HOWARD UNIVERSITY
1867

Emulating Opportunistic Mobile Wireless Networks on ORBIT

Sankar Roy

Postdoctoral Researcher,

Department of Systems and Computer Science, Howard University

HOWARD UNIVERSITY
1867

Outline of the Talk

- An Overview of the Project
- Preliminaries on Pigeon Networks
- Background of ORBIT testbed
- Emulating a Pigeon Network on ORBIT
- Critiques on ORBIT

Introducing our group/project

- Who are we?
  - Two researchers (myself and Professor Jiang Li) at Howard University, DC
- What is the project about?
  - Our focus is on emulation of a special type of DTN (Delay Tolerant Network)
  - To evaluate the usability of ORBIT testbed

HOWARD UNIVERSITY
1867

Outline of the Talk

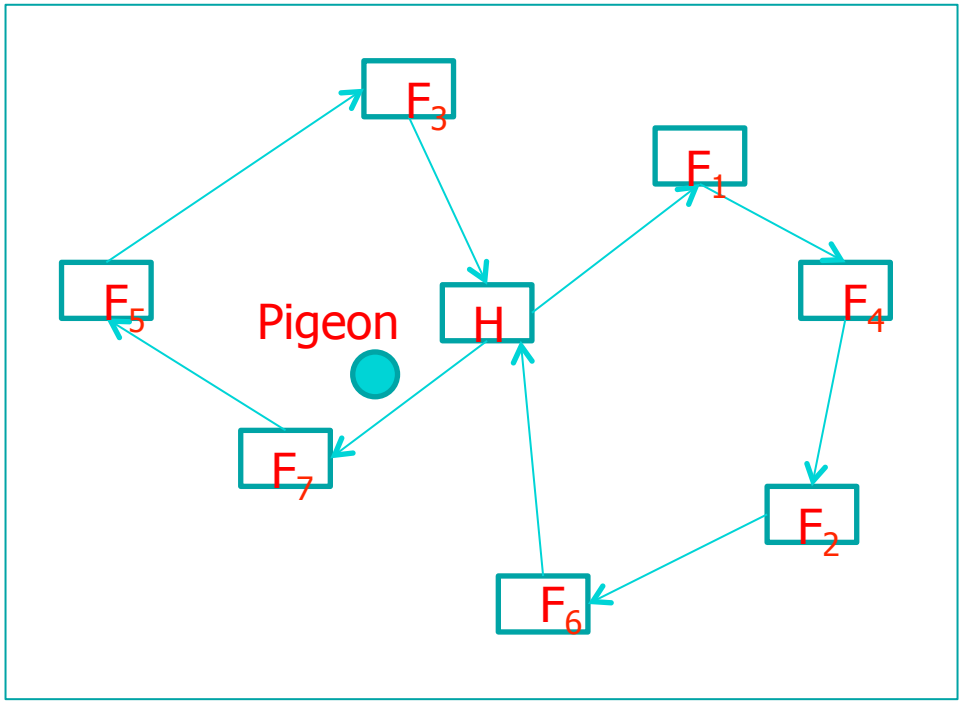
- An Overview of the Project
- Preliminaries on Pigeon Networks
- Background of ORBIT testbed
- Emulating a Pigeon Network on ORBIT
- Critiques on ORBIT

Preliminaries: pigeon network

- It is a special DTN
- The message carrier, called a *pigeon*, is owned by a host node and has dedication to the owner node.
- The owner node of a pigeon is called the *home host*, and other host nodes are known as the *foreign hosts*.
  - Hosts communicate with each other through the mobile pigeon.

A cell of a pigeon network

- Pigeon Network may have multiple cells.
- A cell has three kinds of nodes: the pigeon P, the home host H, and multiple foreign host F_i.
- P is mobile and moves across H and F_i whereas H and F_i are typically static.



HOWARD UNIVERSITY
1867

Outline of the Talk

- An Overview of the Project
- Preliminaries on Pigeon Networks
- Background of ORBIT testbed
- Emulating a Pigeon Network on ORBIT
- Critiques on ORBIT

Background of ORBIT testbed

- One grid (400 computers); multiple sandboxes.
- If the user runs proper emulation programs in some of the computers, each such computer becomes one node of the emulated network.
- The ORBIT testbed provides the **OMF interface** through which the user manages the resources.
- A user writes an **Experiment Description (ED)** script and runs it via the OMF interface.

