Dithus: Device discovery

Design/status

v.

Rune Torbensen

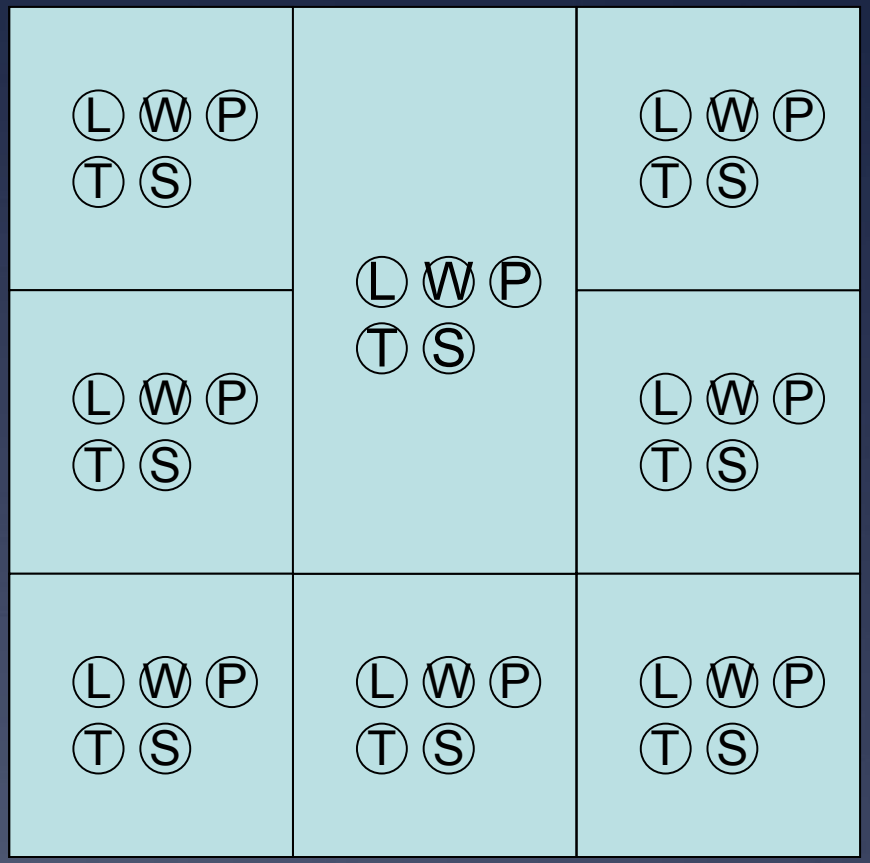
Contents

* Introduction
* Device discovery design
* Experiments: new and existing equipment

Introduction: Case study

Realistic home automation

* Consumer electronics
  * Cost sensitive
  * Reuse of infrastructure
  * No additional functionality
  * No app-logical management for end-devices
* House Case study
  * 8 rooms with each 5 devices.
  * 4 different wireless networks
* Competitive
  * Functional equal end-devices are interchangeable
* Generic communication infrastructure

