Aalto University
School of Science

Experimentally Driven Research

T-110.6130 Systems Engineering in Data Communications Software

Matti Siekkinen

30.10.2012

Outline

- Experimentally driven research
  - What and why?
  - How to do it?
  - What are the challenges?
- Testbeds and tools
  - Currently available
  - Coming in the future
- Conclusions

What is experimentally driven research?

- Study how solutions (protocol, algorithm) work in operational setup
- Goals:
  - Understand how individual solutions behave in real world
  - Understand what are interactions between individual solutions
  - Improve the behavior through redesign or novel solutions
- How?
  - Design, implement, deploy, run, and measure

What is experimentation?

- Q: How is experimentation defined?
- A: It isn't…

What’s the difference?

- Testing
- Experimentation
- Benchmarking
- Trial
- Pilot

- Following slides are an attempt to define terminology by “Experimentally Driven Research” FIREworks (EU-ICT) white paper

Experimentation vs. Testing

- Testing is well defined, e.g. by ETSI
  - Conformance testing, Interoperability testing
- Experimentation (in our discipline) is not well defined
  - The orderly or methodical observation of the variation of facts resulting from artificial stimuli in a reproducible environment that confirms a hypothesis (verification) or rejects it (falsification)
- In short:
  - Modify stimuli … observe impact
  - Hypothesis is valid if we can show that an experiment can be reproduced

"Experimentally Driven Research" FIREworks white paper

Experimentation vs. Testing (cont.)

- Knowledge about the right stimuli and observation points of a system?
  - **Testing:** known list of stimuli and observation points for checking correctness
  - **Experimentation:** search for the right stimuli and observation points for assessment
- Level of maturity of the knowledge about the behavior of a system?
  - Draw a line between **research** (experimentation) and **development** (testing)

“Experimentally Driven Research” FIREworks white paper

Experimentation vs. Benchmarking

- **Benchmarking**
  - Controlled (often optimal) conditions
  - Measure functional and/or performance metrics
  - Compare results to an existing specification
    - E.g. well-accepted industry standard

“Experimentally Driven Research” FIREworks white paper

Ensuring quality in experimentation

- Some important properties of experimentation
  - Verifiability
    - We can find a formal model that is equivalent to experimental model (produces same results than the experiments)
  - Reliability
    - Probability that system will perform its intended function during a specified period of time under stated conditions
  - Repeatability
    - We get always the same result with same parameters and input variables
  - Reproducibility
    - We get same results with same experimentation setup on a different experimental system

"Experimentally Driven Research" FIREworks white paper

Scale and cost of experimentation

- Large scale?
  - In LSI/VLSI circuit design it was large-scale 100-5000 circuit elements very-large-scale (5k-50k), super-large-scale (50k-100k) ultra-large-scale (>100k) …
- **Cost** factor as a measure
  - Cost is a function of complexity, dimension and environmental conditions
  - **Large-scale:** cost exceeds the cost of a laboratory environment by one or more levels of magnitude

“Experimentally Driven Research” FIREworks white paper

Why is experimentally driven research necessary?

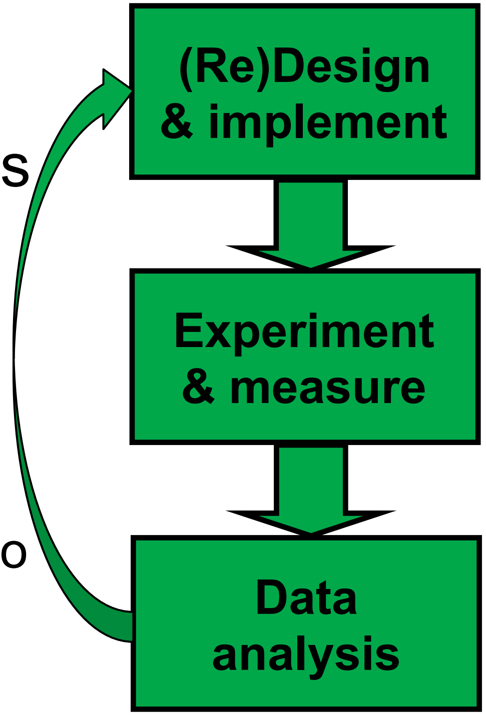
- Evaluate solutions even if…
  - analytical modeling of their behavior is difficult
  - simulation is unfeasible (does not scale enough)
  - suitable simulation tools do not exist
- See non-trivial dependencies between parameters
  - Simulations are only as good as the models they rely on
- Provide input for simulators
  - Think about where workloads and simulation models come from
  - E.g. topologies for simulating routing protocols

Outline

- Experimentally driven research
  - What and why?
  - → How to do it?
  - What are the challenges?
- Testbeds and tools
  - Currently available
  - Coming in the future
- Conclusions

Typical process

- (Re)Design & implement solution and deploy it
  - E.g. overlay routing protocol
- Experimentation and collection of measurements
  - Traffic traces, application logs, etc.
  - Input for data analysis
- Inference / Analysis
  - Can be also integrated into measurement
  - Learn how solution behaves and (optionally) go back to beginning



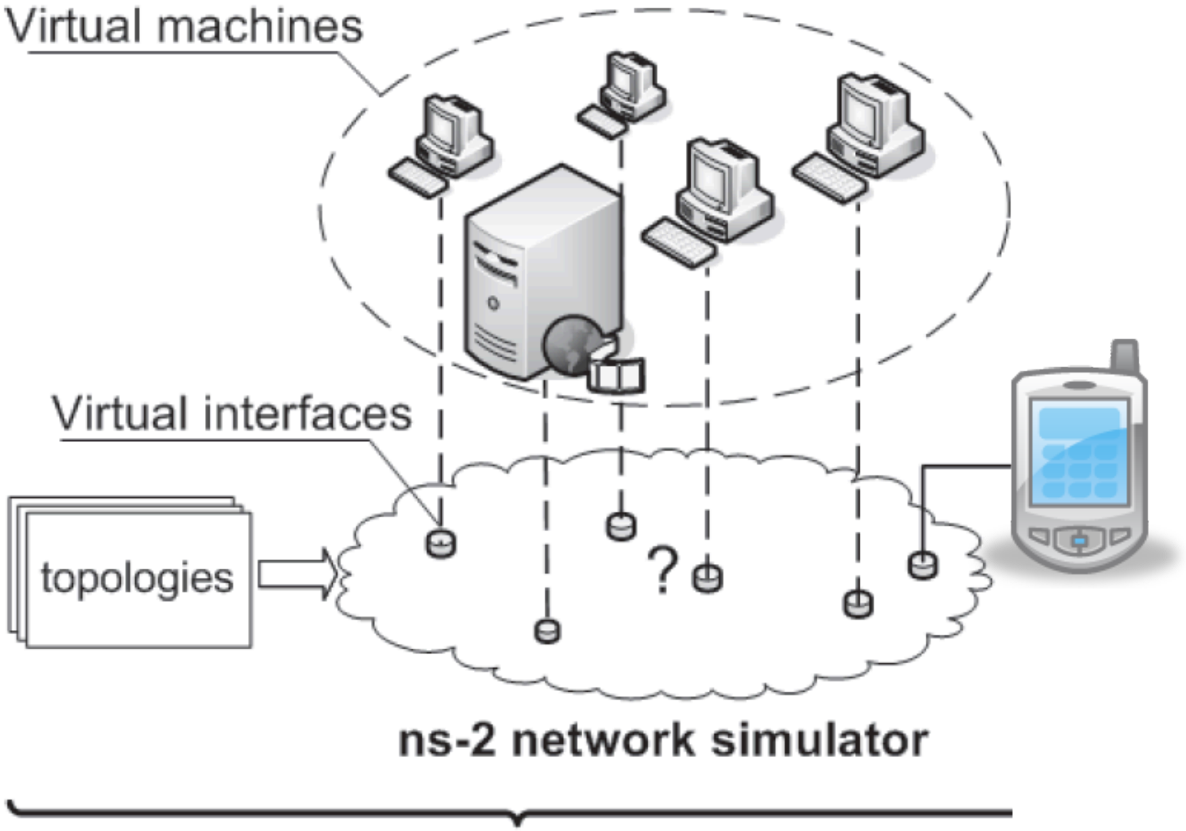
Experimentation approaches

- Emulations
- Closed proprietary testbeds
  - Usually small or medium size
- Open testbeds
  - Small to large scale
- The Internet
  - Huge scale

Emulations

- In our discipline: Something in between simulations and real testbeds
- Part of the system is real
  - Reproduces exactly the original kind of behavior
  - Whereas simulated system behaves according to a model
- Typical case:
  - Real devices (or virtual machines) running a real application
    - As opposed to application workload for simulator
  - Simulated network
    - NS-2/3
- What's the point?
  - Can measure behavior of real devices and applications over controlled network environment
  - Evaluate how network parameters impact device/application behavior

