Virtualization and Slicing of Wireless Mesh Network

Sachin Lal Shrestha, Junhee Lee, Song Chong
School of EECS, KAIST
Daejeon 305-701, Rep. of Korea
sachinls, junhee@netsys.kaist.ac.kr, song@ee.kaist.ac.kr

In this extended abstract, we present the technique for Wireless Mesh Network (WMN) slicing to accommodate several experiments simultaneously.

Introduction

Experiments performed on a network provide an exact performance of network algorithms and protocols. While initial research ideas can be evaluated through analysis, simulations, and emulations, implementation and deployment of these ideas on a realistic environment helps in systematically identifying and addressing many practical problems that systems typically encounter. There is no substitute for a real life environment to capture the complexity of multipath and fading that is inherent to wireless. Therefore, understanding wireless environments requires extensive prototyping and experimentation, in order to uncover new insights that lead to improvements in system design. Research idea approved at simulation level takes greater effort and sometimes impossible to port into a real network. Hence, network protocol development paradigm must change from simulation to testbed. For validating a research idea, constructing a testbed is economically unviable. Our Wireless Mesh Network (WMN) testbed WISEMESH [2] at KAIST is a shared testbed with virtualization capability that can accommodate several experiments in a space division or time division.

The goal of our work is to design and implement a system that virtualizes a wireless network using a large-scale 802.11 mesh testbed. The objective of virtualization is to allow multiple experiments to co-exist on a wireless experimental facility in an efficient manner.

GENI [1], a congress of several (wired and wireless) testbeds scattered around the world, is the forthrunner in promoting virtualization concept through its GENI Management and Control (GMC) platform that threads these testbeds together and facilitates simultaneous multiple experiments. VINI [3] is part of PLANETLAB development that implements virtualization to efficiently support a large number of slices on the limited physical resources of the testbed. Unfortunately, the virtualization techniques that have been used in the existing testbeds do not address the unique needs of wireless environments.

Challenges to Wireless Mesh Network Virtualization

Current WISEMESH slicing with WIVI [2] can achieve space division slicing by stretching the logical space between group of nodes. On these slices, experiments can run in a time multiplexed slices. The method of realizing network slices and virtualization in WIVI are basic and fails to optimize the resource utilization by running simultaneous experiments. The goal of our work is to enhance WIVI to best fit the requirements of experimenters onto a resource restricted WISEMESH. WMN environment introduces keys issues:

- Resource Restriction: Overprovisioning cannot be applied to a key resource in wireless testbeds: the wireless spectrum. Due to the scarceness of wireless spectrum, virtualization needs to accommodate a wider range or resource partitioning models to support a reasonable range of experiments.

- Coherence: When a transmitter of one experiment is active, all of the corresponding receivers and potential sources of interference as defined by the experiment should be simultaneously active on their appropriate channels of operation. This is a network slice synchronization problem.

- Node Heterogeneity: A testbed like WISEMESH that works as a community network as well as a testbed has a high probability that the nodes are not equal. Nodes can differ in wireless capability in setting up several firmware or physical level parameters.

- Topology Control: Wireless experiments heavily depend on the physical attributes such as location of nodes, separation of nodes and ambience. MAC layer and/or PHY layer experiments can give different results with different topologies.

WMN Slicing Schemes and Scheduling Algorithm

Proposed network slicing solves virtualization problems by collectively considering logical overlay networks over a physical network. WISEMESH slicing can accommodate Space Division Multiple Access (SDMA), Time Division Multiple Access (TDMA) along with Frequency Division Multiple Access (FDMA) and the combination of them. But the network slicing of WIVI needs network administrator intervention and thus has severe drawback considering time and resource utilization.

In the proposed model,we choose automated FDMA and TDMA multiplexing as the basis of slicing scheme. This means, multiple experiments share the same resource pool simultaneously in both FDMA and TDMA fashion where a node is virtualized by dividing the 802.11b/g frequencies into orthogonal groups and then these frequency groups are further divided into time slots. In ideal condition, the virtualization enables cent percent utilization of node resources by properly scheduling the experiments onto its network by exploiting the available range of orthogonal frequencies and time slots over the physical network. The scheduler maximizes the network utility function shown in Eq. (1).

$$max \sum_{i=1}^{N} \sum_{f=1}^{M} \int_{0}^{T} P_i^f(t)dt \qquad (1)$$

where $P_i^f(t)$ is the node activation function of node 'i' on an orthogonal frequency 'f'.