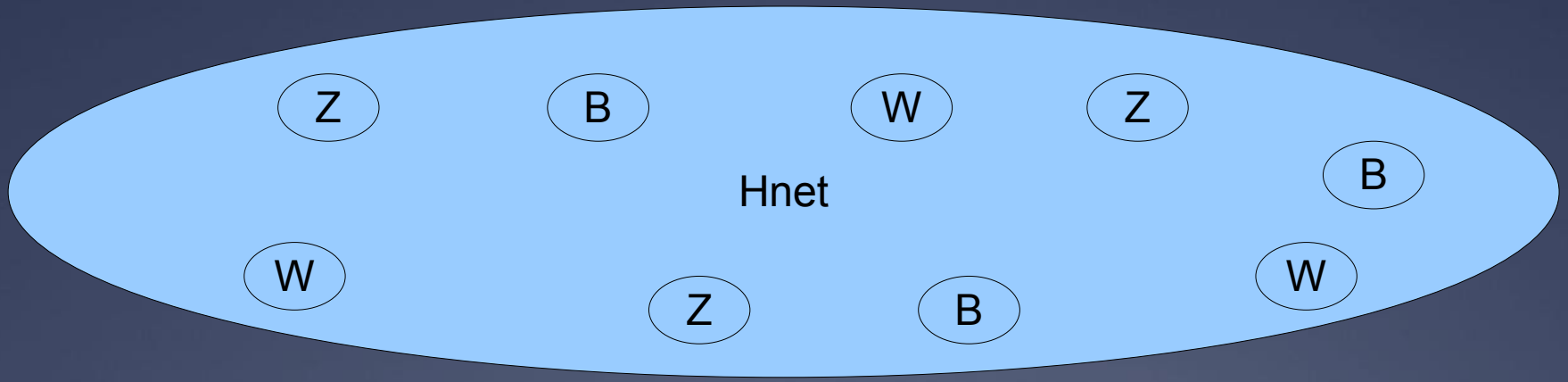


L=Lamp, W=heater, P=PIR,

S=lux-sensor, T=temp-sensor

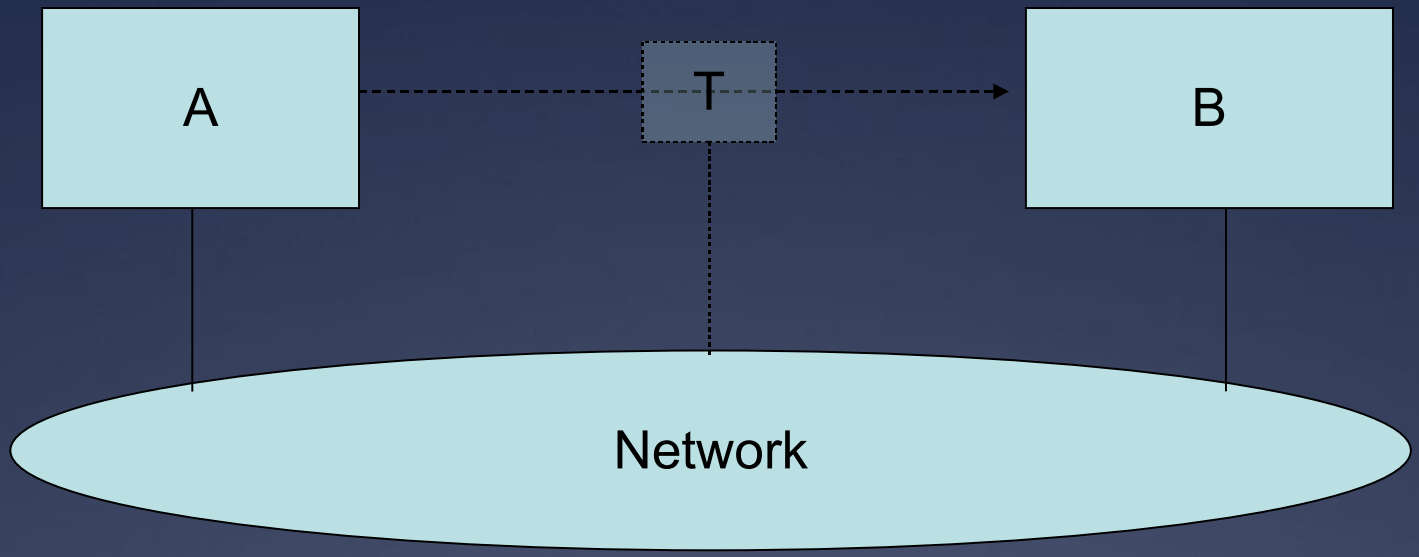
One home network

* Different devices
* Different vendors and protocols
* New device and existing devices
* In one big network