Emulation example



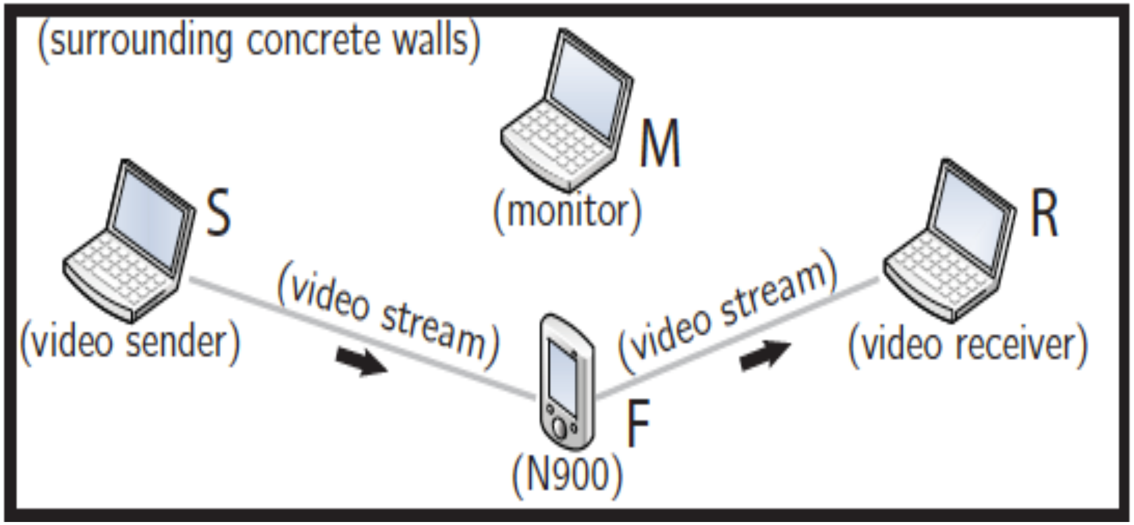
single computer

Closed testbeds

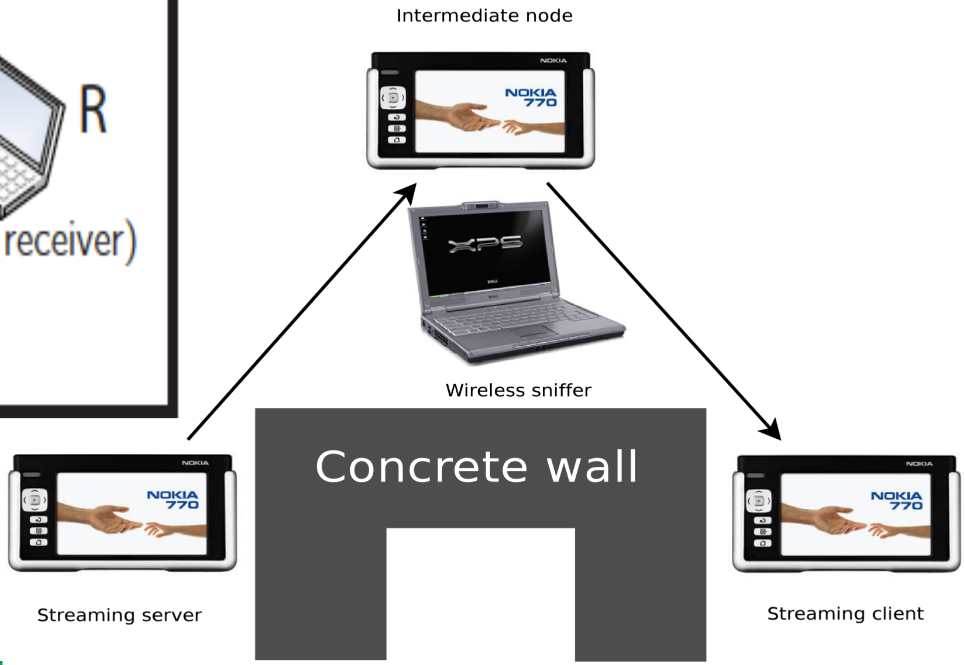
- You can build your own testbed
  - In your own lab
  - Not open to others
  - Often small scale
- Advantages
  - "Easy" to control
  - (Should) know exactly what is going on
  - "Easy" to collect measurements
- Problems
  - Can be substantial amount of work
    - Even if small scale
  - Reproducibility of results can be questionable
    - Can you control all the related parameters?

Example of closed testbed

- Very small multi-hop streaming setups



- Concrete walls for limiting interference



Open testbeds

- Testbeds that are built for use by others as well
- Small to large scale
  - Small/medium: single lab setup
  - Large: cooperative testbed/platform
- There are several available for research purposes
  - Planetlab, emulab, …
- The Internet is an extreme case
  - Open to everyone
  - So, can be viewed as one very large open testbed
  - The most realistic case for solutions targeted for Internet deployment

Outline

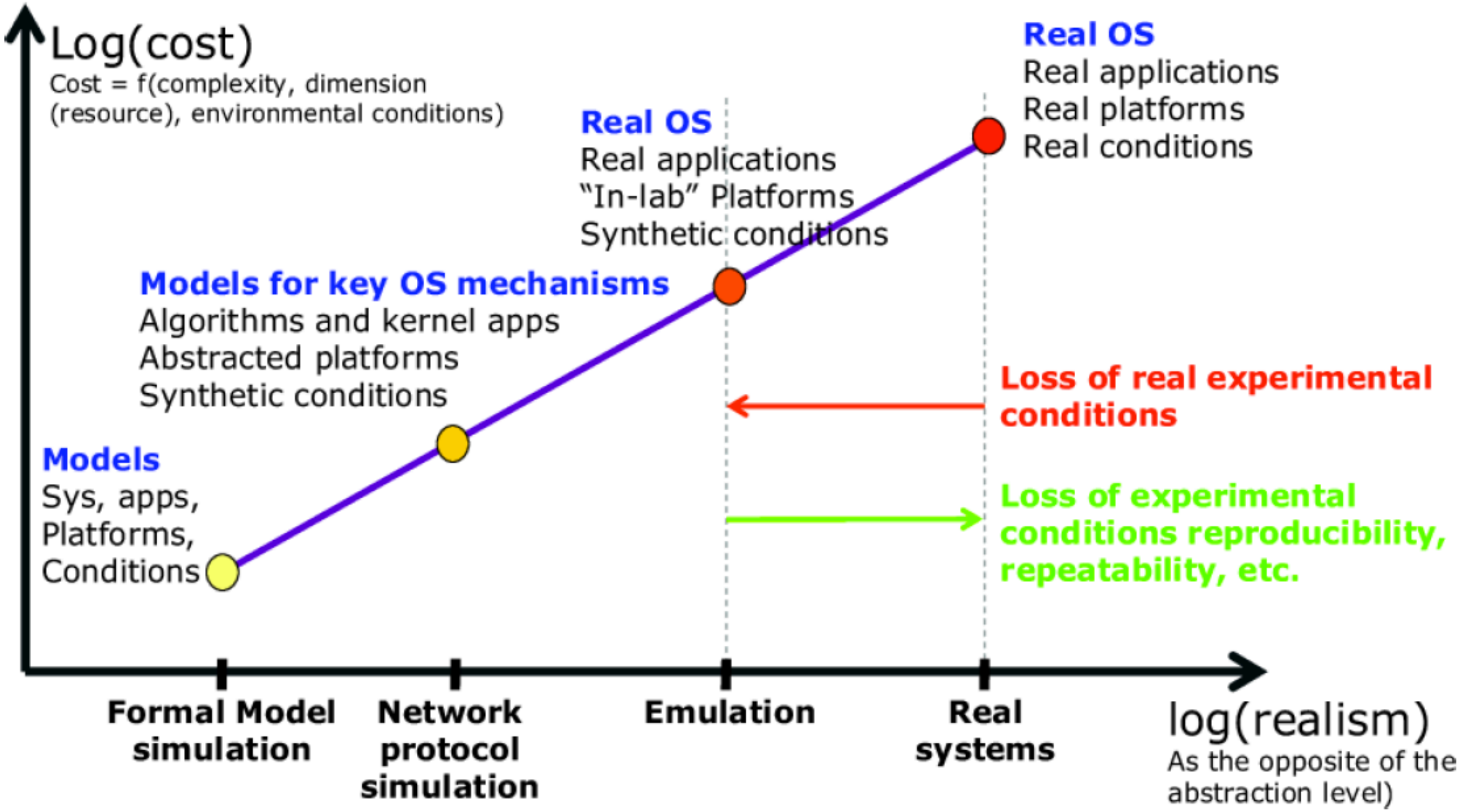
- Experimentally driven research
  - What and why?
  - How to do it?
  - What are the challenges?
- Testbeds and tools
  - Currently available
  - Coming in the future
- Conclusions

Why is experimentation challenging?

- Usually cannot experiment with operational networks
  - No architectural support
  - How to deploy solutions?
  - How to do measurements?
- Cost
  - Need specialized solutions
  - Need large scale deployments
    - We often want Internet like realism from evaluation
  - Both costs, labor and equipment
- Availability of means to do it
  - tools, platforms, networks, etc.
  - cost issue

Simulation → Emulation → Experimentation

- Cost vs. realism



Outline

- Experimentally driven research
  - What and why?
  - How to do it?
  - What are the challenges?
- Testbeds and tools
  - → Currently available
  - Coming in the future
- Conclusions

What is available?

Now

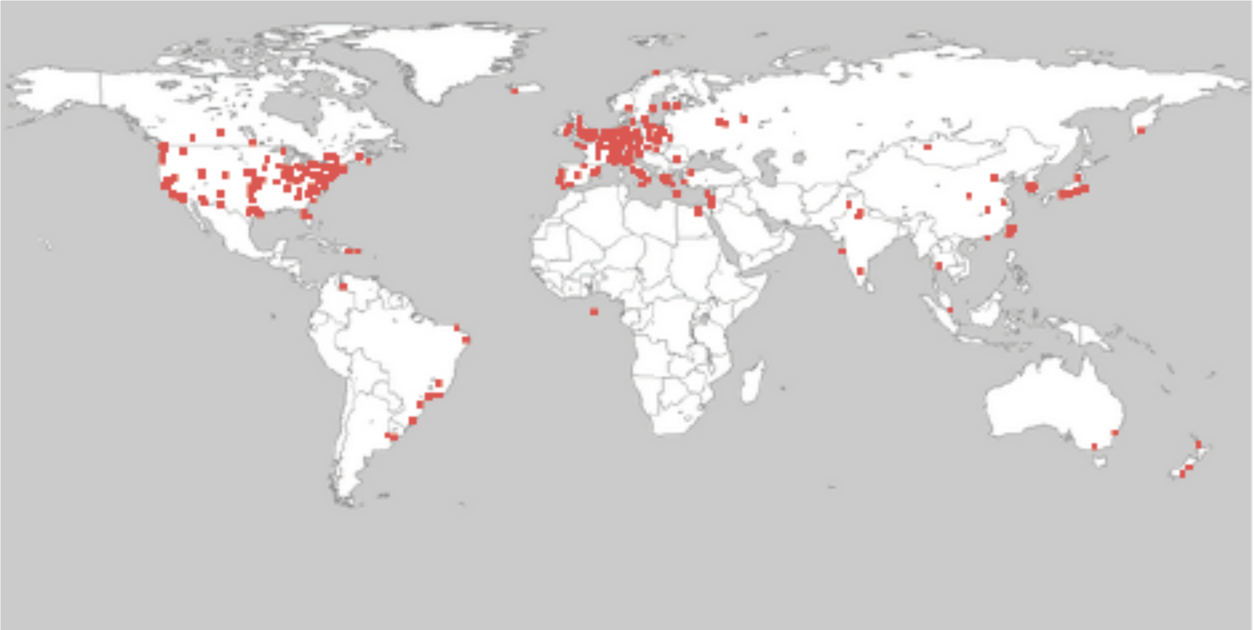
- Testbeds
  - PlanetLab
  - Emulab
  - ORBIT
  - NITOS
  - CitySense
  - WISEBED
- Alternative approaches
  - OpenFlow
  - Click
  - NetFPGA

