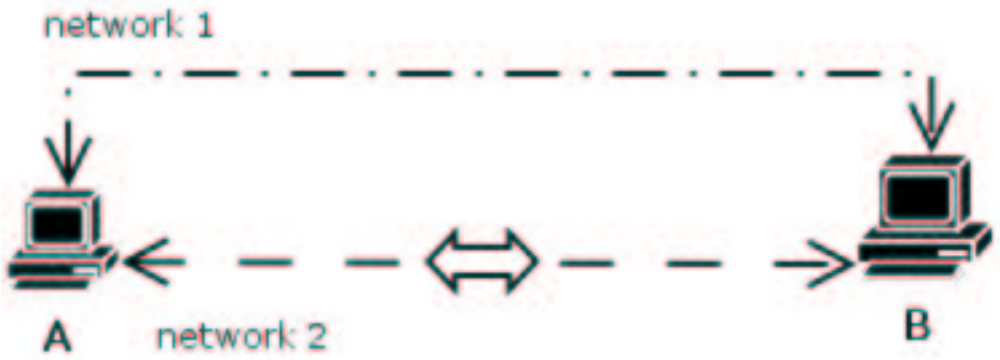
Optimization of a Dynamical Handoff in Mobile Ad Hoc Networks

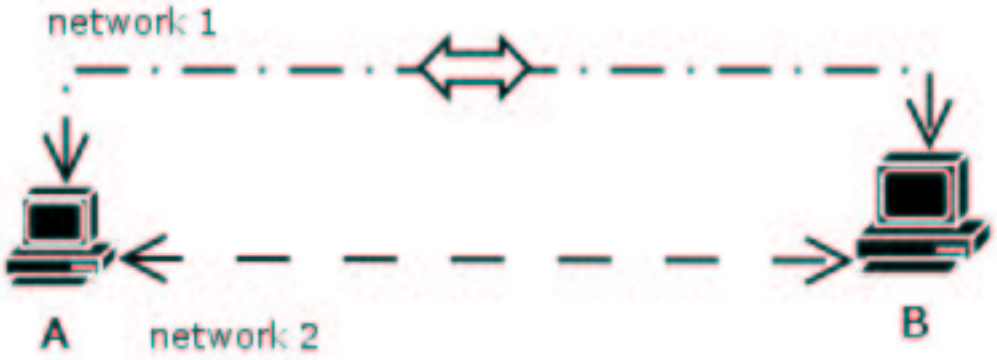
Michael Olbrich

Contents

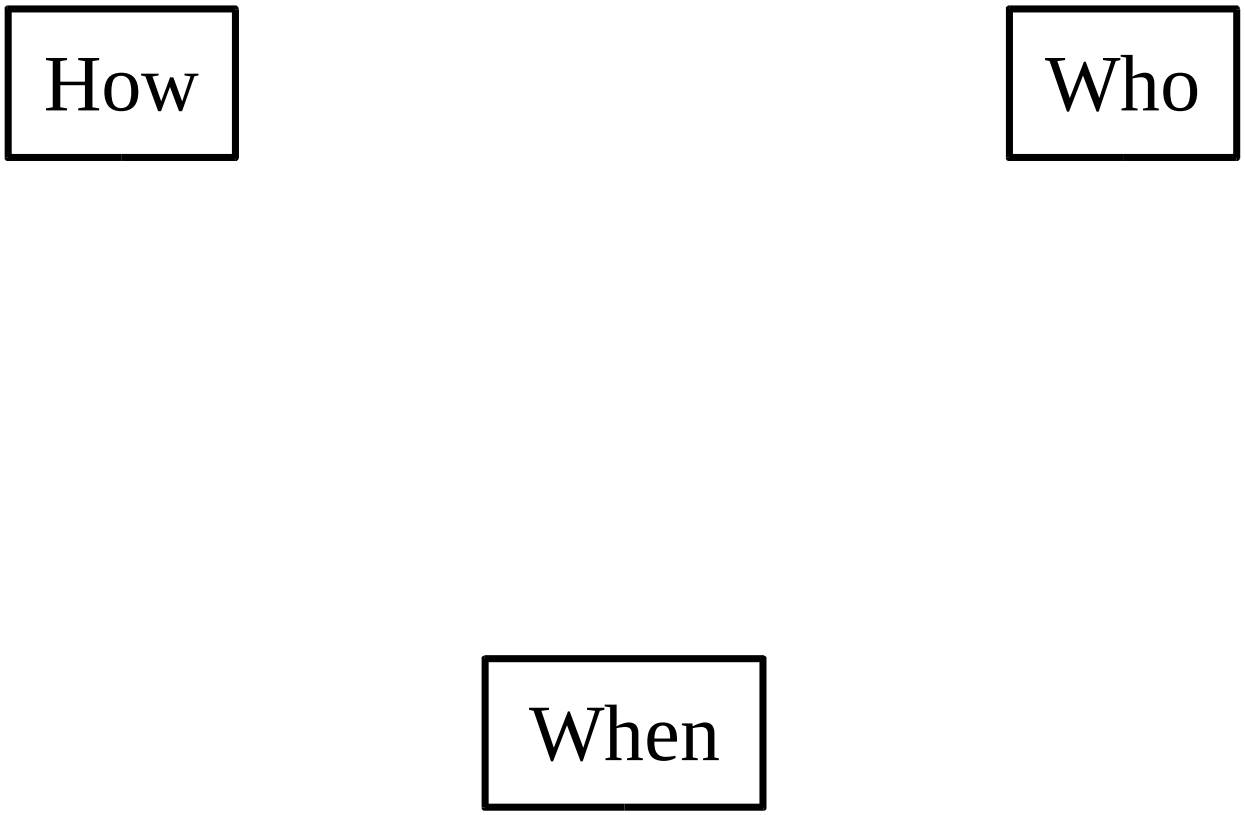
- Introduction
- Dynamic Multiplexer
- Parameter, Strategies & Deciding