Problem domain

* Two arbitrary unfamiliar applications: A using B

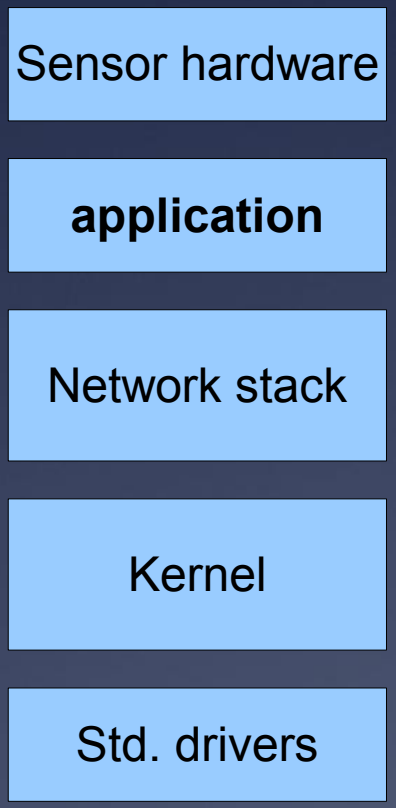


A: Controller application
B: Sensor application

Application models

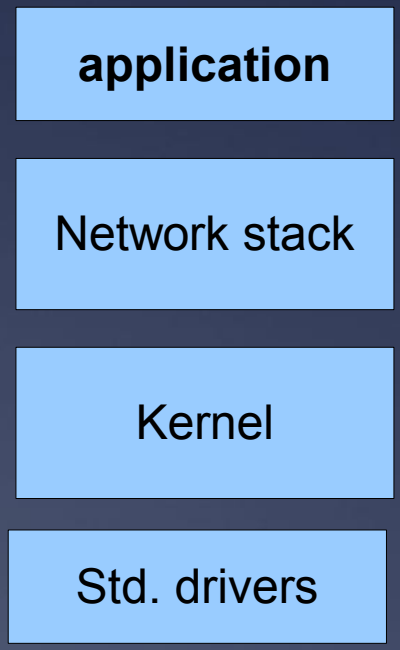
Service provider

Sensor:



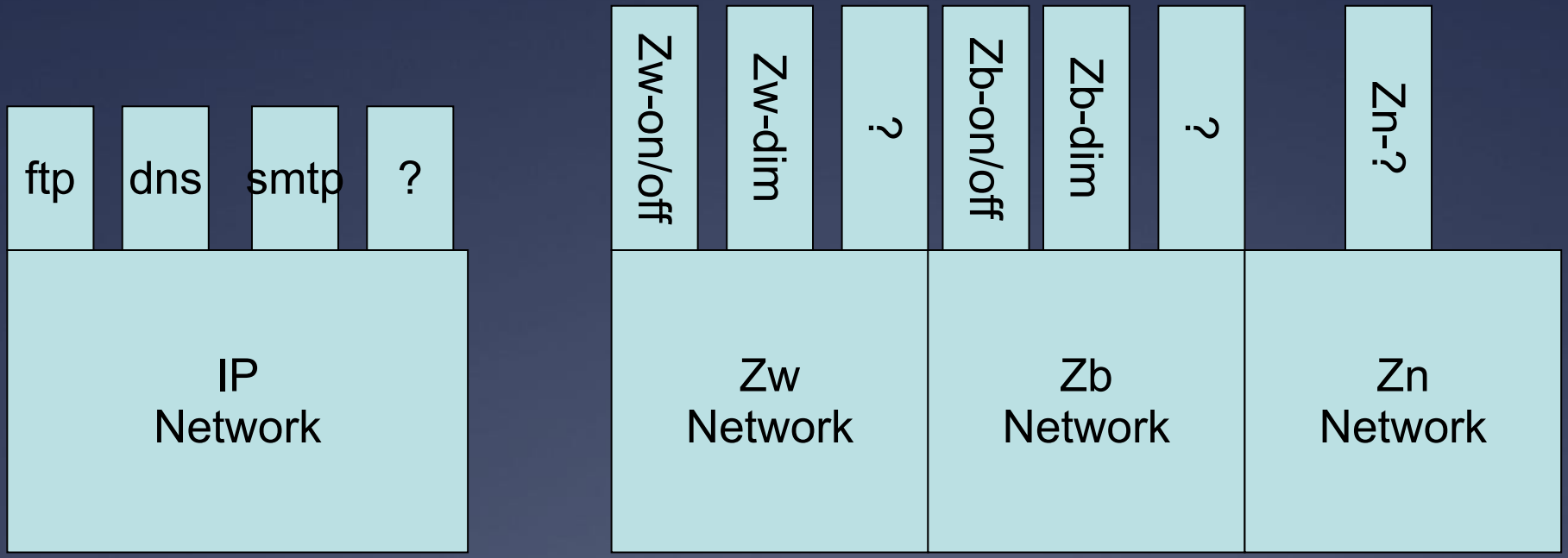
Service consumer

Controller:



