

# Architecture and Framework for Supporting Open-Access Multi-user Wireless Experimentation

---

S.Ganu, I. Seskar, Max Ott, D. Raychaudhuri, S. Paul



<http://www.orbit-lab.org>



# Project Rationale

---

- **Current wireless research**

- Primarily simulation based or small in-house experimental setups
- Difficult to repeat experiments
- Excessive setup and data collection times may hinder rapid prototyping and experimentation

- **Key design goals**

- Support multi-user wireless experimentation
- Remotely accessible, lights-out operation
- Facilitate choreographing of experiments
- Automate measurement collection
- Capture experiment description so as to repeat as often as necessary

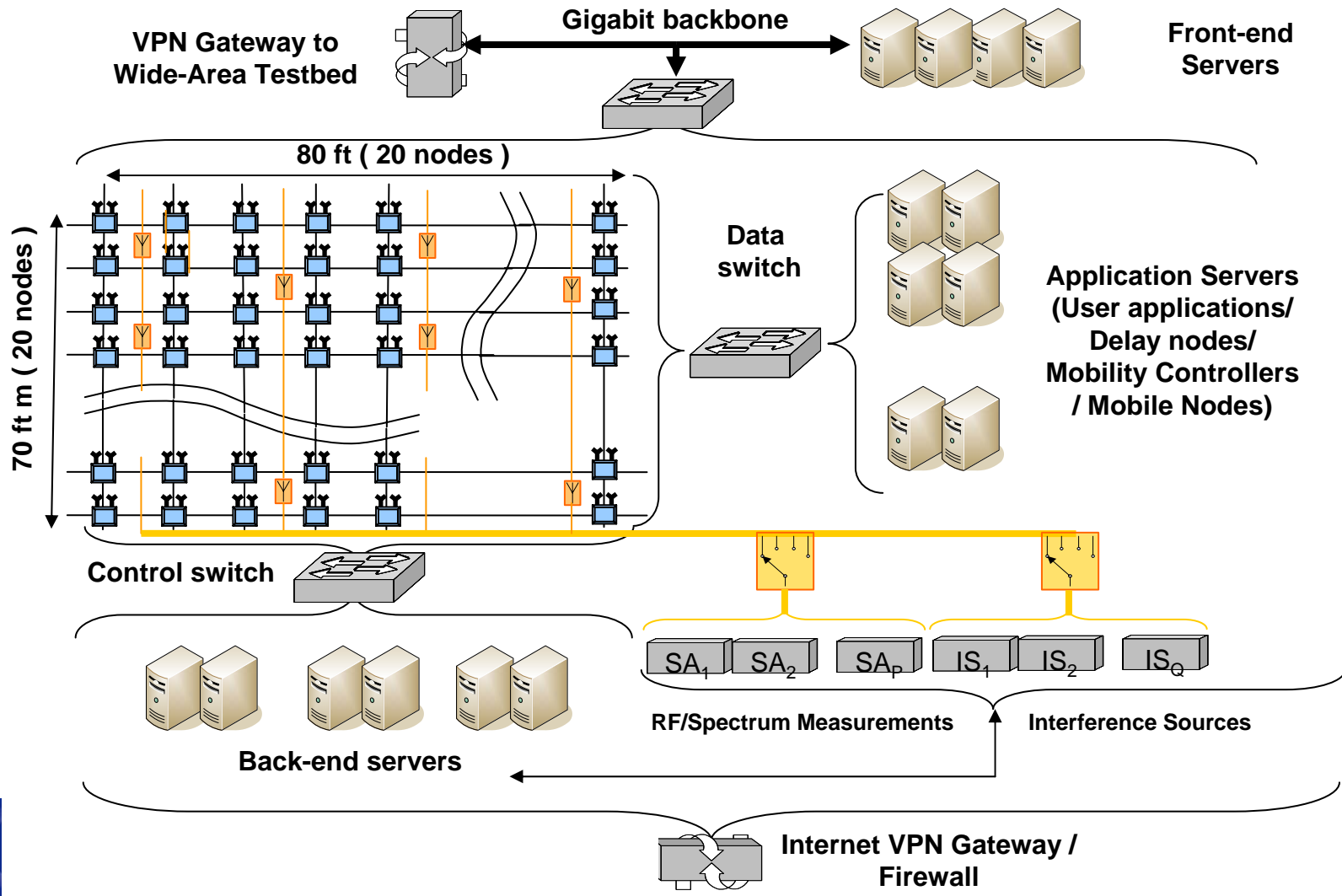
# ORBIT Testbed: Background

---

- Seeded by NSF grant under the Networking Research Testbeds (NRT) program
- Collaborative effort: Rutgers, Columbia, and Princeton, along with industrial partners Lucent Bell Labs, IBM Research and Thomson
- Developed and operated by WINLAB, Rutgers University



# ORBIT: Indoor Grid



# Key Requirements

---

- *scalability*, in terms of the total number of wireless nodes (~100's).
- *reproducibility* of experiments which can be repeated with similar environments to get similar results.
- *open-access flexibility* giving the experimenter a high-level of control over protocols and software used on the radio nodes
- *extensive measurements capability* at radio PHY, MAC and network levels, with the ability to correlate data across layers in both time and space
- *remote access* testbed capable of unmanned operation and the ability to robustly deal with software and hardware failures

# Key Software Considerations

---

- Unlike wired testbeds, difficult to isolate experiments – mainly serial mode of operation
- Need to quickly offload users at the end of the slot
- Reduce start up and clean up times