In the future

- Testbeds
  - GENI
  - SmartSantander
  - …
- Federated testbeds

PlanetLab

- Global distributed system infrastructure
  - platform for long running services
  - testbed for network experiments



Planetlab: facts and figures

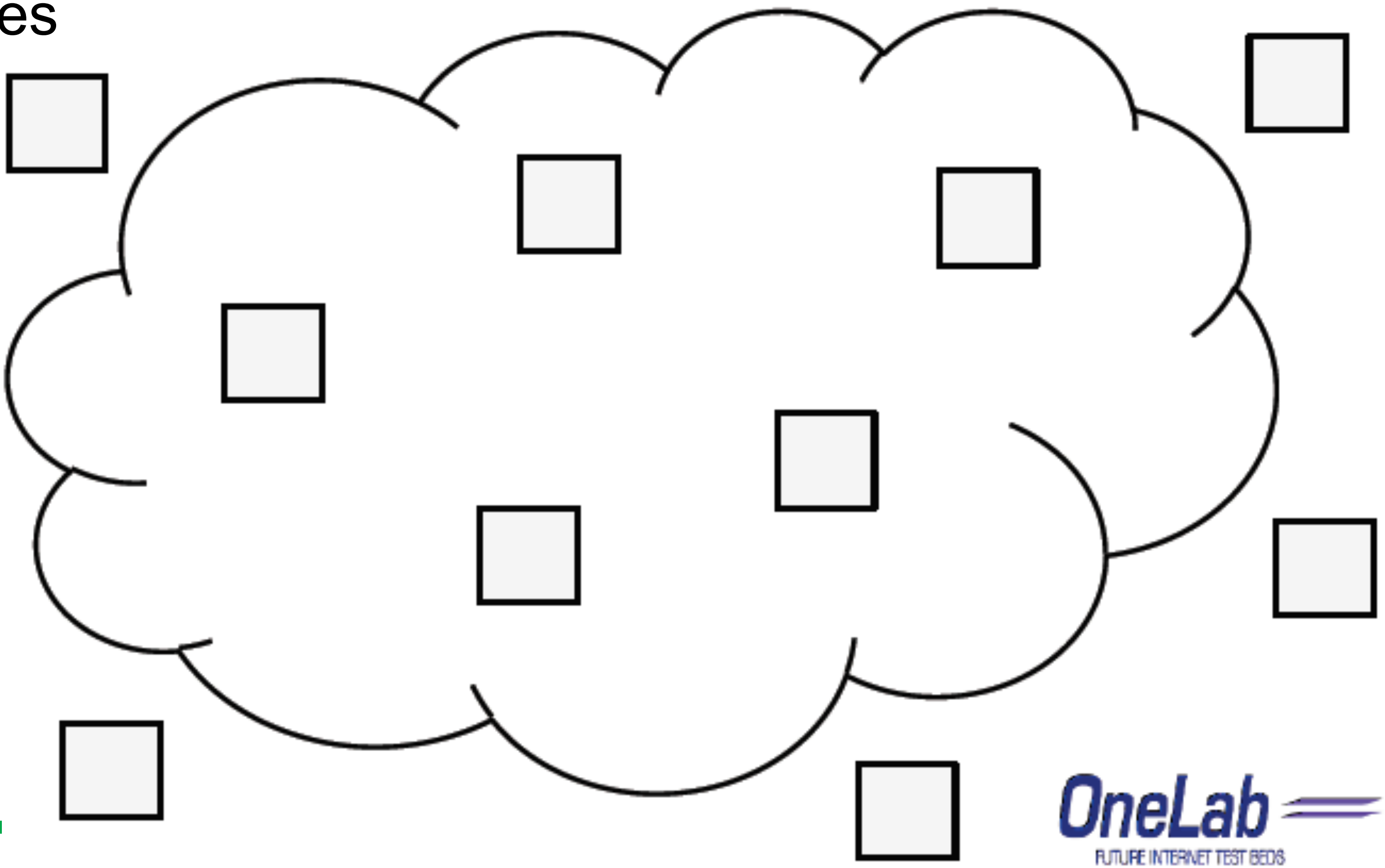
- Launched in March 2002
- 1011 nodes around the world
  - 35 countries
  - 545 sites (universities, research labs)
  - 1143 nodes
- A collection of machines distributed around the globe
  - Most of the machines are hosted by research institutions
- All machines are connected to the Internet
- All machines are administered by a system called MyPLC
  - The software is supported on several flavors of Linux

PlanetLab architecture

- **Site:** physical location of a PlanetLab node
  - E.g. Fraunhofer Institute or UCL
- **Node:** server that runs components of PlanetLab services
- **Slice:** set of allocated resources distributed across PlanetLab
  - UNIX shell access to private virtual servers on a selected set of PlanetLab nodes
  - User assigns a set of PlanetLab nodes to a slice
    - Virtual servers for that slice are created on each of the assigned nodes
- **Sliver:** slice running on a specific node
  - You can use ssh to login to a sliver on a specific node
- Fair share allocation of resources per sliver
  - CPU and link bandwidth

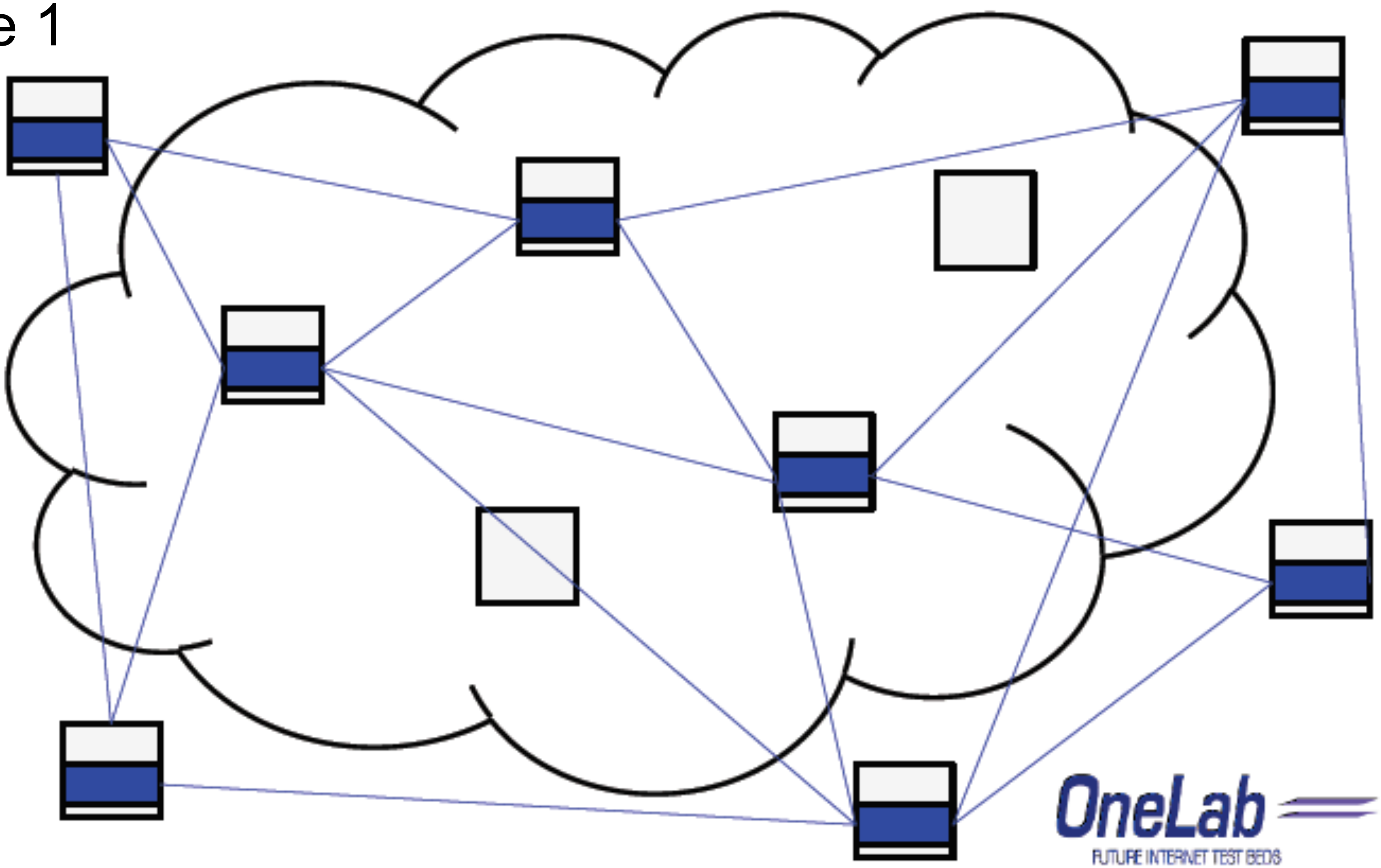
PlanetLab architecture (cont.)

