

"IEEE 802.11s Power Save Modes Support Implementation for Linux"

Dmitry Yuranov
FRUCT Open Source & Linux Lab

Table of Contents

- Introduction
- IEEE 802.11s Power Save Modes
- Power Save Modes Implementation
- Testing
- Conclusion

- Introduction
- IEEE 802.11s Power Save Modes
- Power Save Modes Implementation
- Testing
- Conclusion

Motivation

The main goal of our project called GreenMesh is to improve energy-saving features of Mesh networks.

One of the main issue was to determine how the power saving mechanisms in Mesh networks work.

Implementation of Mesh power save modes in Linux kernel according to IEEE 802.11s standard.

Tasks of the implementation

Based on latest IEEE 802.11s standard the power saving mechanism can be sorted out as next tasks of the implementation:

- indication of mesh STA' power modes in frames;
- a buffering and identification sending frames in mesh STA;
- Mesh Awake Window implementation;
- Peer Service Period implementation.

What's done at the moment

Done: the indication of mesh STA' power modes in Mesh frames.

In process: the support buffering frames in mesh STA and ability a mesh STA to identify peer nodes about presence a buffered frames.

- Introduction
- IEEE 802.11s Power Save Modes
- Power Save Modes Implementation
- Testing
- Conclusion

What is Mesh network?

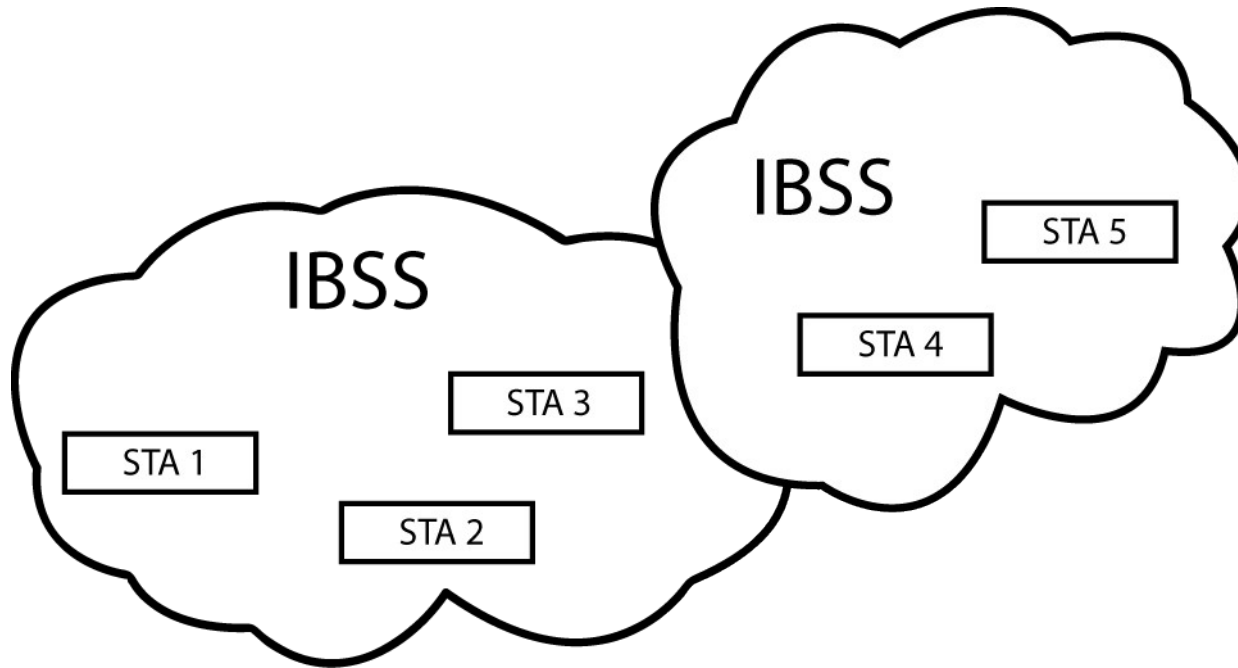
A mesh network is a type of wireless networking that uses redundant and distributed nodes to provide greater reliability and range for any given wireless network.

