Work in Progress: TUNIE: A Large Scale Emulation Testbed for DTN Protocol Evaluation Based on Network Virtualization

Yong Li†, Pan Hui†, Depeng Jin†, Li Su†, Lieguang Zeng†
† Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
‡ Deutsche Telekom Laboratories/TU-Berlin, Ernst-Reuter-Platz 7, Berlin 10587
yong-li07@mails.tsinghua.edu.cn

ABSTRACT
In order to provide communication services in Disruption Tolerant Networks (DTNs) where it lacks of end-to-end paths between the communication sources and destinations, a variety of routing schemes have been proposed. Consequently, the DTN research community needs a credible, flexible and large scale platform to accurately evaluate their performance. Existing simulation tools and wildlife testbeds have drawbacks of either unrealistic DTN environment or not being available for each researcher in terms of large monetary and time investment. In this paper, we propose TUNIE, a large scale emulation testbed for DTN protocol evaluation based on network virtualization. TUNIE is a realistic DTN protocol evaluation environment composing real applications and network nodes with high credibility, flexible controlling, abundant choices and supporting large scale experiments. TUNIE uses virtualization and OpenFlow technologies to generate virtual DTN nodes and simulate the opportunistic links depending on different mobility scenarios of random mobility, human or vehicular mobility trace.

1. INTRODUCTION
Performance evaluations via theoretical models, simulations or experiments are important to algorithm and protocol design in Delay Tolerant Networks (DTN). Recently theoretical analysis frameworks, such as Markov and Ordinary Differential Equation (ODE) models, are used to evaluate the DTN performance in an immediate way. However, since the large underlaying abstraction, their evaluate accuracy usually cannot be guaranteed. Moreover, the ability for these model to evaluate complicate DTN protocols is seriously limited. Therefore, simulation and experiment based performance evaluation are needed. Current simulation approaches can be classified into two types. One type is software-based simulation, for example, using general network simulation software like NS-2 and OPNET, or using specialized DTN simulation tools like The ONE [4] and ORBIT [3]. However, software simulation is hard to emulate the DTN environment. Especially for the wireless link and node mobility properties, it is difficult for them to implement the realistic environments in terms of both physical properties and network phenomenons. On the other hand, some real life DTN testbeds, like UMass DieselNet [1] and ORBIT [3] are built to carry out experiments in evaluating DTN related algorithms and protocols. However, setting up an experiment in such testbed involves in vast investment in terms of monetary and time. It is unrealistic for every researcher to have such testbed. Moreover, realistic testbeds are difficult to support setting large scale experiments.

In order to overcome the shortages existing in both the simulation tools and experiment testbeds, we propose TUNIE, a large scale emulation testbed for DTN Protocol evaluation based on network virtualization. TUNIE is built on the design goals of high credibility, flexible controlling and supporting large scale experiments, which enables setting up hardware-based opportunistic DTN networks with real applications and network nodes. Specifically, we use host virtualization technology to generate virtual DTN nodes, which can be customized in terms of routing protocols, applications and network nodes. Since the high capacity of the multi-core serves, TUNIE is easy to build a large scale DTN network with thousands of nodes. In terms of opportunistic links in DTN, we use OpenFlow [6] to control the link of each node pairs in terms of links up time, link duration and link transmission quality. TUNIE integrates different mobility scenarios including three random mobility models, four human mobility trace and two vehicular trace, and it is convenient for users to set up realistic DTN performance evaluation environments.

2. SYSTEM DESIGN
This section describes the system design of TUNIE. In contrast to existing DTN simulation tools, our design focuses on the goal of providing a reputable and controllable emulation testbed for accuracy DTN protocol evaluation using the technology of network virtualization. We first present the system goals, then describe the system design.

2.1 Design Goals and Choices
The primary goal of TUNIE is to enable controllable and reputable performance evaluation of DTN protocols. First, in order to avoid the inaccurate evaluations, which usually happen when using simulation tools like NS-2 and The ONE, TUNIE should provide an accuracy performance eval-
ulation environment. This environment should include realistic node mobility scenarios, wireless link layer properties, and different network layer implementations. Second, it should afford enough customizations to enable the implementation of new algorithms and deployment of new protocols. Third, it should support large number of nodes in the scales of hundreds and thousands. In TUNIE, we rely on a virtualized wired testbed to implement DTN environments by control the wired links, which behaves as intermittent connectivity with time varying bandwidth and delay. We use virtual machines to run customized routing protocols and applications, and use a centralized control center to control the link behaviors according to different mobility models and scenarios. To enable large scale experiments, TUNIE uses network virtualization to enables many logical nodes to operate on the same, shared physical infrastructure. Moreover, the virtual nodes behave like realistic nodes, and efficiently use the physical infrastructure.

