

The MeshSuite™ Framework for Rapid Prototyping of Large, Enterprise Class Networks and two exemplary use cases. (Click for more)

Far from being a homogenous computing and networking environment, the next generation of Enterprise communications will actually require the integration of an even wider variety of protocols and devices as legacy and IOT applications are finally incorporated into the Enterprise network. Conversely, Enterprises will wish to extend their Software Defined Networking capabilities to encompass these “things” at the edge..

MeshDynamics, in collaboration with our OEM partners, has developed and is testing an evolutionary wireless network concept, where the simulation world and real world are in lock step. Machine learning begins at the simulation end, and drives auto tuning algorithms to meet specific “difficult” use case scenarios.

The refined scripts and applications are uploaded to real mesh nodes to complete the iterative loop. The mesh nodes, through rudimentary low level pub/sub messaging, emulate SDN philosophy and terminology at the network edge. Nodes publish Real time sensor streams. These are aggregated at the root nodes and transmitted to big data servers for analysis.

Developers may customize/tune network behavior with apps running on the mesh nodes, communicating with supervisory apps running on the cloud or a local computer.

At the top of the chain Java source code for the NMS -- and its Heart Beat listening protocol – are offered as an illustration of how customer applications on mesh nodes may generate and listen to their own heart beat protocols. Such heart beats will be broadcasted only to those interested listeners (subscribers), a private communications channel.

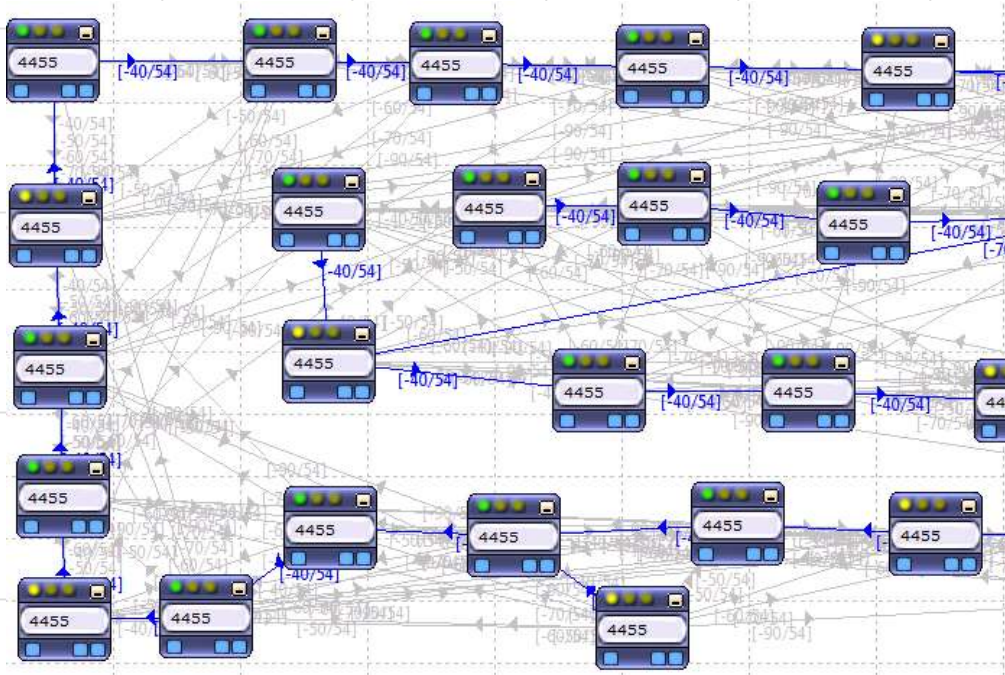
Listeners include mesh node resident apps. (Authorized) Java script apps may access the lower level radio functions, port forwarding rules etc through an intermediary shown as the Application Management Layer above.

The image shows a BitBucket web interface. The top part displays the user profile for Francis daCosta and a list of repositories. The repository 'md6000' is highlighted with a red box. An orange line connects this box to a second screenshot below, which shows the detailed view of the 'md6000' repository. The second screenshot shows a directory structure with various folders and their descriptions.

