System Engineering Summary: GENI WiMax Project

(Points of contact: D. Raychaudhuri, ray@winlab.rutgers.edu; Ivan Seskar, seskar@winlab.rutgers.edu ; Sampath Rangarajan sampath@nec-labs.com)

This document provides a summary of the open GENI WiMax base station under development at WINLAB, Rutgers University with the collaboration of NEC Laboratories, Princeton. The information provided is based on the initial plan and work done prior to Dec 2008, and will be updated periodically during the course of the project.

1. High Level System Outline: This GENI Spiral 1 project is aimed at providing a state-of-the-art IEEE 802.16e WiMax base station as an open, programmable and virtualizable cellular base station node. This open GENI base station node ("GBSN") device is intended to support flexible experimentation in wide-area mobile network service scenarios similar to today's cellular systems. The GBSN will support programmability at both radio link and network routing layers via an "open API", and will work with off-the shelf WiMax handsets and data. The overall goal is to leverage emerging carrier-class cellular wireless and access network platforms to provide a reliable, high-capacity and cost-effective solution for wide-area mobility in GENI.

The emerging WiMax standard (802.16e) is a good technology base for such an open base station node because it represents the state-of-the-art in radio technology (OFDMA, dynamic TDMA with QoS, MIMO and mesh modes, etc.) and is expected to offer cost advantages relative to the cellular → 3G → LTE roadmap, particularly for small deployments. Also, initial WiMax products are IP-based and are typically bundled with much fewer vertical stack protocols than corresponding cellular options such as UMTS and LTE. NEC's Release 1 802.16e base station product is viewed as an excellent starting point for the addition of GENI specific open programmability features. Although the NEC base station was designed for IP-based network applications, we have been able to determine a way to unbundle the basic layer-2 functionality of the device and make it accessible through an external control API. This makes it possible for us to develop Linux-based GENI code on an external PC controller, substantially meeting all the layer 2,3 programmability and virtualization requirements in a manner that is consistent with the approach used for wired GENI routers.

Figure 1 below shows a schematic of the WiMax base station router and its connection to the rest of the GENI network. As shown, the WiMax base station is typically connected to a GENI access network with layer 2 switched connectivity using Ethernet or optical fiber technology. The figure also indicates three distinct interfaces associated with the GENI WiMax base station. These are the GENI control interface for experimenter access to virtual networks (slices) supported by the GBSN. Initially, this project will use the ORBIT management framework (OMF) as the control interface between the outside world and the base station controller (which is a Linux PC). This is the primary external interface relevant to a GENI experimenter. The second interface internal to the GBSN is the R6+ interface by which the base station controller communicates with the base station hardware (which includes its own internal controller running a proprietary NEC operating system and control/management software). The R6+ interface exposes the hardware features such as assignment of MAC/PHY resources (i.e. OFDMA time-frequency slots, power levels, service classification, etc.) to each flow, as well as management interfaces for initial configuration, scheduler policy selection and queue management.

