IPv6 Subnet auto-configuration

for sensor network trees

Benjamin Valentin

ML!PA Consulting GmbH

September 10, 2021

Section 1

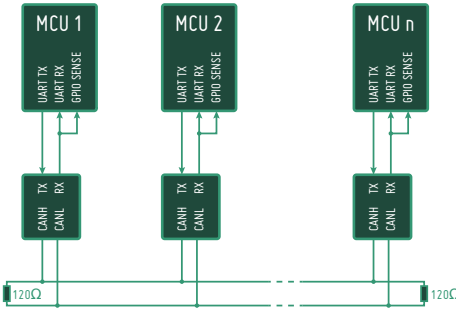
Introduction

About me

- Embedded Engineer at ML!PA Consulting
- Projects for Industrial customers
  - Condition Monitoring, Failure Prediction & Detection, Wear Analysis
- Full Stack Solutions:
  - developing sensor nodes for data collection on installations or vehicles
  - *data analysis in cloud or on edge device*
  - *business data analysis*
  - *frontend development (App or Web based)*

Wired Sensor Networks

- works well for fixed installations
- homogeneous network allows custom protocols
- ‘software Ethernet’ via UART / CAN bus (DOSE)
  - works with any MCU, only needs CAN transceiver (PHY)
  - only a single pair of differential data wires
- no need for dynamic routing algorithm - topology is fixed
- branches in the network may still require routing
- everything should be configured automatically



IPv4

- typically deployed with private addresses & NAT
- multiple routers can create multiple levels of NAT
  - computational overhead on the routers
  - works for HTTP, but not much else

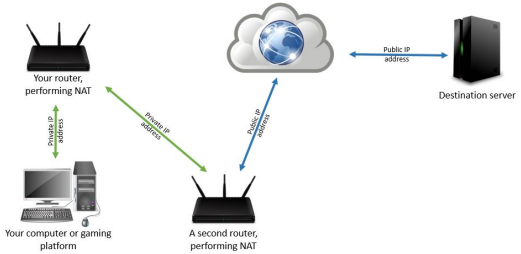


Figure 1: double NAT

IPv4 - bridged

- ‘Solution’: bridge everything on layer 2
  - single broadcast domain

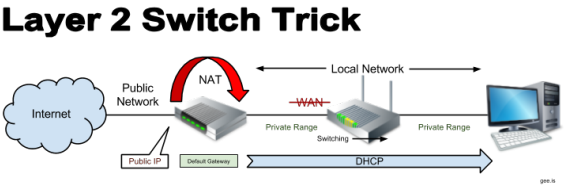
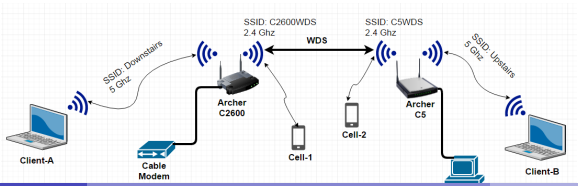
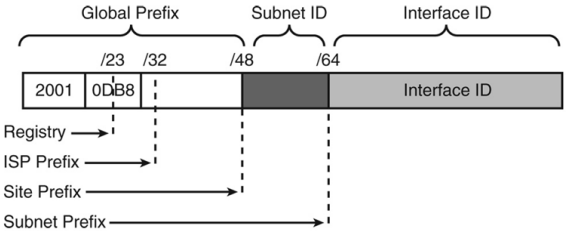


Figure 2: layer 2 routing

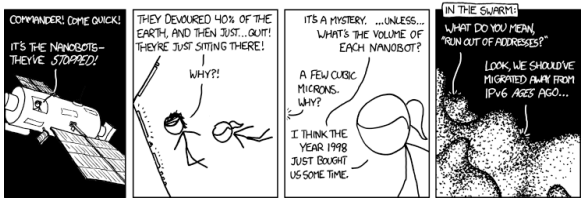
- works for switched networks, but wireless / bus networks require 4-Address mode (source - router - router - destination)



IPv6



- IPv6 has large address space (128 bit) to ease subnet creation and auto-configuration
  - SLAAC (*Stateless Address Autoconfiguration*): Host receives prefix via Router Advertisement, picks last 64 bit of address on it's own
- Provider hands out /56
- /64 needed for SLAAC
  - 2^8 subnets possible on a typical connection
- *Is there something like Stateless Subnet Autoconfiguration?*



ICMPv6

- more than just `ping`
- replaces ARP

Router Solicitation

- Is there any router?