Experiments can be divided into three categories based on the duration of the experiments: Short-term (<1 day), Medium-term (<1 month: few weeks) and Long-term (>1 month: several months). An experiment network slice needs to complete within a defined time period. Therefore, the scheduler fairly allocates physical network resources to the logical slices governed by Eq. 2, which is the fairness criteria.

$$\sum_{j=1}^{s} \tau_e^j \leq T_e \qquad (2)$$

τ_e^j is the time slot j of experiment e. T_e is the total time.

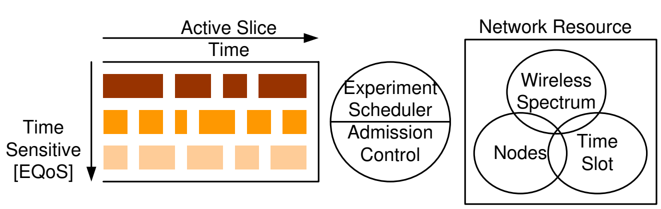

Figure 1. Experiment scheduler model.

Mapping the slices on to a network using FDMA and TDMA for maximum network utilization and timely task completion is an NP-hard problem. We briefly discuss our slicing scheme and scheduling algorithm that provides a suboptimal solution. An experiment/slice queue model with a scheduler and admission control is shown in Fig. 1. Experiment QoS (EQoS) classifies experiments into small delay jitter requirement (100ms), low loss, high throughput. The width of each experiment shows the minimum time an experiment must be active to facilitate a communication process to complete. An experiment is characterized by topology, EQoS, minimum activation period, frequency, node behavior and software. An experiment is admitted if the above requirements along with completion time duration are met by the network.

Once admitted, scheduler prompts respective nodes to initiate the slice with desired frequency, software, and traffic load scenario. Scheduler regulates the logical overlay network changes over a set of nodes to optimize the network utilization and assure fair share of nodes among the slices. The multiplexing of slices takes into consideration the minimum activation time and the EQoS of the experiments.

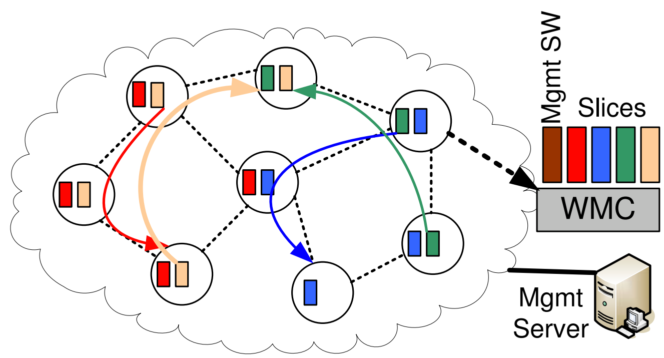

Figure 2. Virtualization Architecture.

Fig. 2 shows the node composition where our virtualization kernel WMC schedules slices that are tightly synchronized with each other. Each node has ethernet connection to WIVI management server.

As discussed earlier, scheduling problem is nothing but a mapping problem of virtual topology onto the physical topology. Let $G_s = (N_s, L_s, R_s)$ be the graph representation of physical network substrate. Let $G_e^i = (N_e^i, L_e^i, R_e^i)$ be the virtual graph representation of experimental topology. N_s is the node substrate, L_s is a link of the substrate network, R_s is the total network resource in terms of time. Similarly, N_e^i, L_e^i, R_e^i are node, link and resource needed by an experiment. R_e^i is the resource time necessary for the experiment to run.

$$Node : f_N^i = N_e^i \rightarrow N_s where f_N^i \in N_s, \forall n \in N_e^i \quad (3)$$

$$Link : f_L^i = L_e^i \rightarrow L_s$$
$$where f_N^i(nm) \in L_s(f_N^i(n), f_N^i(m)), \forall nm \in L_e^i \quad (4)$$

$$Resource : f_R^i = R_e^i \rightarrow R_s$$
$$where f_R^i(n) \leq R_s(n_s), \forall n \in N_e^i, \forall n_s \in N_s \quad (5)$$

Therefore, we define network utilization as, $U_N(v_c, n_s)$, the total node resource utilization with slice/virtualization combination v_c in node n_s.

When we add a virtual node to a node substrate, the node utilization increase is governed by following equation.

$$U_N(v_c, n_s) = U_N(v_c^-, n_s) + R_e(n), \forall n_s \in f_N^i(n) | n \in N_e^i$$

$R_e(n)$ is the resource required by slice virtual node n_s.

Now, we denote $U_N^{'}(v_c, n_c)$ is the remaining resource of substrate node n_s, after assigning a combination of virtual node v_c.