Combines the features of ad hoc and infrastructure networks

Routing on MAC level

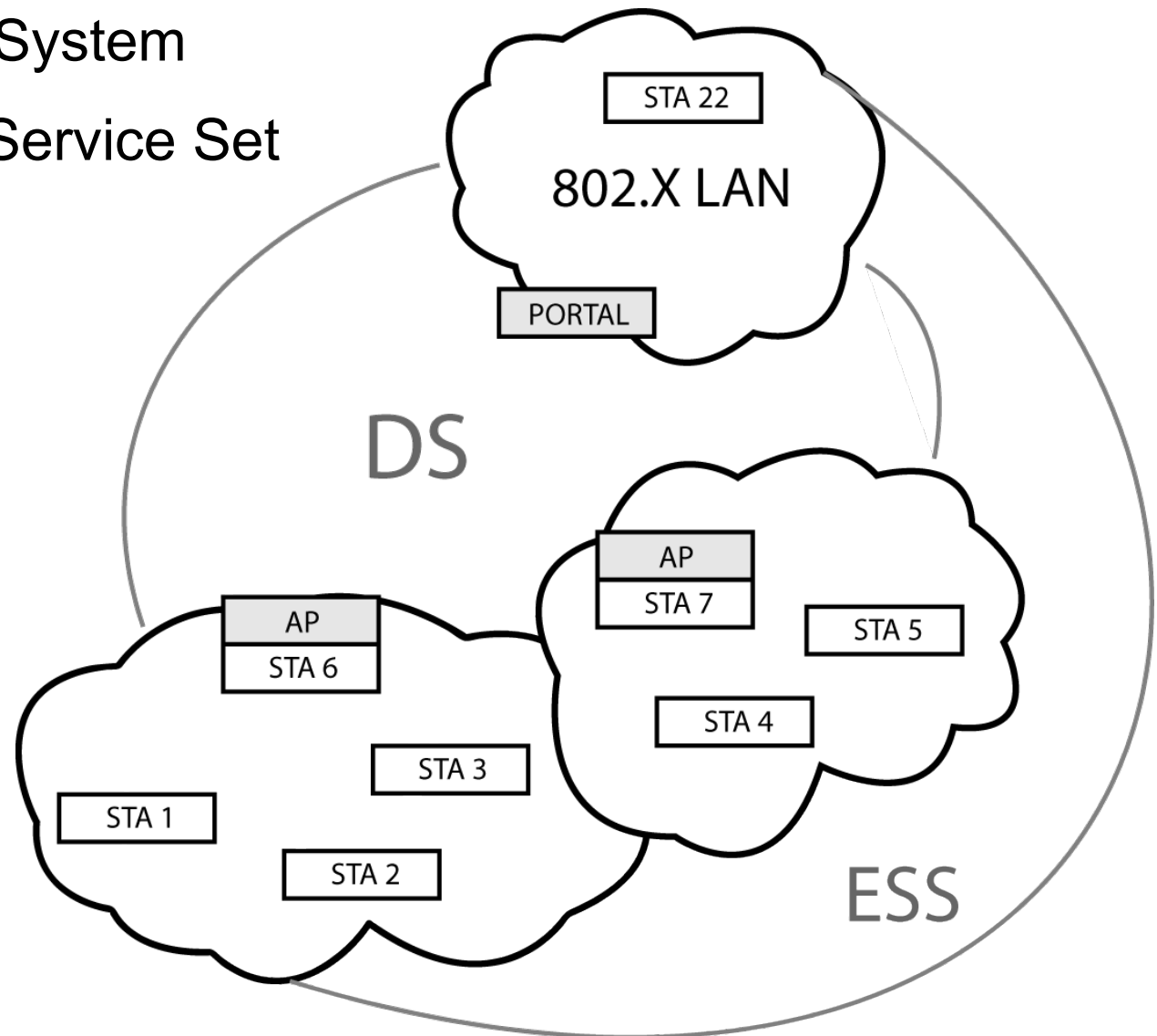
Mesh architecture (1/5)

