KNX IP Simulation

First Insights

Revised after KNX IP TF Meeting 08-11-25

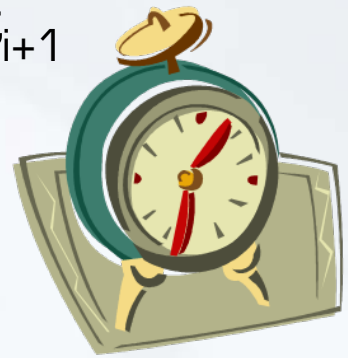
Wolfgang Kastner

Automation Systems Group
Institute of Computer Aided Automation
Vienna University of Technology
Vienna, Austria

www.auto.tuwien.ac.at

Simulation

- Why?
  - To evaluate new network protocols before use
  - To evaluate the performance of (existing) protocols
  - To run large scale experiments
- How?
  - Discrete event system
  - $\alpha = < s_0, (e_0, t_0), s_1, (e_1, t_1), s_2, \ldots >, t_i \leq t_{i+1}$
  - Input parameters
  - Discrete event simulation



Discrete Event Simulator

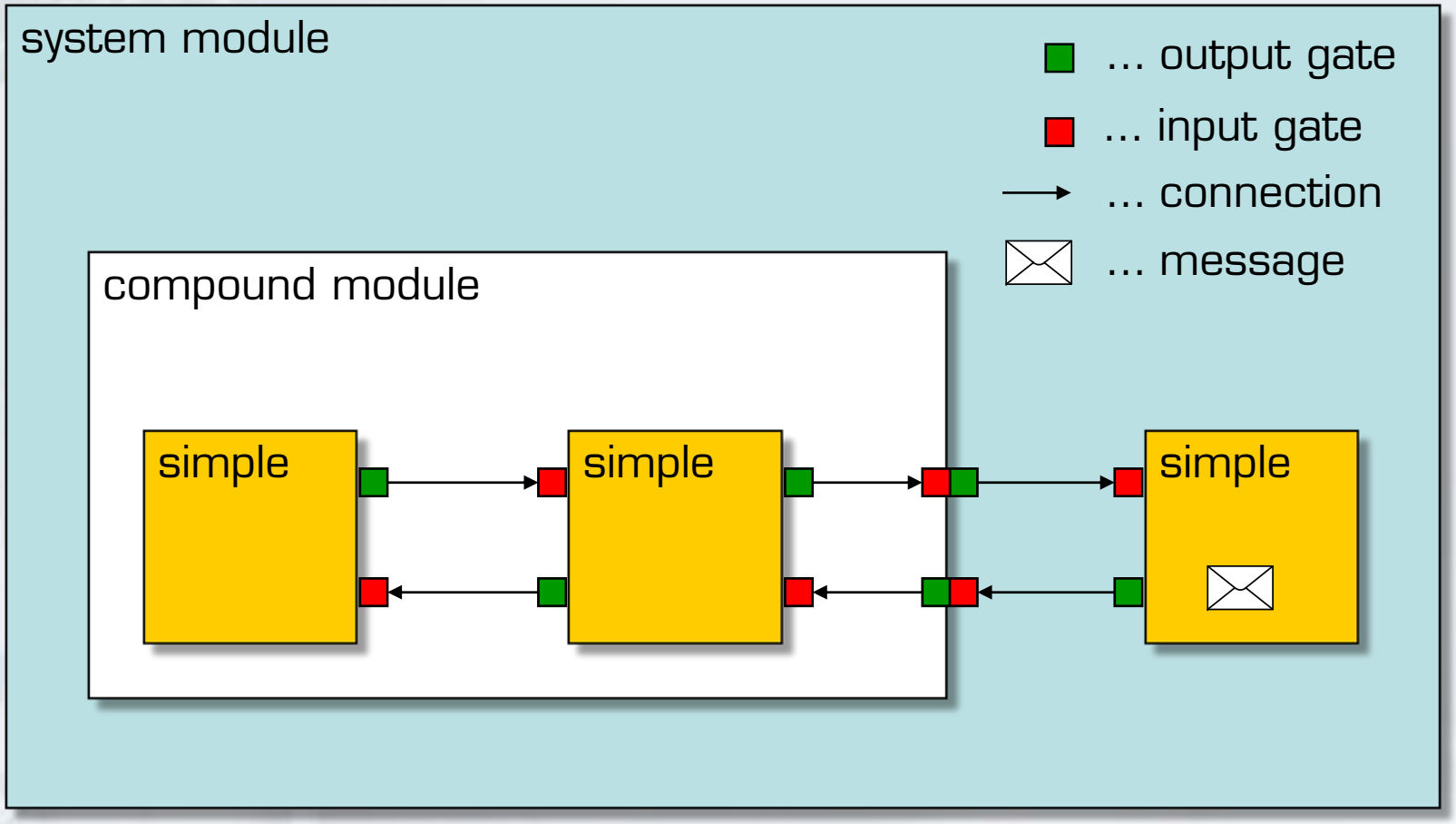
```
init;     // set up internal data structure,
        // insert initial events into the FES
while (FES not empty) {
  retrieve first event from FES;
  timestamp this event;
  process event;          // new events may be
                        // inserted or deleted
}
finish;                       // clean up, write
                  // statistical results, etc.
```