Router Advertisement

- I'm a router!
- My default route is valid for n seconds
- Prefix Information: You can choose an address from `2001:db8/64`
- Route Information: You can reach `2001:db8:c::/62` through me

Neighbor Solicitation

- Who has `2001:db8::8cca:efff:fe8b:d64e`?

Neighbor Advertisement

- I am `2001:db8::8cca:efff:fe8b:d64e`
  - receiver stores l2 source address in Neighbor Information Base

Section 2

Subnet Auto-Configuration

Recapitulation

- SLAAC only uses lower 64 bit of address
- If we have a large prefix (> `/64`) we can sub-divide it
- Hierarchical subnets make for efficient routing
- Subnet-Prefix must not be used by another Subnet

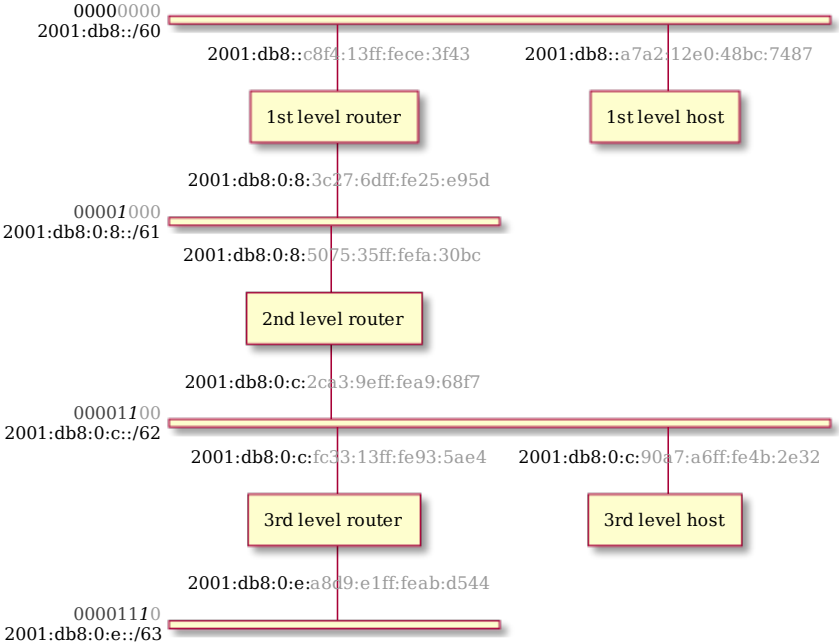
Where to get a large prefix?

- request large prefix via DHCPv6 IA_PD
- configure `radvd` to advertise large prefix