Introduction

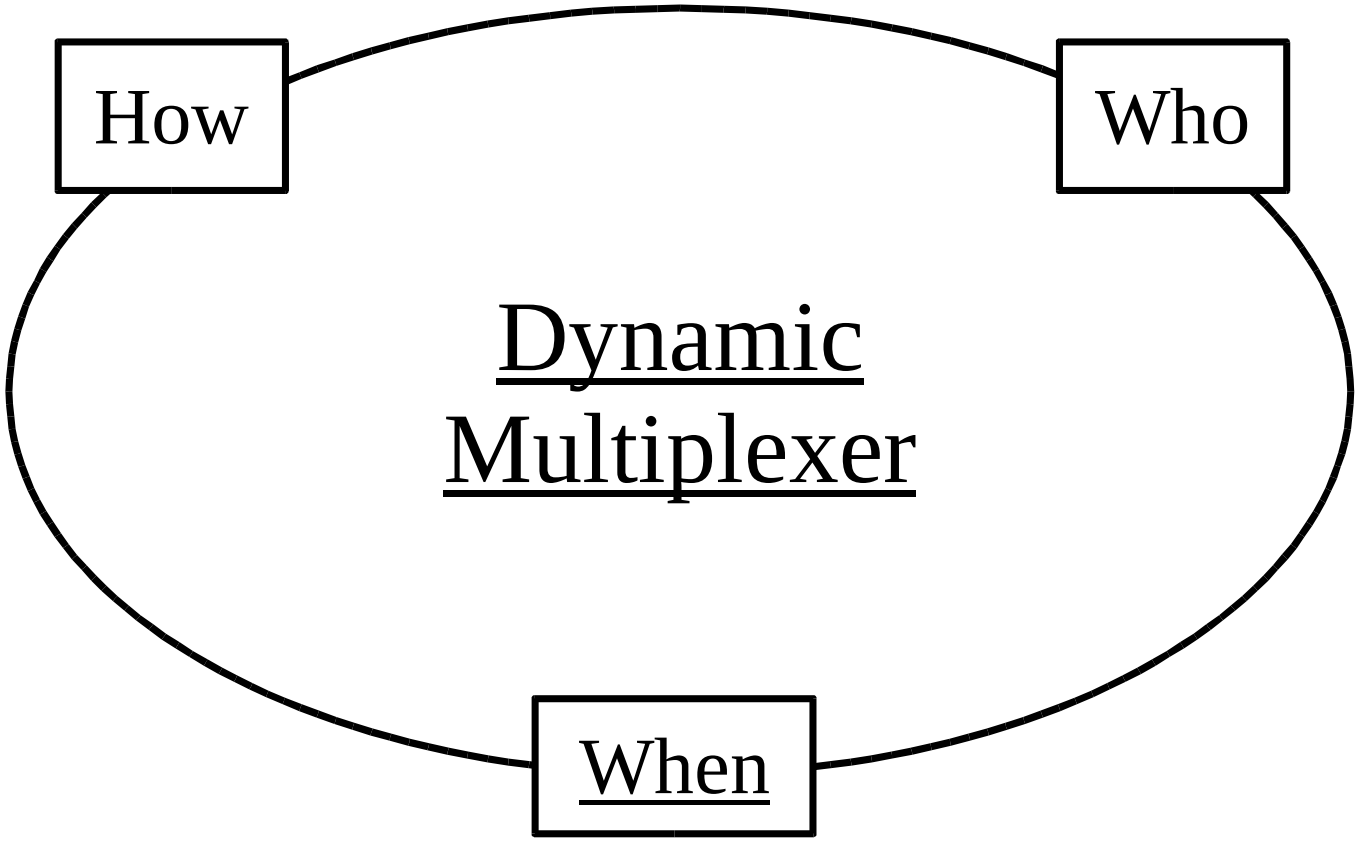




Introduction



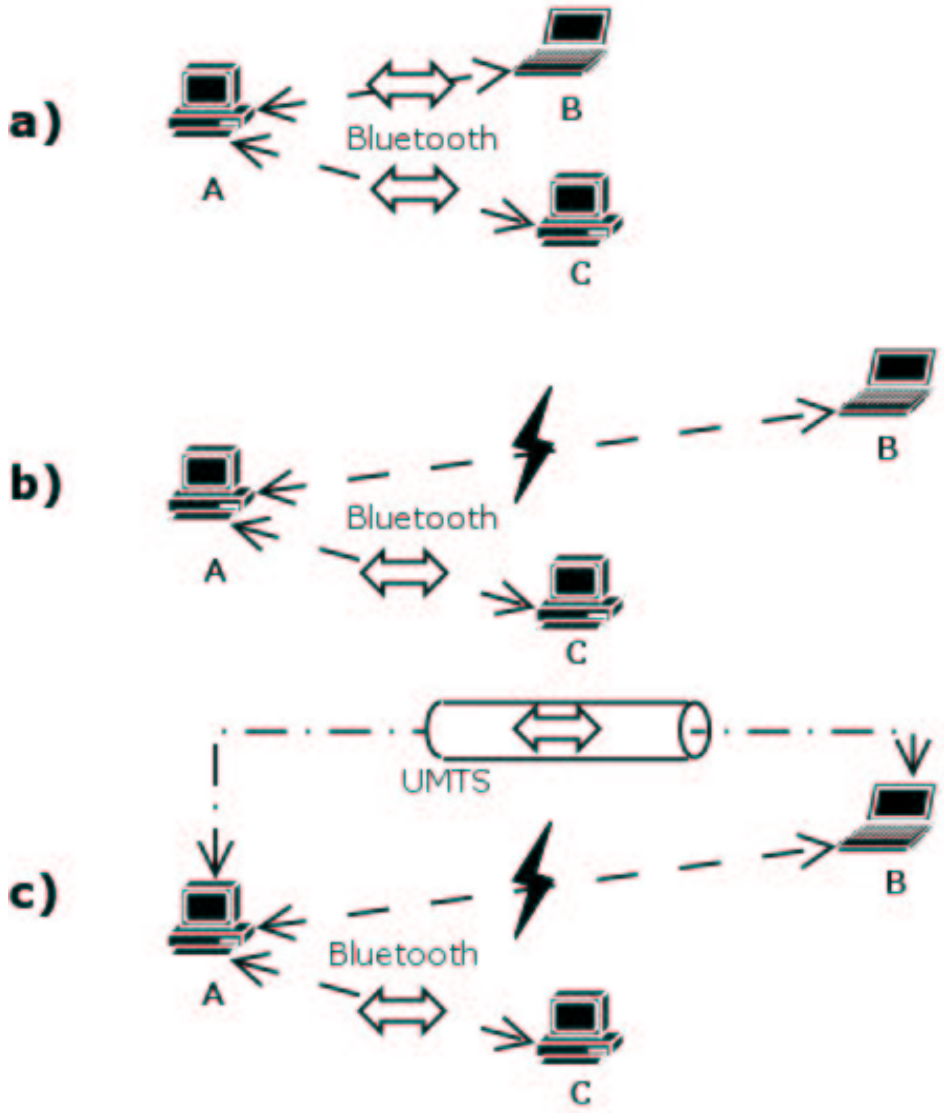
Introduction



Dynamic Multiplexer

- Basic Thoughts
- Architecture

Dynamic Multiplexer - Basics



Dynamic Multiplexer – Basics

Requirements:

- IP based decisions

Dynamic Multiplexer – Basics

Requirements:

- IP based decisions
- Event driven design

Dynamic Multiplexer – Basics

Parameter:

- Static
  - Device specific
  - IP specific
- Dynamic
  - Device specific
  - IP specific

Alternative IP's:

- File
- Incoming tunnels
- SLP

Dynamic Multiplexer – Basics

Requirements:

- IP based decisions
- Event driven design
- Modular design

Dynamic Multiplexer – Basics

How to decide:

- Simple approach: One independent value for each IP address. Highest value “wins”.
- Problem: e.g.
  - Bluetooth good alternative for UMTS
  - Bluetooth NO good alternative for WLAN

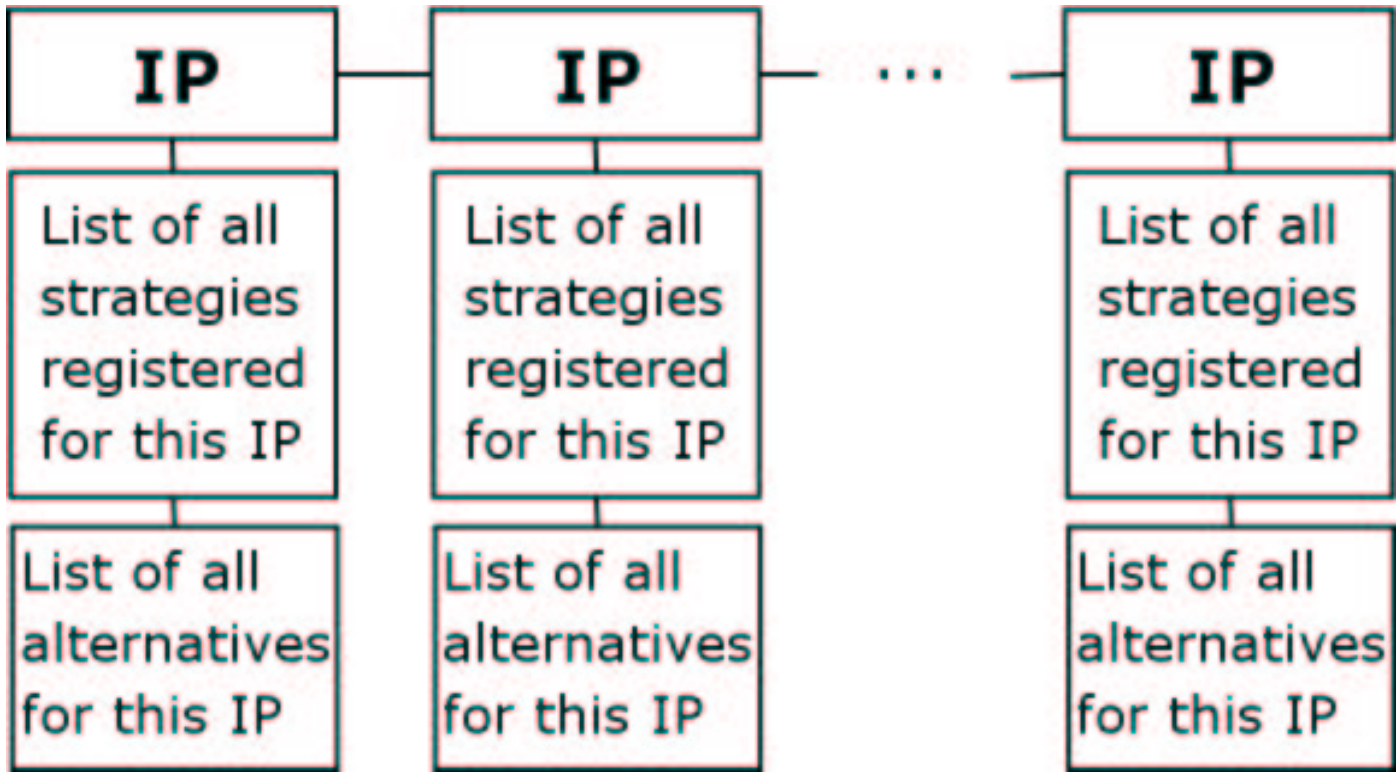
=> One value for each IP address
But: take the IP to be tunneled into account.