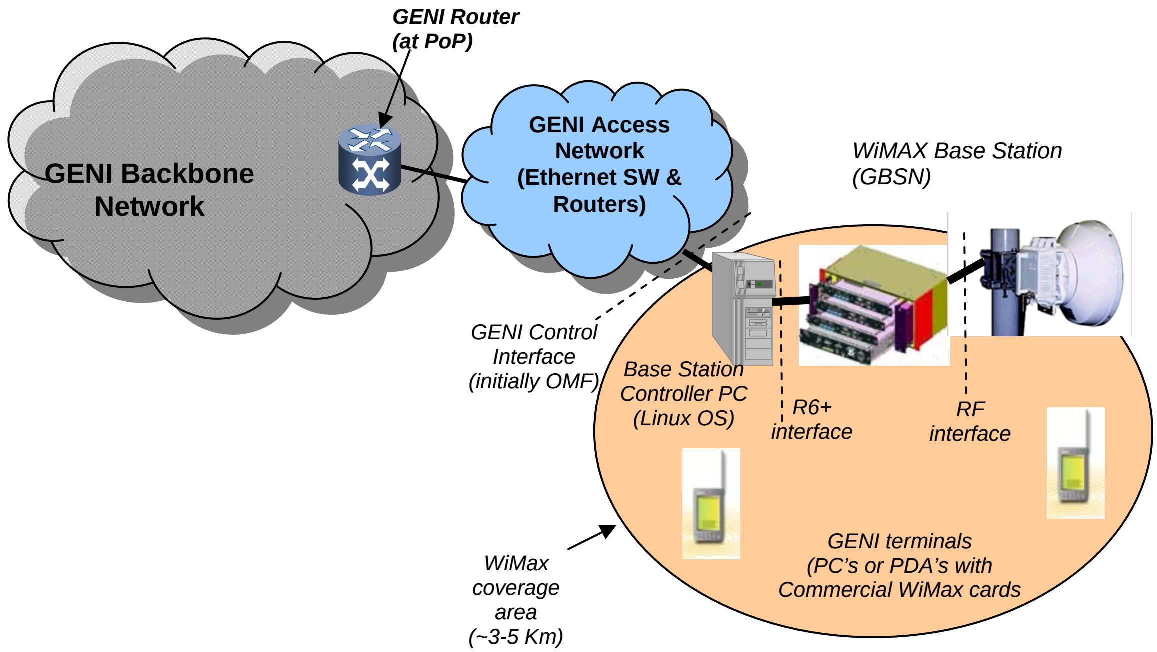

Fig. 1. High level diagram of WiMax base station and its interfaces to the GENI network

The base station will initially be set up at a cellular collocation site at Rutgers University's Busch campus in Piscataway, NJ. As shown, the coverage area is expected to be about 3-5 Km radius, covering the entire campus and also some parts of Highland Park, New Brunswick and Piscataway. In terms of end-user devices supported by the WiMax network, the 802.16e base station's signal is compatible with a growing number of WiMax mobile platforms (for example, the Samsung M8000 or Nokia N800 and Motorola WiMax handsets).

2. Hardware Details - 802.16e Base Station with Open API: The NEC Release 1 WiMAX base-station hardware (photo in Fig. 2) is a 5U rack based system which consists of multiple Channel Cards (CHC) and a Network Interface Card. The shelf can be populated with up to three channel cards, each supporting one sector for a maximum of three sectors. The BS operates in the 2.5 Ghz or the 3.5 Ghz bands and can be tuned to use either 5, 7 or 10 Mhz channels. At the MAC frame level, 5 msec frames are supported as per the 802.16e standard. The TDD standard for multiplexing is supported where the sub-channels for the Downlink (DL) and Uplink (UL) can be partitioned in multiple time-frequency configurations. The base-station supports standard adaptive modulation schemes based on QPSK, 16QAM and 64QAM. The interface card provides one Ethernet Interface (10/100/1000) which will be used to connect to the high performance PC. The base station has been tested for radio coverage and performance in realistic urban environments and is being used in early WiMAX deployments – typical coverage radius is ~3-5Km, and peak service bit-rates achievable range from 15-30 Mbps depending on operating mode and terrain. Note that these service bit-rates are significantly higher than those achievable with first generation cellular technology (such as EVDO), and should be sufficient to support advanced network service concepts to be investigated over GENI.