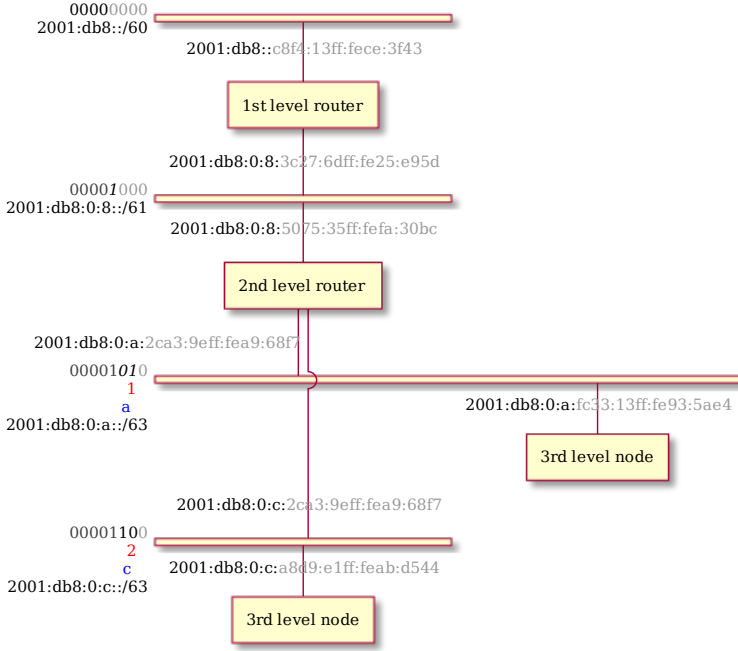
‘Skinny Tree’ / Linear Topology

- Only a single router on each level (link)
- Routers can have multiple downstream interfaces
- any amount of leaf nodes
- no coordination between routers needed outside standard ICMPv6 messages
- When prefix is received create subnets for each downstream interface
  - consume as many bits as required to map number of subnets
  - count up starting with 1, shift to end of prefix
- `gnrc_ipv6_auto_subnets_simple`