# Software components

---

## Experiment Controller

- Choreograph experiments

- Capture experiment details to facilitate repetition

## Measurement Framework

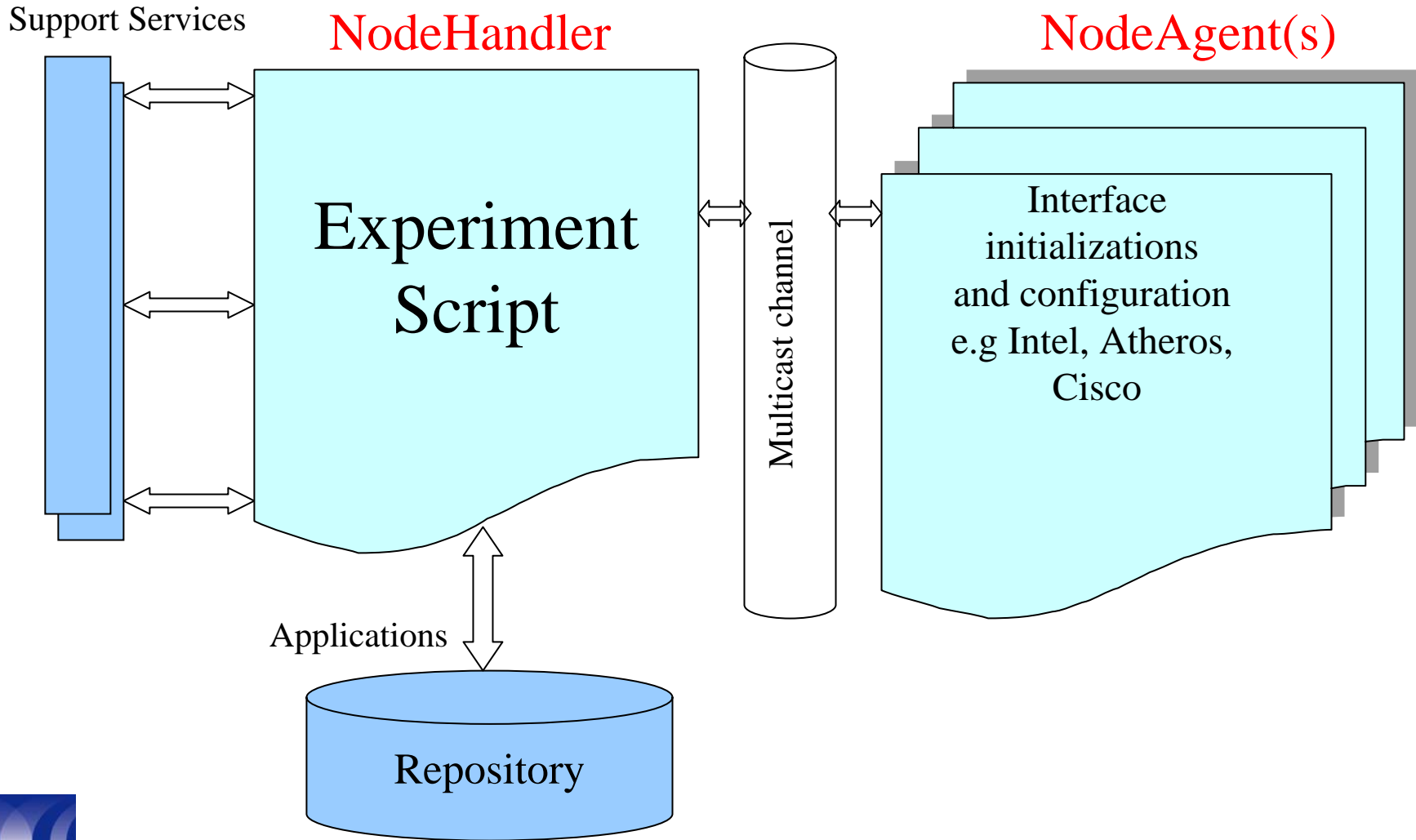
- Efficient measurement collection at run-time

- Avoids delays at end of experiment to collate measurements

## Libmac

- Provides driver independent hooks to the application developers to collect measurements from at radio PHY, MAC layers

# Experiment Controller (NodeHandler)





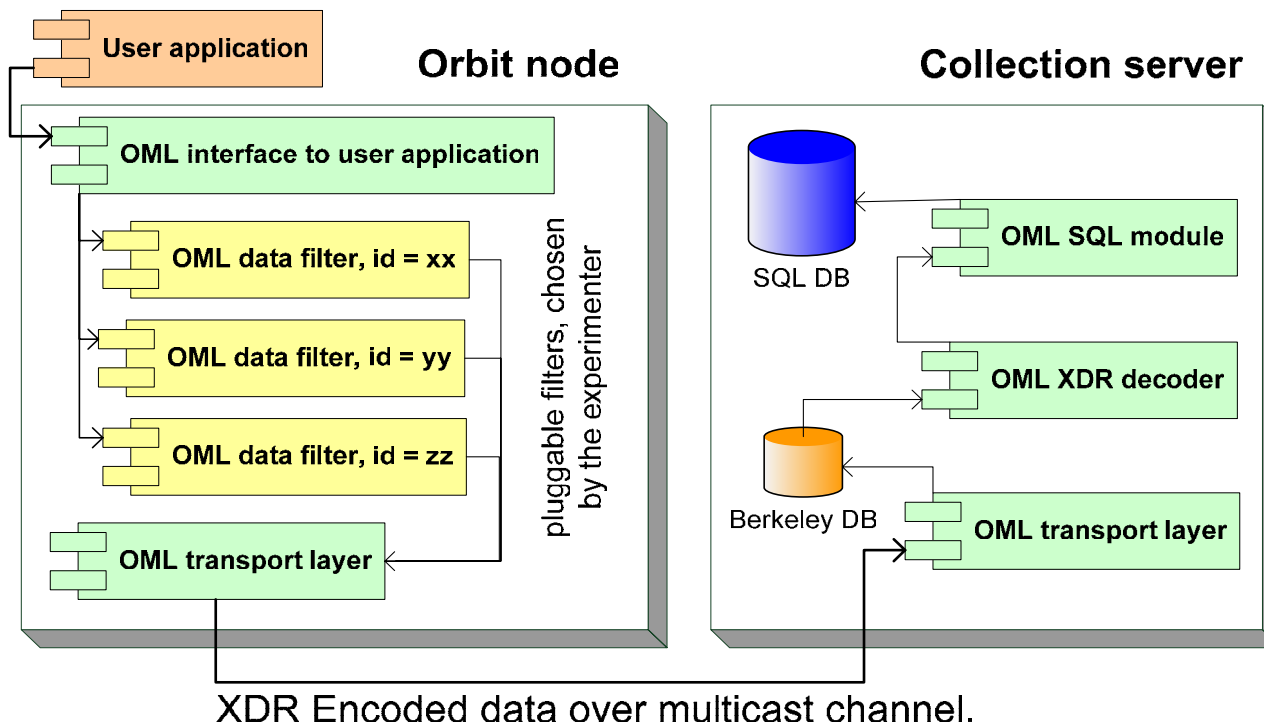
# Experiment Controller

---

- A central **NodeHandler** process communicates with **NodeAgents** (present on each active node in the experiment)
- Instructs nodes to configure interfaces, launch applications etc.
- Communication
  - Over multicast – scalable
  - Using experiment scripts

# OML: Orbit Measurement Library

- Experiments are about collecting measurements
- How to collect them efficiently in a distributed environment like ORBIT?