- IBSS – Independent BSS network (ad hoc)
- STA - Station

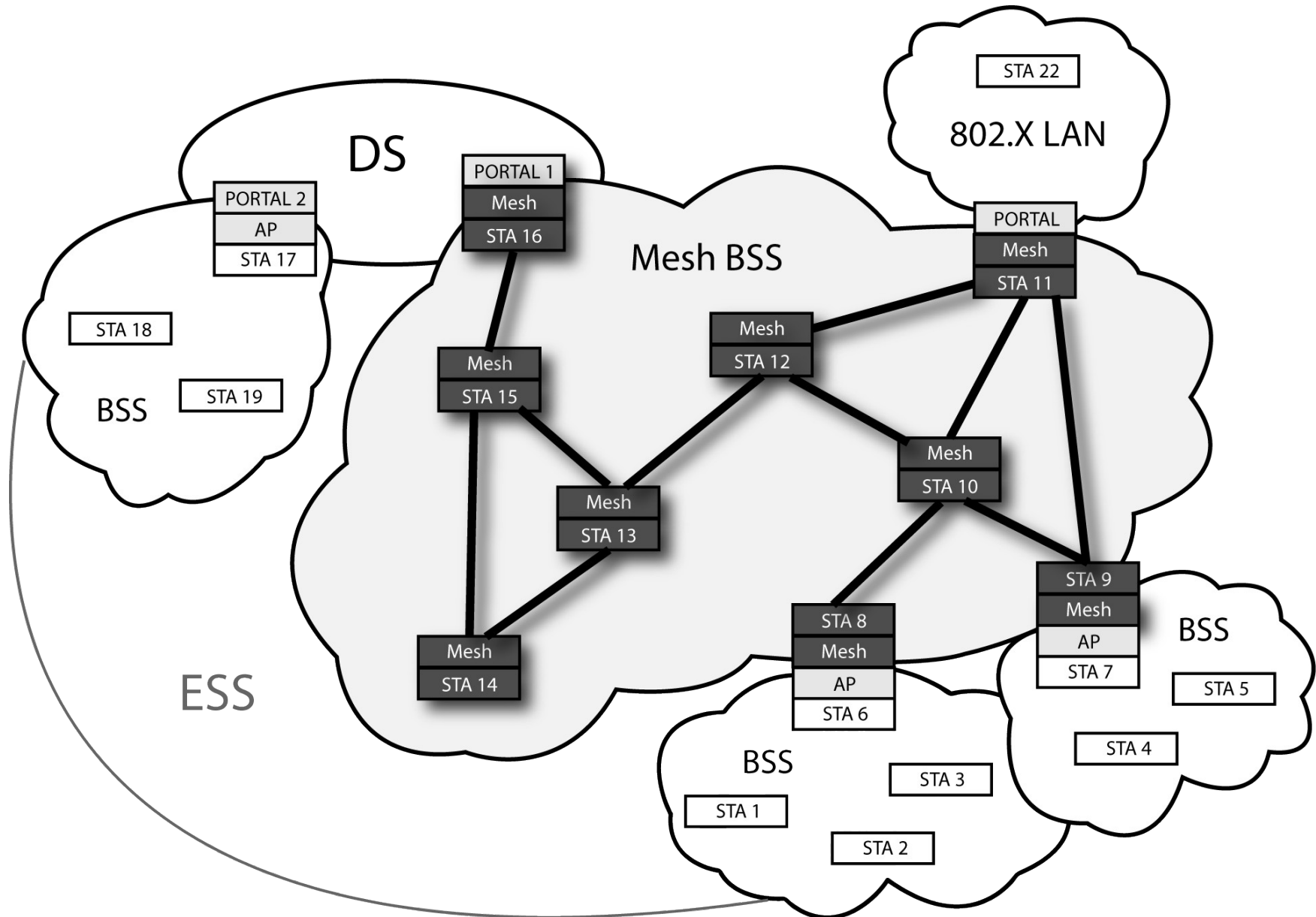


Mesh architecture (2/5)

