SOSI Wireless Mesh Research Testbed Development Project

SOSI, CSC, ECE

NCSU
• Many emerging security and network research areas in network availability, reliability etc.
  – Solutions are often through routing, opportunistic MAC, etc. research areas i.e. low down and across stack
  – Availability of testbed a plus for enabling more successful and more practically useful research
• Need a testbed that allows separation of experimental concerns
  – Data transport, E2E control
  – Signaling and control
  – Management algorithms, “knowledge plane”
• Some originally conceived research projects, envisioned continued use beyond
Objective

• Provide a highly flexible testbed environment to users (wireless network/security researchers)
  – Provide modular programming library to researcher-user
  – Allow researchers to code only the part(s) of the stack they want to experiment with

• Enable network design and security research involving routing, power, channel, scheduling control

• Develop an extensible environment
Phases

• Phase 1: Laboratory testbench
  – Purpose: Develop initial software architecture, create initial common codebase

• Phase 2: Pushcart nodes
  – Purpose: Benchmark wireless conditions, prove developed software and envisaged hardware strategies in outdoor settings, survey multiple locations for permanent installation

• Phase 3: Poletop installations
  – Purpose: Provide a (semi-)permanent testbed installation for researcher-users of SOSIMesh
• No wired backbone, 802.11 link based
• Use simple commodity hardware for platform
  – Standard PCs, Atheros cards (2 / 4 per node)
• Use open source software, develop open source software
  – MadWiFi driver, Linux
• Modularize and cleanly separate functionality whenever possible
  – Separate data, control, management planes
  – Allow plugging in/out specific software modules
Software Architecture

- Central control/management node
- All control/signaling traffic is channeled through the **communicator** module
  - Embeds control plane policies such as spectrum usage
  - Uses TCP connections between mgmt and mesh nodes
  - Uses automatic pub/sub mechanism locally, manual/programmatic pub/sub across nodes (planned extension to complete automatic pub/sub in future)
• Mgmt node contains a **data repository** (MySQL)
  – Acts as decoupling intermediary between various data collection/dissemination procedures
• Management software modules implemented in general as paired **managers** and **agents**
  – Managers and agents are clients to communicator
  – Managers interact with data repository
Software Architecture

- Servers for typical OAM & repository
  - Clients may run on control node or elsewhere on Internet
  - Visualizing currently implemented this way
  - Scripting planned to work this way (will adapt from past existing testbed implementation)
Software Architecture

- Some enabling mgmt modules included as system modules
  - Can be replaced but not eliminated
  - Neighbor discovery
    - Agent collects and sends information to manager
    - Manager enters into database
  - Bootstrap routing
    - Manager computes connected topology and distributes to agents for actuation
    - Can be computed from database, currently manual

- Other core mgmt modules
  - Routing
  - Channel/radio control
  - Power control
  - Scheduling (coarse-grain)

- Separate recovery partition with only system modules
Status

• Initial architectural design of software completed
• System modules implementation complete
  – Communicator (signaling plane)
  – Neighbor discovery, basic routing substrate
  – Data exchange repository, visualization and management
• Placeholder implementations of core modules complete
  – Routing, Channel control, Power control
• Other core module placeholders in development
  – Scheduling
• Physical design of pushcart nodes complete – in early testing
• Publications: Survey on Mesh Design submitted to IEEE CST Journal, one more in preparation
• Funding: DURIP awarded, MRI not selected at NCSU
Early Testing of Pushcarts
Upcoming Tasks

• Complete initial cut of code, package (08/09)
• Implement dynamic control (scheduled or adaptive) of radio channel and power (12/09)
• Complete and release SOSIMesh website (on or linked to SOSI website) (12/09)
• Collaborate with researchers inside and outside NCSU to identify users and projects of SOSIMesh
• Develop access scheduling and general resource management plan (adapt existing) (06/10)
• Continue conversation with NCSU Comtech to finalize Phase 3 deployment plan
• Procure and deploy Phase 3 equipment (06/10)
• Continue to improve software platform with user feedback
• Collaborate with SOSI security researchers to continue hardening testbed code to the extent required
Potential Research Projects

• Diverse Routing
  – Geographically diverse multiple routes
    • Also spectrum diverse
  – Split routing - improve confidentiality
  – Redundant routing - improve performability/reliability

• Secure mobility architecture
  – Current technology and use cases can be matched
    • Some are mismatched
  – Identify mismatches and security vulnerabilities
  – Develop protocols in “sub-stack” to converge

• Other projects in mobility, ad-hoc routing, E2E controls e.g. congestion, ...
Team

• Rudra Dutta (CSC), Mihail Sichitiu (ECE)
• Parth Pathak, JunBum Lim, Umang Patel, Mani Pandia, Gaurish Deuskar
• John Bass
• Phil Emer, Dennis Kekas, Mladen Vouk,
  SOSI board of advisors
Thanks!

SOSI Wireless Mesh Research Testbed Development Project

SOSI, CSC, ECE
NCSU