HOWARD UNIVERSITY
1867

OML and Result Service

- To help the user save and analyze the experiment data, ORBIT testbed provides a measurement library called **OML**.
- The ORBIT **result service** allows the user to retrieve the experiment data over the Internet in real time or after the experiment.
- The user can view the data on a web browser window on his/her local machine.

Emulating Node Mobility on ORBIT?

- Three ways to emulate a "virtual mobile node" on an ORBIT node:
  - noise injection/controlling attenuation
  - madwifi driver (no longer supported)
  - packet filtering e.g. iptables.
- On sandbox sb-4, a small mobile network (e.g. 5 nodes) can be emulated (by controlling attenuation).
- To emulate a large network (e.g. 30 nodes), packet filtering is practically the best way.

HOWARD UNIVERSITY
1867

Outline of the Talk

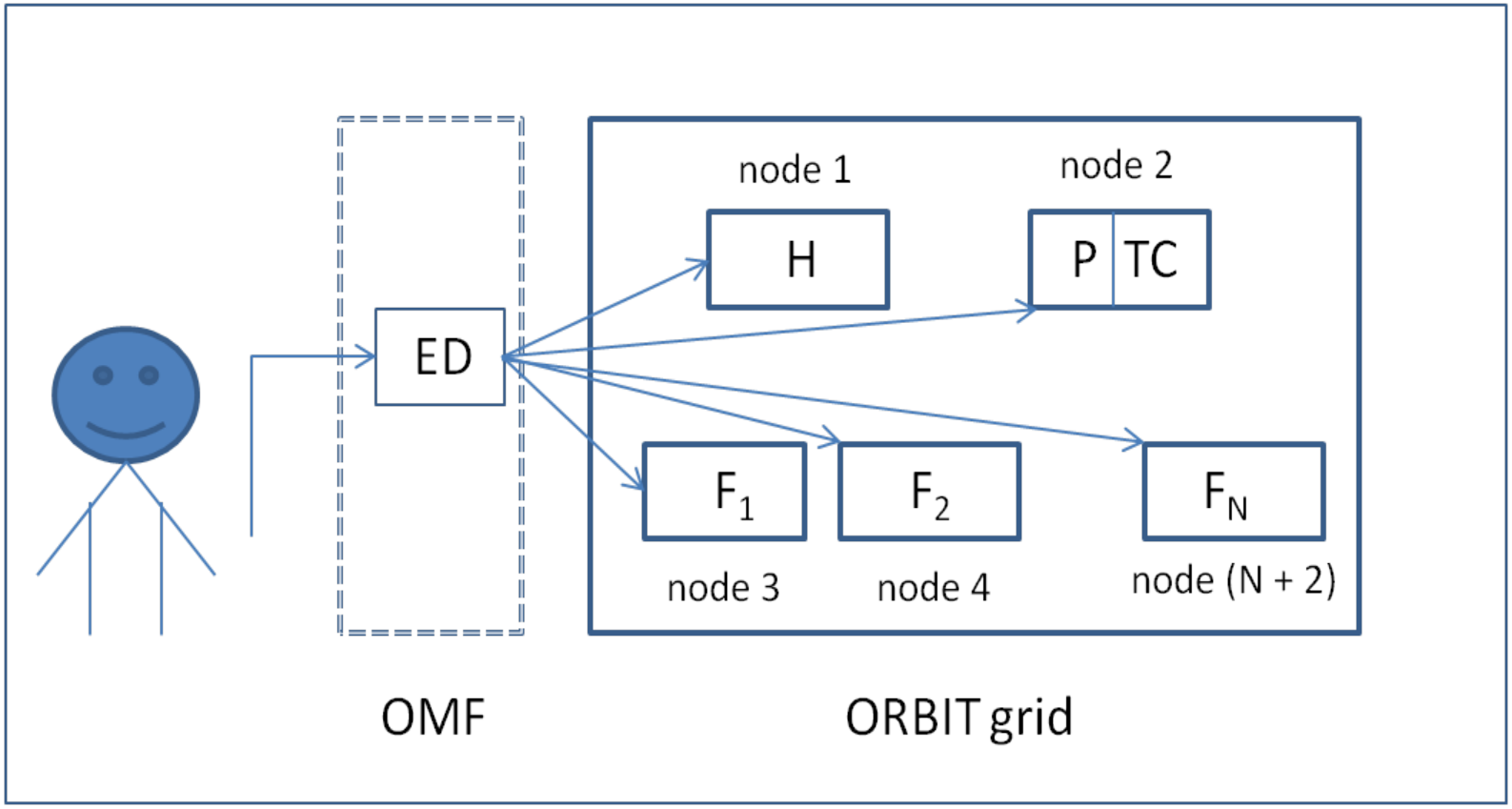
- An Overview of the Project
- Preliminaries on Pigeon Networks
- Background of ORBIT testbed
- Emulating a Pigeon Network on ORBIT
- Critiques on ORBIT