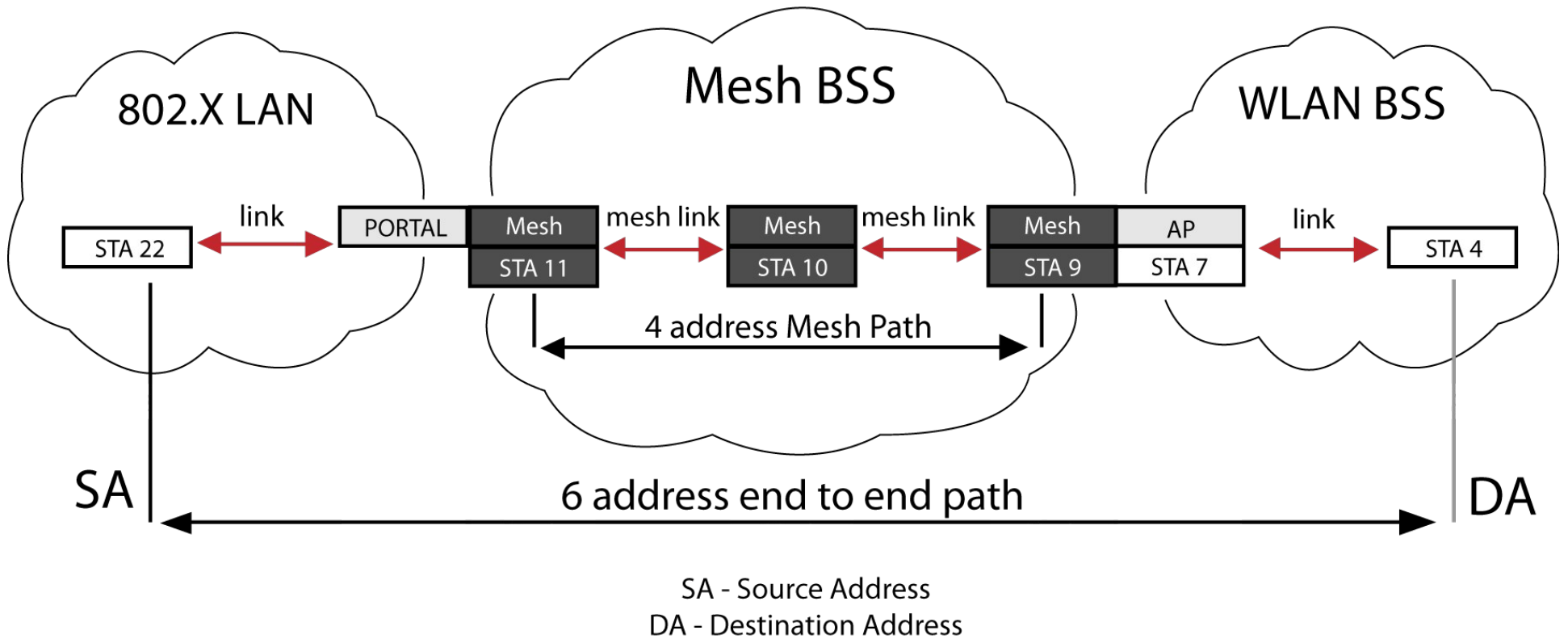
- DS – Distribution System
- ESS – Extended Service Set



Mesh architecture (3/5)



Mesh architecture (4/5)



Mesh architecture (5/5)

- Routing on MAC level.
- Hybrid Wireless Mesh Protocol (HWMP)
- The Airtime Link Metric is used to calculate each pairwise link within the mesh and is defined to be the amount of channel resources consumed by transmitting the frame over a particular link.

The problem of energy saving

- Why is it so important?
- How does it work in Mesh?

Power Save Modes in Mesh (1/4)

States: Awake and Doze

Modes:

- Active Mode - A mesh STA shall be in Awake state all the time.
- Light Sleep Mode - A mesh STA alternates between Awake and Doze states
- Deep Sleep Mode - A mesh STA alternates between Awake and Doze states. May choose not to listen to the Beacons from its peer mesh STA.

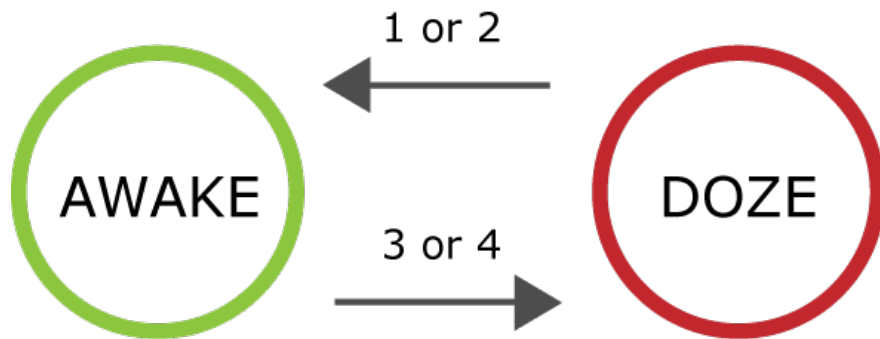
Power Save Modes in mesh (2/4)

The used mesh power mode shall be indicated by the Power Management field and Mesh Power Save Level field.

