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

Changbin Liu*, Rick Correa*, Harjot Gill*, Taveer Gill*, Xiaozhou Li*
Shiv Muthukumar*, Taher Saeed*, Boon Thau Loo*  Prithwish Basu†

*University of Pennsylvania  †Raytheon BBN Technologies

netdb.cis.upenn.edu/puma
Motivation

- **Technological/social trends:**
  1. Devices with multiple wireless radios and tunable RF channels
  2. Commercial interests in software defined radio technologies
  3. FCC opens up “white spaces” spectrum to unlicensed devices
  4. Community mesh networking gains traction
Motivation

- **Problem**: can we automatically configure it given performance goals and constraints?
  - Intelligent channel selection
  - Adaptive routing that leverages channel diversity

- **Prior work**
  - Many channel selection and routing protocols
  - Specific choice of protocol is dictated by deployment scenarios
  - A *one-size-fits-all framework* does not exist

Our solution—**PUMA**: Policy-based Unified Multi-radio Architecture

A multi-radio multi-channel wireless mesh network
PUMA Overview

- A novel *declarative* platform for policy-based *channel selection and routing*
  - Declarative: focus on the “what” instead of the “how”
    - Achieve flexibility and avoid *stove-piped* implementations
    - Orders of magnitude reduction in code size [SIGCOMM’05]
- **Users specify in PawLog declarative language**
  - Channel selection policy goal and constraints
  - Routing protocols
- **PUMA automatically generates channel selection and routing configurations to achieve desired goals**
Roadmap

- **PUMA framework**
  - Channel selection
  - Routing
- **Implementation**
- **Evaluation**
- **Summary**
PUMA: Channel Selection

- **Wireless interference:**
  - Links interfere if running on nearby channels

- **Ideally, all links use non-interfering channels**

- **However, constraints exists:**
  - # of channels, interfaces
  - Primary users

- **Optimization under goal and constraints** ➔ **Constraint Optimization Problem (COP)**
PUMA: Channel Selection

**Channel manager:**
- Channel selection ➔ COP
- Input: policy constraints and optimization goal
- Output: determine channel assignments with a constraint solver
- Interacts with channel abstraction layer and declarative networking engine
Declarative Channel Selection: Centralized

Channel selection as COP (one-hop interference model)

goal minimize C in totalCost(C)
var assignChannel(X,Y,C) forall link(X,Y)

s1 \( \text{cost}(X,Y,Z,C) :\) assignChannel(X,Y,C1), assignChannel(X,Z,C2), Y!=Z, C=1, \(|C1-C2|<F\_mindiff\).

s2 \( \text{totalCost}(\text{COUNT}<C>) :\) cost(X,Y,Z,C).

\[
\begin{align*}
\text{min} & \sum_{l_{xz} \in E} \text{cost}(c_{xy}, c_{xz}) \\
\text{cost}(c_{xy}, c_{xz}) & = \begin{cases} 1 & \text{if } |c_{xy} - c_{xz}| < F\_mindiff \\ 0 & \text{otherwise} \end{cases}
\end{align*}
\]

\[
\begin{align*}
\forall l_{xy} \in E, c_{xy} & \in A_x & \text{(3)} \\
\forall l_{xy} \in E, c_{xy} & \notin P_x & \text{(4)} \\
\forall l_{xy} \in E, c_{xy} & = c_{yx} & \text{(5)} \\
\forall x \in V, \bigcup_{l_{xx} \in E} c_{xy} & \leq i_x & \text{(6)}
\end{align*}
\]

Natural mapping: COP \(\rightarrow\) declarative specifications

PawLog rules in PUMA
Channel Selection: Centralized vs. Distributed

• **Centralized**
  – Straightforward, easy to implement
  – Complexity: generally NP hard

• **Distributed**
  – Better scalability, approximated solution
  – Principle: each node solves a local-COP based on neighborhood’s information

**Example:**
Perform iterative coloring until all links are colored
Declarative Channel Selection: Distributed

Channel selection as COP

\[
\text{min} \quad \sum_{l_{xu}, l_{xz} \in E, y \neq z} \text{cost}(c_{xy}, c_{xz})
\]

\[
\text{cost}(c_{xy}, c_{xz}) = \begin{cases} 
1 & \text{if } |c_{xy} - c_{xz}| < F_{\text{mindiff}} \\ 
0 & \text{otherwise}
\end{cases}
\]

\[
\forall l_{xy} \in E, c_{xy} \in A_x 
\]

\[
\forall l_{xy} \in E, c_{xy} \notin P_x 
\]

\[
\forall l_{xy} \in E, c_{xy} = c_{yx} 
\]

\[
\forall x \in V, | \bigcup_{l_{xy} \in E} c_{xy} | \leq i_x 
\]