# OML: Orbit Measurement Library

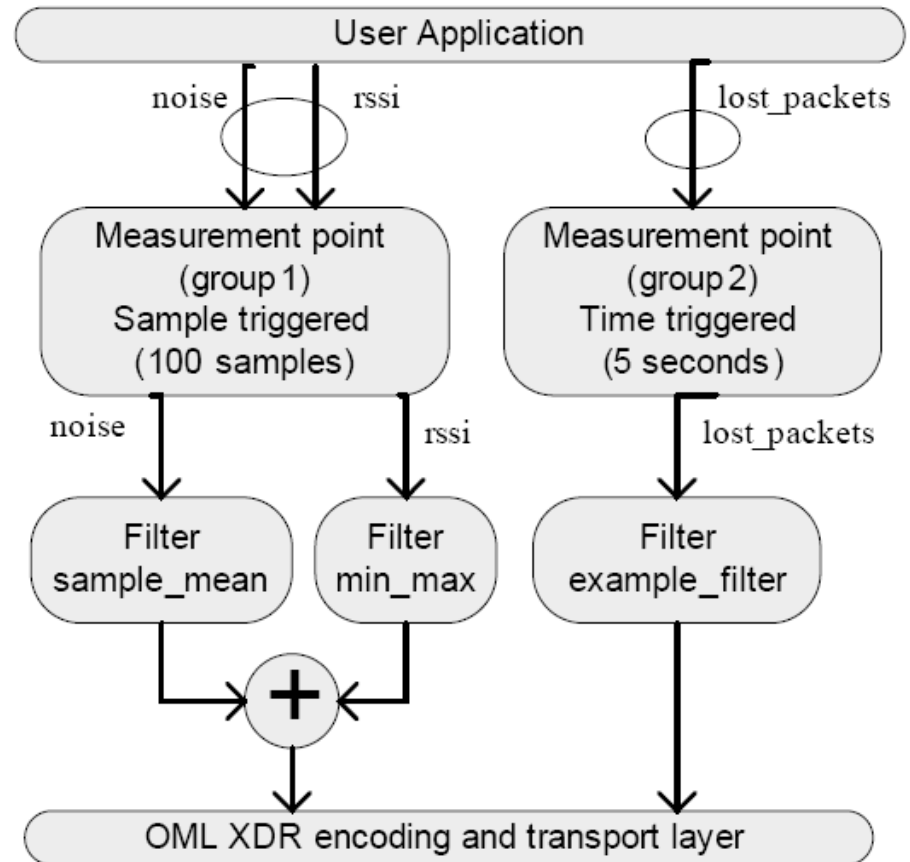
---

- Client
  - Simple API for application writers
  - Filters reduce the amount of reportable data
  - XDR encoded data over multicast channel
- Collection Server
  - Berkeley DB used for scalability
  - SQL database for persistent for data archiving
  - One multicast channel per experiment for logical segregation of data, and scalability



# OML: Pluggable Filters

- Not all measurements may be needed
- Allow dynamic preprocessing before reporting to database
- Experimenter can choose the granularity (per packet or every N packets, per second or every N seconds)



# Real time Statistics

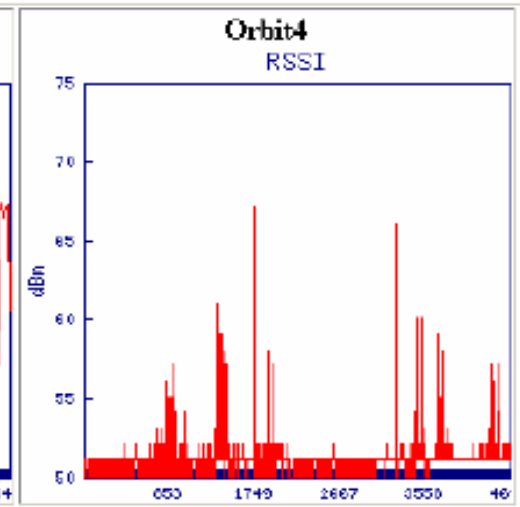
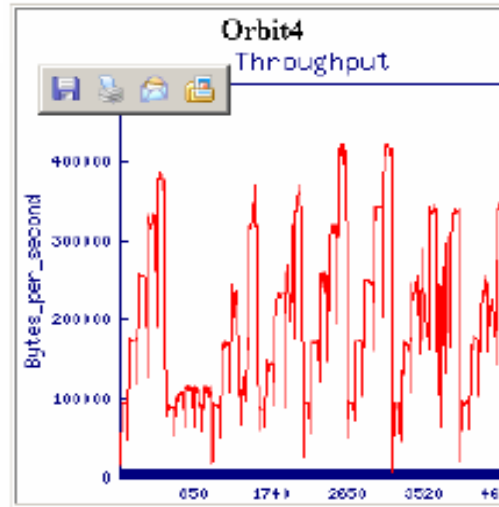
```
mysql> select * from metrics;
```

id	metric	units	type
1	rsssi	db	float
2	noise	db	float
3	throughput	bps	float
4	offered_load	bps	float

4 rows in set (0.00 sec)

```
mysql> select * from metrics_values where id=3 and node_id="orbit4";
```

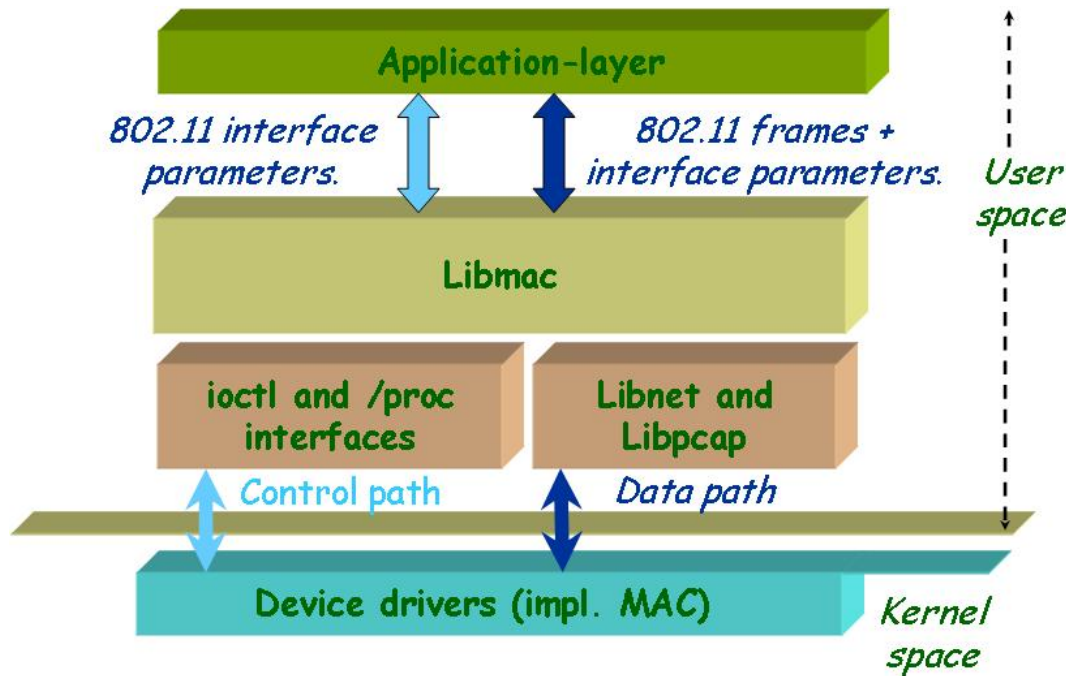
id	value	packet_size	node_id	sequence_no	timestamp
3	1086464	256	orbit4	4	1084223553
3	1515520	256	orbit4	10	1084223554
3	1503232	256	orbit4	16	1084223555



