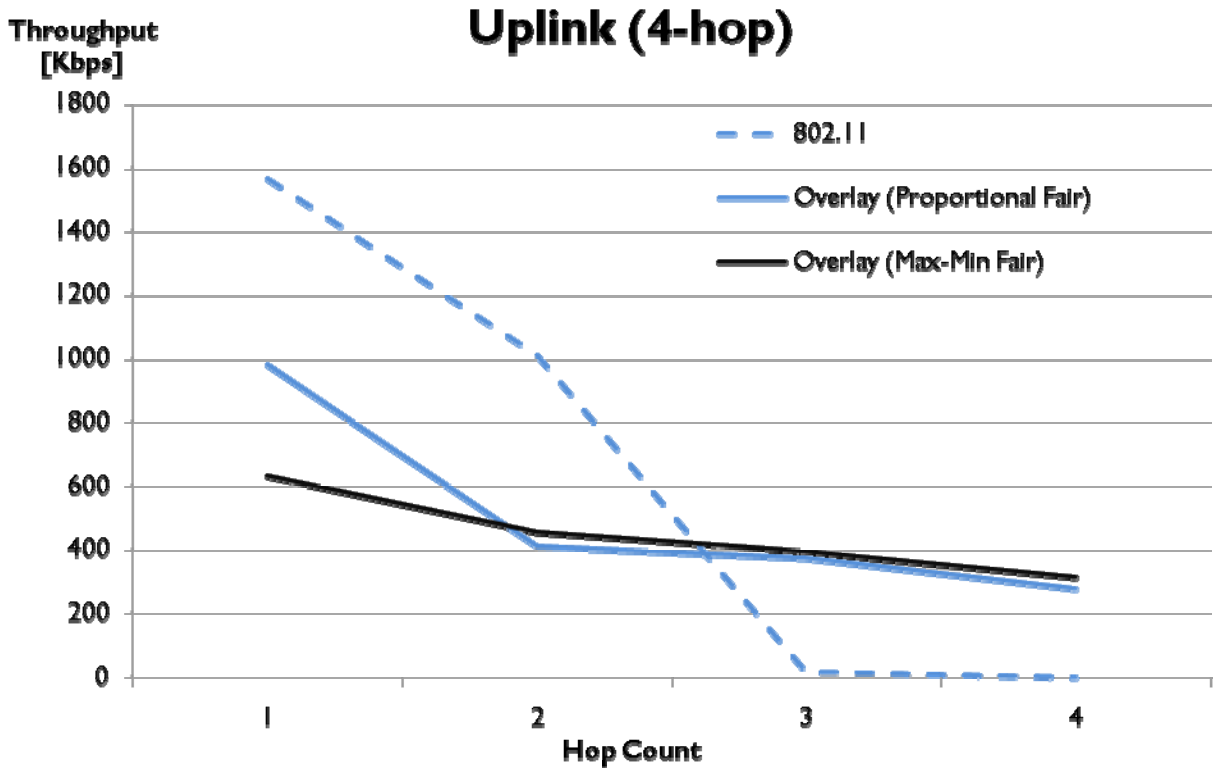


KAIST Wireless Mesh Testbed



Experimentation

- Network Engineering by Utility Function



References

- [Feldmann07] A. Feldmann, "Internet Clean-Slate Design: What and Why?," ACM SIGCOMM Computer Communication Review, Vol.37, No.3, 2007, pp.59-64.
- [Dovrolis08] C. Dovrolis, "What would Darwin Think about Clean-Slate Architectures?," ACM SIGCOMM Computer Communication Review, Vol.38, No.1, Jan. 2008.
- [Kropff06] M. Kropff, T. Krop, M. Hollick, P. S. Mogre, and R. Steinmetz, "A Survey on Real World and Emulation Testbeds for Mobile Ad Hoc Networks," TRIDENTCOM 2006.
- [Krop07] T. Krop, M. Bredel, M. Hollick, and R. Steinmetz, "JiST/MobNet: Combined Simulation, Emulation, and Real-World Testbed for Ad hoc Networks," WINTECH 2007.
- [Barr04] R. Barr, "JiST: Embedding Simulation Time into a Virtual Machine," In Proceedings of EuroSim Congress on Modeling and Simulation, Ithaca, USA, 2004.
- [Raych05] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracua, H. Liu, and M. Singh, "Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols," Proc. of WCNC 2005.
- [Peter02] L. Peterson, T. Anderson, D. Culler, and T. Roscoe, "A Blueprint for Introducing Disruptive Technology into the Internet," Proc. of HotNet-I, October 2002.
- [Judd04] G. Judd and P. Steenkiste, "Repeatable and Realistic Wireless Experimentation Through Physical Emulation," ACM SIGCOMM Computer Communications Review, 34(1):63-68, January 2004.
- [Kohler00] E. Kohler, "The Click Modular Router," Ph.D. thesis, MIT, November 2000.
- [Johnson05] D. Johnson, T. Stack, R. Fish, D. Flickinger, R. Ricci, and J. Lepreau, "TrueMobile: A Mobile Robotic Wireless and Sensor Network Testbed," Technical Report FTN-2005-02, University of Utah Flux Group, April 2005.
- [Tiss92] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," IEEE Trans. Automatic Control, vol. 37, no. 12, Dec 1992.
- [Chen06] L. Chen, S.H. Low, M. Chiang, and J.C. Doyle, "Cross-Layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks," IEEE INFOCOM 2006.

References [Contd.]

- [Lin06] X. Lin, N.B. Shroff, "The Impact of Imperfect Scheduling on Cross-Layer Rate Control in Wireless Networks," IEEE INFOCOM 2006.
- [Leo01] E. Leonardi, M. Mellia, F. Neri, and M.A. Marsan, "On the stability of input-queued switches with speed-up," IEEE/ACM Transactions on Networking, vol.9, no.1, pp.104-118, Feb. 2001.
- [NS2] http://www.isi.edu/nsnam/ns/
- [NS3] http://www.nsnam.org/
- [GloMoSim] http://pcl.cs.ucla.edu/projects/glomosim/
- [QualNet] http://www.scalable-networks.com/
- [OPNET] http://www.opnet.com/
- [WiMesh] http://143.248.238.6/~sachin/realization/
- [RoofNet] http://pdos.csail.mit.edu/roofnet/docu.php