SAMSUNG
Open Source Group

IoT related MAC layers, header compression and routing protocols

Netdev 2.1

2017-04-08, Montreal

Stefan Schmidt
stefan@osg.samsung.com
Samsung Open Source Group

Agenda

- Intro
- IPv6 over Bluetooth (Luiz Von Dentz / Stefan Schmidt)
- Mesh Link Establishment (Alexander Aring)
- Routing in lossy networks (Michael Richardson)

Scope

- Feels like an Alien topic at Netdev :-)
- Not about performance, data-center, hw offload, virtualization, etc.
- Workshop to create some awareness on the niche we are working in

Scope

- The “IoT” networking space is full of vendor specific solutions
- Ignoring as much as we can here and focus on things with public specifications (IEEE, IETF, etc)
- TLDR; IPv6 all the way to the sensor

Subsystems

- net/bluetooth/: IPv6 over Bluetooth and Bluetooth mesh
- net/{ieee,mac}802154/: short range low-power wireless
- net/6lowpan/: IPv6 adaptation and header compression
- Userspace Unstrung: RPL lossy routing

IPv6 over Bluetooth

Luiz Von Dentz

RFC 7668

- 48 bit MAC addresses, RFC 7668, Section 3.2.2
  - The host part, EUI-64, is formed by concatenating the first 3 bytes of the MAC address, 0xFF, 0xFE and the last 3 bytes of the MAC address. No flipping of the universal/local bit!
  - Source/Target Link Layer Address Option in Neighbor Solicitation/Discovery and Router Solicitation/Advertisement messages contains the 6 byte BT MAC address, with the option length being 1.
  - Compare with 802.15.4, which already has an EUI-64 address assigned or, if a short address is used, creates an EUI-64 address by concatenation as specified in RFC 4944, Section 6. And RFC 4944, Section 8. further documents the Source/Target Link Layer Address Option to have length 2 if an EUI-64 address is used (contains 6 bytes of zero padding) or length 1 if an 16 -bit address is used (contains 4 bytes of zero padding).

RFC 7668

- Multilink subnet, RFC 7668, section 3.2.1
  - Central acts usually as 6lowpan border router (6LBR)
  - Peripheral acts usually as 6lowpan node (6LN)
  - Ignore a 6LN that created an identical random MAC address
  - 6LBR maintain a dedicated connection to each 6LN
  - Direct 6LN to 6LN communication using link-local addresses not possible
  - IPv6 prefix needed for 6LN to 6LN communication

RFC 7668

- IPv6 link-local address autoconfiguration
- Any mechanism for creating addresses in the announced prefix(es)
  - Router solicitation
  - Address creation, e.g. IPv6 privacy, etc.
- Neighbor discovery, Section 3.2.3:
  - Address registration option (ARO) from RFC 6775 included in neighbor advertisement options, central learns other addresses than link-local
  - RFC 6775, Section 2, describes the ARO neighbor messages as updating the neighbor cache, whereby the neighbor cache becomes a lookup table for device addresses
  - A Bluetooth LE 6LN MUST NOT register its link-local address.

6loTUN network driver

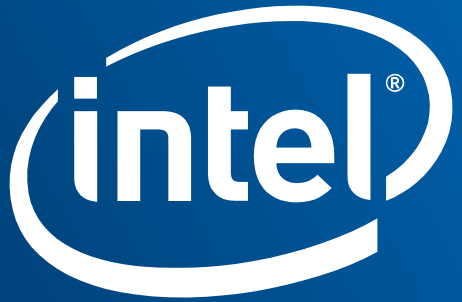
- Allow userspace to create interfaces that uses 6lowpan layer.
- Move L2CAP channel to userspace, probably in bluetoothd, which has the following advantages:
  - Can use authorization agent directly
  - Use common code to handle profiles/services
  - Async IO
  - No new kernel APIs needed
- How to do routing in case there are multiple peers connected?
  - Match the link address from neighbor discovery cache, this might only be possible if done from kernel.
    - 15.4 has dealt with neighbor discovery cache by introducing hardware headers which are then filled in with link addresses from cache, but userspace would have to remove these once sending over the air.
  - Create a different interface for each peer and attach them to a bridge.