**MATLAB, Excel for Mysql allows easy post processing**



# Libmac

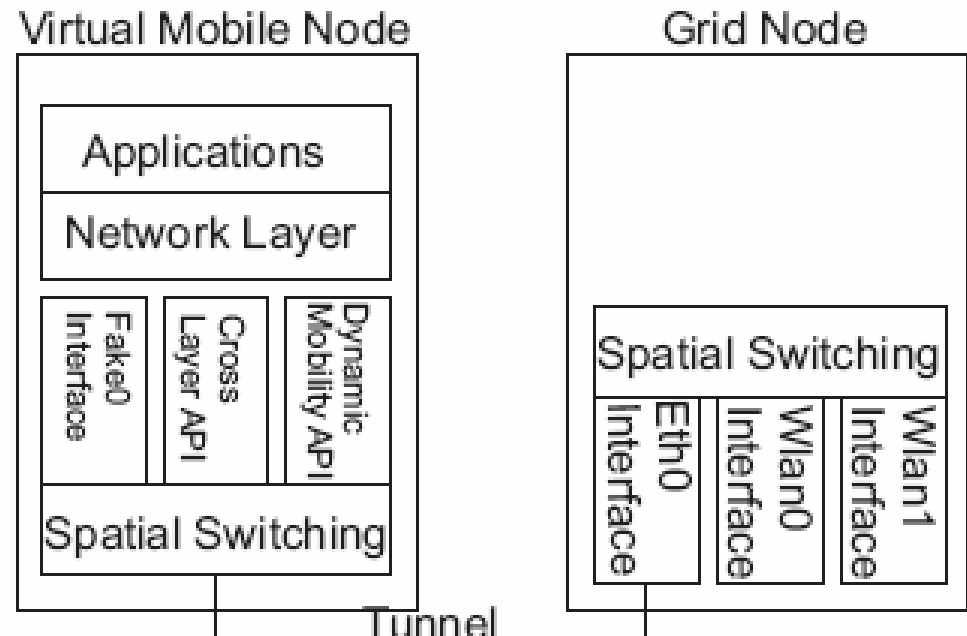


## User-space C library

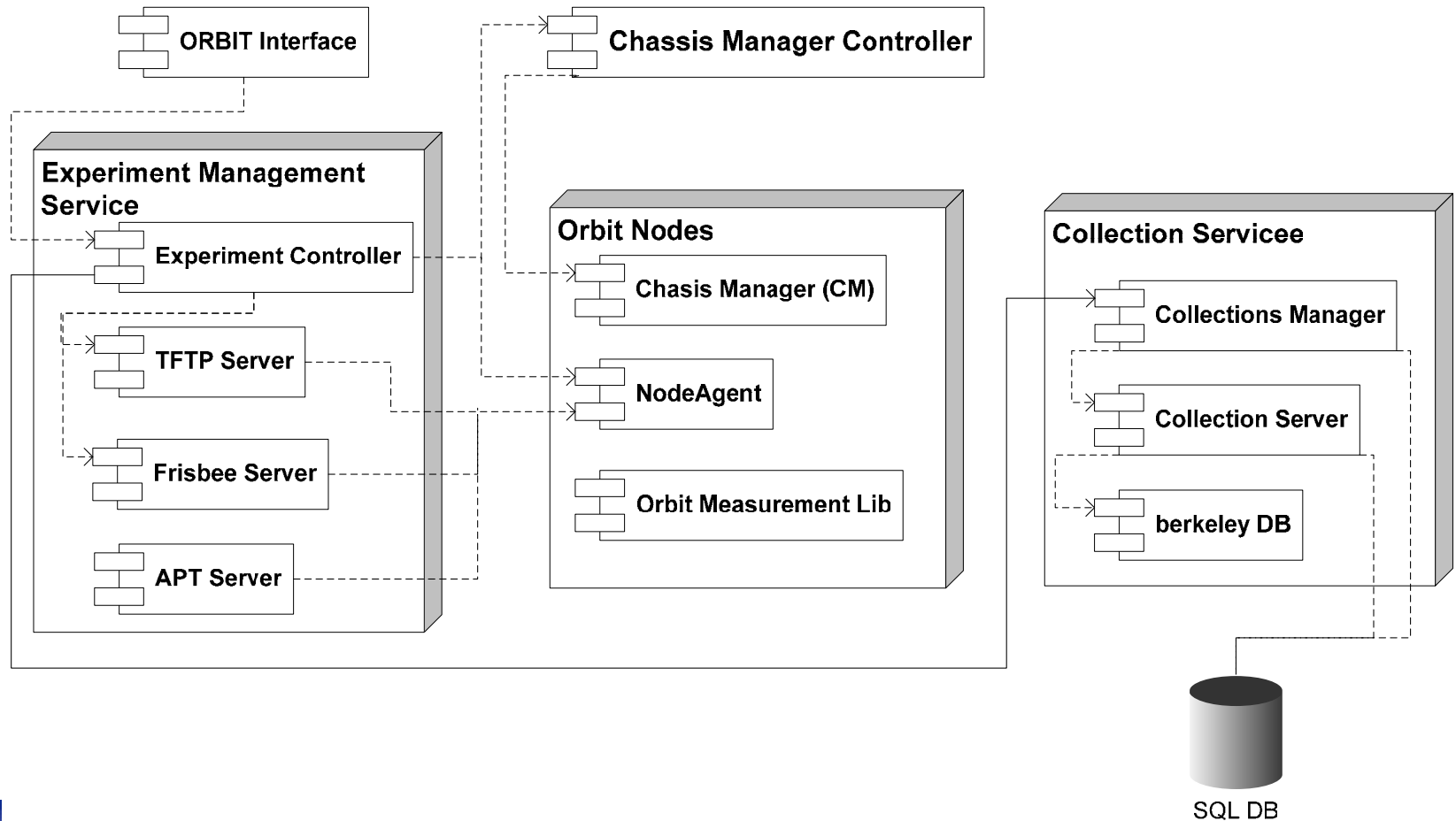
- To inject and capture MAC layer frames.
- To manipulate wireless interface parameters at both aggregate and per-frame levels.
- To communicate wireless interface parameters over the air, on a per-frame level
- Allows application developers to interface with driver measurements through simple function call