OMNeT++

A discrete event simulation environment

- Mainly focused for communication networks
- Flexible (C++ based) programming model
- Clear separation among simulation kernel and developed models
- Several random number generators
- GUI support
- Open (complete source code available)
- Free for academic and non-profit use
- Several component add-on libraries available

OMNeT++

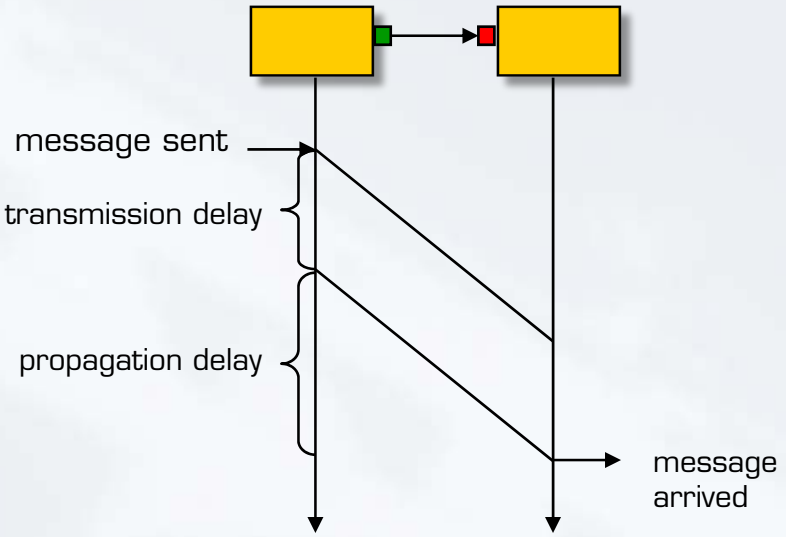

system module
... output gate
... input gate
... connection
... message
compound module
simple
simple
simple

OMNeT++

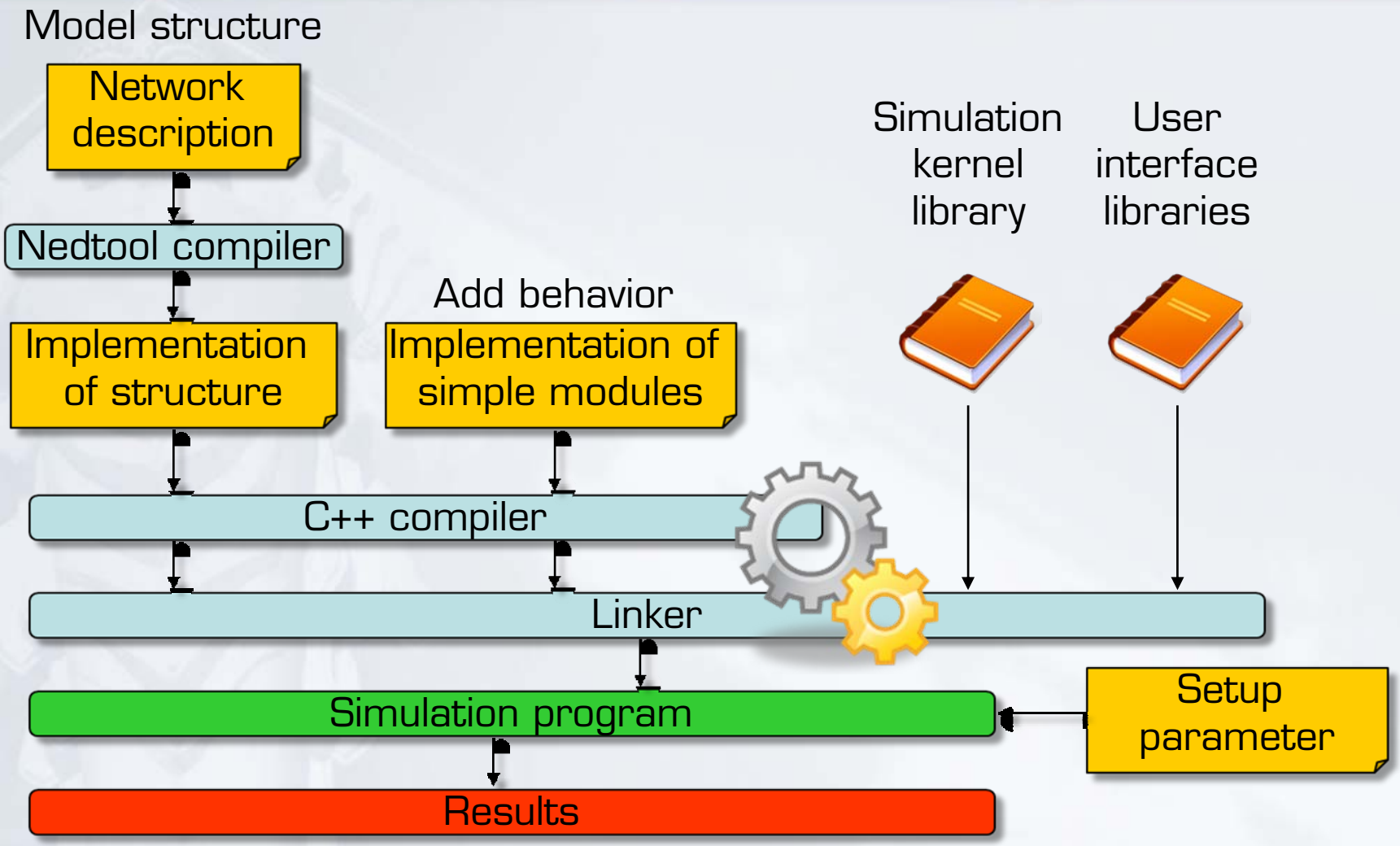
- Simple module
  - C++ using the OMNeT simulation class library
  - Class derived from `cSimpleModule`
  - Redefine virtual member functions `initialize(), handleMessage(), finish()`
  - NED description of interface
- Compound module
  - Unlimited nesting
  - NED only: parameters, gates and connections