6loTUN over GATT (Low Priority)

- iOS and Android still don't support L2CAP CoC (Connection oriented Channels) which is used by ipsp
- Custom GATT service acting as the transport channel in userspace.
- Requires fragmentation/reassembly to work properly
- Tunnel L2CAP over GATT aka. L2GATT:
  - This enables any profile/service using L2CAP CoC to be tunneled over GATT, including the already defined IPSP and OTP/OTS.
  - It does cause some overhead, 7 bytes per packet compared to L2CAP CoC, and may require some reimplementation of L2CAP layer on other OSes, anyway similar overhead would happens regardless of the chosen protocol.
  - Other solutions that require per channel service don't scale very well as they turn out to be very expensive in terms of memory as for each new channel a number of handles/UUIDs have to be allocated in the GATT database, also discovering multiple instances of a service with the same UUID may require several more round trips, not to mention it may confuse stacks.

Contact

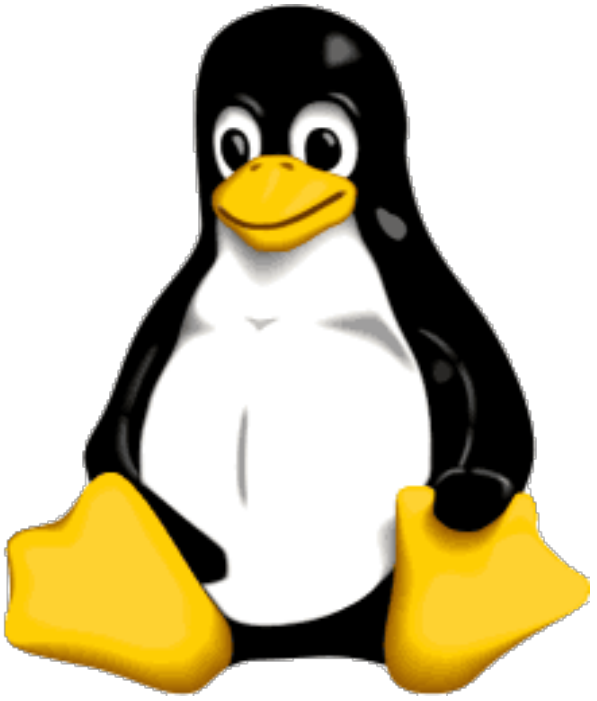
email: luiz.von.dentz@intel.com
irc: vudentz@freenode.org


intel

10 minutes about my MLE experience

netdev 2.1

Alexander Aring
Pengutronix
Hochschule Rhein-Main
<aar@pengutronix.de>



Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Pengutronix.

What's MLE?

Mesh Link Establishment

- Currently IETF Draft for 802.15.4 only
  - Developed by ZigBee IP
  - UDP Protocol
  - Marked as "dead"
  - Moved to 6lo WG → no activity yet

- Used in Thread Specification
  - They name it MLE
  - But it is NOT MLE (a fork)

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

What does MLE?

Three Major Tasks

- Link Establishment (Security)
- Link Quality Detection
- Network Parameter Distribution (not interested)