# Mobility Emulation: Our Approach

- Uses software spatial switching
- Emulates trajectory by switching to different radio and antenna positions as time progresses



# Essential Orbit Services





# Chassis Manager Controller

- Web/Program interface for remote control of nodes
- Provides facilities for power on, reboot, hard/soft power off
- Console access to node
- Logging of node state (on/off), temperature, and voltage

## Main Grid Node Status

(CM IP: 10.1.x.y)

1,20	2,20	3,20	4,20	5,20	6,20	7,20	8,20	9,20	10,20	11,20	12,20	13,20	14,20	15,20	16
1,19	2,19	3,19	4,19	5,19	6,19	7,19	8,19	9,19	10,19	11,19	12,19	13,19	14,19	15,19	16
1,18	2,18	3,18	4,18	5,18	6,18	7,18	8,18	9,18	10,18	11,18	12,18	13,18	14,18	15,18	16
1,17	2,17	3,17	4,17	5,17	6,17	7,17	8,17	9,17	10,17	11,17	12,17	13,17	14,17	15,17	16
1,16	2,16	3,16	4,16	5,16	6,16	7,16	8,16	9,16	10,16	11,16	12,16	13,16	14,16	15,16	16
1,15	2,15	3,15	4,15	5,15	6,15	7,15	8,15	9,15	10,15	11,15	12,15	13,15	14,15	15,15	16
1,14	2,14	3,14	4,14	5,14	6,14	7,14	8,14	9,14	10,14	11,14	12,14	13,14	14,14	15,14	16
1,13	2,13	3,13	4,13	5,13	6,13	7,13	8,13	9,13	10,13	11,13	12,13	13,13	14,13	15,13	16
1,12	2,12	3,12	4,12	5,12	6,12	7,12	8,12	9,12	10,12	11,12	12,12	13,12	14,12	15,12	16
1,11	2,11	3,11	4,11	5,11	6,11	7,11	8,11	9,11	10,11	11,11	12,11	13,11	14,11	15,11	16
1,10	2,10	3,10	4,10	5,10	6,10	7,10	8,10	9,10	10,10	11,10	12,10	13,10	14,10	15,10	16
1,9	2,9	3,9	4,9	5,9	6,9	7,9	8,9	9,9	10,9	11,9	12,9	13,9	14,9	15,9	16
1,8	2,8	3,8	4,8	5,8	6,8	7,8	8,8	9,8	10,8	11,8	12,8	13,8	14,8	15,8	16
1,7	2,7	3,7	4,7	5,7	6,7	7,7	8,7	9,7	10,7	11,7	12,7	13,7	14,7	15,7	16
1,6	2,6	3,6	4,6	5,6	6,6	7,6	8,6	9,6	10,6	11,6	12,6	13,6	14,6	15,6	16
1,5	2,5	3,5	4,5	5,5	6,5	7,5	8,5	9,5	10,5	11,5	12,5	13,5	14,5	15,5	16
1,4	2,4	3,4	4,4	5,4	6,4	7,4	8,4	9,4	10,4	11,4	12,4	13,4	14,4	15,4	16
1,3	2,3	3,3	4,3	5,3	6,3	7,3	8,3	9,3	10,3	11,3	12,3	13,3	14,3	15,3	16
1,2	2,2	3,2	4,2	5,2	6,2	7,2	8,2	9,2	10,2	11,2	12,2	13,2	14,2	15,2	16
1,1	2,1	3,1	4,1	5,1	6,1	7,1	8,1	9,1	10,1	11,1	12,1	13,1	14,1	15,1	16

Orbit CMC Main Grid Control - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://cmc.orbit-lab.org/controlgrid.html

Customize Links Free Hotmail Windows Media Windows

1,15	2,15	3,15	4,15	5,15	6,15	7,15	8,15	9,15	10,15	11,15	12,15	13,15	14,15	15,15	16,15	17,15	18,15	19,15	20,15
1,14	2,14	3,14	4,14	5,14	6,14	7,14	8,14	9,14	10,14	11,14	12,14	13,14	14,14	15,14	16,14	17,14	18,14	19,14	20,14
1,13	2,13	3,13	4,13	5,13	6,13	7,13	8,13	9,13	10,13	11,13	12,13	13,13	14,13	15,13	16,13	17,13	18,13	19,13	20,13
1,12	2,12	3,12	4,12	5,12	6,12	7,12	8,12	9,12	10,12	11,12	12,12	13,12	14,12	15,12	16,12	17,12	18,12	19,12	20,12
1,11	2,11	3,11	4,11	5,11	6,11	7,11	8,11	9,11	10,11	11,11	12,11	13,11	14,11	15,11	16,11	17,11	18,11	19,11	20,11
1,10	2,10	3,10	4,10	5,10	6,10	7,10	8,10	9,10	10,10	11,10	12,10	13,10	14,10	15,10	16,10	17,10	18,10	19,10	20,10
1,9	2,9	3,9	4,9	5,9	6,9	7,9	8,9	9,9	10,9	11,9	12,9	13,9	14,9	15,9	16,9	17,9	18,9	19,9	20,9
1,8	2,8	3,8	4,8	5,8	6,8	7,8	8,8	9,8	10,8	11,8	12,8	13,8	14,8	15,8	16,8	17,8	18,8	19,8	20,8
1,7	2,7	3,7	4,7	5,7	6,7	7,7	8,7	9,7	10,7	11,7	12,7	13,7	14,7	15,7	16,7	17,7	18,7	19,7	20,7
1,6	2,6	3,6	4,6	5,6	6,6	7,6	8,6	9,6	10,6	11,6	12,6	13,6	14,6	15,6	16,6	17,6	18,6	19,6	20,6
1,5	2,5	3,5	4,5	5,5	6,5	7,5	8,5	9,5	10,5	11,5	12,5	13,5	14,5	15,5	16,5	17,5	18,5	19,5	20,5
1,4	2,4	3,4	4,4	5,4	6,4	7,4	8,4	9,4	10,4	11,4	12,4	13,4	14,4	15,4	16,4	17,4	18,4	19,4	20,4
1,3	2,3	3,3	4,3	5,3	6,3	7,3	8,3	9,3	10,3	11,3	12,3	13,3	14,3	15,3	16,3	17,3	18,3	19,3	20,3
1,2	2,2	3,2	4,2	5,2	6,2	7,2	8,2	9,2	10,2	11,2	12,2	13,2	14,2	15,2	16,2	17,2	18,2	19,2	20,2
1,1	2,1	3,1	4,1	5,1	6,1	7,1	8,1	9,1	10,1	11,1	12,1	13,1	14,1	15,1	16,1	17,1	18,1	19,1	20,1