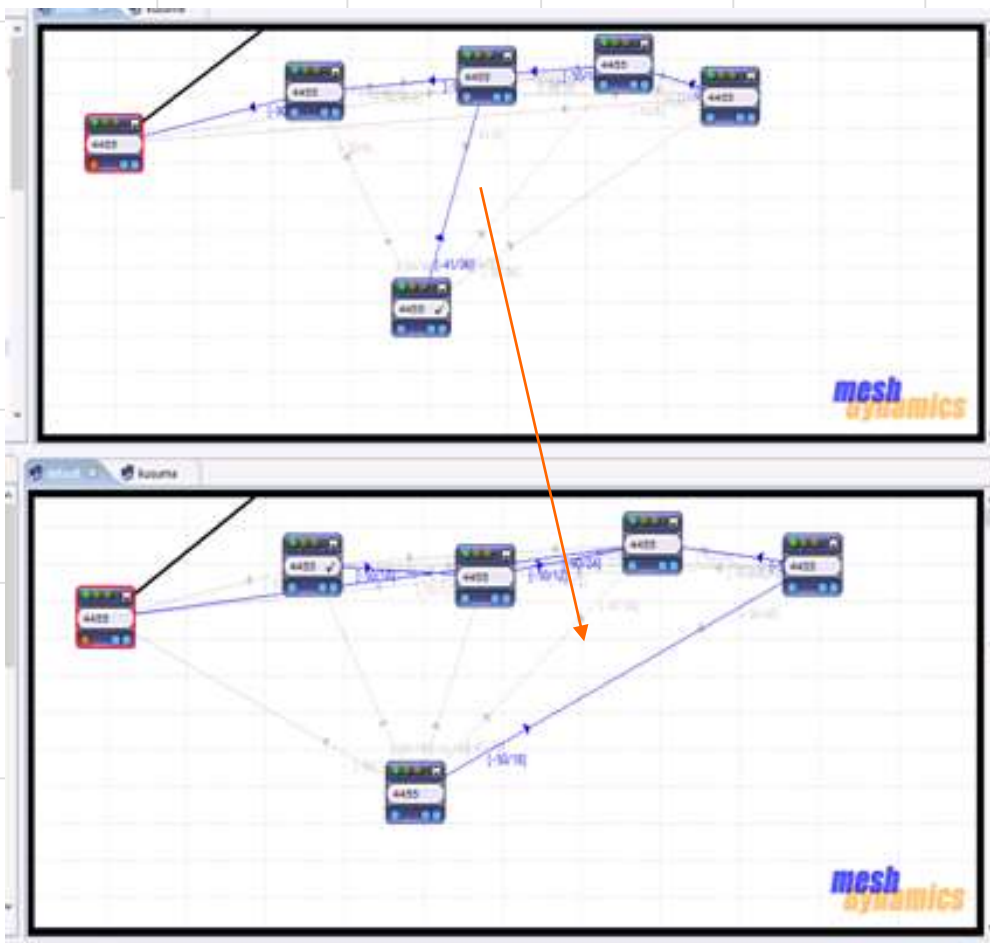
Repository Name	Description
md6000	OpenWRT Images For Multiple Board-Radio Ensembles
mdman-build-server	Build OpenWRT images for Cavium, IMX, and X86 (Simulator)
meshd	Mesh Routing and Control, includes Real time IOT framework
md-man	Manufacturing Dashboard for placing orders/tracking
md-automation	Test automation toolkit and Jenkins for automated builds
mesh-nms	Java NMS source code for integration in J2EE enterprise. API

Folder Name	Description
md6000 / md_es_1408	
..	
TestPlan/MD6000	
compat-wireless	OpenWRT Supported 11ac Radio Card: Compex
design_doc	
embeddedstudios	
gateworks-openwrt_1408	OpenWRT Supported Board Families Gateworks
gateworks-openwrt_1602	
md-mac80211	OpenWRT MAC80211 Abstraction Layer
md_configs_1210	
meshap_patches	
simulator	OpenWRT Images For X86 Simulation
wifi-sim_host	

- Simulator uses X86 image of same code compiled for embedded network processors.
- Automated Testing of Network Performance for Expected Use Cases from Library
- Mesh Nodes Heart beats data logs analyzed. Accelerated Product Development.