Dynamic Multiplexer – Basics

Requirements:

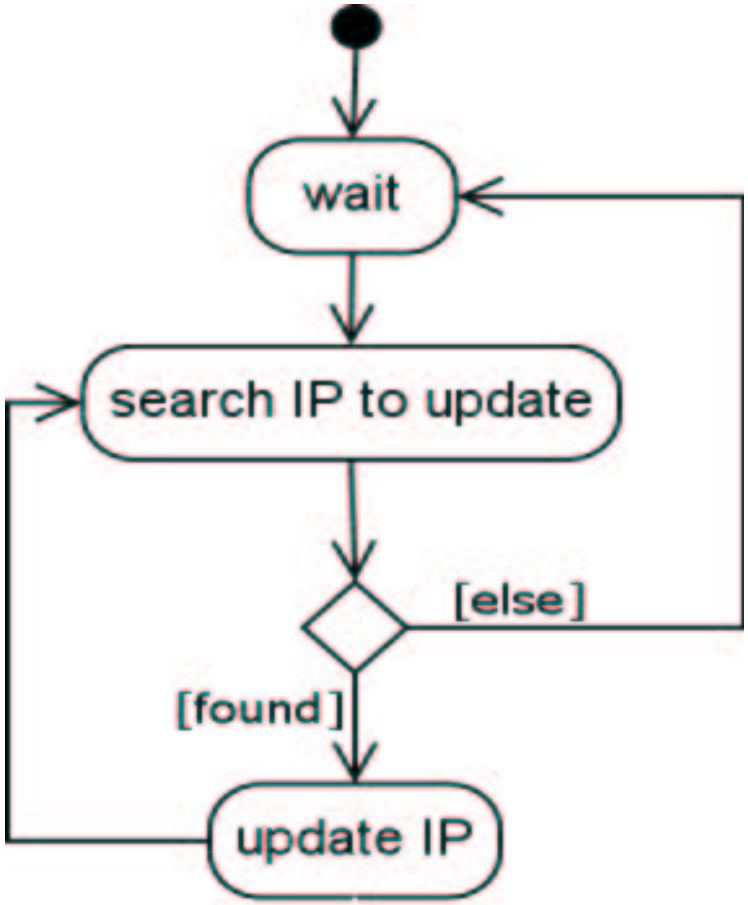
- IP based decisions
- Event driven design
- Modular design
- Deciding by comparing