Activity level	Mesh Power Mode	Power Management field	Mesh Power Save Level field
Highest	Active Mode	0	Reserved
Middle	Light sleep Mode	1	0
Lowest	Deep sleep Mode	1	1

Power Save Modes in Mesh (3/4)

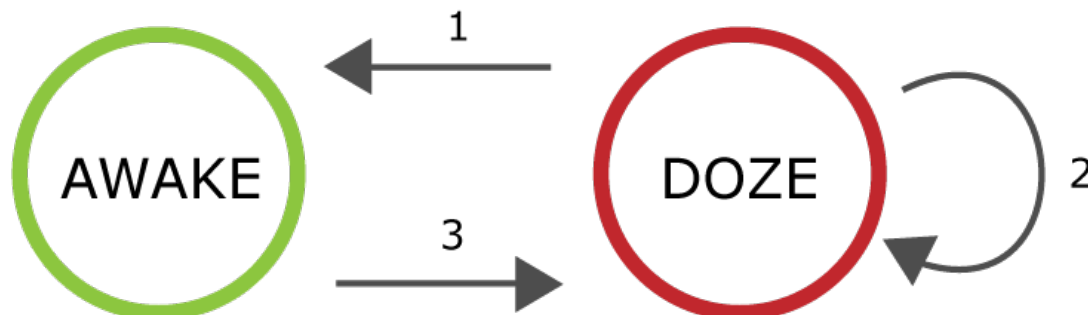
Acting in Light Sleep Mode



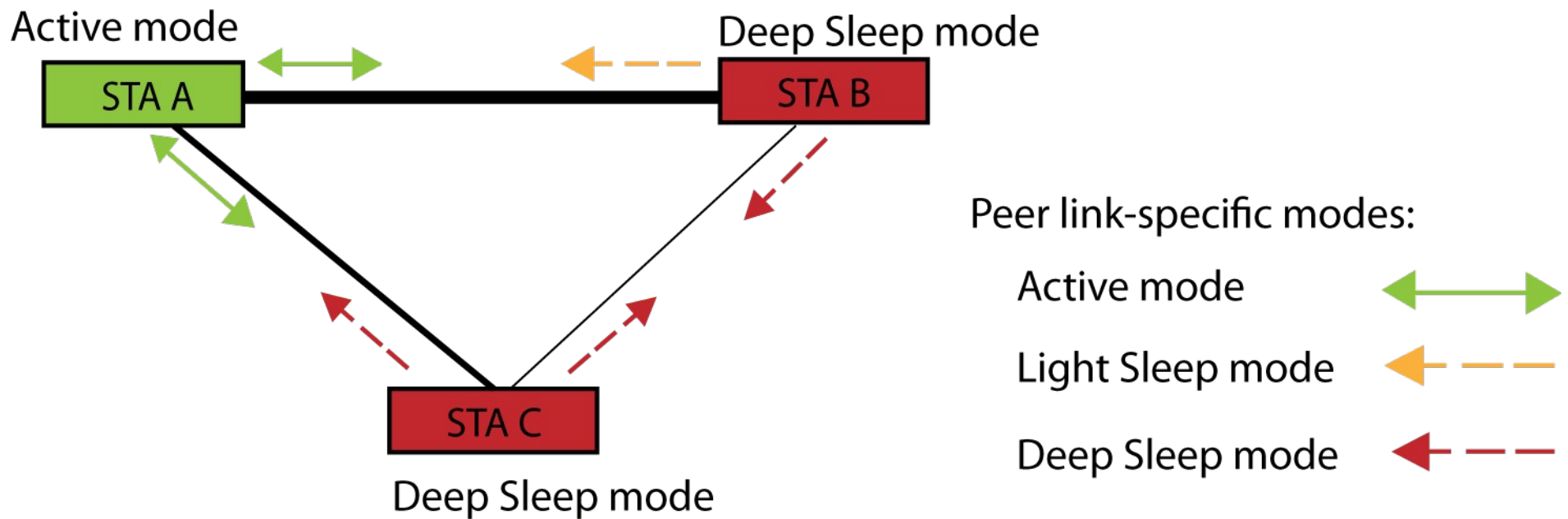
1. Sending DTIM Beacon
2. Listening TIM Beacon

3. DTIM Beacon has sent and Awake Window has been finished
4. Finished listening TIM Beacon

Acting in Deep Sleep Mode



Power Save Modes in Mesh (4/4)



Mesh Awake Window

Mesh Awake Window is a period of time where the mesh STA operates in Awake state.