ALL Reset All Power ON All Power OFF Hard All Power OFF Soft ALL Identify

Select Node from Grid above

Reset       OFF\_Hard  
 ON       OFF\_Soft  
 Identify       STATUS

Execute Command Clear

# Frisbee\*

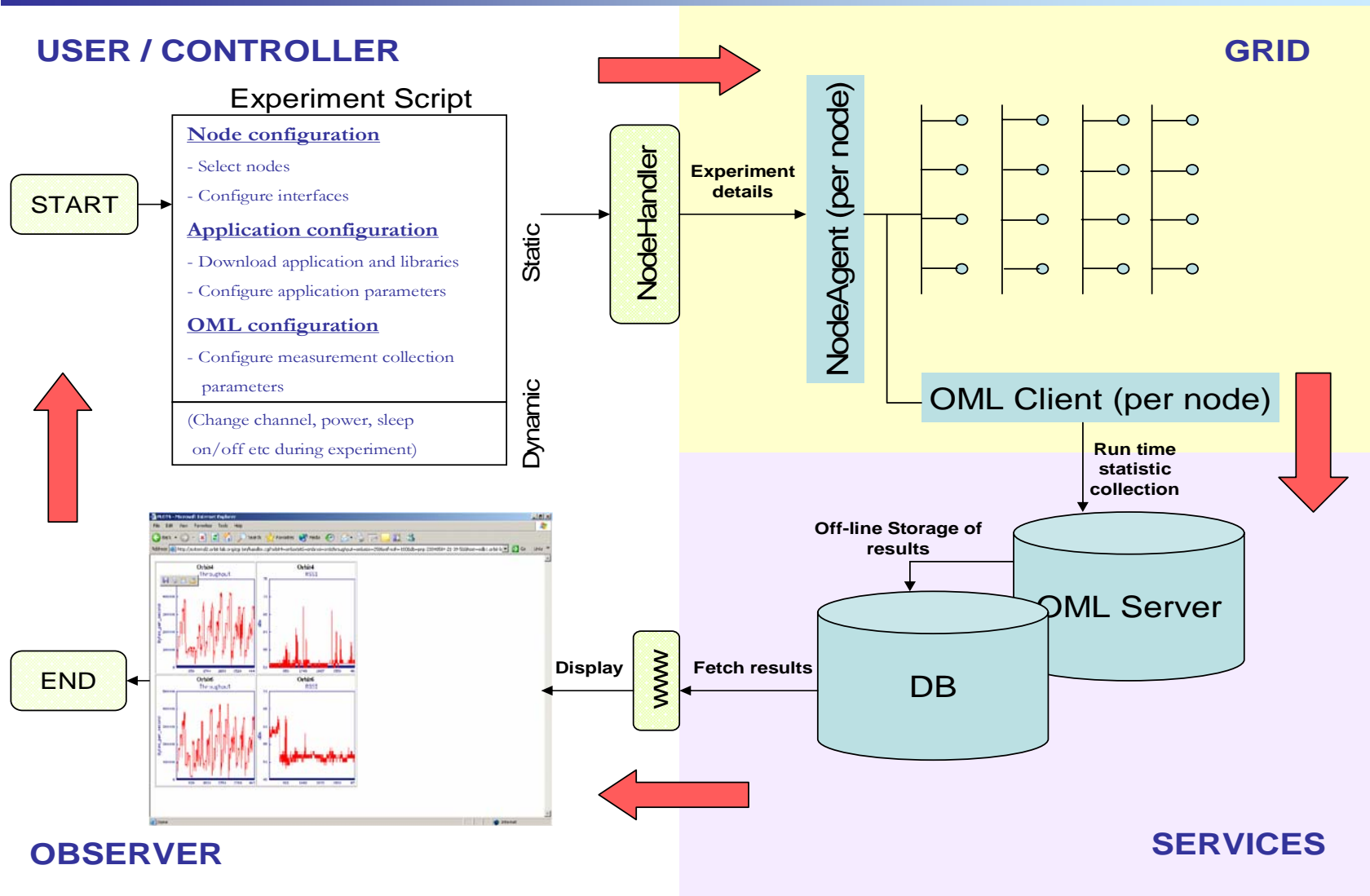
---

- Fast and automated way to image any number of nodes
- Frisbee – Client/Server application that facilitates fast transfers of entire disk images.
  - Baseline Node Image (300 MB) currently takes ~5 minutes to install on all 64 grid nodes

\* From Emulab Testbed, University of Utah



# Putting it all together

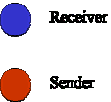
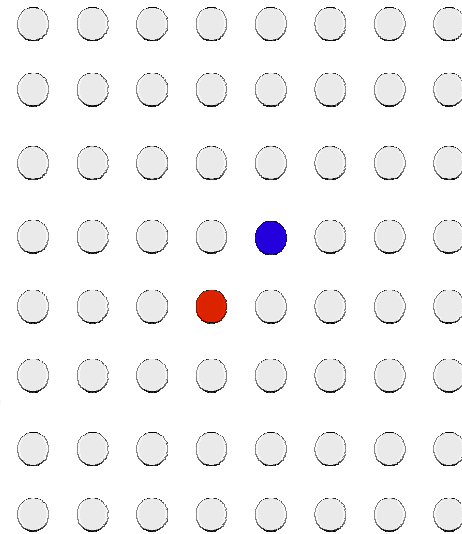


# Sample Experiments using ORBIT

---

- Sender-Receiver Experiment

- Node 4-3 sends to Node 5-4
  - 11 Mbps
  - ‘b’
  - Ad-hoc (or Master-Managed)
  - 3 Mbps offered load
  - Measure RSSI, Throughput at receiver and offered load at sender