- Nodes



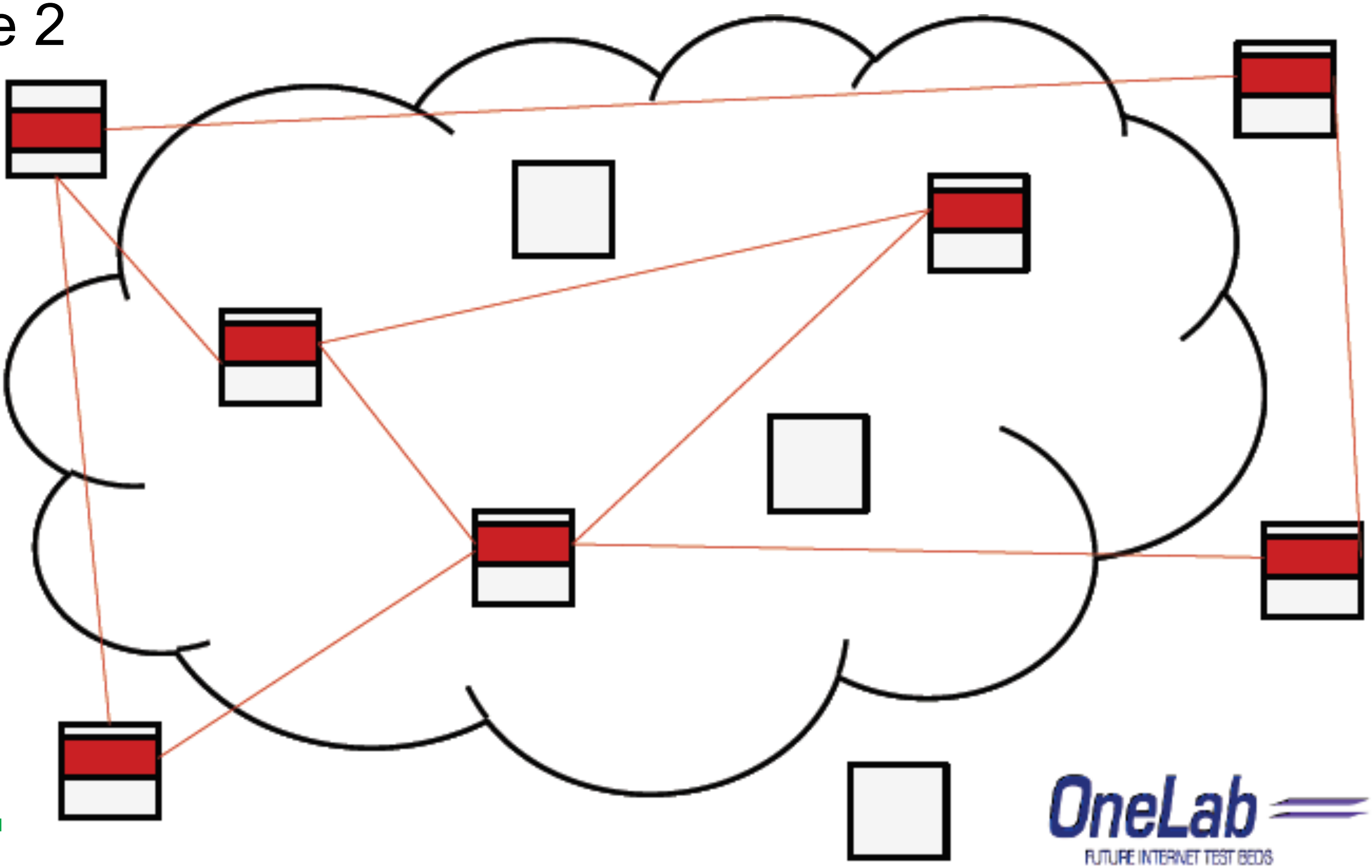
PlanetLab architecture (cont.)

- Slice 1



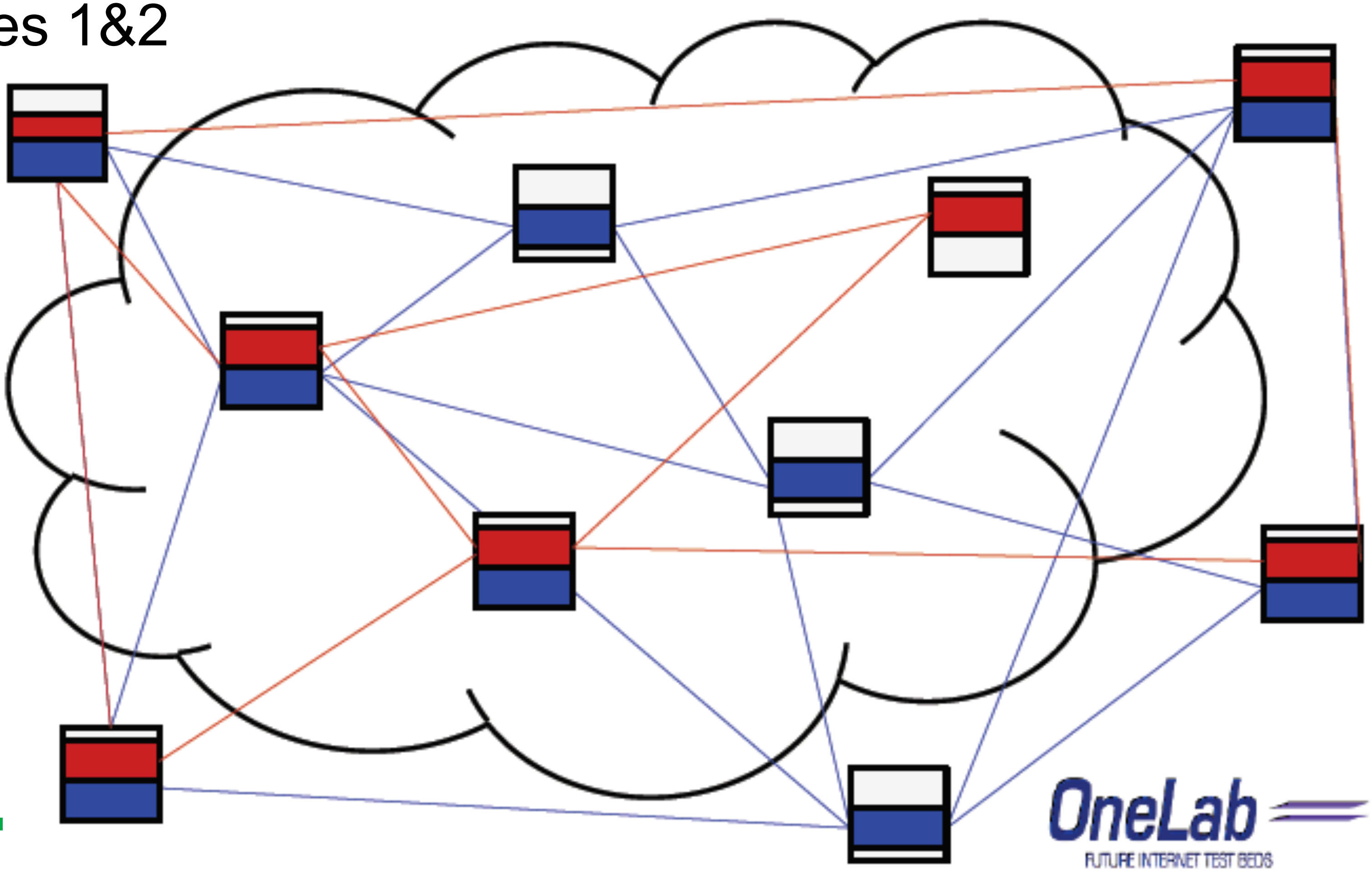
PlanetLab architecture (cont.)

- Slice 2



PlanetLab architecture (cont.)

- Slices 1&2



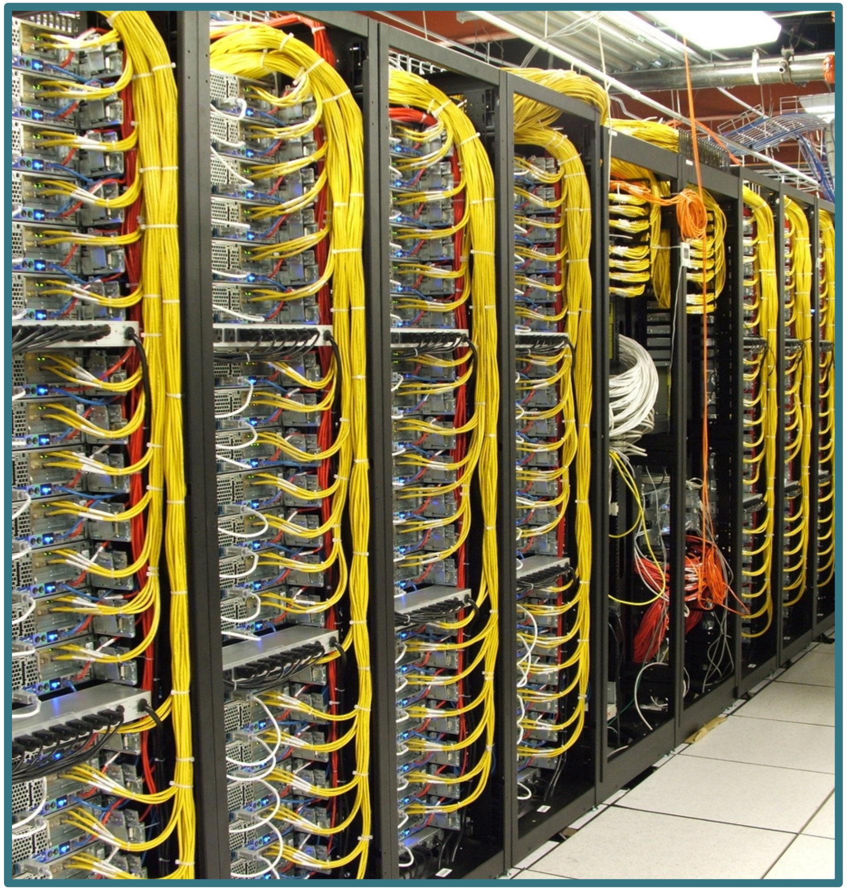
Using PlanetLab

- Central Website that manages
  - All accounts
  - All nodes
  - All resources
- Registering with one of 3 PLCs (your PlanetLab Central)
  - PL USA (planet-lab.com)
  - PL Europe (planet-lab.eu)
  - PL Japan (planet-lab.jp)
- Different kinds of memberships available depending on kind of institution
  - HIIT and Aalto (Comnet@ELEC) are members

PlanetLab: pros and cons

- Pros:
  - Access to large scale distributed resources
  - Run experiments with complete control over each node
    - Restricted root access
  - Connectivity over real Internet paths
  - Scale from one to few to many nodes
  - Monitor CPU and network traffic
  - Deploy long-running experimental services
- Cons:
  - Shared resources
    - Crowded machines can cause biased experiments
  - Stability
    - Nodes go down
    - Management often not high priority

Emulab

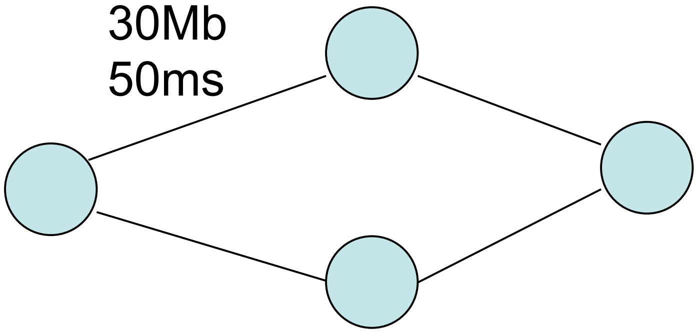


- Network testbed
  - Provides remote access to custom emulated networks
- Developed at University of Utah around the year 2000
  - Currently many deployments around the world exist
  - Private, open, specific purpose (e.g. teaching, research, security research) …
- Who can use it?
  - Open deployments can be used by external researchers

