**Motivation -**
- Designing modular and extendable testbed platform for supporting wireless mesh network research
  - Testbed based development and evaluation of protocols generate more realistic results and better reproduce unpredictable wireless environment
  - Current testbeds lack modular and flexible design which makes it difficult to extend implementations and repeat experiments
  - Many implementations are often hardware specific and customized for particular protocol of interest

- While in CentMesh
  - Common operations of protocols (e.g., information gathering and dissemination) are separated from its core functionality
  - Clean modularization of network management algorithms and control plane functionality
  - Portable communication API for mesh networking

**Introduction -**
- CentMesh/SOSIMesh/MeshBed/CWMN
  - MeshBed contains a set of management procedures and their common control plane operators
  - Example of various management processes are routing algorithm, topology management, channel assignment etc.
  - It provides a platform for development and integration of new management protocols without significant overhead

- CentMesh features -
  - “Communicator” API provides reliable communication between network management tasks and enables distributed design in loosely coupled manner
  - Publish/subscribe relationship is used instead of client/server model which allows various management processes to send and receive messages based on protocol topic irrespective of node’s current role in network
  - No dedicated backhaul network or control channels are used which allows remote management simultaneously with other network operations

**Software architecture**
- Provides set of interfaces for protocol/application developers
- Abstraction of complexity of communication by hiding the way underlying message processing works
- Mechanisms and policies clearly separated in terms of management and control processes
- Network is managed using cooperation of several management processes
- All management processes record and use collected/calculated information in XML Database

**Management and Control Processes -**
- Fundamental management processes provided in MeshBed suite
  - Neighbor manager - collects neighbor information periodically
  - Topology Manager – builds network topology based on nodes’ neighbor information
  - Channel manager - allocates channels to links of various nodes based on their proximity and interference
  - Routing manager – generates end-to-end routing paths
- Control Processes – communicator API
  - Reporter and disseminator collect and distribute control information and management decisions

**Hardware and Router Configuration -**
- Hardware and OS
  - Desktop computers – 1.4 GHz processor, 512 MB memory
  - Matrix outdoor kits – Soekris 4826 embedded board
  - Atheros chipset AR5413 (extended range 5006x) AR5212
  - Fedora Core 9 and Madwifi driver
  - Outdoor experimentation using batteries, inverters and moving cart with custom stand for antennas

- Router configuration and failure handling
  - Every mesh router has 3 kernel images and 3 respective root file systems – 1 for failsafe operations and 2 for experimentation
  - Developers can modify any of 2 experimental images/file systems for their implementation
  - Any software failures, hang-ups or kernel crashes are handled using rebooting in failsafe mode

- Fail-safe Mode
  - In case of any software failure, router can be remotely rebooted in failsafe image and required changes can be made
  - Failsafe mode provides privileged access to both experimental images and file systems to deal with failures
  - At least one wireless interface on all routers are used (not dedicated) for fail-safe mode operations
  - Static channel, subnet and routing is used for remotely (multi-hop) accessing any router in fail-safe mode which takes away the trouble of hard reboot especially in outdoor deployment