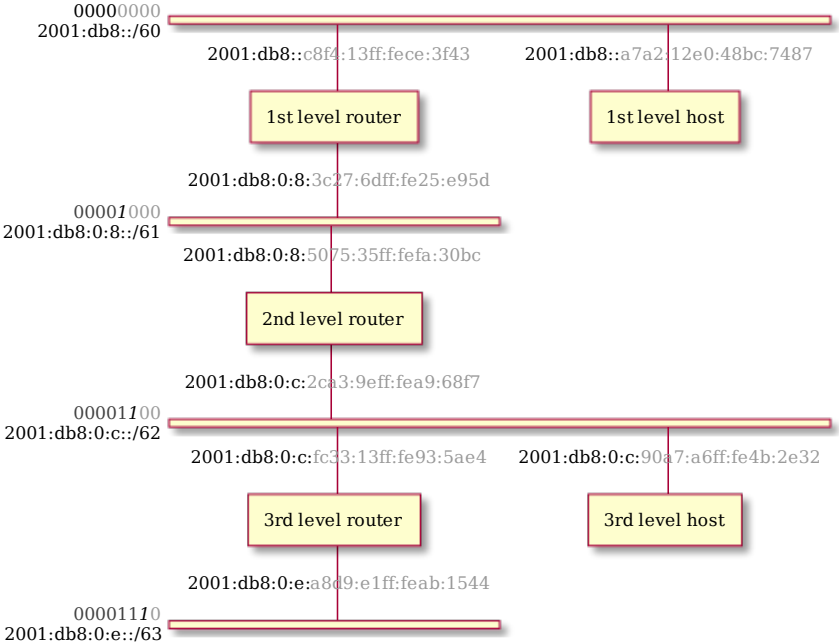
Linear Network



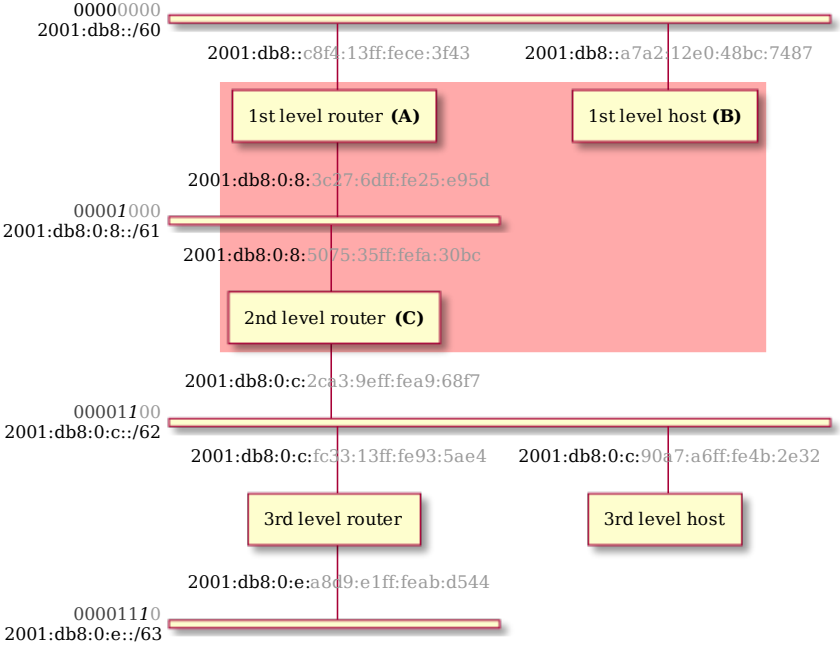
Multiple Downstream Interfaces



Linear Network



Linear Network



ICMPv6

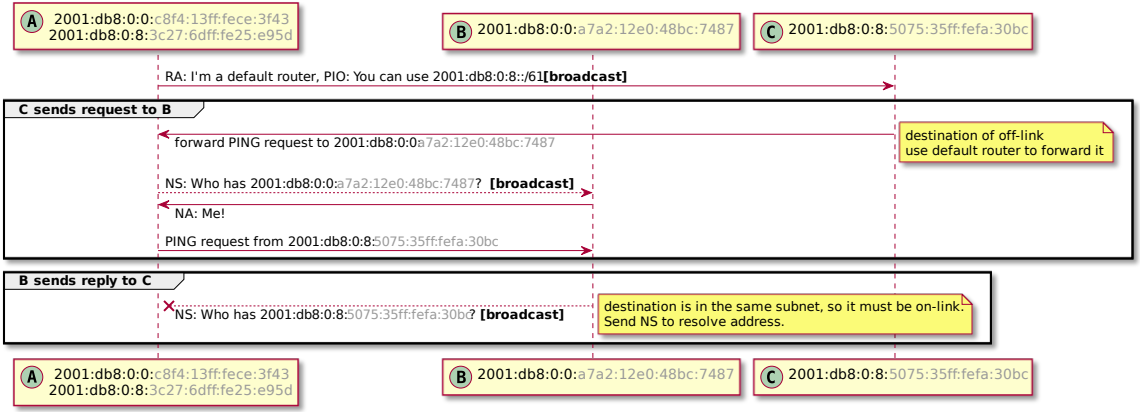


Figure 8: missing downstream route

ICMPv6

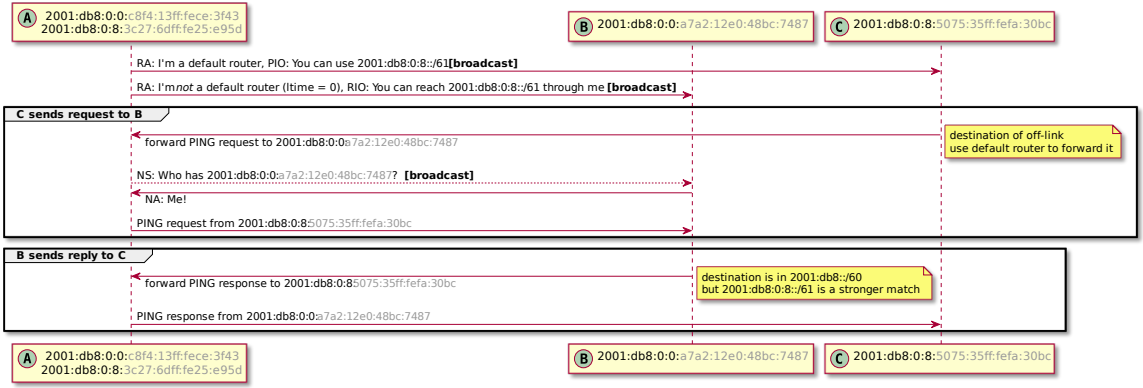
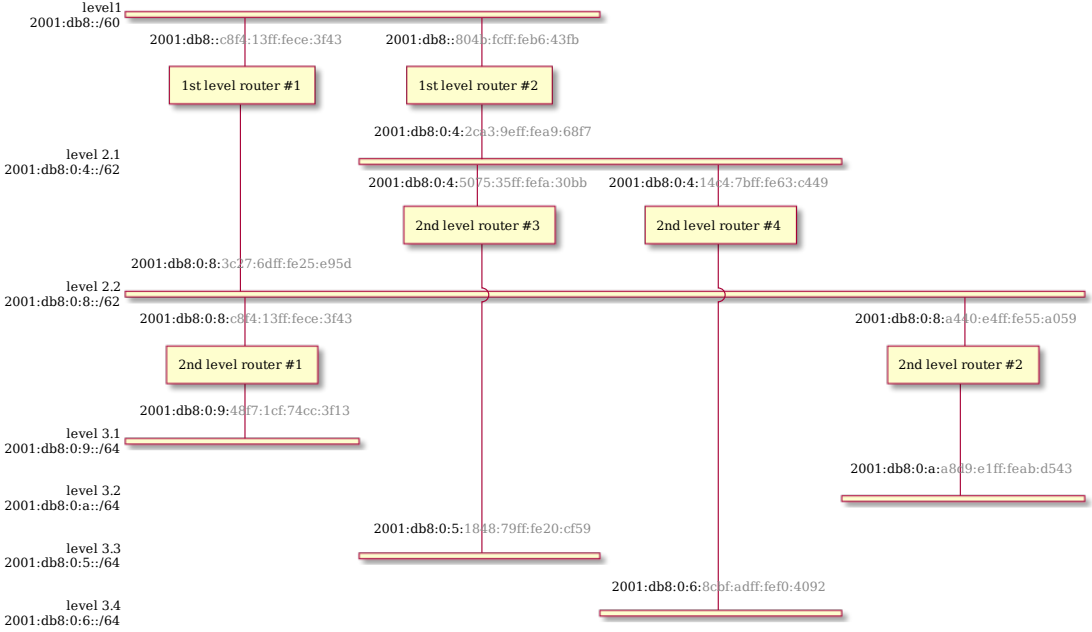