Emulating a Pigeon Network on ORBIT

- To perform the experiment, the experimenter mainly needs two things: (a) an ED script which is run on the OMF interface, and (b) the emulation programs which are run on the grid nodes.
- The network is built using (N+2) real nodes: one emulates the home host, one emulates the pigeon , and each of the rest emulates one foreign host.

Emulating a Pigeon Network on ORBIT (contd.)

- An ED script is run by the OMF to control the emulation programs

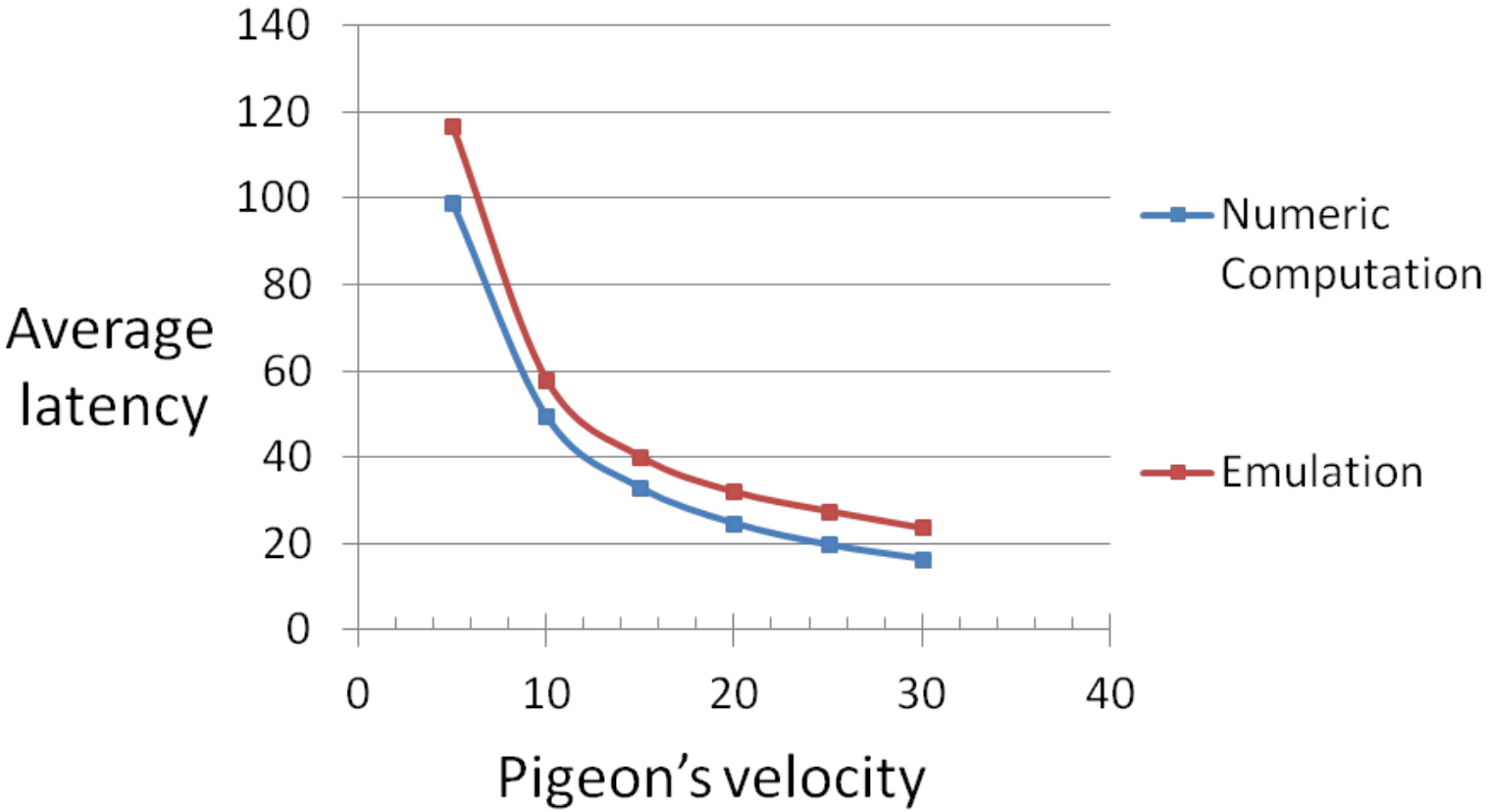


Emulating a Pigeon Network on ORBIT (contd.)

- User needs to produce different emulation programs for building the pigeon network.
- On the pigeon node, in addition to the regular emulation program we also run one emulation program called topology controller (TC).
- The TC uses a packet filter to control the connectivity of the pigeon with the host nodes.
- Use **OML APIs** inside the emulation programs to inject observed data into the OML server.

Emulation Results

As an example, we present one plot from the emulation. It also compares the result with the numerical computation.



HOWARD UNIVERSITY
1867

Outline of the Talk

- An Overview of the Project
- Preliminaries on Pigeon Networks
- Background of ORBIT testbed
- Emulating a Pigeon Network on ORBIT
- Critiques on ORBIT

HOWARD UNIVERSITY
1867

Strengths of ORBIT

- Most of the time the testbed is in working condition.
- Helpful administrators
- The web service to reserve the resources works perfectly
- Multiple sandboxes: an experimenter needs not reserve the whole grid for development.
- Isolation (RF enclosure) in sandboxes is helpful.

HOWARD UNIVERSITY
1867

Limitations of ORBIT

- Multiple users cannot do experiments simultaneously on the grid (BIG PROBLEM)
- No solution better than packet filter for emulating tens of nodes on the grid
- Typically, the "omf-5.2 load image" takes long time. Also, many nodes may fail to load
- Seldom we observe that experiment does NOT stop after the intended time interval
- Lack of detailed/consistent information on ORBIT wiki