OMNeT++

- Messages
  - Sent among modules
  - Attributes: name, kind, length, source and destination modul and gate, sending and arrival times
  - Structure may be defined in a msg file and then automatically generate C++
- Channel Characteristics
  - Propagation delay
  - Bit error rate
  - Data rate

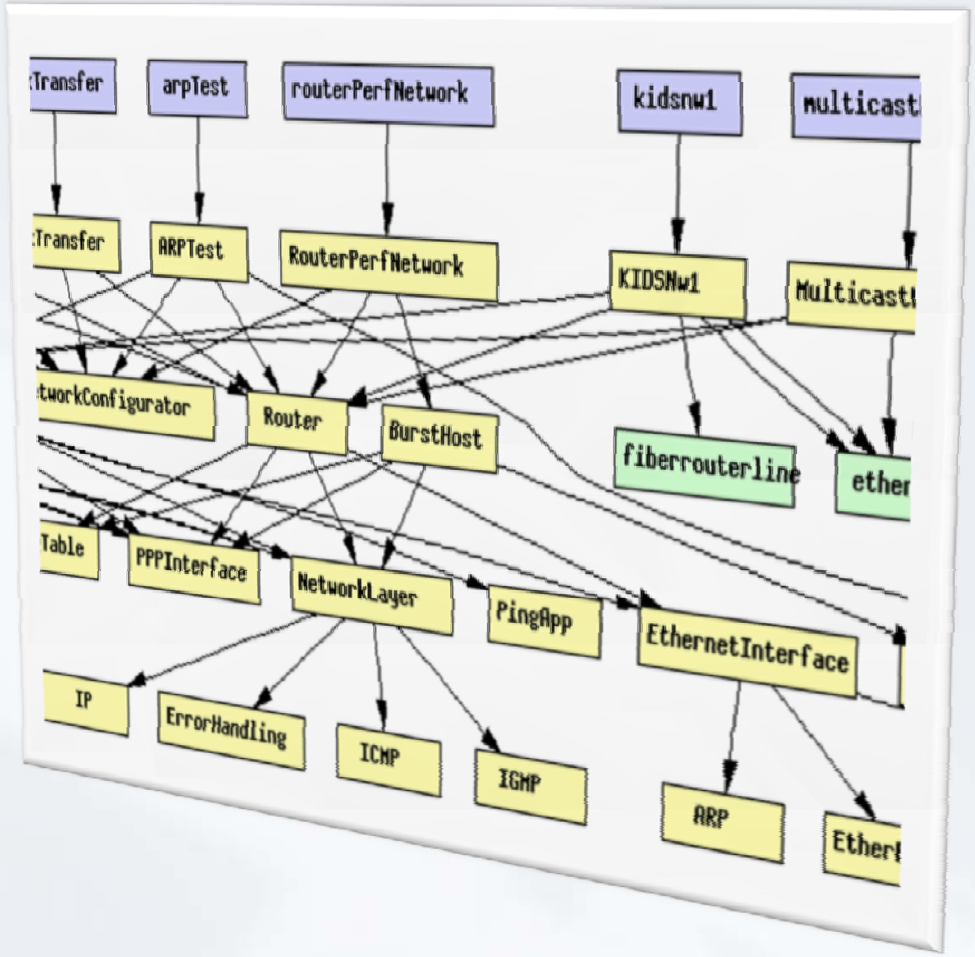


Build and Run Simulations



INET Framework

- Network interfaces
  (e.g. Ethernet, 802.11)
- Network protocols
  (e.g. IP, IPv6, ARP, ICMP)
- Transport protocols
  (e.g. TCP, UDP)
- Application models
  (e.g. EthernetAPP,
  PingApp, TCPApp, UDPApp)
- Nodes
  (e.g. Hub, Router, Switch,
  StandardHost)



