PUMA: Policy-based Unified Multi-radio Architecture for Agile Mesh Networking

Tanveer Gill* Changbin Liu* Lu Ren* Harjot Gill* Boon Thau Loo* Prithwish Basu†
*University of Pennsylvania †Raytheon BBN Technologies
{changbl, tgill, gillh, luren, boonloo}@seas.upenn.edu, pbasu@bbn.com

I. Introduction

Recently, the following trends have emerged in wireless networking: (1) transceivers supporting multiple tunable RF channels are becoming common; (2) devices with multiple wireless interfaces are becoming ubiquitous; (3) software defined radio technologies have developed into an active area of research with commercial uses [13]; and (4) the Federal Communications Commission (FCC) has opened up "white spaces" spectrum to unlicensed devices.

Another wireless networking technology that is gaining popularity is community mesh networking – a cost-effective mechanism for providing high speed wireless Internet connectivity to rural and urban communities where broadband wireless connectivity is unavailable or too expensive. Instead of dealing with mobility or minimizing power usage, the focus here is to increase the network capacity by reducing the interference [7]. Multi-radio multi-channel solutions have the potential to facilitate high throughput scalability in dense mesh network deployment scenarios to meet user needs.

In light of the above technological trends, we propose the demonstration of PUMA (Policy-based Unified Multi-radio Architecture) [8], [4], a platform that aims to develop intelligent network protocols that simultaneously control parameters for dynamic (or agile) spectrum sensing and access, dynamic channel selection and medium access, and data routing with a goal of optimizing overall network performance.

In PUMA, channel selection policies are formulated as constraint optimization problems (COP) that can be succinctly specified using the PawLog declarative language. These policy specifications are then compiled into efficient constraint solver [1] code for execution. The conciseness and customizability of PawLog allow the providers a great degree of flexibility in the specification and enforcement of local and global channel selection policies.

In addition to support policy specifications, PUMA integrates a constraint solver with a declarative networking engine [10]. This enables one to use PawLog to specify the mechanism for distributed channel selection protocols and implement multi-hop declarative routing protocols [9].

II. System Overview

Figure 1 shows an overview of PUMA from the perspective of a single PUMA node. A detailed description of the PUMA system is available in [8].

Channel Manager. The role of the channel manager is to assign available channels to wireless links to satisfy a performance *goal* (e.g. minimize interference in the network, minimize the number of unique channels) while subjected to *constraints* (e.g. regional policies on spectrum usage [12]). In PUMA, we use the PawLog language for declaratively expressing goals and constraints as a *constraint optimization problem* (COP) [14]. These specifications are compiled into executables within Gecode constraint solver [1]. The channel manager takes as additional input *network status* information, which includes network topology and the set of channels available to each node.

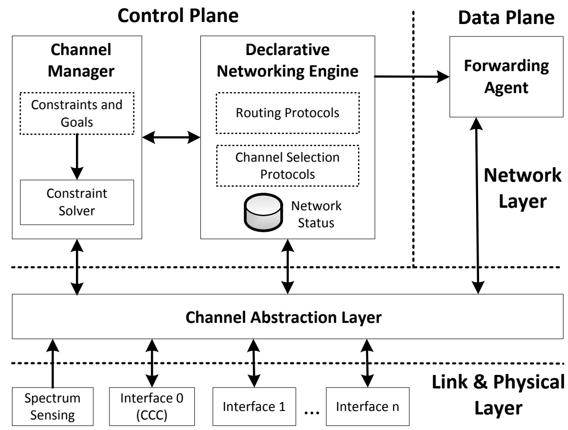

Fig. 1. Components of a PUMA node. The components in dotted lines indicate PawLog inputs.

The channel manager can be deployed either in a *centralized* or *distributed* mode. In the centralized mode, all nodes send their local neighborhood and channel availability information to a centralized channel manager which performs channel assignment for the entire network. In the distributed mode, each node makes individual channel assignment decisions (*local* COP) using its own solver, with only information gathered from neighbors within the vicinity. Nodes then exchange channel assignments results with neighbors to perform further COP computations until all links have been assigned with a channel.

Declarative Networking Engine. At the network layer, the RapidNet declarative networking engine [5], [11] is deployed within the control plane to implement a variety of neighbor discovery and routing protocols also expressed in PawLog. Moreover, channel selection protocols enable nodes to exchange status information among themselves while performing channel assignment using the constraint solver. All network status computed by PUMA (e.g. neighbor discovery, routing, channel availability and assignments) are maintained and stored as RapidNet tables, and made available to other components via callbacks.

Channel Abstraction Layer. Each PUMA node runs a number of multi-channel wireless radio devices (interfaces). Typically, the first interface operates on the *common control channel* (CCC), reserved solely for routing and channel selection protocol messages. A *spectrum sensing* component is able to detect channels available for each interface by periodically scanning a wide range of spectrum. The set of available channel information is then made available to the channel manager through the channel abstraction layer [6], which interacts with multiple radios and presents upper layers with a uniform communication interface. In order for packets to be routed to neighbors using appropriate interface/channel, the output of the channel manager is then used to initialize the channel assignment table at the channel abstraction layer.

Forwarding agent. Lastly, the output of declarative routing is a forwarding table (next-hop for each destination) used by the forwarding agent. Given a destination, the forwarding agent queries the channel abstraction layer to determine the corresponding interface/channel for the next-hop, and forwards the packet accordingly.