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

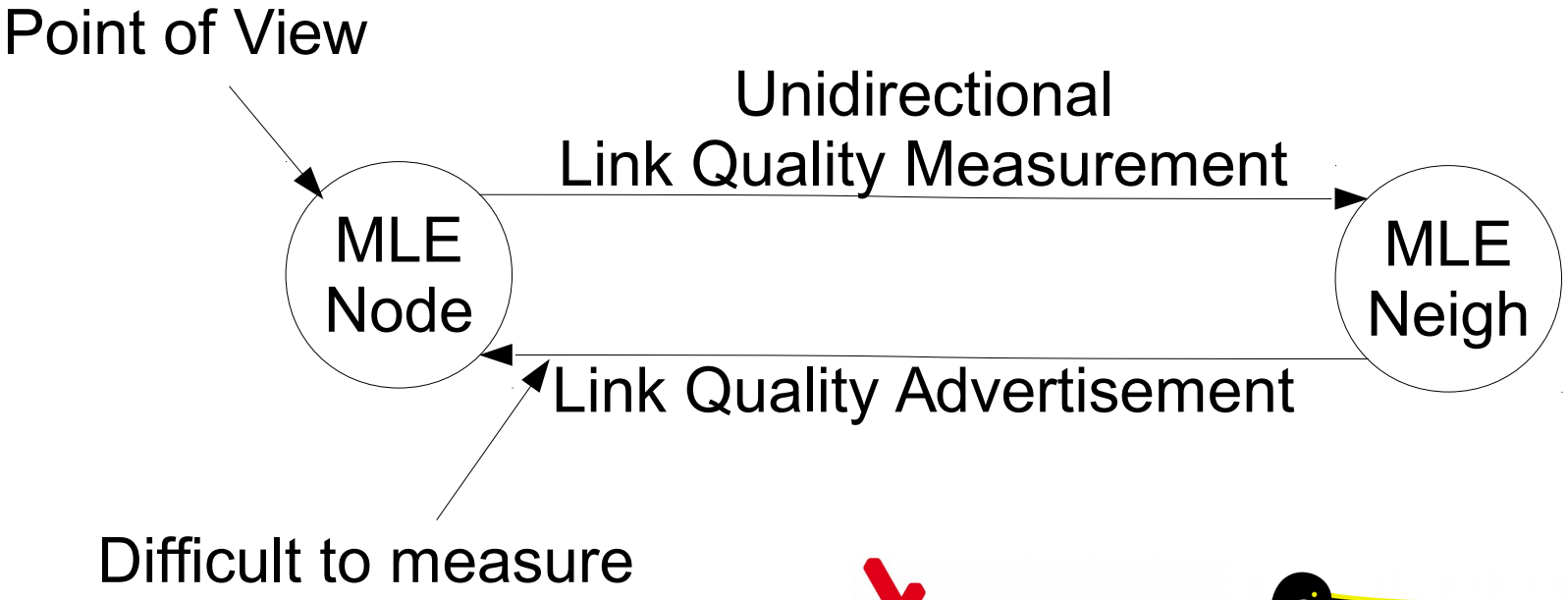
Link Establishment

- Blocks all non-MLE Traffic until Link Establishment to Neighs
- Security Parameter exchange
  - For Peer To Peer not solved by IEEE
  - Handshake: Response & Challenge
  - “Frame Counter” for Replayed Message Protection

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

Link Quality Distribution

- Periodic Messages
  - Getting better Link Quality Values
  - For Bidirectional Link Quality Measurements (asymmetric links)



Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Pengutronix.

Solved issues

MLE Messages needs read/set MAC Header Information on UDP
(common issue in 6LoWPAN)

- CMSG Data recvmsg/sendmsg
- IPV6_RECVPKTINFO_L2
- Link Layer specific Attribute

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Final Opinion

Big challenge
Link Establishment

- Without it?
  - Nothing work
  - Solve IEEE 802.15.4 Issues
  - Get new issue: Key Distribution
- Thread MLE (closed Spec.)
  - Provided by OpenThread?
  - IPv6 Neighbor Discovery Handling?

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

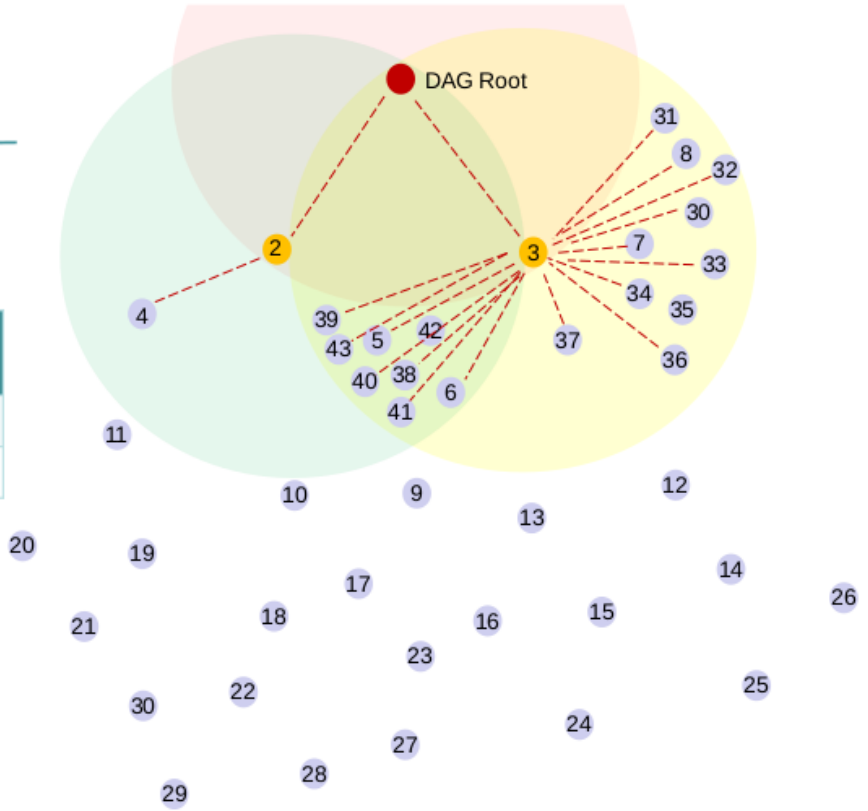
LLNs and Linux
IETF ROLL (RPL)
And 802.15.4