# Define Sender

```
#
# Define nodes used in experiment
#
nodes([4,3], 'sender') { |node|
  node.image = nil # Use default disk image

# experiment property space
node.prototype("test:proto:sender", { # use prototype "sender"
  'destinationHost' => '192.168.5.4', # Set it's property "destinationHost"
  'packetSize' => Experiment.property("packetSize"),
  'rate' => Experiment.property("rate") # bind the remaining properties to defaults
}) # Can be overridden later
node.net.w0.mode = "master"
node.net.w0.type = 'b'
node.net.w0.essid = "helloworld" # Set wireless parameters
node.net.w0.ip = "%192.168.%x.%y"
node.net.w0.rate = "11m"
}
```

w0, w1 are interpreted by nodeAgent according to the card being used  
e.g Intel w0= eth2, w1= eth3  
Atheros w0=ath0, w1= ath1



# Define Receiver

---

```
#  
# Define nodes used in experiment  
#  
nodes([5,4], 'receiver') { |node|  
  node.image = nil          # assume the right image to be on disk  
  node.prototype("test:proto:receiver" , {  
    'hostname' => '192.168.5.4',  
    'protocol' => 'udp_libmac'  # Use Libmac to report RSSI  
  })  
  node.net.w0.mode = "managed"  
  node.net.w0.type = 'b'  
  node.net.w0.essid = "helloworld"  
  node.net.w0.ip = "%192.168.%x.%y"  
}
```



# Script..

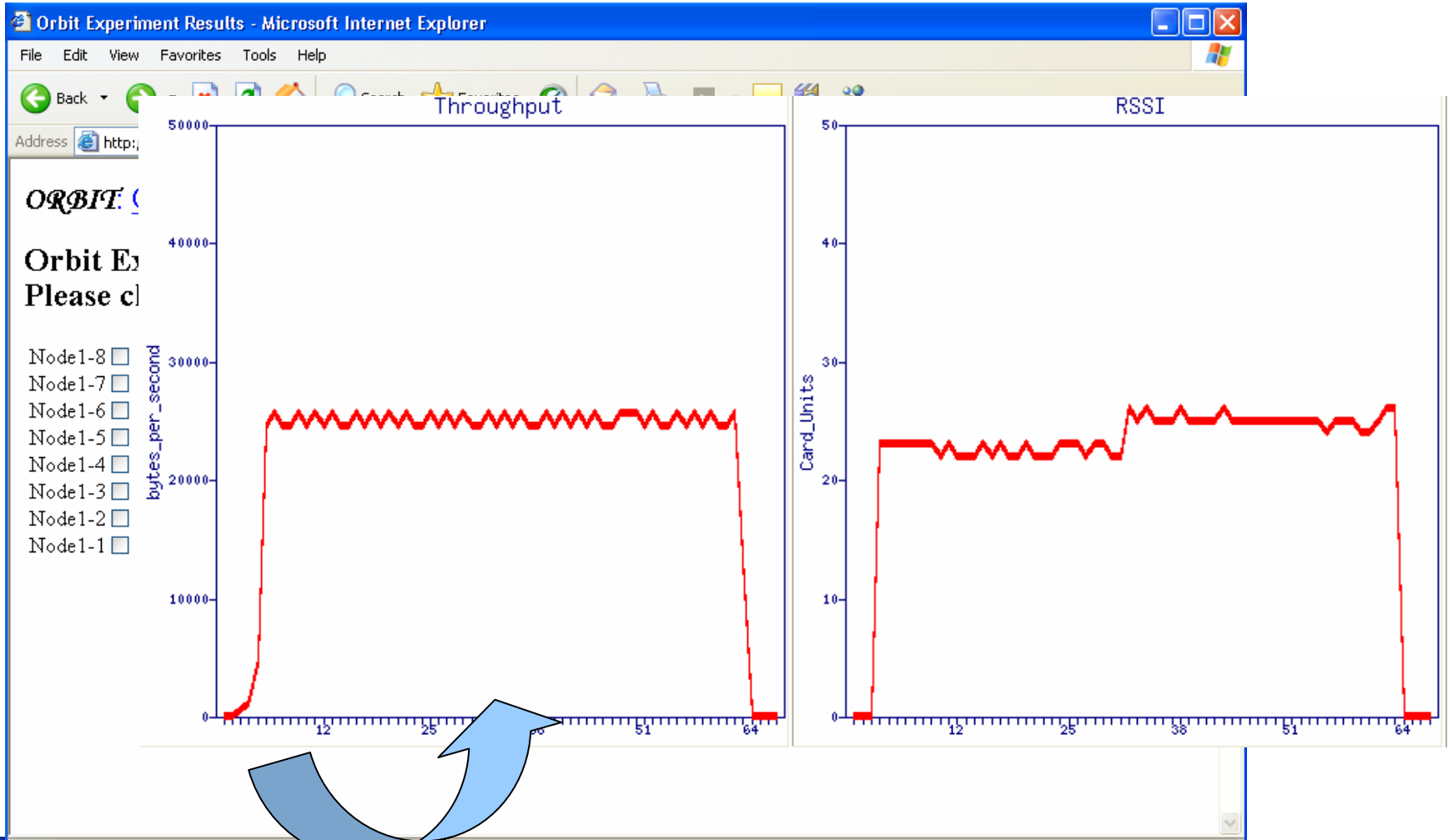
```
# Now, start the application
whenAllInstalled() {
    allNodes.startApplications

    #Set packet size to 1024 bytes
    # and packet rate to 3000 Kbps
    NodeSet['sender'].send(:STDIN, 'proc/otg', 'size 1024')
    NodeSet['sender'].send(:STDIN, 'proc/otg', 'rate 3000')

    # Run the experiment for 60 seconds
    wait 60

    # Stop the applications
    allNodes.stopApplications
    Experiment.done
}
```