III. Demonstration Plan

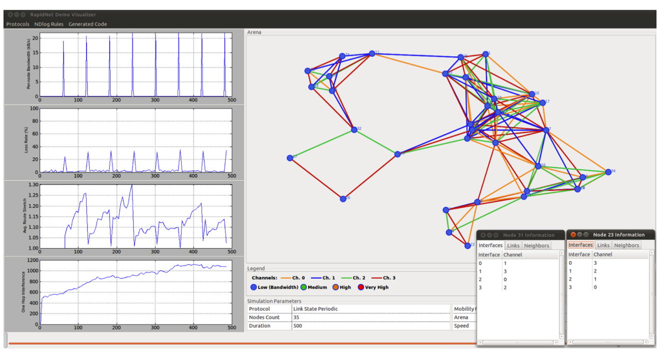

Fig. 2. Screenshot of the PUMA visualizer.

Our demonstration takes as input declarative specifications which are automatically compiled to PUMA code for execution. PUMA is developed using the RapidNet declarative networking engine and the Gecode constraint solver. Our platform is integrated with ns-3 [2] network simulator. PUMA supports multi-radio multi-channel capabilities via the use of the *channel abstraction layer* [6]. In addition to ns-3 simulations, PUMA supports an *implementation mode* that enables us to use actual sockets capable of multi-radio multi-channel wireless communication.

In our demonstration, for ease of deployment, we plan to conduct our live demonstration primarily in the simulation mode. We will showcase both centralized and distributed channel selection policies based on the one-hop and two-hop [15] interference models, and that PUMA results in improved network throughput and lowered loss rates compared to alternative solutions that use single channel or a static partitioning scheme that fixes each interface to use a particular channel. PUMA channel selection policies will be used in conjunction with a variety of declarative routing protocols [9] (e.g. link-state, HSLS, OLSR) that will continuously update routing state in the presence of mobility. Network traces obtained from actual PUMA execution runs are directed to the PUMA visualizer that will display the actual status of nodes during the simulation, side-by-side with actual performance statistics of the protocol. To compliment our live demonstration, we will also present our evaluation results based on measurements obtained on the ORBIT testbed [3].

Our setup will involve two laptops, one running simulation, and the other displaying complimentary evaluation results. We will also use a poster to present the background and technical content of PUMA. Figure 2 shows an example execution of the current version of our demonstration. A declarative link-state protocol is executing, and the graphs on the left show performance characteristics of the routing and channel selection protocols (e.g. bandwidth utilization, packet loss rates). The figure also shows the tables for two nodes' interface-channel mappings. The link colors denote the channels selected by PUMA for communication among each pair of nodes. In our actual demonstration, we will also show other channel selection performance statistics, such as network throughput and data packet delivery ratio.

We will also demonstrate that our use of the PawLog language results in concise specifications in the form of policy goals and constraints. This results in orders of magnitude reduction in code size. For instance, a typical COP formulation for channel selection requires multi-thousand of lines of manually written C++ solver code in Gecode, as compared to a handful of PawLog rules which achieves the same behavior when compiled into solver code. The compact specifications further facilitate policy customizations and enable us to rapidly explore and deploy a variety of channel selection protocols.

IV. Acknowledgments

This work is supported in part by NSF grants CCF-0820208, NSF CNS-0845552, and NSF CNS-1040672.

References

[1] Gecode constraint development environment. http://www.gecode.org/.
[2] Network Simulator 3. http://www.nsnam.org/.
[3] ORBIT Wireless Network Testbed. http://www.orbit-lab.org/.
[4] PUMA. http://netdb.cis.upenn.edu/puma/.
[5] RapidNet. http://netdb.cis.upenn.edu/rapidnet/.
[6] C. Chereddi, P. Kyasanur, and N. H. Vaidya. Design and implementation of a multi-channel multi-interface network. In *REALMAN*, 2006.
[7] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom*, 2004.
[8] C. Liu, R. Correa, H. Gill, T. Gill, X. Li, S. Muthukumar, T. Saeed, B. T. Loo, and P. Basu. PUMA: Policy-based Unified Multi-radio Architecture for Agile Mesh Networking. In *The Fourth International Conference on COMmunication Systems and NETworkS (COMSNETS)*, 2012.
[9] C. Liu, R. Correa, X. Li, P. Basu, B. Loo, and Y. Mao. Declarative policy-based adaptive MANET routing. In *ICNP*, 2009.
[10] B. T. Loo, T. Condie, M. Garofalakis, D. E. Gay, J. M. Hellerstein, P. Maniatis, R. Ramakrishnan, T. Roscoe, and I. Stoica. Declarative Networking. In *Communications of the ACM (CACM)*, 2009.
[11] S. C. Muthukumar, X. Li, C. Liu, J. B. Kopena, M. Oprea, and B. T. Loo. Declarative toolkit for rapid network protocol simulation and experimentation. In *SIGCOMM (demo)*, 2009.
[12] F. Perich. Policy-based Network Management for NeXt Generation Spectrum Access Control. In *DySPAN*, 2007.
[13] C. Santivanez, R. Ramanathan, C. Partridge, R. Krishnan, M. Condell, and S. Polit. Opportunistic spectrum access: Challenges, architecture, protocols. In *ACM WiCon*, Boston, MA, 2006.
[14] E. Tsang. *Foundations of Constraint Satisfaction*. Academic Press, 1993.
[15] Y. Yi and M. Chiang. Wireless Scheduling Algorithms with O(1) Overhead for M-Hop Interference Model. In *IEEE ICC*, 2008.