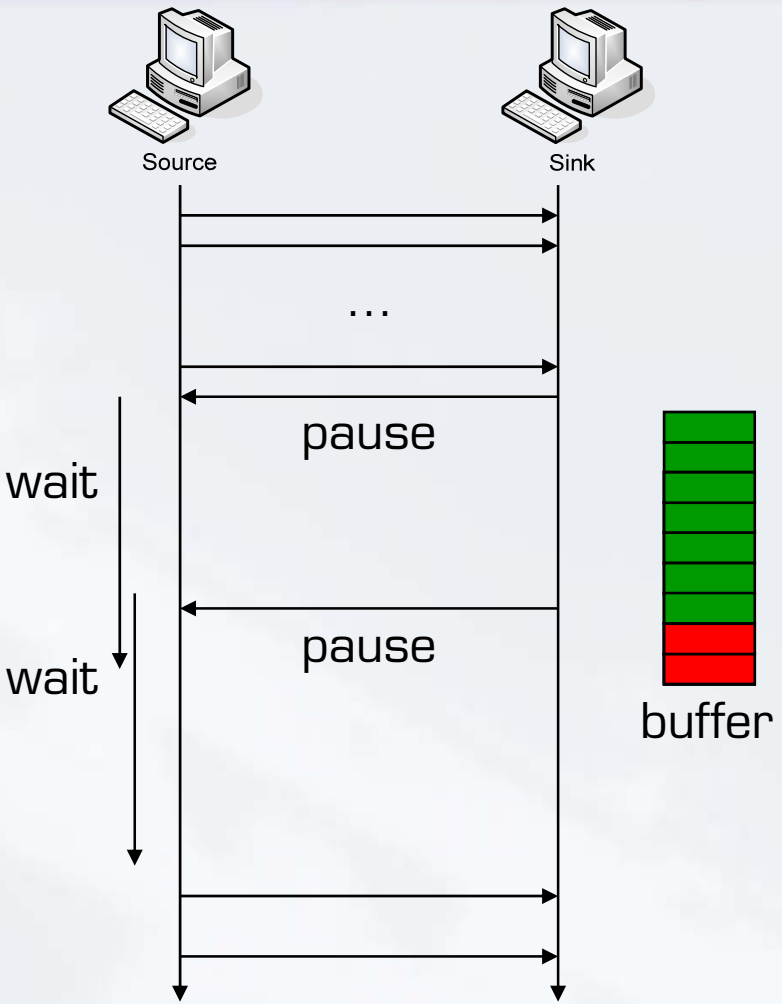
KNX IP Device Modeling

- Assumption:
  - MAC part (Ethernet)
  - µP part (KNX IP)
  - Buffer in between

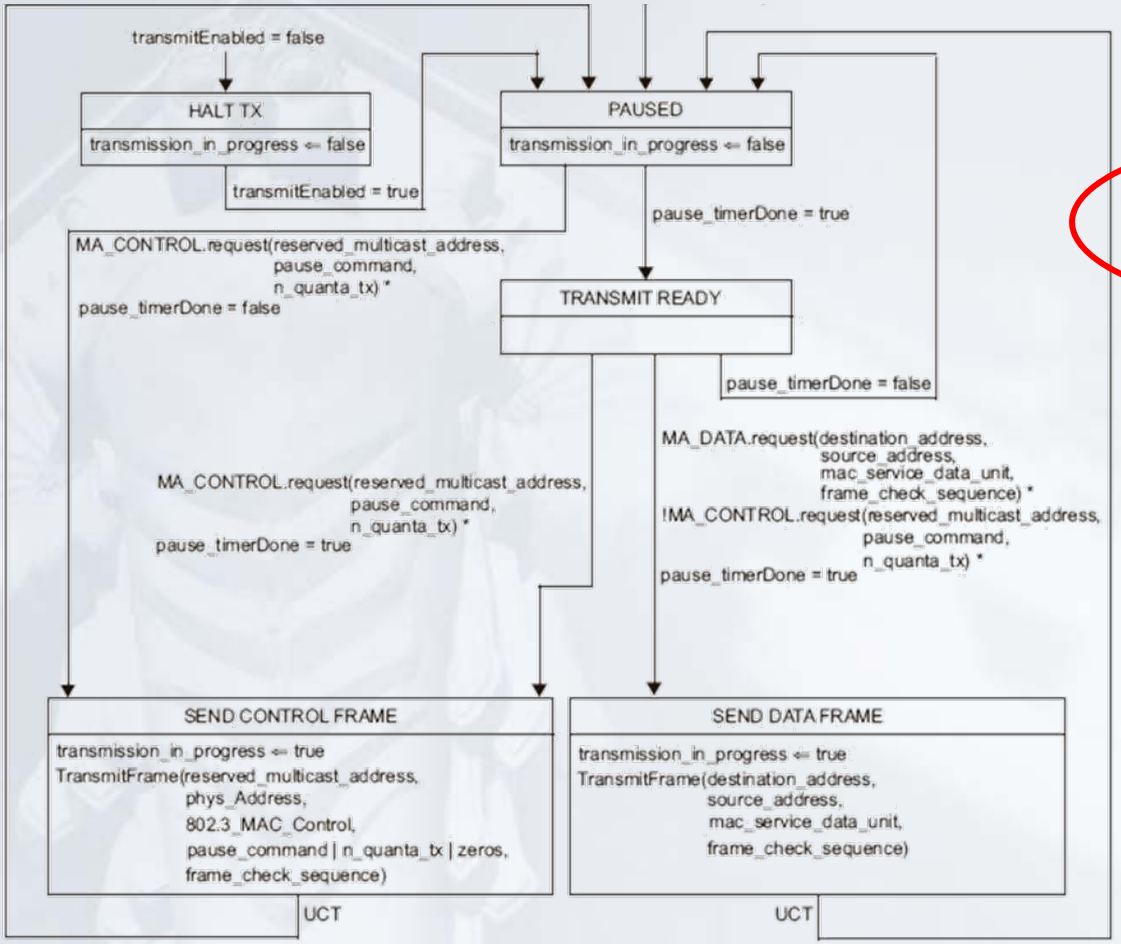
- Specific parameters:
  - KNX IP device conformance class
  - Buffer size and data transmission rate MAC / µP
  - KNX IP flow control