Home Automation protocols

Application protocols:
Internet protocols and end-device application protocols



Device discovery design

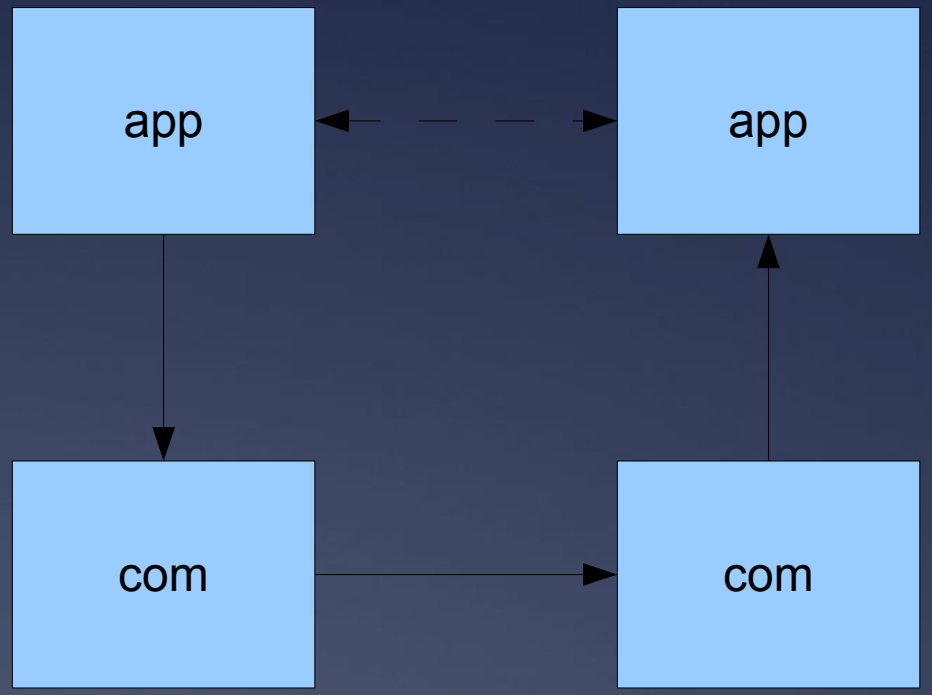
Device discovery

* Ability to enroll new devices into a running system
* Minimal user involvement
* Includes previously unknown device types

Device discovery: Basic idea

Separating problems:

* Application translation
* Communication bridging



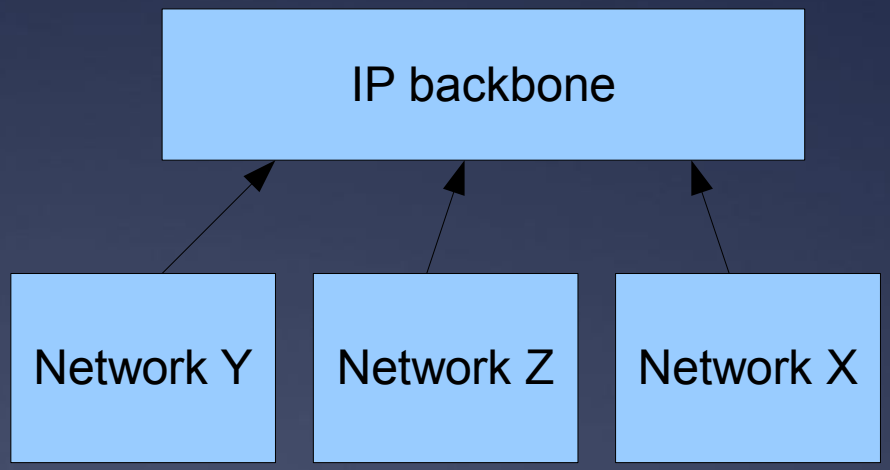
Heterogeneous network

One network:

* IP based backbone
* All nodes reachable

Sub networks

* Y,Z,X etc.



Sub networks

Required Methods

* Include devices in to the network
* Exclude devices from the network
* List of visible/registered devices on the network
* Send/receive data to/from a device on the network.
* Device descriptor from a device.

Device discovery stack

Rich description language

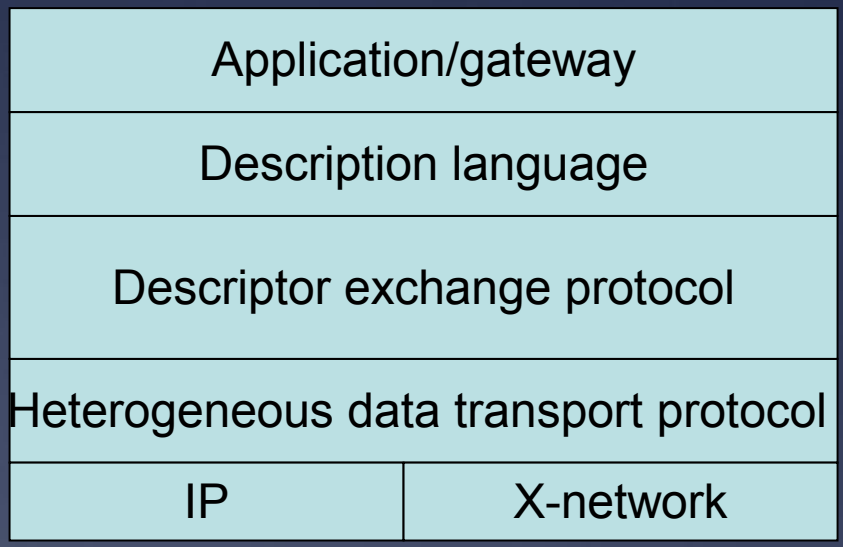
* Range of application protocols
* Support new concepts (SI units)

Descriptor exchange protocol

* any file based rich description language
* Simple description – i.e. type id.