Testing the scaling of the mesh networks as more nodes join the network



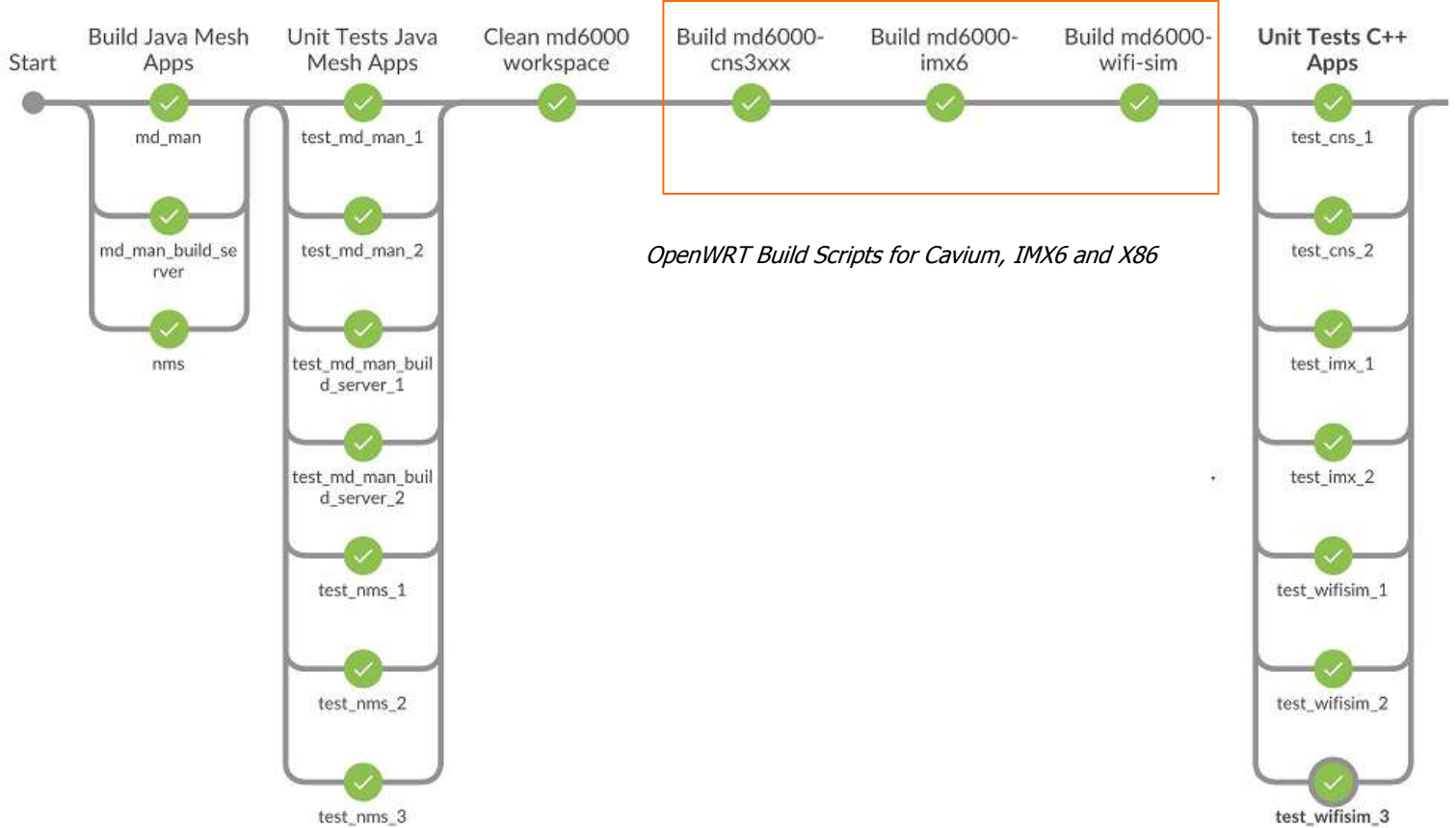
Testing the dynamic Parent selection transition for mobile mesh nodes

iPad 3:27 PM 60%

all 1 Pipeline Changes Tests Artifacts Logout

Branch: — 4h 24m 1s No changes

Commit: — 24 minutes ago Started by user marco



NOTE: Developers may prefer to run Jenkins from the terminal or in scripts: `curl -X POST http://hostname:8080/jenkins/job/nms/build --user XXXX:PASS` The check box for CSRF needs to be unchecked see: <https://wiki.jenkins.io/display/JENKINS/Remote+access+API>