  - Ethernet flow control
  - Network parameters

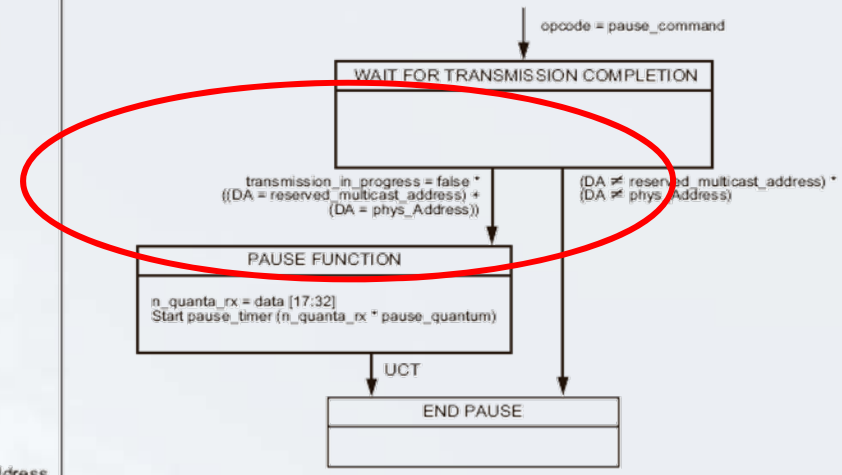
IEEE 802.3 Annex 31B

- “MAC Pause Operation“
- Reserved multicast address
- The specification states “…it is not required that an implementation be able to transmit PAUSE frames."
- However, a node has to respond to PAUSE frames