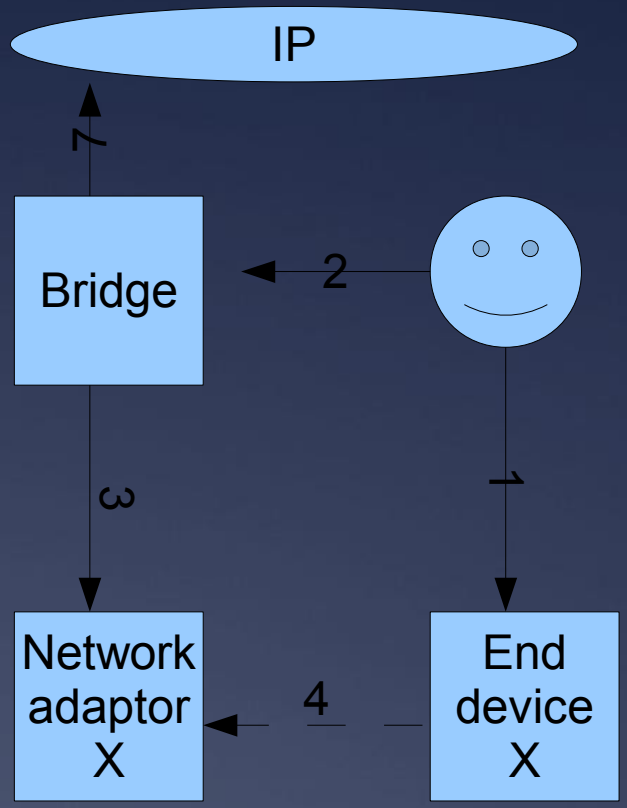
Heterogeneous net (Hnet) data transport

* No data semantics assumptions
* Hnet address (ip:port)
* Abstracts wireless network technology
* Methods for H-net address discovery.



End-device enrollment sequence

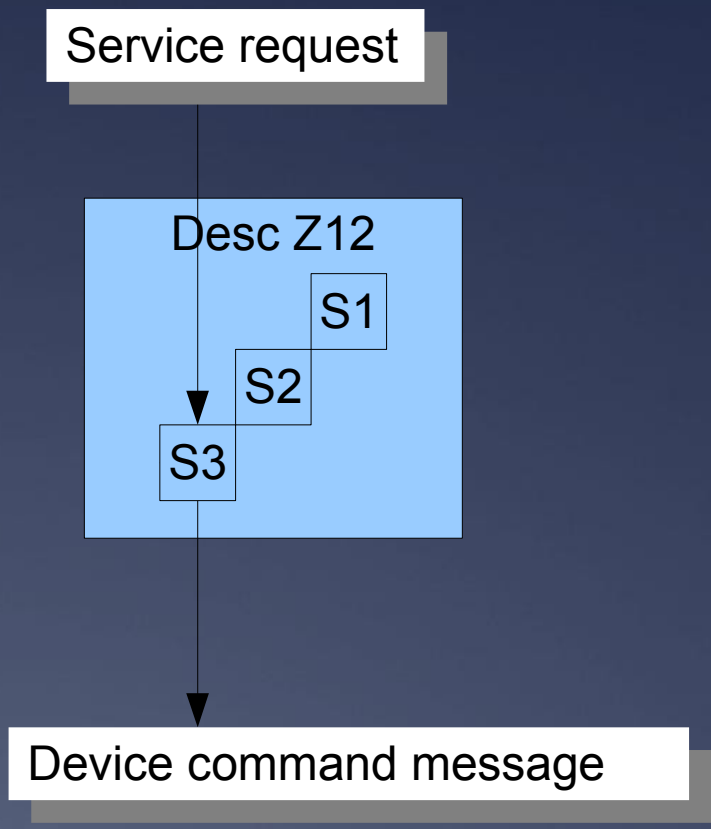
1. User: join button on end-device
2. User: include button on bridge
3. Bridge sends include command to network adaptor.
4. End-device joins X-network adaptor (x methods)
5. Bridge sends Get_Descriptor to new device.
6. Bridge receives and buffers new descriptor.
7. Bridge sends UDP broadcast: Services_updated