A slice with virtual node n such that $f_N^i(n) = n_s$ can only be realized as virtual network over the substrate provided that $U_N^{'}(v_c, n_c) - R_e > 0$

$v_c = \{n | n \in N_e^i\}$ is the combination of virtual nodes at node n_s. Therefore, ideally, the scheduler objective is to search for a combination of virtual nodes v_c on a physical node such that $U_N^i(v_c, n_s)$ is minimized for that combination.

$$\arg\min_{v_c} U_N^{'}(v_c, n_s) \quad (7)$$

Globally, we have to minimize the network wide remaining resource, therefore,

$$\min \sum_{n_s \in N_s} U_N^{'}(v_c^{n_s}, n_s) \quad (8)$$

where, $v_c^{n_s}$ is the virtualization n_s of node

Upon the arrival of a virtual network request, assign its topology to the substrate network to achieve high utilization of substrate nodes. A special case of the virtual network assignment problem can be formulated as an unsplittable flow problem which is NP-hard. Therefore, the VN assignment problem is intrinsically difficult and a heuristic will be used to solve the problem [4].

Before discussing the algorithm, we look close at experiment types. The experiments can be divided into two types; first, fixed node basis and second, link basis. In the first one, experiments are characterized by selected nodes, while in the second, experiments are based on link properties, such as number of hops, link length and link channel characteristics (obstacles, etc). Fixed node basis experiments are rigid for the change in nodes and just needs to be implemented onto the physical substrate whereas the link based experiments are flexible, in a sense that the nodes with similar link characteristics can be selected. Thus, mapping an experiment onto the physical substrate is to search and allocate the virtual topologies in such a way that aggregate nodes' utilization is maximized.

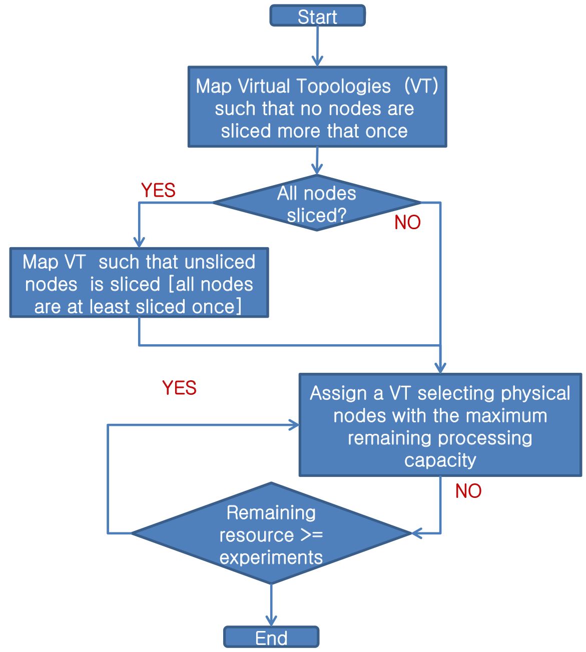

Figure 3. Slicing Algorithm.

Fixed node virtual topologies are assigned onto the physical substrate (Fig. 3). The first assignments are carried on in such as way that none of the nodes undergo more than one slicing. In the second slicing the nodes that are not sliced gets a virtual node assigned of an experiment slice. The second phase of assignment uses both types of experiments. Once all the nodes are assigned with at least a virtual node, the initialization process ends. The mapping that follows maximizes the processing utilization of all the nodes. Until the network resource is greater than required by the experiments and provides that the virtual topology is satisfied, physical substrate is assigned virtual nodes. The mapping hereafter starts with the nodes that have maximum processing resource remaining. The assignment continues till experiments are exhausted or the network resource can not accommodate anymore experiments.

Conclusion

Resource sensitive scheduling optimizes the network utilization and completes experiments in timely fashion.

Node selection based on its suitability to the experiment requirements can facilitate larger number of experiments to coexist together on a shared network like WISEMESH. The present work is under development. We are currently working on node level virtualization capability. Briefly, we are focusing on asynchronous time sharing between several slices on a node. We have developed a concept of MAC and PHY layer marriage. We have dual radio, so that, two different MACs can be tested. Virtualization layer is build on top of MAC layer. The protocol suit is being developed in application layer (user space) where each experiment has its own protocol suit, lookup tables, and network settings. Isolation is key issue that can be handled with this type of virtualization without using VSERVER, UML based linux, etc.

References

[1] Geni: Global environment for network innovations. *http://www.geni.net/*.

[2] Wisemesh: Wireless scalable and efficient mesh network. *http://wisemesh.kaist.ac.kr/*.

[3] A. Bavier. In vini veritas: Realistic and controlled network experimentation. *ACM SIGCOMM 2006*, September 2006.

[4] Y. Zhu and M. Ammar. Algorithms for assigning substrate network resources to virtual network components. *IEEE INFOCOM*, MARCH 2006.