IEEE 802.3 Annex 31B



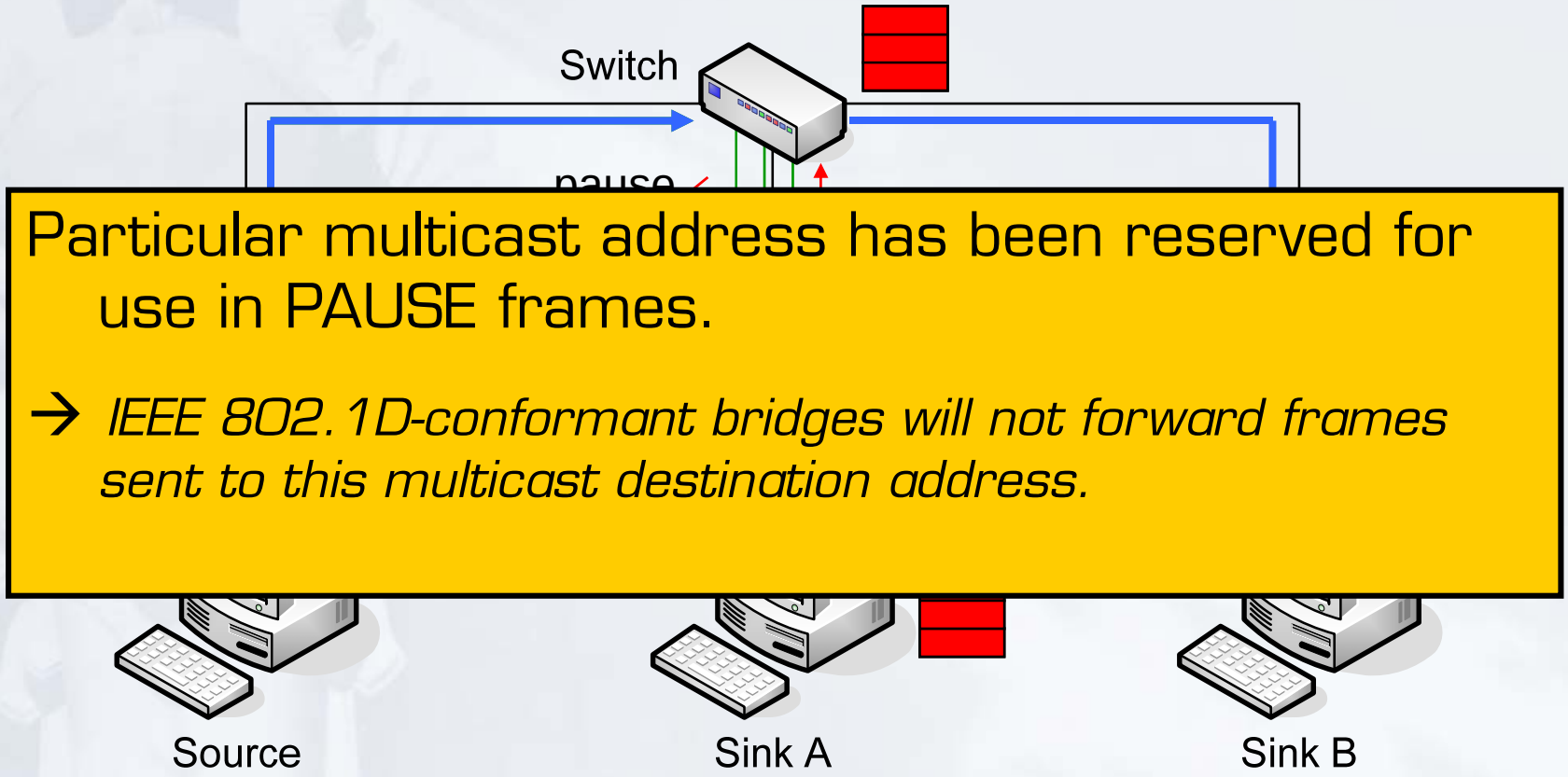


Assumption:

Destination address has to correspond to the reserved multicast address

IEEE 802.3 Annex 31B

A method for flow controlling an individual link
→ it is not a method for end-to-end flow control!