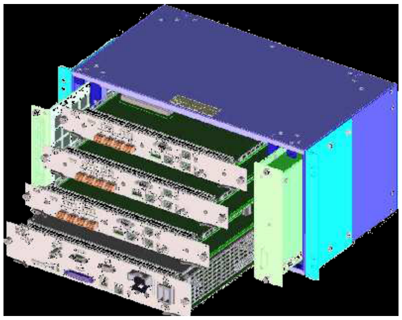

Fig. 2 NEC's Rel 1 802.16e BS

The 802.16e base station allocates time-frequency resources on the OFDMA link with a number of service classes as specified in the standard – these include unsolicited grant service (UGS), expedited real time polling service (ertPS), real-time polling service (rtPS), non-real time polling (nrtPS) and best effort (BE), as shown in Fig. 3. The radio module as currently implemented includes scheduler support for the above service classes in strict priority order, with round-robin, or weighted round-robin being used to serve multiple queues within each service class. These packet queuing and service scheduling features are expected to provide adequate granularity for virtualization of radio resources used by each slice in GENI. It is noted here that OFDMA in 802.16e with its dynamic allocation of time-frequency bursts provides resource management capabilities qualitatively similar to that of a wired router with multiple traffic classes and priority based queuing. The GENI slice scheduling module to be implemented in the external PC controller is responsible for mapping "*Rspec*" requirements (such as bandwidth or delay) to the available 802.16e common packet layer services through the open API. Slices which do not require bandwidth guarantees can be allocated to the nrtPS class, while slices with specific bandwidth requirements (for quantitatively oriented experiments, for example) can be allocated to the UGS category.