Figure 9: Route Information Option advertises downstream route

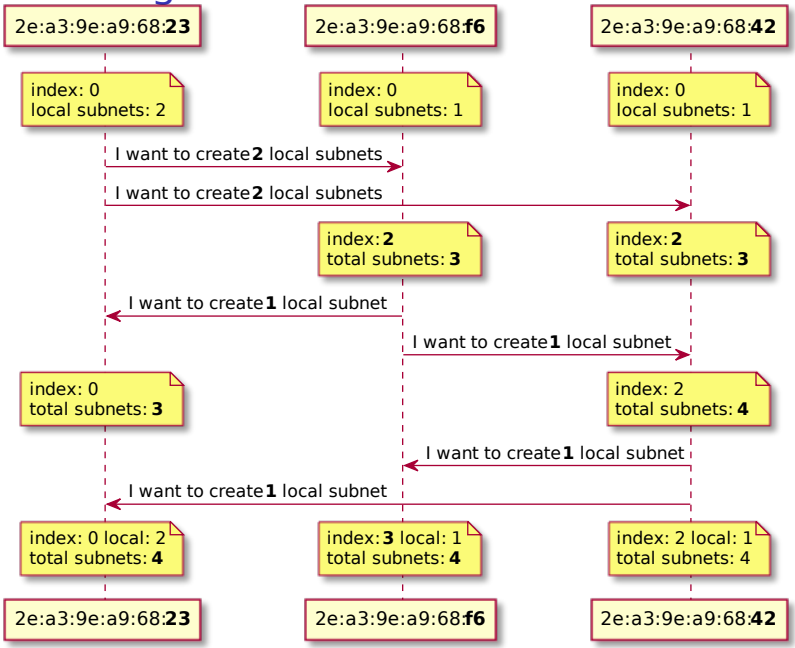
General Tree Topology

- Multiple routers on each level (link)
- Routers can have multiple downstream interfaces
- any amount of leaf nodes
- Routers have to coordinate so the same subnet does not get used twice
  - no standard solution :(
  - custom UDP protocol, but very simple
  - routers announce the number of subnets they want to create, l2 address order used to determine prefix ID
  - remove invalid prefixes if new router joins
    (send ICMPv6 RA:PIO with lifetime 0)
- `gnrc_ipv6_auto_subnets`

General Tree Topology



Synchronisation Algorithm



Questions?

Section 3

Demo