Michael Richardson
mcr@sandelman.ca
http://unstrung.sandelman.ca/

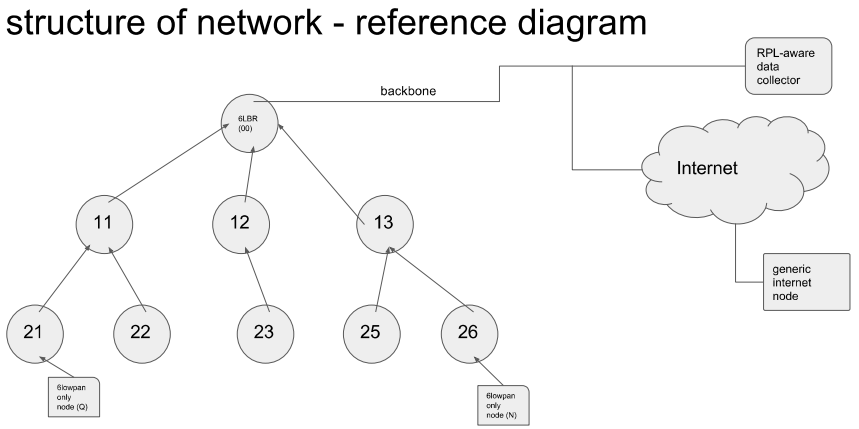
Some mesh network images

Unbalanced load (cont.)

Parent Node	Number of Children	Children from shared area
2	1	0
3	~~10~~ 18	~~2~~ 8



What are LLNs and ROLL



- Specifications
  - IEEE 802.15.4
  - RFC7228: constrained nodes
  - RFC4919: 6lowpan
  - RFC6550: ROLL
  -