2.2 System Overview

In order to support large scale experiments and node customization, we use operation system virtualization technology to set up virtual nodes as DTN nodes. In the node architecture, we implement the usually used routing and forwarding protocols like epidemic, two-hop forwarding, spay and wait, etc. To simulate the realistic communications in DTN paradigm, which uses the intermittent opportunistic connection to transmit packets, we use OpenFlow to control the link events such as link up and down according to different inputs of the mobility scenarios. Existing DTN simulations like The ONE [4] omits the details of the wireless link characteristics and simply assume that any two nodes can communicate with each other when they are in the transmission range of each other. However, the bandwidth of wireless link varies with the time and spaces. In TUNIE, we use the link virtualization technologies to control the real-time transmission rate of each link according to the wireless link behaviors.

The system architecture is shown in Fig. 1. We can observe that there are two important features. First, we use host virtualization to emulate DTN nodes. More specifically, we use XEN based virtualization to run a serial of virtual machines as DTN nodes, and allow users to program each nodes to customize their own designed algorithms and protocols. Second, OpenFlow is used to control the link of all DTN nodes to let them behave like opportunistic links. Specifically, a controller running in any PC transforms the mobility settings by the experimenter to the link up and down events, and then transmits these events to the OpenFlow switches, which connect all the virtual nodes. The OpenFlow controls the connectivity between all nodes pairs, and let the nodes together behave like a DTN network.

Virtual link is a key part of TUNIE. We use the link virtualization technology to enable the simulation of wireless opportunistic links. First, for each virtual nodes, we set up with one virtual interface, which is a tap device. Then, we use the software bridge in XEN virtualization environment to connect all the tap devices with the physical interface in the host node, which is connected by the OpenFlow switches. Then, the OpenFlow switch can let any two virtual node set up a link at any time, and also block any connections at any time. These link controlling is achieved by the flow management in OpenFlow, which is further controlled by the centralized controller. The link view of the TUNIE is shown in Fig. 2. It shows that OpenFlow switch connects the physical multi-core serves, and then each virtual nodes also are connected with each other. By the OpenFlow link controlling function, all the links between any two nodes are controlled by the system. Therefore, it is easy for experimenters to simulate all kinds of mobility scenarios.

3. EXPERIMENT WORKING FLOW

The experiment working flow of TUNIE is described in Fig. 3. When we set up an experiment in TUNIE, we first need to configure the controller by setting the overall network parameters like the number of nodes, mobility model, etc. Then, TUNIE will generate the virtual nodes by communicating with the required multi-core serves. Then, users need to login the virtual nodes to configure and prototype their algorithms and protocols. In TUNIE, the controller will translate the experiment setting into link control events, and through the OpenFlow flow rules to control the link between virtual nodes to behave as opportunistic links. In the following subsections, we will introduce our considered mobility scenarios, routing and application settings in TUNIE.
3.2 Routing and Applications

The largest vehicular data trace available was the taxi trace, which contains the longitude and latitude coordinates of the taxi’s location, timestamps, and speed. Specifically, we utilized the GPS devices to collect taxi traces and records every 15 seconds for moving taxis. The specific information contained in such a report includes: the taxi’s ID, locations and timestamps, and GPRS modules to report the specific locations.

In collecting the taxi trace, we used the mobility trace logs obtained from 27,000 participating taxis, which were equipped with GPS devices. The specific information contained in such a report includes: the taxi’s ID, locations and timestamps, GPRS modules to report the specific locations, and the specific information contained in such a report includes: the taxi’s ID, locations and timestamps, GPRS modules to report the specific locations. Packet can be generated by copies when they are received, which is similar to the request-response type applications. The packets generated in TUNIE are used to emulate the DTN opportunistic links. TUNIE opens lots of choices in terms of mobility scenarios, routing protocols and applications to users to simplify setting up experiments.

4. CONCLUSION

In this paper, we design and implement TUNIE: a network virtualization based DTN testbed. TUNIE integrates XEN virtualization and OpenFlow technologies, where XEN enables setting up large scale DTN network and OpenFlow is used to emulate the DTN opportunistic links. TUNIE opens lots of choices in terms of mobility scenarios, routing protocols and applications to users to simplify setting up experiments.

5. REFERENCES