Description/translation language

Simple devices:

```
<!-- setvalue(1, ”Light Intensity”, 25);
 -->
<service type="set" desc="Light
   Intensity">
      <parameters>
        <parameter type="int" min=""
   max="" step="" scale=""
   unit="Lumen"/>
      </parameters>
      <write>
        <byte>01</byte>
        <byte>04</byte>
        <byte module="abx" arg1="9"
   arg2="17" arg3="{parameter:1}"/>
      </write>
</service>
```

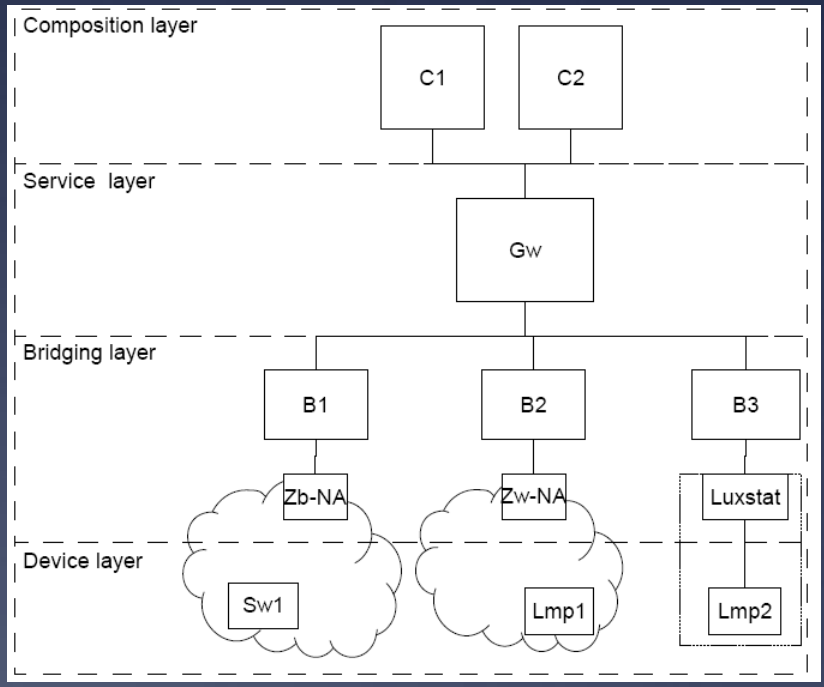


Experiments:
New and existing equipment

System overview

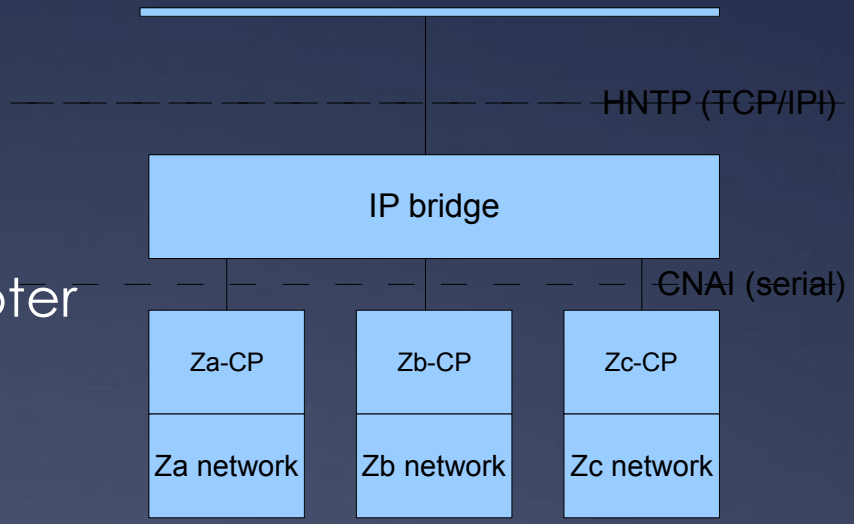
Embedded IP devices on LAN

* Controllers (just software)
* Translator/service gateway
* IP bridges
* Wireless network adapters
* End-devices:
  * Push button
  * Lamps wall plugs
  * Light control subsystem
* Test of end-devices



Bridge model

- HNTP: Heterogeneous network transport protocol
- Two parts:
  - Generic IP bridge part
  - Network adapters:
  - Z n - networks
- CNAI: Common network adapter interface
  - En network manager
  - Virtuelle adresser (1-255)
  - Payload-tunnel to end-device
    - Vport, len, payload..
  - C. network commands (via 0)
    - Include/exclude