Dynamic Multiplexer – Architecture



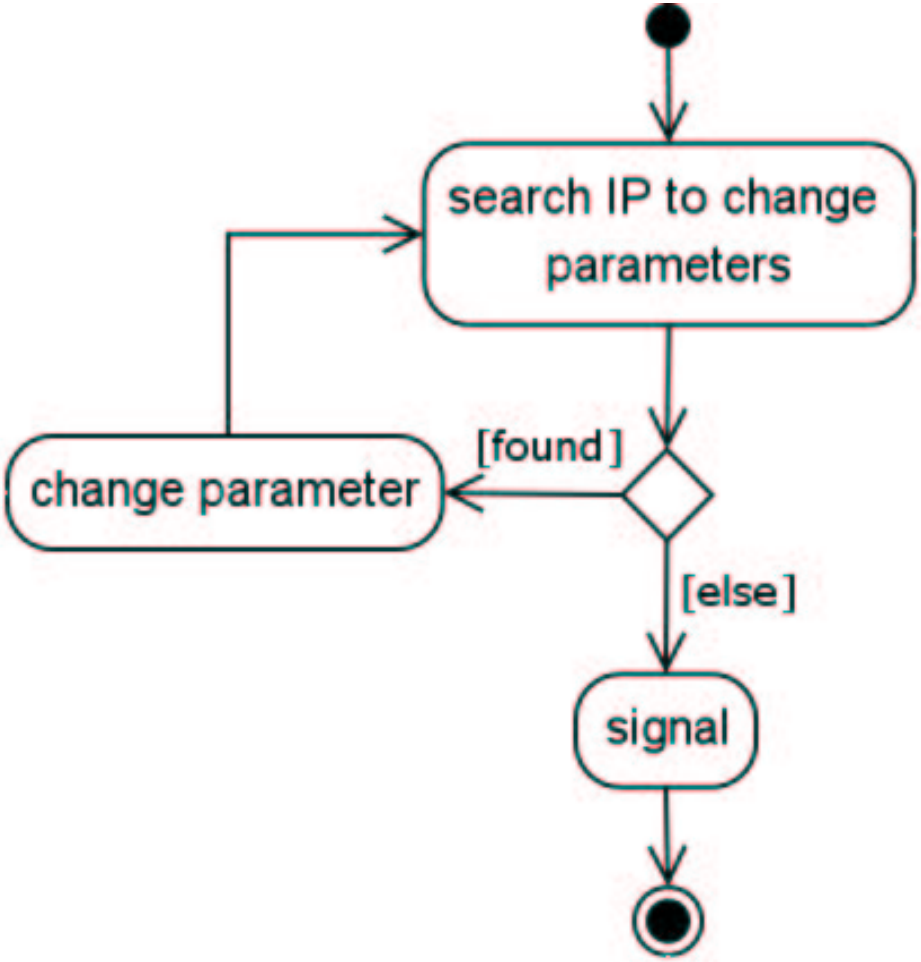
Dynamic Multiplexer – Architecture

Core Loop



Dynamic Multiplexer – Architecture

Strategy Loop



Dynamic Multiplexer – Architecture

Parameter Trust Level

Several strategies may provide the same parameter
=> one value has to be chosen.

Each strategy provides a trust level for each parameter to show how exact the value is.

Example: maximum data rate

- Static value, maximum data rate of the device:
  maxDataRate = 11 Mbit/s trustLevel = 1