Emulab Characteristics

- Physical viewpoint: Large switched LAN with control software
- How it works
  - Creates custom network topologies specified by users in NS
  - Software manages PC cluster, switching fabric
- User can
  - Replace node OS software
  - Configure link topology
    - Uses Virtual LANs to implement arbitrary and isolated topologies
  - Control “physical” link characteristics
    - Shape latency/bandwidth/drops/errors
    - Done via invisibly interposed “shaping nodes”
- Also has a set of wireless 802.11 capable nodes
  - Both infrastructure and ad-hoc mode supported

Emulab: Example Topology Configuration

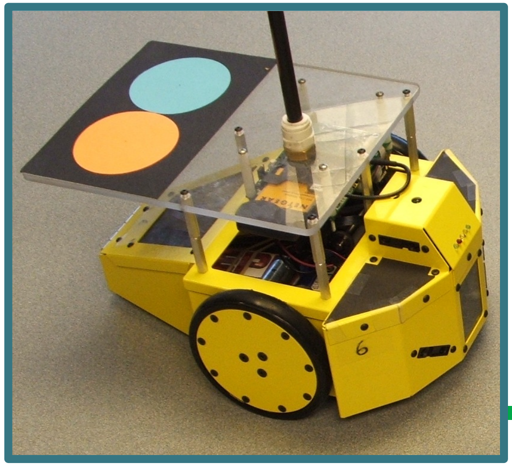


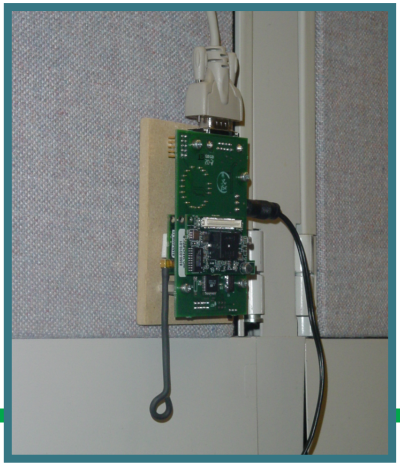
```
set ns [new Simulator]
source tb_compat.tcl

set nodeA [$ns node]
set nodeB [$ns node]
set nodeC [$ns node]
set nodeD [$ns node]

set link0 [$ns duplex-link $nodeA $nodeB 30Mb 50ms DropTail]
set link1 [$ns duplex-link $nodeA $nodeC 30Mb 50ms DropTail]
set link2 [$ns duplex-link $nodeC $nodeD 30Mb 50ms DropTail]
set link3 [$ns duplex-link $nodeB $nodeD 30Mb 50ms DropTail]

$ns rtproto Static
$ns run
```

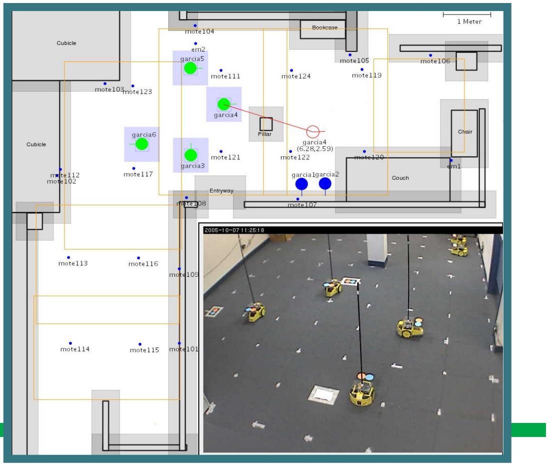
- Network testbed mapping problem
  - Map virtual network to shared physical resources
    - I.e. select hardware on which to instantiate network experiments
  - Optimization problem with set of constraints
  - Ricci et al: “A Solver for the Network Testbed Mapping Problem”. In SIGCOMM CCR. 2003.

Emulab: Mobile Sensor Additions

- Several user-controllable mobile robots
  - Onboard PDA, WiFi, and attached sensor mote
- Many fixed motes surround motion area
  - Simple mass reprogramming tool
  - Configurable packet logging
  - …







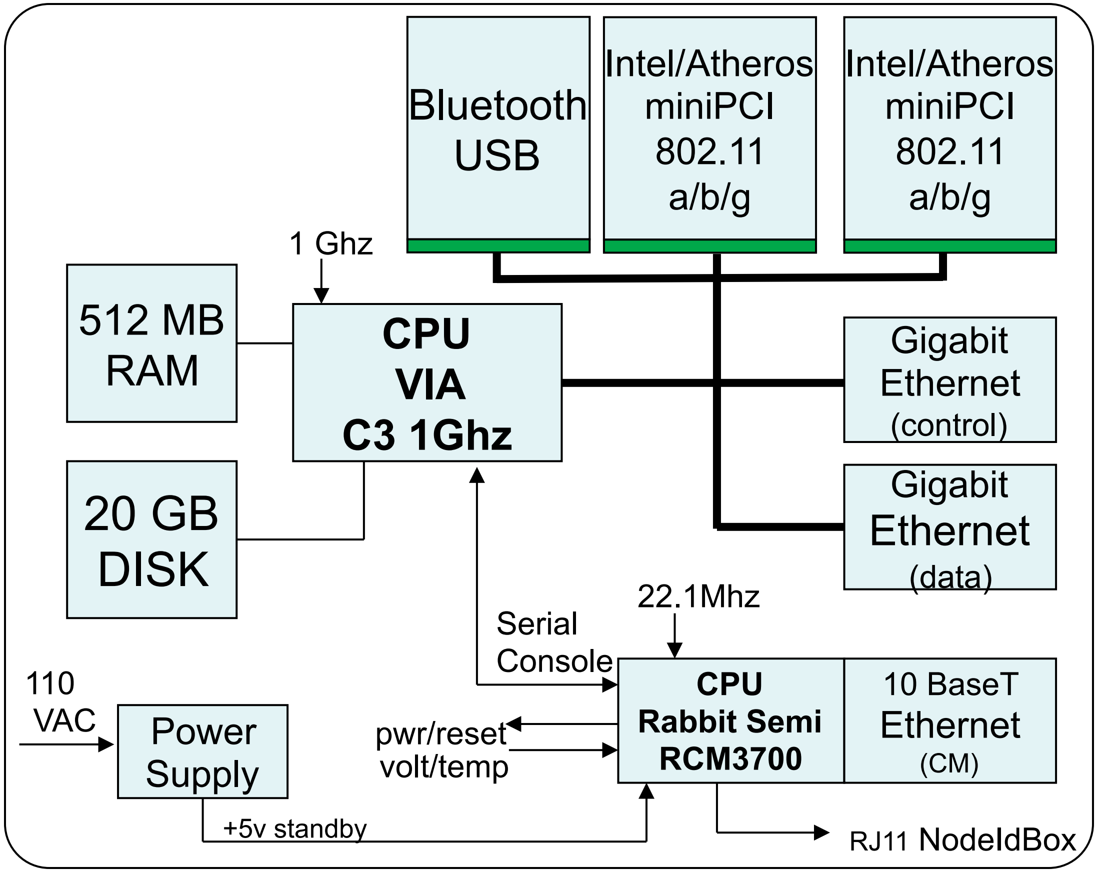
ORBIT

- Open-Access Research Testbed for Next-Generation Wireless Networks
- Developed (2005) and operated by WINLAB, Rutgers University

ORBIT characteristics

- Wireless testbed
  - Scale: total nodes ~100's
- Controlled environment
  - Reproducible experiments
  - User can control protocols and software used on the radio nodes
- Measurement capabilities on radio PHY, MAC and network levels
- Testbed is remotely accessible
  - unmanned operation
  - able to deal with software and hardware failures
- Who can use it?
  - Universities, industrial research labs
  - Both US and non-US
- Seems to be still active based on user mailing list
  - Seems also not to have completely automatic failure handling…

Orbit radio node



Bluetooth USB

Intel/Atheros miniPCI 802.11 a/b/g

Intel/Atheros miniPCI 802.11 a/b/g

1 Ghz

512 MB RAM

CPU VIA C3 1Ghz

Gigabit Ethernet (control)

20 GB DISK

Gigabit Ethernet (data)