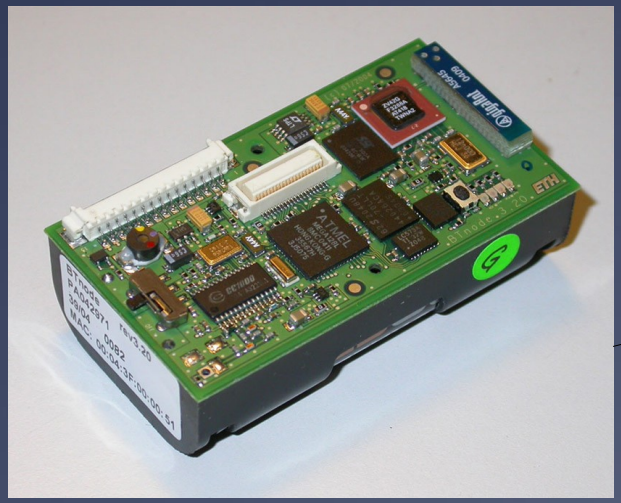
Udviklingsplatform

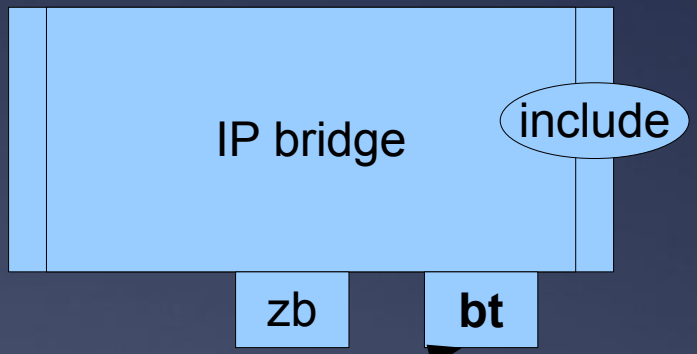
- Linux boks
  - IP stack og usb port
  - Serial port to usb konverter
- God udviklingsplatform:
  - Gcc, Gdb debugger, Eclipse.
  - Mange pakker.
  - NSLU: rigelig hukommelse og hastighed.
  - Stadig "embedded" device.
- Software
  - IP til Seriel port protocol conv.
    - Event UDP broadcast
    - Telnet command

Experiment: New End-devices

Bluetooth network adaptor:

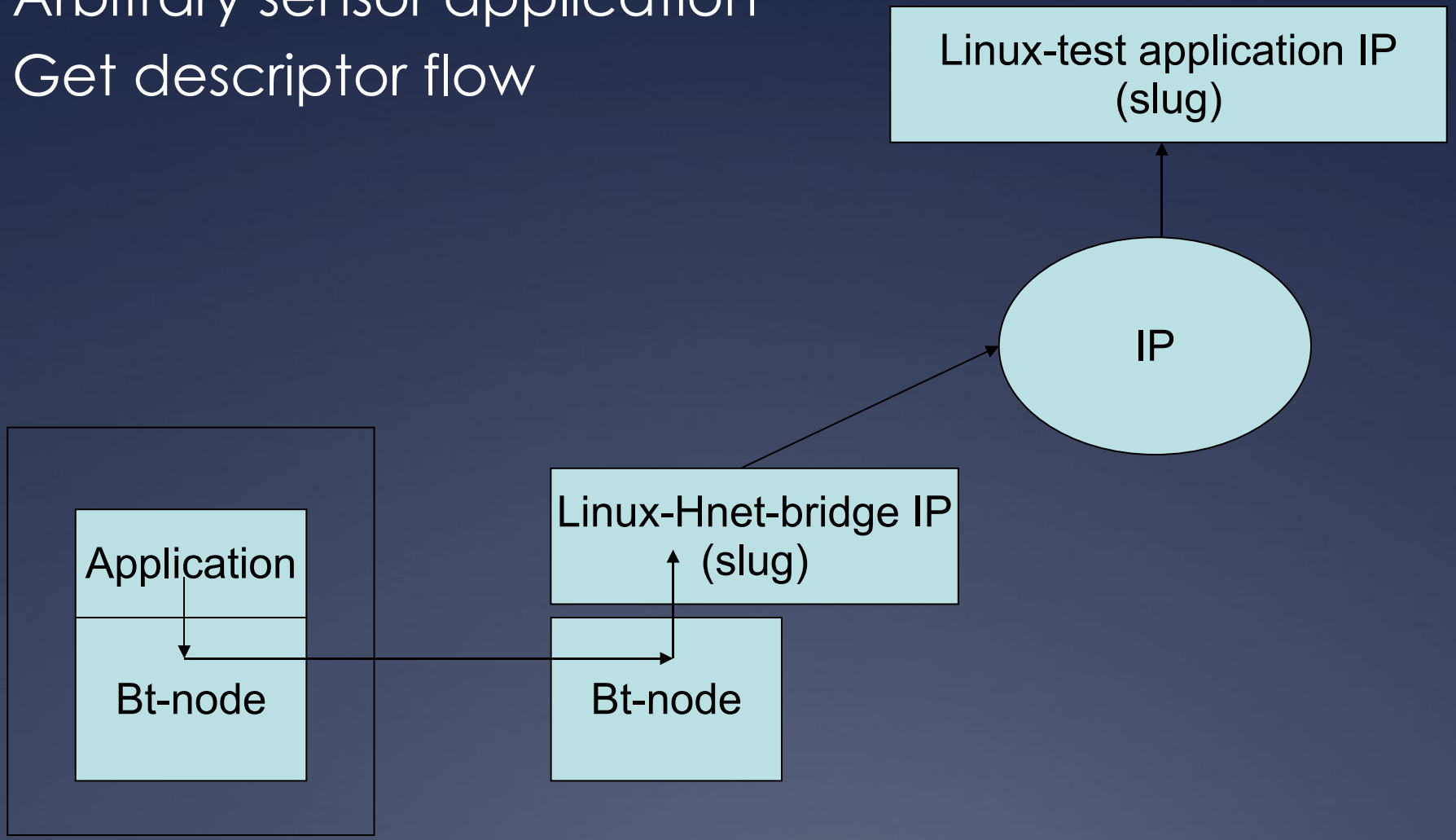
* Send message
* Include method
* Get descriptor