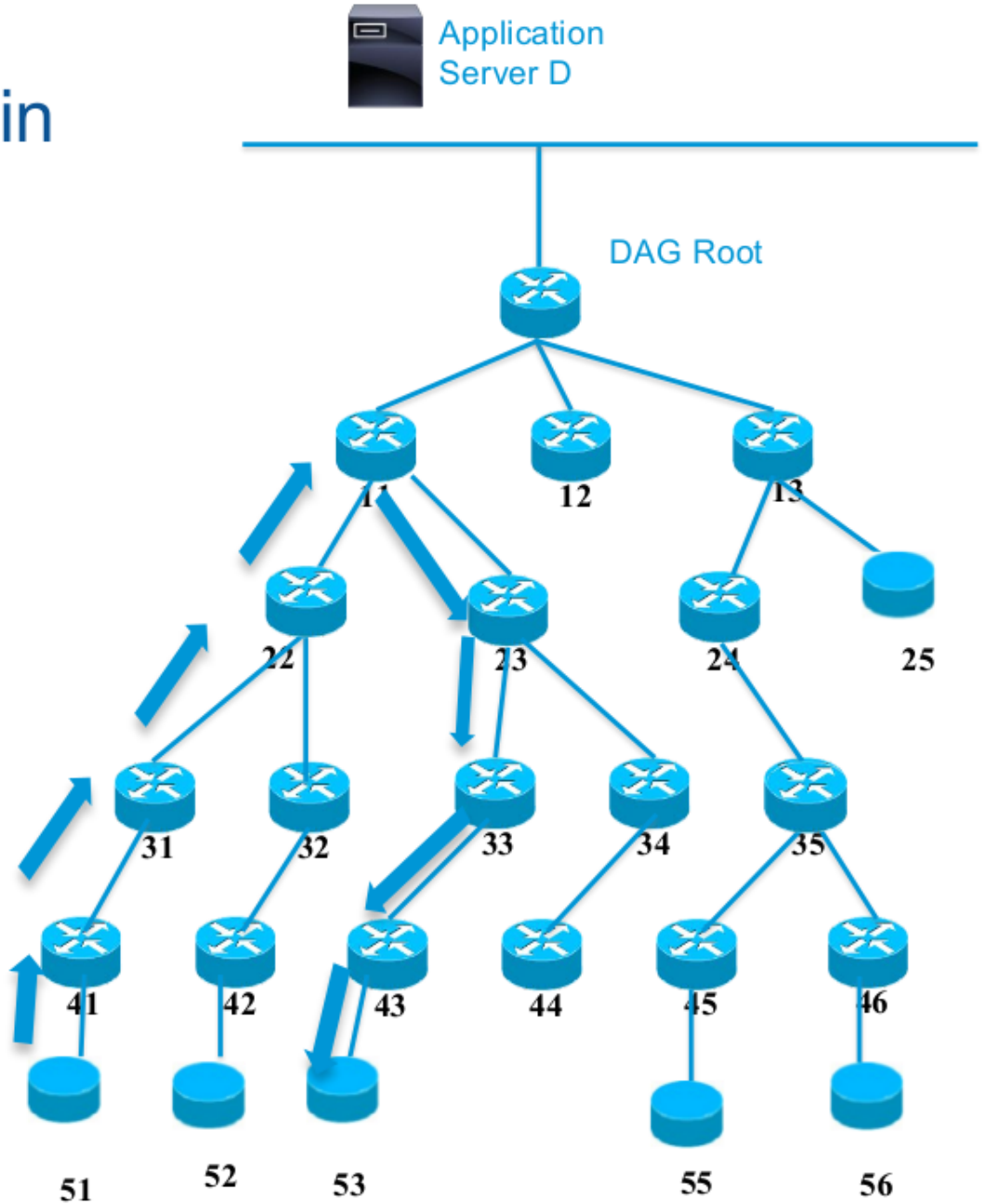
- http://unstrung.sandelman.ca Is my RPL implementation
  - In use in a number of labs

RPL – reactive routing

- Reactive
  - No updates when no traffic.
  - Forms a loop-free Destination oriented Directed Acycling Graph (DODAG)
  - More about Point to Multipoint.
- Also p2prpl
- Pronounced “RIPPLE”
- Comes in Storing and Non-Storing