22.1Mhz

Serial Console

110 VAC

Power Supply

pwr/reset

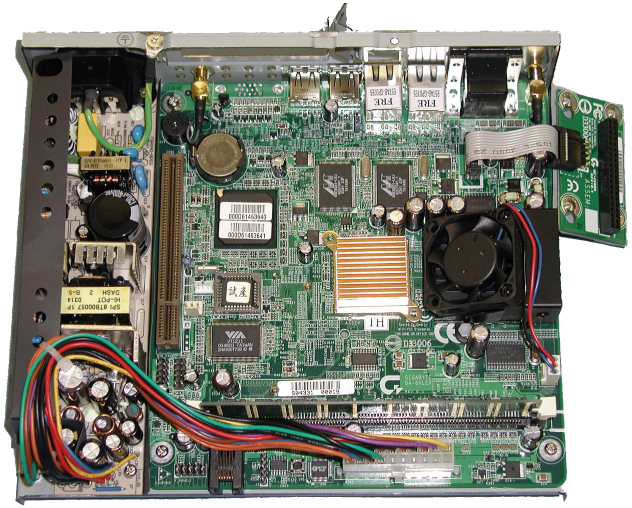
volt/temp

CPU Rabbit Semi RCM3700

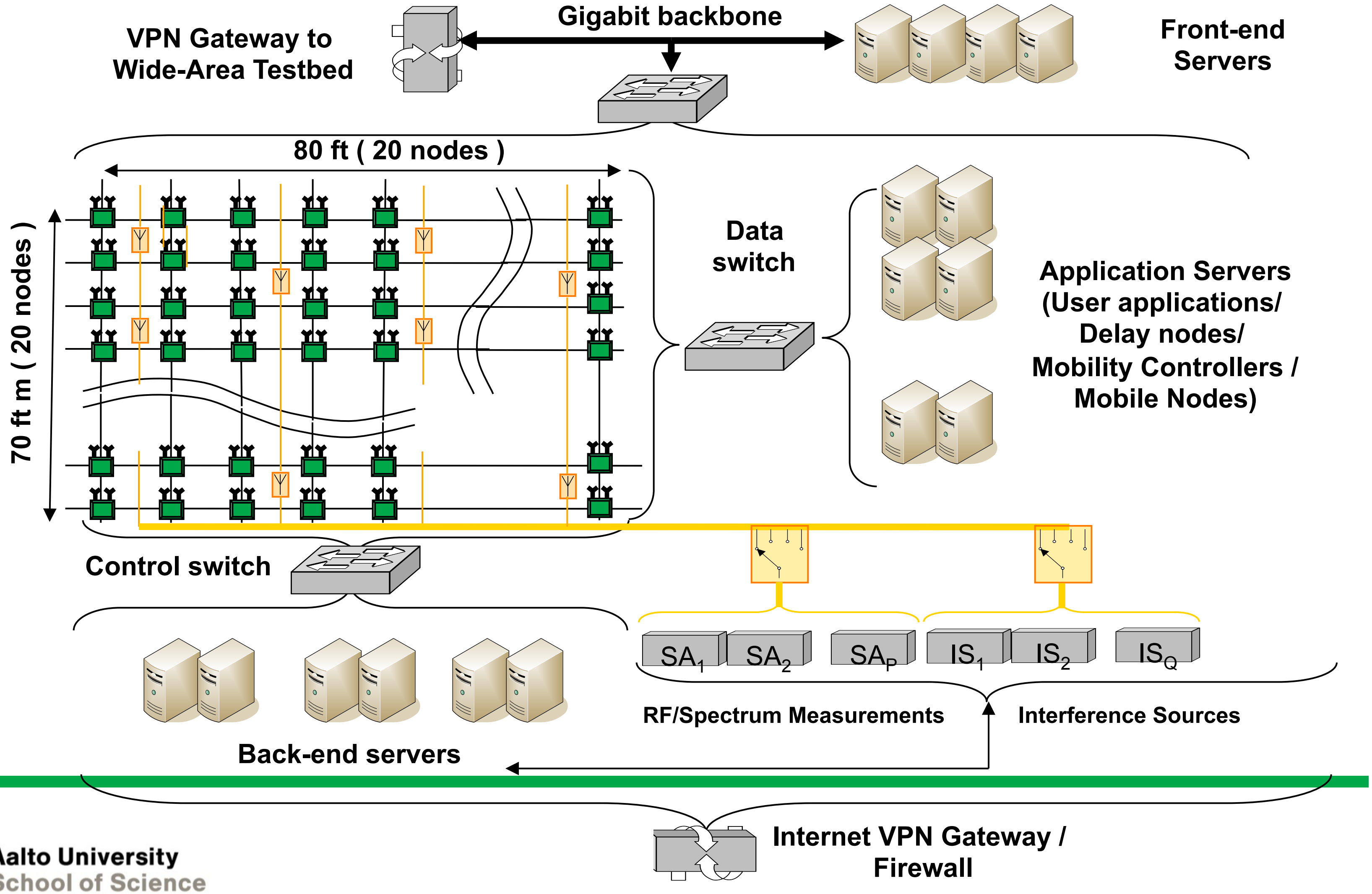
10 BaseT Ethernet (CM)

+5v standby

RJ11 NodeIdBox



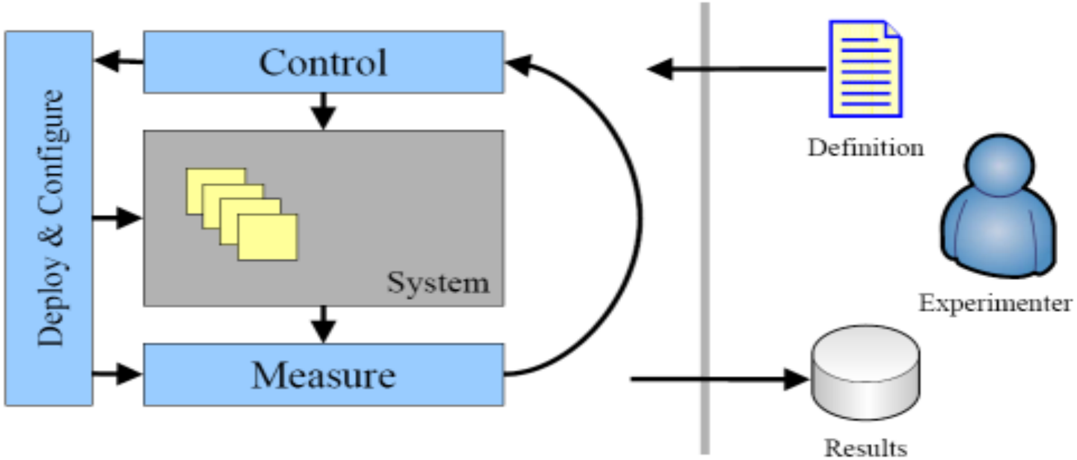
ORBIT: Indoor Grid



Aalto University
School of Science

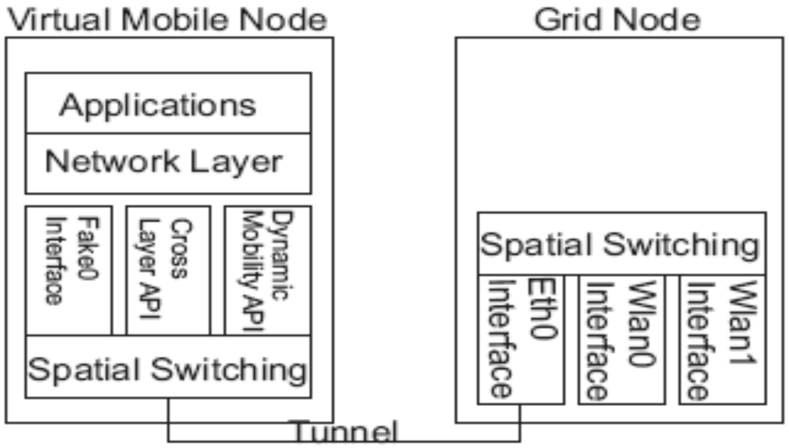
Running experiments

- Unlike wired testbeds, difficult to isolate experiments
  - Serial mode of operation
  - Quickly offload users at the end of the slot
- Scheduling system for requesting and allocating slots for experiments



What about mobility in ORBIT?

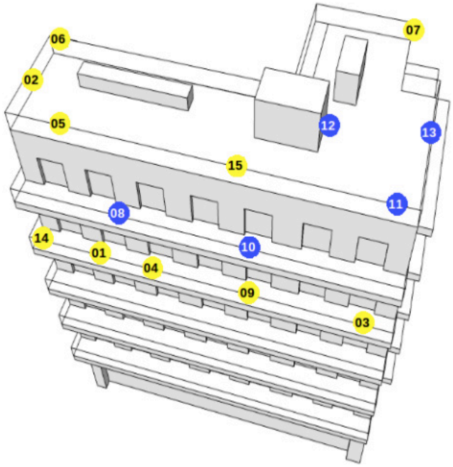
- Virtual mobility
- Emulates trajectory by switching to different radio and antenna positions as time progresses
- Discretized grid mobility
  - Virtual mobile node appears to be at location i by using radio grid node i
  - Uses grid radio node j when it "moves" to grid location j
  - Nothing moves physically



Other existing testbeds

- Wireless testbeds built on OMF
  - Open source ctrl & mgmt software
    - NICTA and Winlab (Orbit)
  - NITOS
    - Wireless (802.11) testbed at NICTA
    - Orbit nodes and diskless nodes (45 total)
    - Multi-user experimentation
      - Spectrum slicing (channel allocation)
  - Many others, mostly small closed testbeds
- CitySense
  - Urban-Scale Sensor Network Testbed
  - 26 weather, CO2, and noise pollution sensor nodes deployed in Cambridge, MA
    - Programmable by users
  - Seems inactive since a couple of years

University of Thessaly's campus building (Greece)



Other existing testbeds (cont.)

- WISEBED
  - “Multi-level infrastructure of interconnected testbeds of large-scale wireless sensor networks for research purposes”
  - 9 testbeds around Europe
    - Indoor WSNs
    - PHY/MAC: mostly 802.15.4, also some non-standard solutions used
  - Web based clients to manage experiments
    - Have developed APIs to expose the WSN services
  - EU project
    - Already finished
    - Not sure if these are really usable…

Outline

- Experimentally driven research
  - What and why?
  - How to do it?
  - What are the challenges?
- Testbeds and tools
  - Currently available
    - → Use cases
  - Coming in the future
- Conclusions

PlanetLab use cases

- EGOIST [1-4]
  - Overlay routing solution
  - Evaluated on PlanetLab over a period of two years
  - Extensive measurements of paths between 50 PL nodes
  - Demonstrated
    - superiority of EGOIST's neighbour selection primitives over existing heuristics
    - effectiveness under significant churn, resistance to cheating, and small overhead
- Radar: Internet topology[5-7]
  - "Internet radar" which "draws" the Internet topology as it evolves over time
  - runs periodic tree-like manner traceroutes to a set of destinations
    - reduces the induced traffic and data redundancy
  - 150 PlanetLab nodes as monitors around the world
    - Observes differences between geographically diverse parts of the network

Emulab use cases

- VOID (Vvireless Online Interference Detector) [8]
  - Non intrusive interference detection for 802.11 networks
    - Apply statistical methods on throughput summaries at upstream wired routers
    - Figure out which devices are the interferers -> necessary to know before can minimize the interference
  - Showed effectiveness with Emulab’s wireless features
- Scarlett [9]
  - Skewed content popularity causes performance bottlenecks in MapReduce file systems
    - Uniform data replication
    - Contention for slots on machines storing more popular blocks
  - Scarlett replicates files based on their access patterns
    - Spreads out to avoid hot spots
  - Part of evaluation done with Hadoop running on DETER testbed (Emulab at USC ISI and UC Berkeley mainly for security research)

ORBIT use case

- Evaluating TCP Simultaneous Send Problem over 802.11 [10]
  - Problem: high MAC contention between data and ACK packets during bulk transfer
  - Solution: ACK skipping
  - Both from earlier analytic and simulation studies
  - Use ORBIT to evaluate problem and solution
  - Results
    - Confirm severity of problem
    - Confirm that ACK skipping alleviates problem
    - However, not as much as simulations suggested
      - Due to TCP implementation differences

Use case references

[1] G. Smaragdakis, N. Laoutaris, P. Michiardi, A. Bestavros, J. W. Byers, M. Roussopoulos. Distributed Network Formation for n-way Broadcast Applications. IEEE Transactions on Parallel and Distributed Systems, 21(10), 2010.

[2] G. Smaragdakis, N. Laoutaris, V. Lekakis, A. Bestavros, J. W. Byers, M. Roussopoulos. EGOIST: Overlay Routing using Selfish Neighbor Selection. ACM CoNEXT 2008.