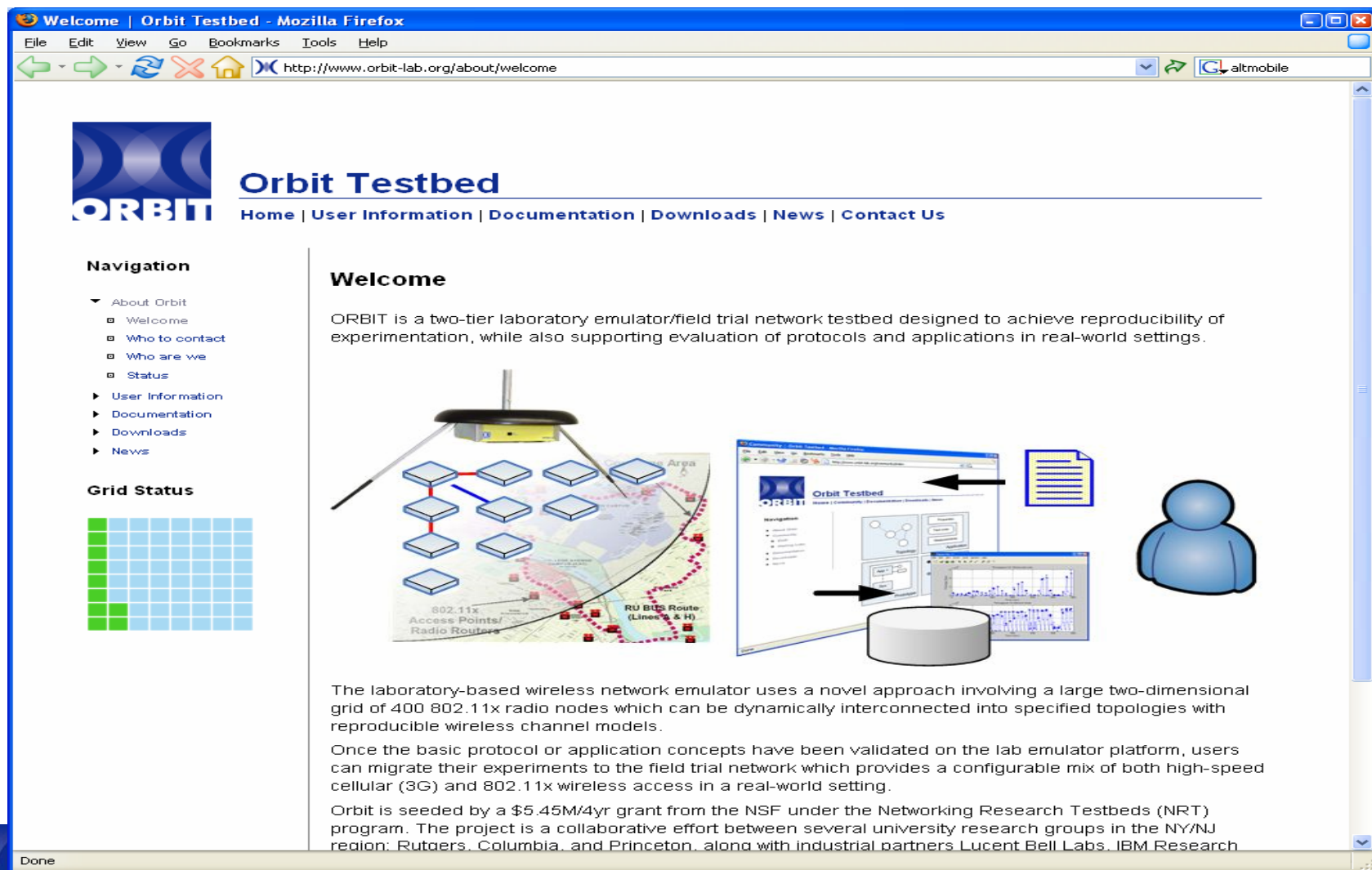
# View Results (during exp.)



Cross layer effects! Throughput, RSSI vs time



# www.orbit-lab.org




The screenshot shows a Mozilla Firefox browser window displaying the Orbit Testbed website. The browser's address bar shows the URL <http://www.orbit-lab.org/about/welcome>. The website features a blue header with the ORBIT logo and navigation links: Home, User Information, Documentation, Downloads, News, and Contact Us. A left sidebar contains a 'Navigation' menu with expandable sections for 'About Orbit' (Welcome, Who to contact, Who are we, Status), 'User Information', 'Documentation', 'Downloads', and 'News'. Below this is a 'Grid Status' section with a grid of colored squares. The main content area is titled 'Welcome' and contains a paragraph about the testbed's purpose, a diagram of the network setup, and a paragraph describing the laboratory-based wireless network emulator. The diagram shows a physical antenna connected to a grid of nodes on a map, with labels for '802.11x Access Points/ Radio Routers' and 'RU BUS Route (Lines & H)'. To the right of the diagram is a computer monitor displaying the website interface, a document icon, a database cylinder, and a person icon. The browser's status bar at the bottom shows 'Done'.

Welcome | Orbit Testbed - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

<http://www.orbit-lab.org/about/welcome> altmobile



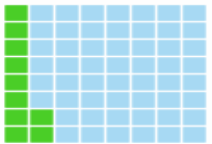
## Orbit Testbed

[Home](#) | [User Information](#) | [Documentation](#) | [Downloads](#) | [News](#) | [Contact Us](#)

### Navigation

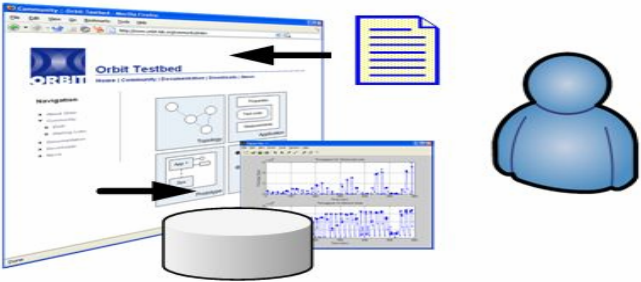
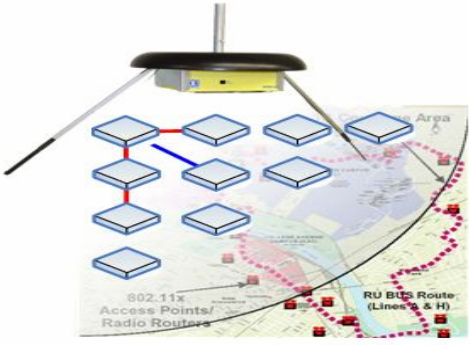
- ▼ About Orbit
  - ▣ Welcome
  - ▣ Who to contact
  - ▣ Who are we
  - ▣ Status
- ▶ User Information
- ▶ Documentation
- ▶ Downloads
- ▶ News

### Grid Status



### Welcome

ORBIT is a two-tier laboratory emulator/field trial network testbed designed to achieve reproducibility of experimentation, while also supporting evaluation of protocols and applications in real-world settings.



The laboratory-based wireless network emulator uses a novel approach involving a large two-dimensional grid of 400 802.11x radio nodes which can be dynamically interconnected into specified topologies with reproducible wireless channel models.

Once the basic protocol or application concepts have been validated on the lab emulator platform, users can migrate their experiments to the field trial network which provides a configurable mix of both high-speed cellular (3G) and 802.11x wireless access in a real-world setting.

Orbit is seeded by a \$5.45M/4yr grant from the NSF under the Networking Research Testbeds (NRT) program. The project is a collaborative effort between several university research groups in the NY/NJ region: Rutgers, Columbia, and Princeton, along with industrial partners Lucent Bell Labs, IBM Research

Done