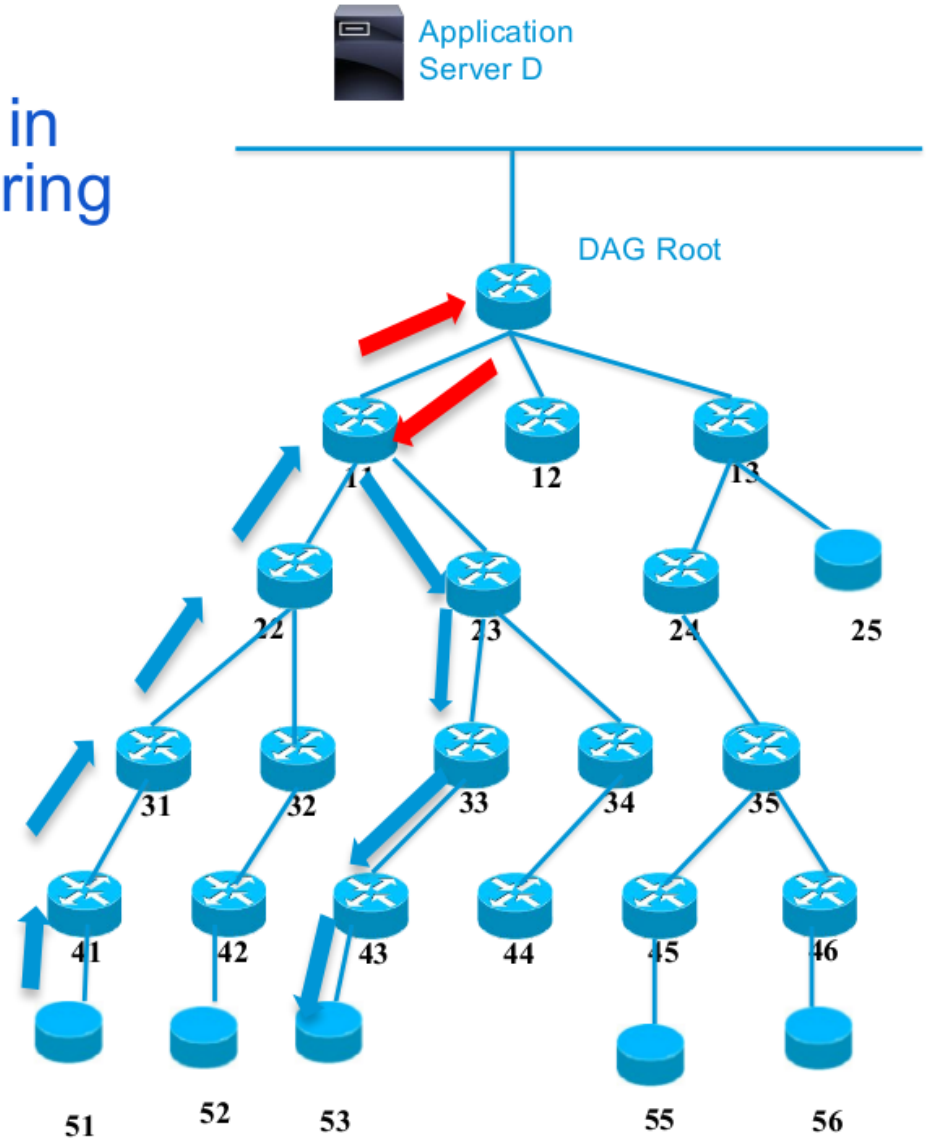
- BABEL, OSPF, RIP
  - Periodic updates
  - Proactively finds failures and repairs them
  - Can be made equal-cost multipath
  - BABEL used in HOMENET
  - OSPF common as IGP