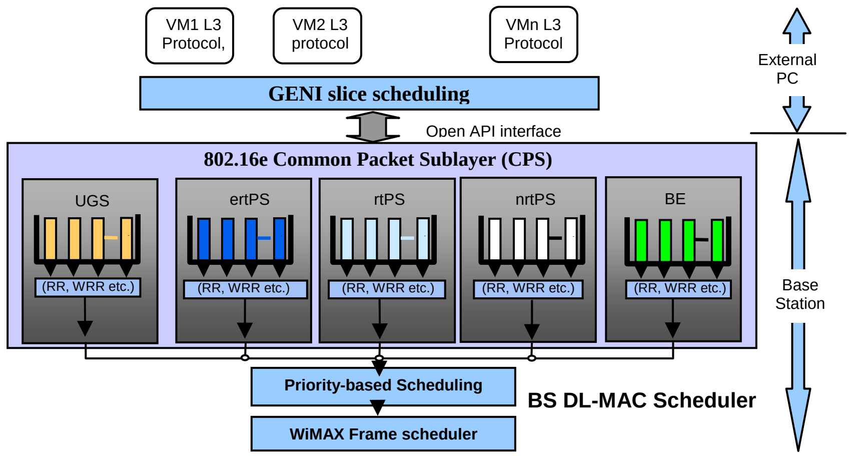

Figure 3. Packet Scheduling in the 802.16e BS & Interface to GENI Slices

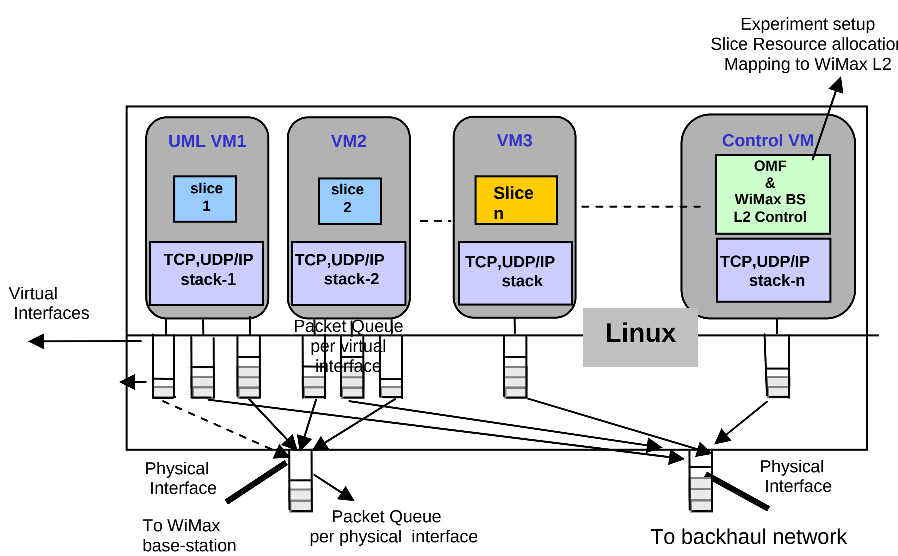

Figure 4. External Base Station Controller Architecture

3. GENI Control Interface: As mentioned earlier, the GBSN includes an external controller that runs Linux. In the initial prototype, we will use the ORBIT Management Framework (OMF) software to interface the base station to other parts of the GENI network. The Linux controller is implemented using UML virtualization, although we will later consider upgrades to other VM platforms. The OMF software is implemented via "Orbit grid services" on the control slice shown in Figure 4. For more information on OMF, please refer to documentation at http://www.orbit-lab.org/wiki/Documentation .

The controller provides support for multiple slices assigned to the GENI WiMAX node. Each slice will run within its own virtual machine (using software such as UML – User Mode Linux) as shown in Fig. 4. Each VM will be capable of providing multiple virtual interfaces, so that programs loaded on a slice that runs within a virtual machine can emulate its own router and perform IP routing. Virtual interfaces will be mapped to physical interfaces based on the next hop for a virtual interface. The controller will receive IP packets from the base-station on the R6+ interface mentioned earlier. When a packet is received, it will be forwarded to the appropriate slice for further processing. The outgoing IP packets from a slice will be placed on queues specific to a virtual interface. Outgoing packets on virtual interfaces mapped to the layer 2 interface of the WiMAX base station will be tagged so that they can be assigned traffic class and bandwidth parameters (BE, ertPS, rtPS etc.) as determined by the flow CID (connection ID).

The L2 base station control software on the external controller provides APIs to both control Layer 2 parameters and also to receive L2 specific information from the base station. An experimenter's program within a slice (obtained through the OMF control interface) can use these APIs to modify L2 parameters as well as receive L2 specific data both at load time and also at run time. Slice scheduling and resource allocation at the external controller (Layer 3) and at the base-station (Layer 2) will be specified using mechanisms similar to the *RSpec* command under consideration for the GMC.

4. Running a GENI Experiment: The WiMax base station described will be integrated into the ORBIT management framework. An experimenter will be able to access the WiMax network through the ORBIT portal and use available ORBIT scripting, experiment control, management and measurement tools to run their experiment. Of course, the experimenter will also have to set up and physically deploy necessary end-user equipment (PC's, mobile devices, laptops) within the coverage area. OMF facilities will be extended to provide software downloads for Linux-based end-user devices. For more details on running an ORBIT experiment, refer to http://www.orbit-lab.org/wiki/Tutorial .

5. Experimental Measurements: We also plan to use features of the ORBIT measurement library for collecting real-time measurements for experiments. The library will run on the GBSN controller and record per flow or per packet measurements and maintain databases for each experiment, and make it available through the ORBIT experiment control portal. The framework will handle both Layer 2 and 3 measurements. The collection library will aggregate the measurements and send them to a collection server running in the OMF management system.

References:

[1] "Wireless virtualization in GENI", GDD-06-17, 2006, at www.geni.net/GDD/GDD-06-17.pdf

[2] Larry Peterson, Steve Muir, Timothy Roscoe, and Aaron Klingaman, "PlanetLab Architecture: An Overview", May 2006, at http://www.planet-lab.org/files/pdn/PDN-06-031/pdn-06-031.pdf

[3] "*GENI Management and Control GMC"*, GDD-06-15, 2006, at http://www.geni.net/GDD/GDD-06-15.pdf

[4] A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford "In VINI Veritas: Realistic and Controlled Network Experimentation," in *Proceedings of ACM SIGCOMM, Pisa, Italy*, 2006.

[5] D. Raychaudhuri, et al, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *Proceedings of WCNC* 2005. See also: www.orbit-lab.org