A mesh STA in light or deep sleep mode shall be in Awake state during its own Mesh Awake Window.

Beacon frame with Mesh Awake Window element from a peer STA indicates that there are a buffered traffic for a mesh STA.

Peer Service Period

When a mesh STA turns into *Awake* state during *Mesh Awake Window*, it may send a trigger frame in order to initiate *a peer service period*.

This contiguous period is used for successful transmission buffered frames towards mesh STAs that operates in light or deep sleep mode.

- Introduction
- IEEE 802.11s Power Save Modes
- Power Save Modes Implementation
- Testing
- Conclusion

Changes in the kernel space

- Adding non-peer power modes
- Adding peer power modes
- Tracking the power modes
- Changes in cfg80211 subsystem and nl80211 interface

Adding non-peer power modes

This has been done in mac80211 subsystem.

- Power Management field of Frame Control is set up in `ieee80211_beacon_get_tim` function during header initialization if power mode is not active.
- Mesh Power Save Level field is set up in `mesh_mgmt_ies_add` function during mesh capability installation if power mode is the deep sleep.

Adding peer power modes

- In transmission function `ieee80211_xmit` power mode is checked.
- Power Management field of frame control and Mesh Power Save Level field of QoS control field are set if necessary to indicate peer power mode.

Tracking power modes

- Unicast data frames and QoS Null frames are tracked to record current peer power mode in `ieee80211_rx_h_sta_process` function.
- A peer power mode is indicated in `sta_info` structure. For this purpose `peer_ps_mode` field is added to this structure.

Changes in cfg80211 subsystem and nl80211 interface

This changes are aimed to:

- getting and setting local and peer link-specific mesh power modes;
- getting and setting mesh non-peer power mode.

Changes in the user space

As we have new parameters in `cfg80211` component we had to add ability to work with them from userspace using `iw` utility.

iw utility (1/2)

1. To work with non-peer power modes:
 - was defined the name and the nl80211 type of a new parameter: «mesh_power_mode».
 - was written two functions to parse and to print power modes.

iw utility (2/2)

2. To work with link-specific (peer) power modes station handler defined in station.c file has been updated.

Summary: what have been done

In kernel space:

- local, peer and non-peer power modes;
- tracking the power modes;
- changes in cfg80211 subsystem and nl80211 interface.

In user space: changes in iw utility

- Introduction
- IEEE 802.11s Power Save Modes
- Power Save Modes Implementation
- Testing
- Conclusion

Build kernel

Hwsim tool is used for testing the mesh stack.

After the configuration of the kernel, it is need to make sure that next features are checked:

```
CONFIG_MAC80211=m  
CONFIG_MAC80211_MESH=y  
CONFIG_MAC80211_HWSIM=m
```

It is needed to apply “Send-To-Self (loop)” patch which implements routing of traffic between local IP addresses externally via ethernet interfaces.

Setup Mesh

In our testbed we created three devices using a `mac80211_hwsim` software simulator.

Each created device doesn't have any hardware, but is represented in the system with according device.

Testing the network (1/2)

To test the network iw utility was used.

Checking that nodes have connection between each other:

```
#iw dev mesh7 station dump | grep -e Sta -e plink  
Station 42:00:00:00:02:00 (on mesh7)  
mesh plink: ESTAB  
Station 42:00:00:00:01:00 (on mesh7)  
mesh plink: ESTAB
```

Testing the network (2/2)

After that an ability of sending packets between nodes was checked.

And final step was to test that power save modes are set. Before that it is needed to check that enum `nl80211_meshconf_params` in `nl80211.h` in `iw` and the kernel are identical.

```
# ./iw dev mesh5 get mesh_param mesh_power_mode  
active
```

Final testing script

We have developed a testing script.

The script gets one optional parameter - the number of mesh points to create.

If testing succeeds script returns a zero value and it exits with nonzero error code if an error has occurred during testing process.

- Introduction
- IEEE 802.11s Power Save Modes
- Power Save Modes Implementation
- Testing
- Conclusion

Future plans

- To extend our implementation with Mesh Awake Window and Peer Service Period.
- To develop a driver notification system which aims to schedule Awake and Doze states for active and sleep modes.

Questions & Answers

Dmitry Yuranov,

d.yuranov@gmail.com

Open Source & Linux Lab,

<http://osll.furct.org>, osll@fruct.org