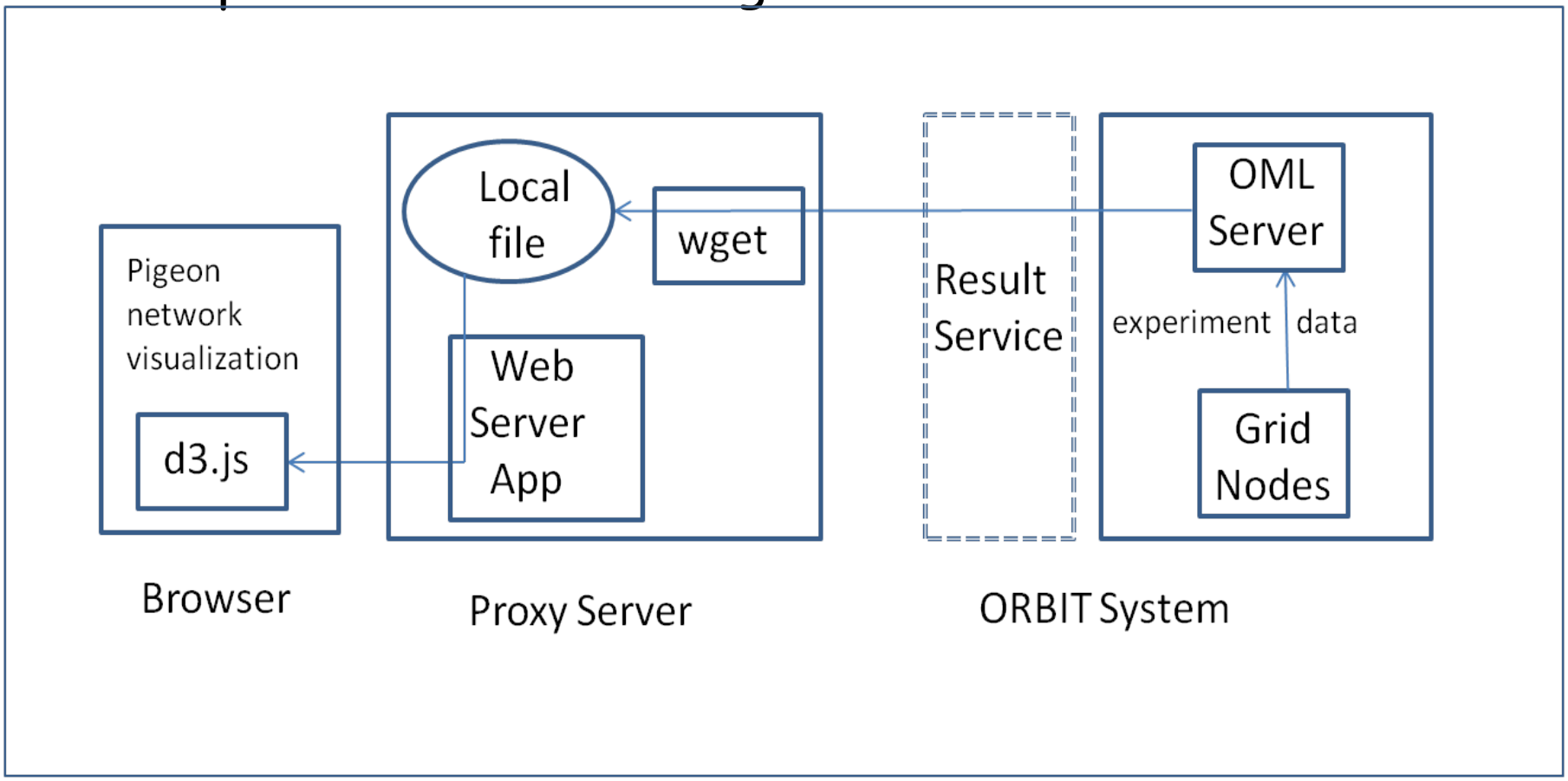
HOWARD UNIVERSITY
1867

Limitations of ORBIT (Contd.)

- Needs to improve the reliability of a long experiment (involving tens of nodes).
- Unavailability of support (to the best of our knowledge) for running an experiment over multiple instances (BIG PROBLEM)
- Sometimes our experiments involving tcp data transfer (with big messages) between nodes halt unexpectedly.
- No readymade support for topology visualization

Our Real-time Topology Visualization Module

- The visualization module (d3.js library) retrieves the real-time experiment data through the ORBIT **result service**.



Our suggestions as an experimenter

To automate an experiment with multiple instances

- Allow the user to submit a generic resource request to the experiment controller (EC) for a set of nodes (say 100 nodes)
- Allow the user to define a group with generic set of nodes (say 20 nodes).
- A special *for loop* construct could be made available as an ED feature, in which each iteration might correspond to one instance of the experiment.

HOWARD UNIVERSITY
1867

Our suggestions as an experimenter (Contd.)

- While data is injected to OML, the experiment's current instance number should be recorded in the database.
- Provide a preparation clause in ED to allow the user to specify the parameters for different instances
- Allow ED to start two applications (of the same group) one after the other with some delay in-between, if needed.

HOWARD UNIVERSITY
1867

Our suggestions as an experimenter (Contd.)

- Wiki to have updated and consistent information
- Wiki should have an efficient search tool
- Wiki could have more non-trivial example code

Questions??

Thank You