[3] G. Smaragdakis, N. Laoutaris, P. Michiardi, A. Bestavros, J. W. Byers, M. Roussopoulos. Swarming on Optimized Graphs for n-way Broadcast. IEEE INFOCOM 2008.

[4] N. Laoutaris, G. Smaragdakis, A. Bestavros, J. W. Byers. Implications of Selfish Neighbor Selection in Overlay Networks. IEEE INFOCOM 2007.

[5] A. Hamzaoui, M. Latapy, C. Magnien. Detecting events in the dynamics of ego-centered measurements of the Internet topology. Proceedings of International Workshop on Dynamic Networks (WDN), in conjunction with WiOpt 2010.

[6] C. Magnien, F. Ouedraogo, G. Valadon, M. Latapy. Fast dynamics in Internet topology: preliminary observations and explanations. Fourth International Conference on Internet Monitoring and Protection (ICIMP 2009), May 24-28, 2009, Venice, Italy.

[7] M. Latapy, C. Magnien, F. Ouédraogo. A Radar for the Internet. Proceedings of ADN'08: 1st International Workshop on Analysis of Dynamic Networks, in conjonction with IEEE ICDM 2008.

[8] Cai, K., Blackstock, M., Feeley, M. J., and Krasic, C. 2009. Non-intrusive, dynamic interference detection for 802.11 networks. In Proceedings of the 9th ACM Internet Measurement Conference. IMC 2009.

[9] Ganesh Ananthanarayanan, Sameer Agarwal, Srikanth Kandula, Albert Greenberg, Ion Stoica, Duke Harlan and Ed Harris. Scarlett: coping with skewed content popularity in mapreduce clusters. In Proceedings of the sixth conference on Computer systems (EuroSys '11), 2011.

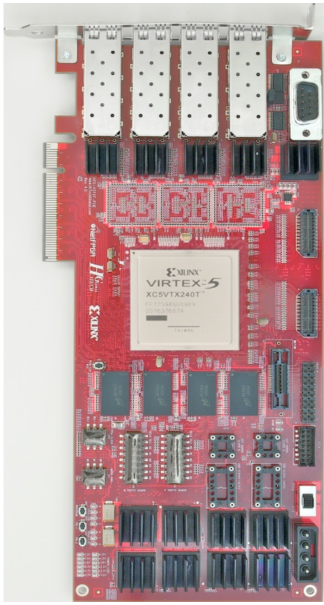
[10] Sumathi Gopal and D. Raychaudhuri, "Experimental Evaluation of the TCP Simultaneous Send Problem over 802.11 Wireless Local Area Networks", Proceedings of the ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-Wind) 2005.

Outline

- Experimentally driven research
  - What and why?
  - How to do it?
  - What are the challenges?
- Testbeds and tools
  - Currently available
    - → Alternative approaches
  - Coming in the future
- Conclusions

Alternative approaches (or how to build your own testbed)

- Instead of building testbeds, provide customizable boxes
- Click Modular Router
  - Extensible toolkit for writing packet processors
  - Way to create a custom software router
  - Performance is of course limited…
- NetFPGA
  - Targeted for teaching and research
  - Low-cost PCI card with a user-programmable FPGA for processing packets
  - Basic is 4 ports of Gigabit Ethernet (only…)
    - 1199$ (commercial) or 599$ (academic)
  - 4x10GE version was announced end of 2010
    - $1,675 (academic)
  - More powerful than software
  - More complex to use
- OpenFlow



OpenFlow: Motivation

- Experimenters want to try out their solutions on real networking devices
- Vendors don’t want to open interfaces inside their boxes
  - Reliability, competition
  - Experimenters dream = vendors nightmare
- OpenFlow sort of strikes a balance
  - Leave production traffic untouched
  - But allow controlling how other traffic is handled
  - Do this without vendors having to open their boxes

OpenFlow: Overview

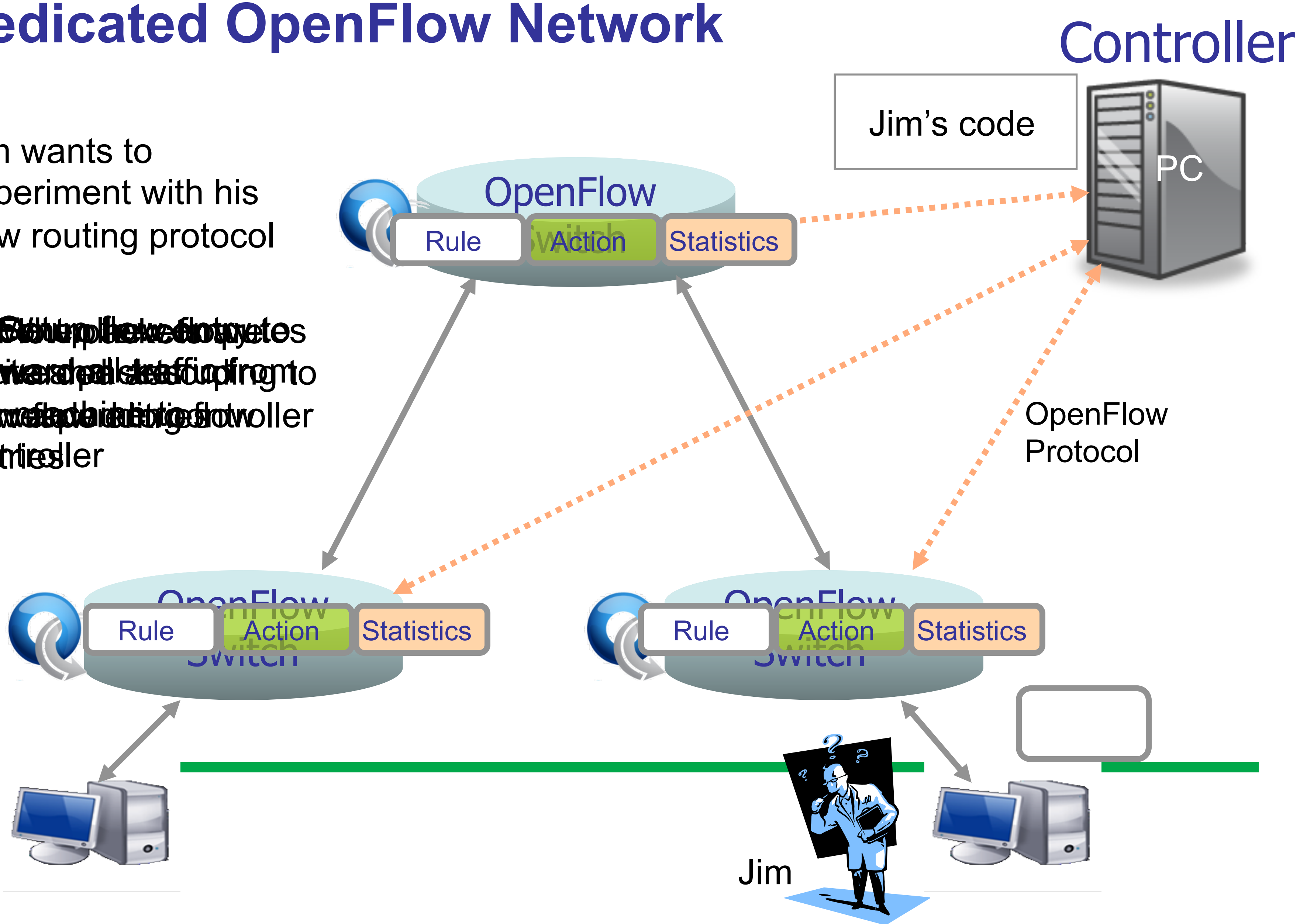
- OpenFlow enables flow-level control of forwarding
- Implementable on COTS hardware
  - Exploits flow-tables existing in modern switches
    - Flow tables are used normally by FWs, QoS, NAT,…
  - Normal production traffic handled as usually
- Make deployed networks programmable
- Goal (experimenter's perspective):
  - No more special purpose test-beds
  - Validate your experiments on deployed hardware with real traffic at full line speed

OpenFlow Usage

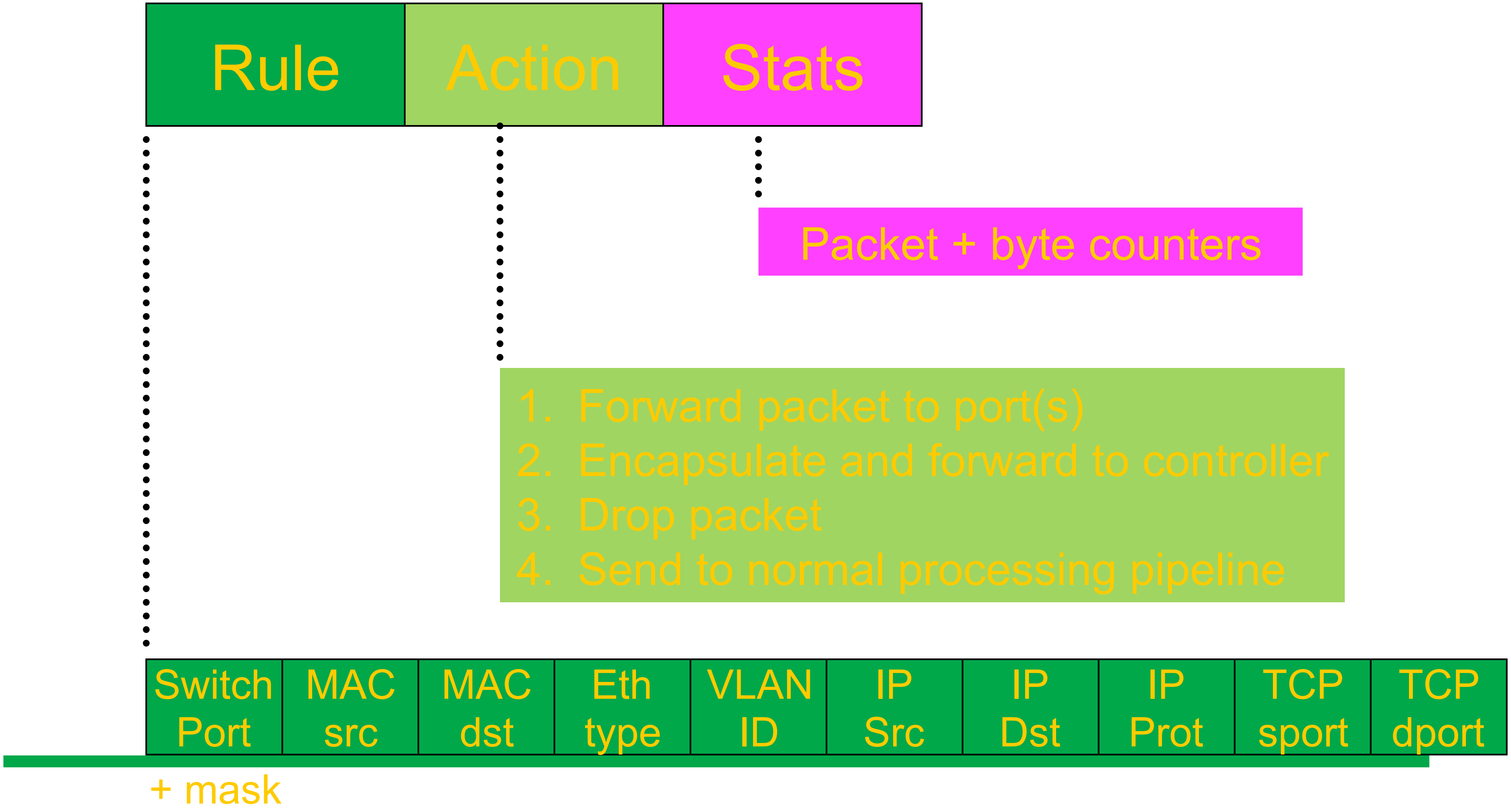
Dedicated OpenFlow Network

Jim wants to experiment with his new routing protocol

[illegible]