Goal: minimize \( C \) in \( \text{totalCost}(C) \)

\[
\text{var assignChannel}(@X,Y) \text{ for all } e\text{SetLinkChannel}(@X,Y)
\]

// trigger the start of the solver

\[
d1 \quad \text{eStartSolver}(@X) :- e\text{SetLinkChannel}(@X,Y).
\]

// two-hop assignments

\[
d2 \quad \text{twoHopChannels}(@X,Y,Z,C) :- \text{link}(@X,Y), \text{assignChannel}(@Y,Z,C).
\]

// propagate channels to ensure symmetry

\[
d3 \quad \text{assignChannel}(@Y,X,C) :- \text{assignChannel}(@X,Y,C).
\]

\[
ds1 \quad \text{cost}(@X,Y,Z,W,C) :- \text{twoHopChannels}(@X,Z,W,C1), \text{assignChannel}(@X,Y,C2), W \neq X, Z \neq Y, W \neq Y, C = 1, |C1-C2| < F_{\text{mindiff}}.
\]

\[
ds2 \quad \text{totalCost}(@X,COUNT<C>) :- \text{cost}(@X,Y,Z,W,C).
\]

\[
dc1 \quad \text{assignChannel}(@X,Y,C) \rightarrow \text{link}(@X,Y), \text{availChannel}(@X,C,F,St1), \text{availChannel}(@Y,C,F,St2).
\]

\[
dc2 \quad \text{availChannel}(@X,C,F,St) \rightarrow \text{primaryUser}(@X,C).
\]

Leverage the distributed characteristics of declarative networking
PUMA: Routing

Declarative networking engine:
- Implement routing protocols (e.g., LS, OLSR)
- Adapt routing policies to switch between protocols [ICNP09]
- Leverage channel diversity to improve routing
- Generate forwarding table
Cross-layer Optimization

- **Channel selection and routing**
  - Both in PawLog declarative language
  - Interact with each other

- **Key idea: make them inter-dependent to co-optimize**
  - Routing: leverage channel diversity based on channel selection results
  - Channel selection: give more interference cost to the links which appears more on the best routes
  - Dependencies are easily captured in PawLog declarative rules

- **A distributed cross-layer protocol**
Implementation

- **PUMA prototype**
  - RapidNet ([netdb.cis.upenn.edu/rapidnet](http://netdb.cis.upenn.edu/rapidnet)) declarative networking engine, integrated with ns-3
  - Gecode constraint solver
  - Multi-radio multi-channel wireless capabilities
  - PawLog declarative language

- **Protocols**
  - Centralized and distributed channel selection protocols
  - Cross-layer protocol

<table>
<thead>
<tr>
<th>Protocol</th>
<th>PawLog</th>
<th>Imperative (C++)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized</td>
<td>35</td>
<td>3229</td>
</tr>
<tr>
<td>Distributed</td>
<td>48</td>
<td>4445</td>
</tr>
<tr>
<td>Cross-layer</td>
<td>59</td>
<td>5817</td>
</tr>
</tbody>
</table>

**Table I**

*PawLog and Compiled C++ Comparison*
Evaluation

• **RapidNet simulation in ns-3**
  • Controllable parameters:
    – Network topology
    – # of nodes, radios, channels

• **ORBIT wireless testbed**
  • Two radios
  • IEEE 802.11a/b/g
Simulation in ns-3

• **Settings**
  – 30 nodes in 600m X 600m arena
  – Wireless range: ~ 100m
  – 3 radios with 8 channels

Figure 4: Aggregate network throughput (30 nodes).

Cross-layer > Distributed > Centralized > Identical-Ch > 1-Interface
ORBIT Testbed Evaluation

• **Settings:**
  
  – Node: 1 GhZ VIA Nehemiah, 64KB cache, 512 MB RAM
  – 30 nodes in a 8m X 5m grid
  – 2 radios with 10 channels

Performance trend is consistent with simulation results
Summary

• **PUMA: a policy-based unified multi-radio architecture for wireless mesh networks**
  – Declarative networking engine + Constraint solver
  – Channel selection and routing
  – Cross-layer optimization
  – PawLog: orders of magnitude code size reduction
  – Evaluation in simulation and Orbit testbed
Thank you

netdb.cis.upenn.edu/puma

See you at our demo at Esquire Hall