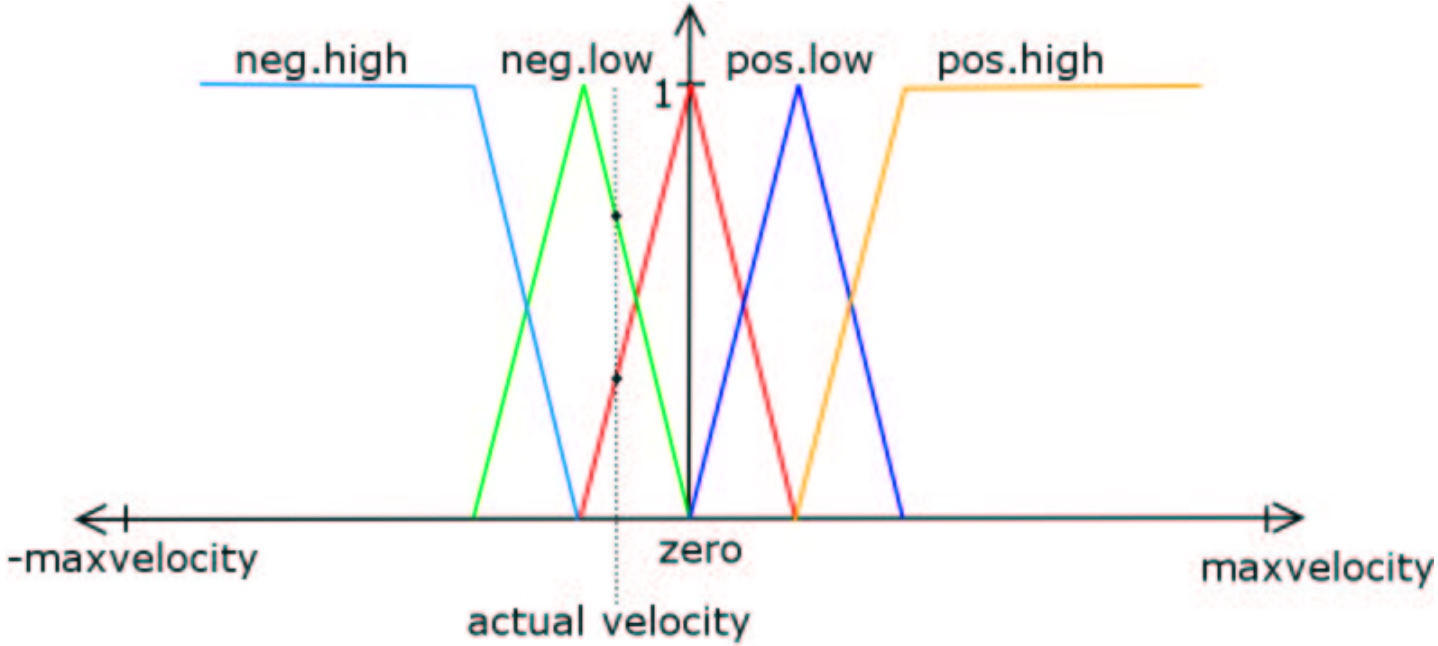
- From a strategy watching the device getting deactivated:
  maxDataRate = 1Bit/s trustLevel = 1000

Parameter, Strategies & Deciding

- Fuzzy Logic
- Basic and generic parameters
- Example: load

Parameter, Strategies & Deciding

Fuzzy Logic



If velocity is neg.high then danger is high.
If velocity is neg.low then danger is medium.
If velocity is zero then danger is low.
If velocity is pos.low then danger is medium.
If velocity is pos.high then danger is high.

Parameter, Strategies & Deciding

Basic and Generic Parameter

Basic Parameter:

- device state
- Maximum data rate
- IP date rate
- lost packages
- ...

Generic Parameter:

- load
- link quality
- ...

Parameter, Strategies & Deciding

Example: load

Basic parameter: ipDataRate, deviceDataRate, maxDataRate

Generic parameter:

- ipLoad = ipDataRate/maxDataRate
- deviceLoad = deviceDataRate/maxDataRate
- load = deviceLoad

- estimatedIpLoad = source.ipDataRate/maxDataRate
- EstimatedLoad = estimatedIpLoad + load

If load is low then loadImprovement is low
If load is high and estimatedLoad is high then loadImprovement is low
If load is high and estimatedLoad is medium then loadImprovement is medium
If load is high and estimatedLoad is low then loadImprovement is high

Parameter, Strategies & Deciding

Example: load

Problem: deviceLoad doesn't take the load of other devices on a shared medium into account

=> use of the output queue size as an additional source of information:

Generic Parameter: outputQueue

If deviceLoad is high or outputQueue is full then load is high
...

Additional abstraction layer of generic parameter
=> only local changes necessary when adding/changing basic parameters

Outlook

What is done

- framework
- a few parameters
- simple strategies

What needs to be done

- enhancing the framework
- more complex decission process
- more parameters
- "active" parameter collection(?)
  - messure delays
  - messure packet loss