Flow Table Entry
“Type 0” OpenFlow Switch



Rule	Action	Stats

Packet + byte counters

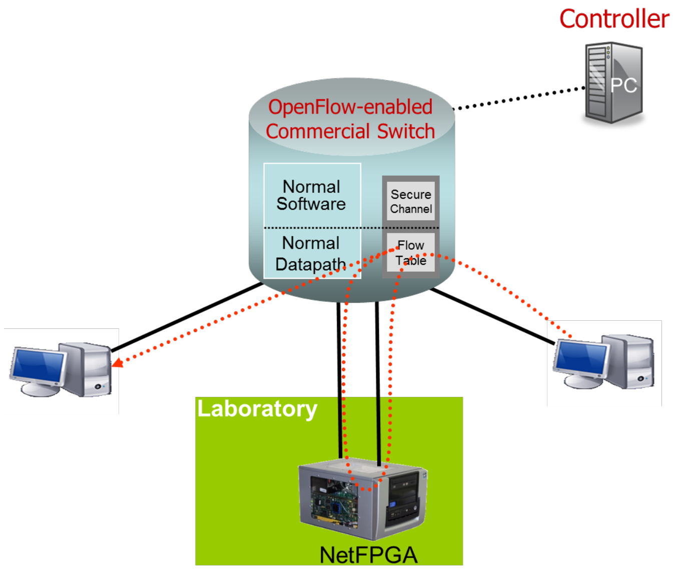
1. Forward packet to port(s)
2. Encapsulate and forward to controller
3. Drop packet
4. Send to normal processing pipeline

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport

+ mask

OpenFlow limitations

- Per-packet networking is more challenging
  - Mechanism is based on exploiting flow-tables
    - -> Per-flow handling
  - Possible to do per-packet processing
    - Via controller (slow)
    - Redirect flow through e.g. NetFPGA (fast)
- Forwarding remains the same
  - Can’t use new forwarding primitives
  - Can’t use new packet formats/field definitions
- Delay in new flow setup
  - ~10ms delay in Stanford deployment
  - Can push down flows proactively to avoid delays
  - Issue when delays are large or new flow-rate is high
- Need still deployment
  - Target is campus environments



Outline

- Experimentally driven research
  - What and why?
  - How to do it?
  - What are the challenges?
- Testbeds and tools
  - Currently available
  - → Coming in the future
- Conclusions

Coming in the future: GENI

- GENI: Global Environment for Network Innovations
  - US based NSF funded initiative/project
  - Virtual laboratory for exploring future Internets at scale
- Some design concepts
  - Programmability
    - Researchers can control how GENI-nodes behave
    - Download software into GENI-nodes
  - Federation
    - Different parts of the GENI suite owned and/or operated by different organizations
  - Slice-based Experimentation

GENI status

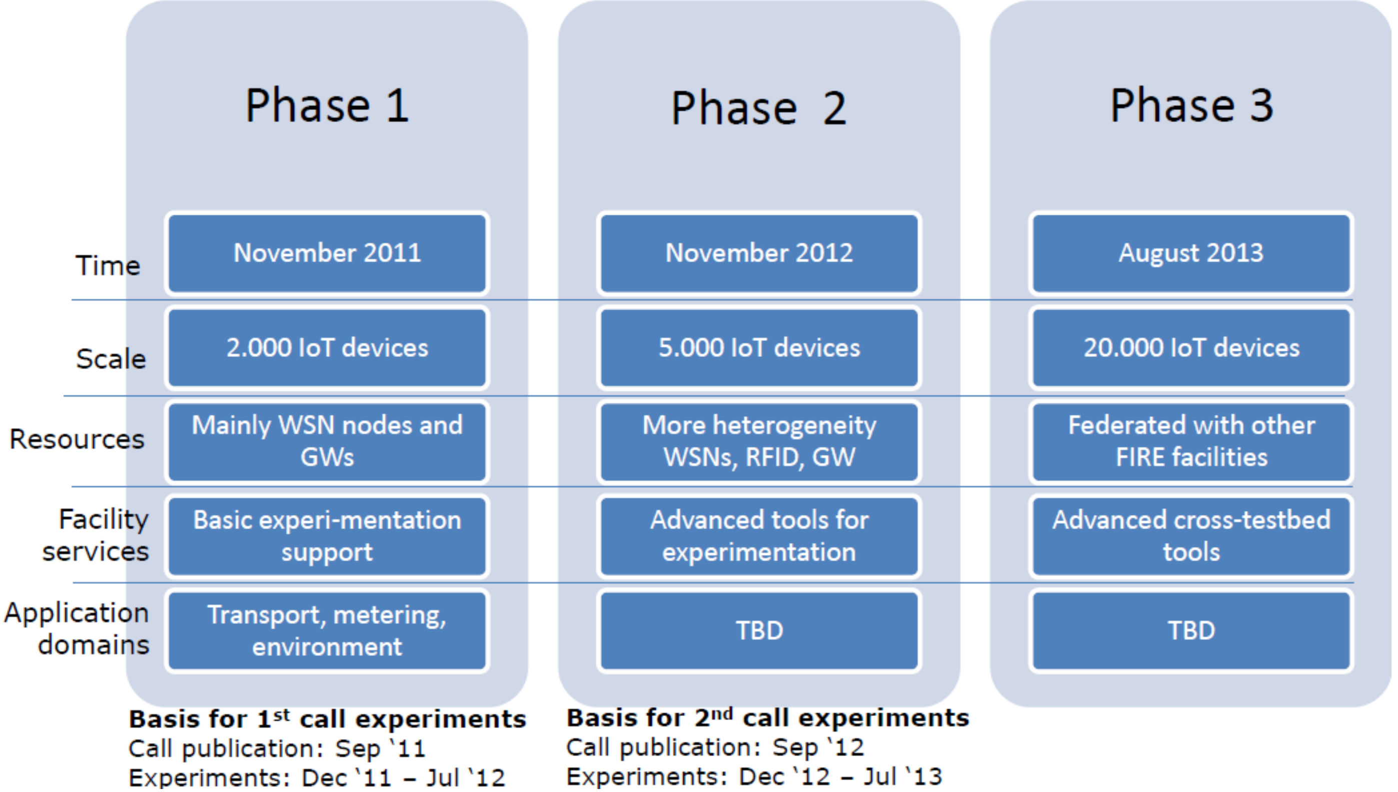
- Program proceeds in spirals
  - One spiral lasts 12 months
- Now on spiral 4
  - Transition from a rapid-prototyping effort to a "real GENI" that supports network research experimentation
  - Goals
    - Ramp up the number of experimenters using GENI by providing better tools and services including 24x7 support
    - Grow the scale of GENI by deploying GENI racks and by GENI-enabling campuses using OpenFlow and WiMAX technologies
    - Create the first rev of GENI instrumentation and measurement systems
- At least 6 spirals defined
  - Let's see what comes out eventually…

Living Labs: SmartSantander

- Smart Santander aims at deploying an Internet of Things infrastructure
  - For experimental research
  - In the framework of a city (Santander, Spain)
- Kind of instantiations of the Living Lab concept
  - Hot topic at the moment
- Infrastructure to support
  - Research community
  - End-users (inhabitants, local authorities,...)
  - Service providers

SmartSantander (cont.)

- Phased roll-out and deployment



	Phase 1	Phase 2	Phase 3
Time	November 2011	November 2012	August 2013
Scale	2.000 IoT devices	5.000 IoT devices	20.000 IoT devices
Resources	Mainly WSN nodes and GWs	More heterogeneity WSNs, RFID, GW	Federated with other FIRE facilities
Facility services	Basic experi-mentation support	Advanced tools for experimentation	Advanced cross-testbed tools
Application domains	Transport, metering, environment	TBD	TBD

Basis for 1st call experiments
Call publication: Sep '11
Experiments: Dec '11 – Jul '12

Basis for 2nd call experiments
Call publication: Sep '12
Experiments: Dec '12 – Jul '13

SmartSantander (cont.)

- Device deployment
  - 3000 IEEE 802.15.4 devices
  - 200 GPRS modules
  - 2000 joint RFID tag/QR code labels
  - 400 parking sensors
  - …
  - Locations
    - Static (streetlights, facades, bus stops)
    - On-board of mobile vehicles (buses, taxis)
- Use cases:
  - Environmental monitoring
  - Parking area management
  - Traffic monitoring
  - Participatory sensing
  - …

SmartSantander (cont.)

- Real-world environment as open platform for experimentation
  - Protocol experimentation, data and knowledge engineering, WSN mgmt, services and applications
- Open calls for experimentation
  - Can ask funding to do experiments in SmartSantander
  - Second call open now
- Early stage currently
  - Time will tell how successful it will be

Conclusions

- There is a time and place for experimental research
  - Not necessarily the best solution every time
  - Understand the limitations wrt. other methods
    - Analytical evaluation, simulations, etc.
- Select your tools and testbeds well
  - Several choices available
  - Experiments take a lot of time and efforts!
    - Cannot try everything
- Design experiments well
  - You'll save time by getting it right the first time
- Interpret your results well and with honesty
  - Data needs to be "cleaned"
    - Experiments will give you anomalies like outliers etc.
  - More about these in other lectures