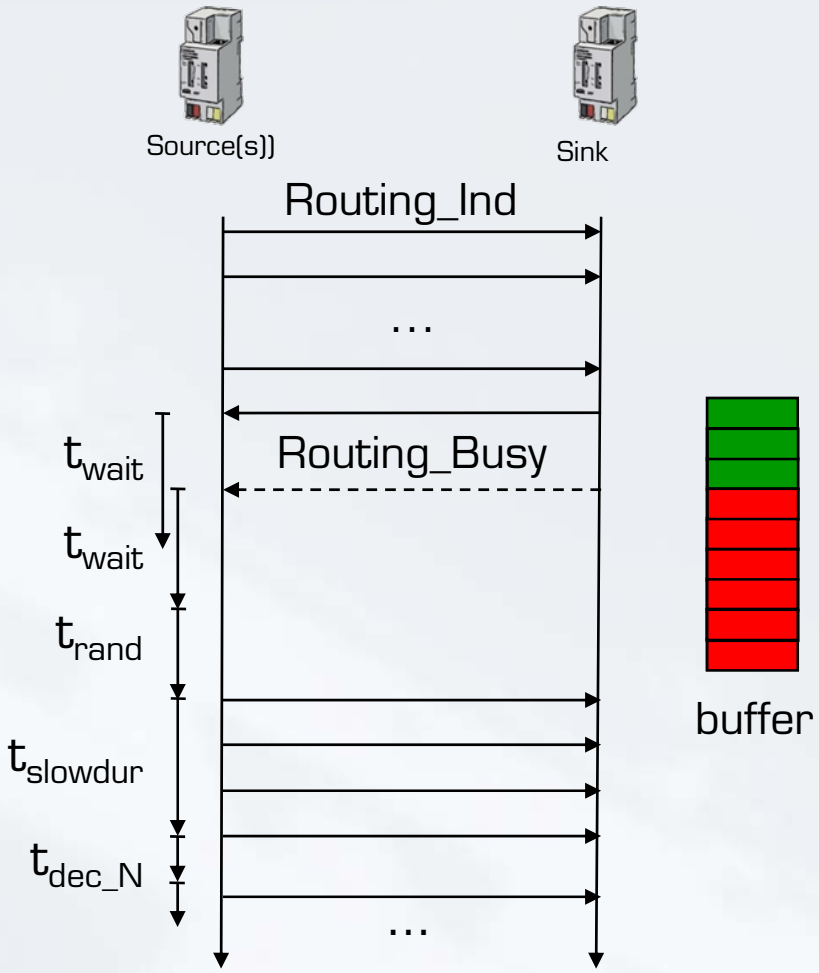


Particular multicast address has been reserved for use in PAUSE frames.

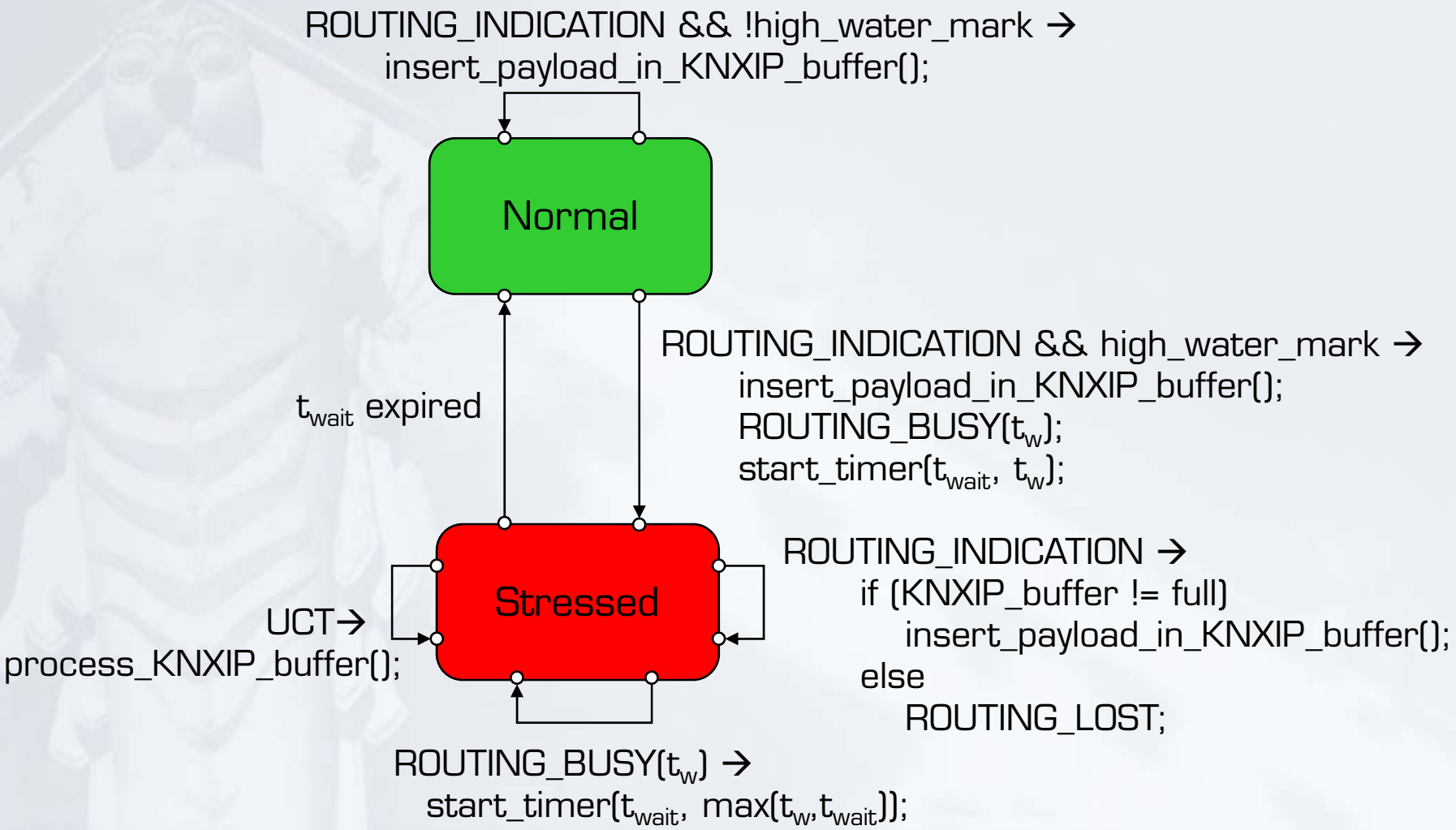
→ *IEEE 802.1D-conformant bridges will not forward frames sent to this multicast destination address.*