Stretch in storing mode



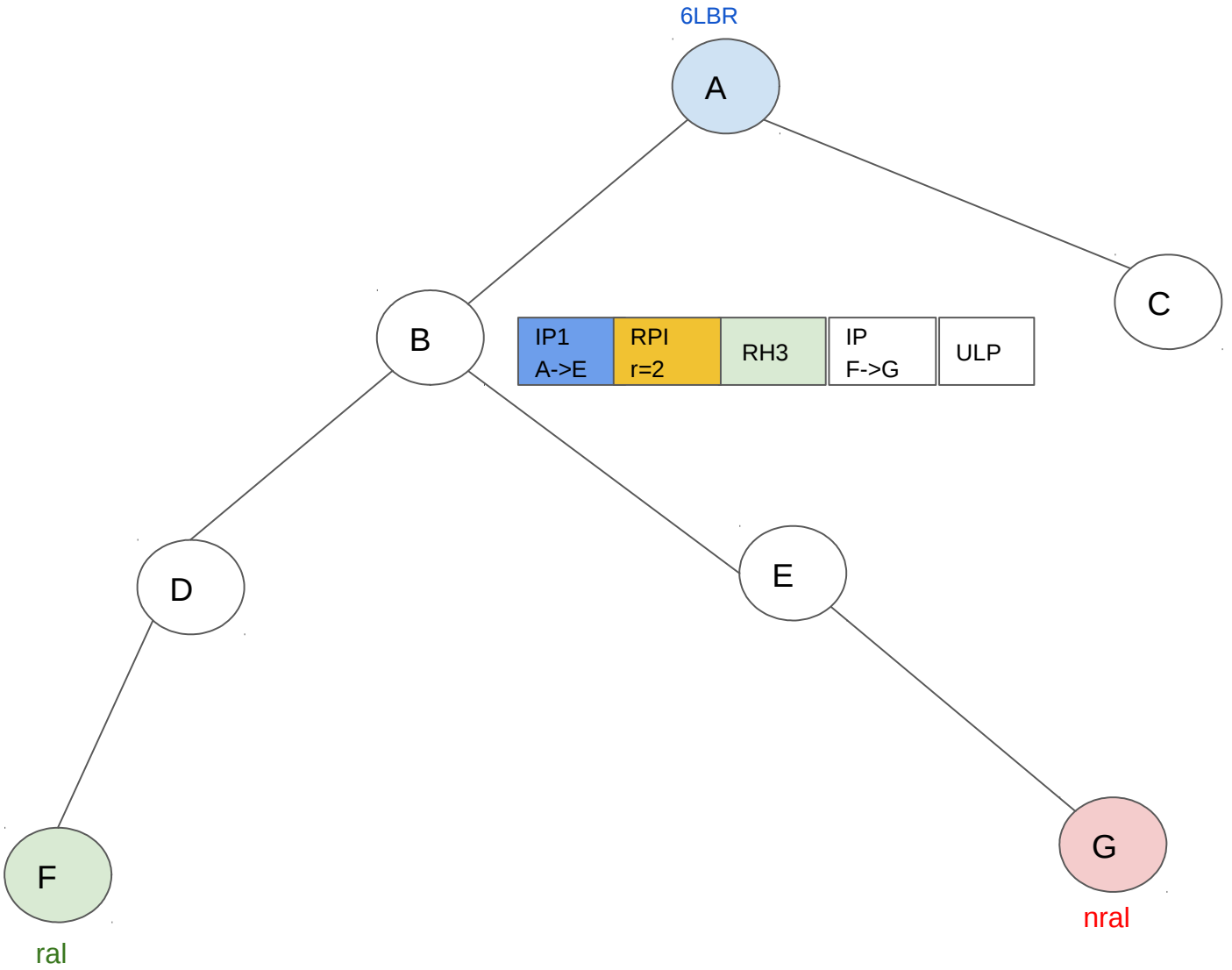
RPL in non-storing mode

Stretch in non-storing mode

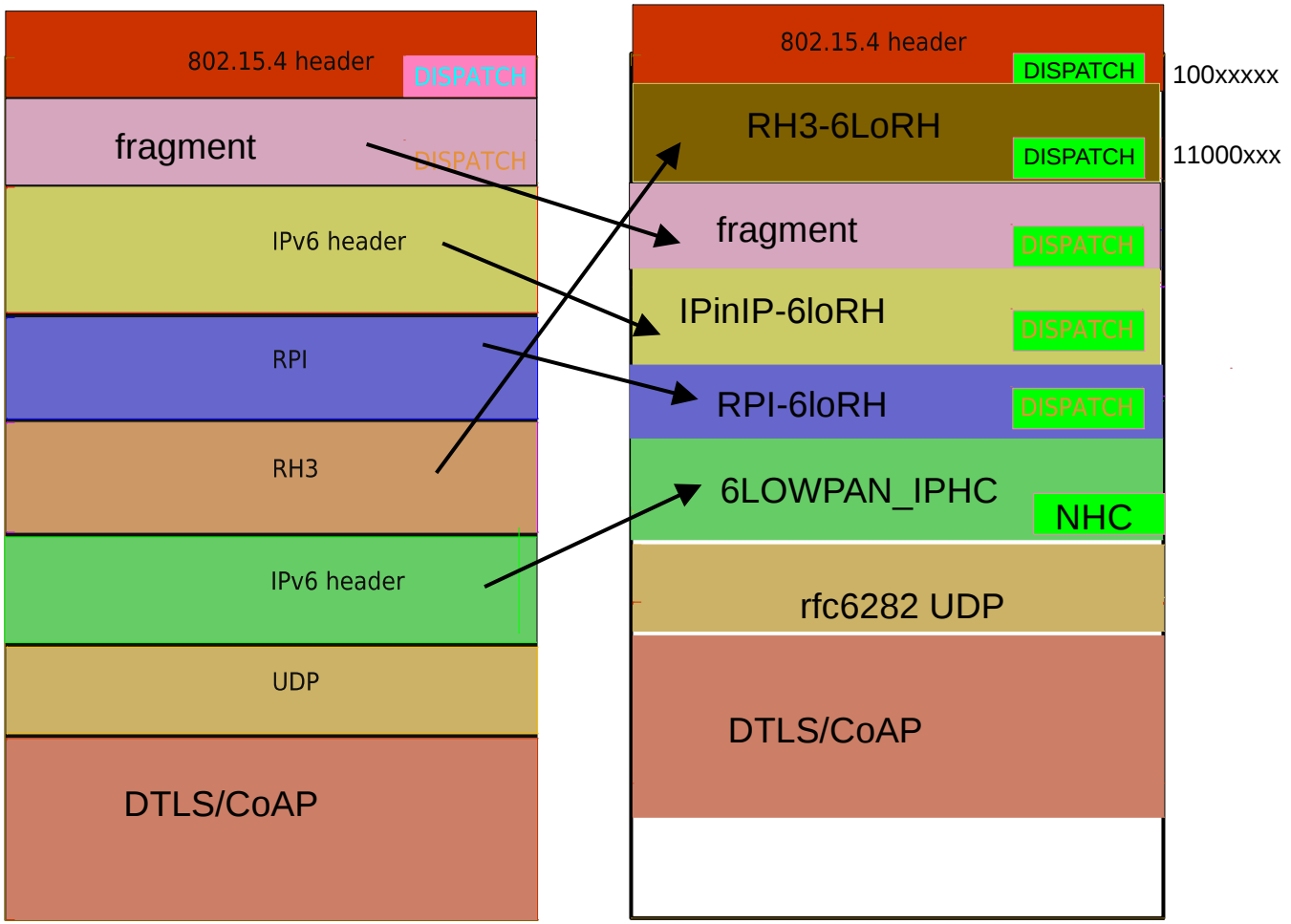


Non-Storing mode: without the inclusion of 2460bis proposal

Form ral (rpl-aware-leaf) to nral (not-rpl-aware-leaf)



Dispatch: architectural view



BYTE SIZES ARE NOT TO SCALE

What we want, what we really really want

Source routes:

```
ip route add 2001:db8::0012 via
fe80::10,fe80::11,fe80::12 table
1234
```

Per-neighbor keying and tx-power control:

```
ip neigh add 2001:db8::0012 lladdr-
short 12:34 key 0x2345678 tx-power
19
```

Access to rplInstanceID (aka VRFs), TX and RX power from userspace

- Recvmsg() -- need to see RX power from wireless interface.
  - So it has to be passed up through skb from driver.
- Recvmsg() -- needs to tell us RPLInstanceID.
  - Yet to be written RPI HbH processing has to map RPLInstanceID to route table ID.
    - Or should rpl instance ID be considered akin to VLAN tag? 802.15.4 does not support 802.1q... but 802.15.10 adds ethernet types. Besides, RPL can run over ethernet already.
  - Maybe network namespace is a better mapping?
- Sendmsg() -- ability to set TX power to use. Essential for probing how far away a node is, and conserving power.
- Sendmsg() -- ability to set RPLInstanceID.
  -
- More comments in linux-wpan list:
  - http://www.sandelman.ca/mcr/unstrung/linux-rpl-needs.txt

Existing and Future Work

- Original 6lowpan work, as updated recently by Alex Aring.
  - New IETF work, 6loRH needs to me implemented.