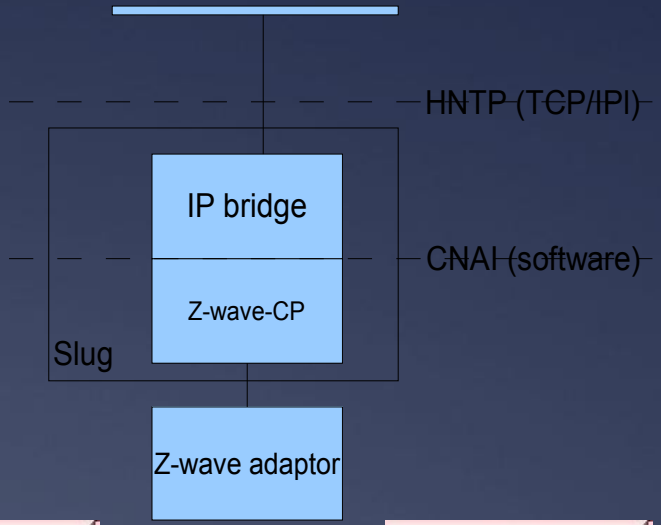
Device discovery on Btnodes

* Arbitrary sensor application
* Get descriptor flow



Experiment: Existing equipment

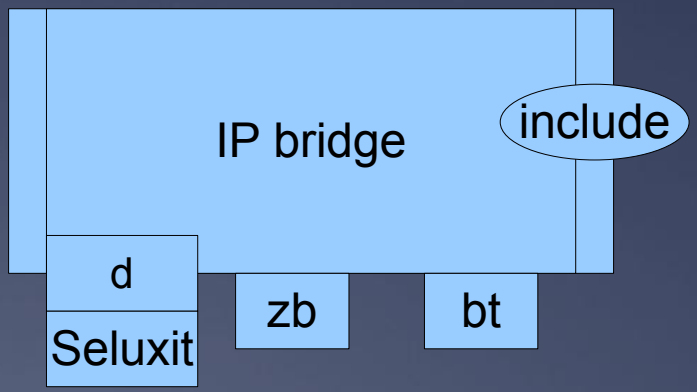
- Seluxit z-wave adapter
- Z-wave common interface sw driver module
- Serial port protocol
- Wall plugs on/off
- PIR sensor event
  - Befolket
  - Ubefolket



Experiment: Existing equipment

Seluxit adaptor driver (d):

* (Send message method)
* Include method -> network equivalent
* Exclude method -> n.e.
* Get network list -> n.e.
* Get_descriptor -> n.e.



The End

* Questions?