  - https://codestand.ietf.org Is new site to help connect implementers and mentors.
- Need to write RPI HbH option processor. This needs to be in the forwarding path. Could be a netfilter module.
- Need to write RH3 processor, this needs to avoid/override the routing step; not clear how to best do this yet.
- Do all of the above, while keeping all the 6lo packets in compressed form. Can it be all be done by extracing required info akin to hardware offloads?
  - While data rates are not high, energy is unavailable; sleep states are better.
-

Low-Power Wide-Area Network

- Star topology with router connected to backbone network
- Couple of kilometers cell radius (3-10 km)
- Link layers: LoRa, SigFox, ...
- Very restricted bandwith, typically 10s of bytes as MTU and only a few hundred bytes per day
- Different IPv6 adaptation and compression schemes needed
- The direct device to router links of the star topology allow for a bigger shared context which is used by Static Context Header Compression (SCHC) to reduce IPv6 and UDP headers down to a context identifier

UAPI

- Configuration interface for various header compression modules (on/off, per node)
- Expose information for route-over and mesh-under protocols (LQI, RX in recvmsg and tx in sendmsg, source routes)

Misc

- Proposed RPL implementation within the kernel from João Pedro Taveira
- Many more 6LoWPAN based link-layer adaptations work in progress (IETF 6lo group)
- IPv6-over-foo: NFC, DECT ULE, Z-wave BacNet, MS/TP, ...

Missing input

- Figuring out what various userspace customers would need
- LWM2M
- IoTivity
- OpenThread
- ZigBee IP