KNX IP Flow Control

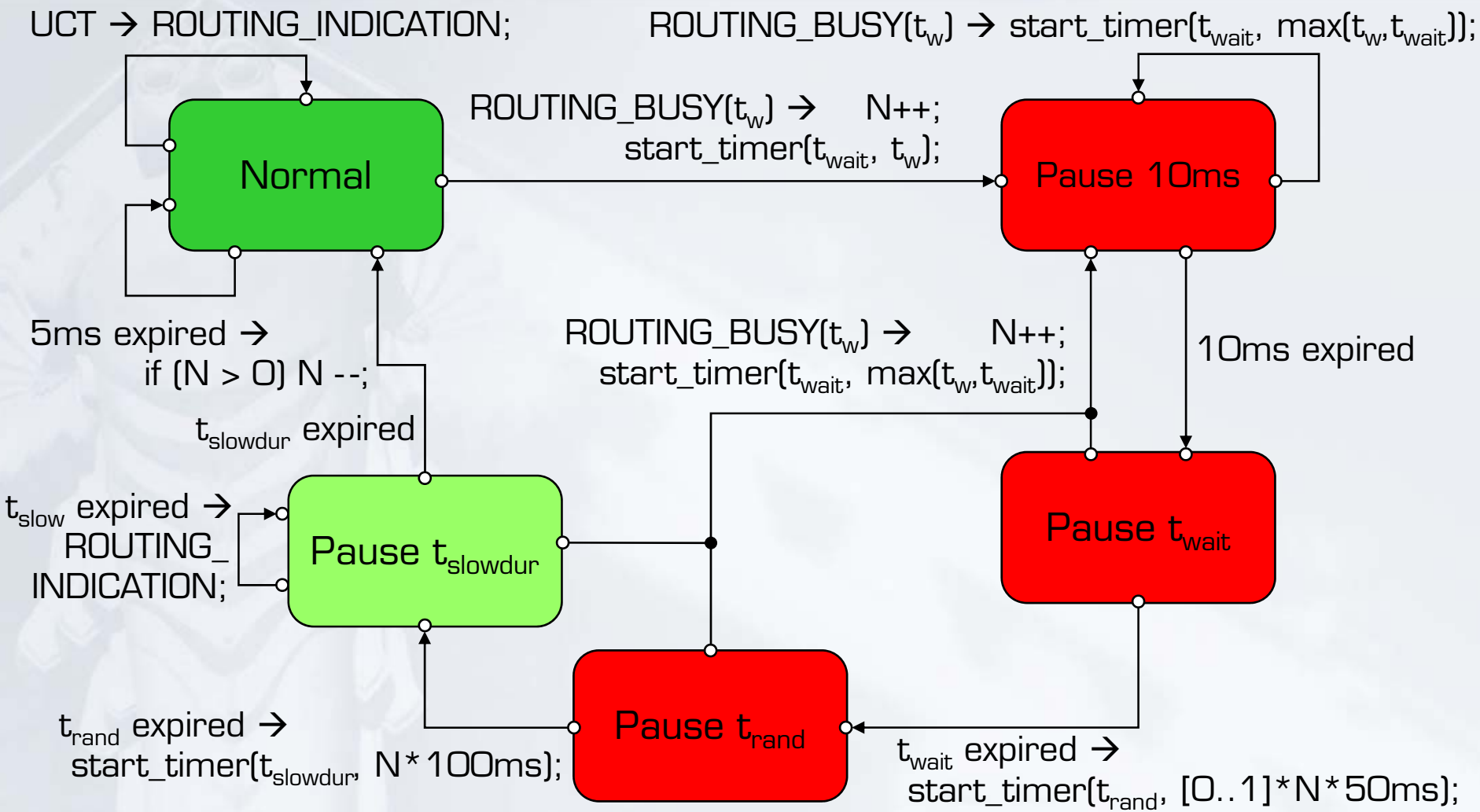
- KNX TF IP / AN 117
- Each KNX IP device may generate 50 routing datagrams per second
- Parameters:
  - t_{wait} ≤ 100ms
  - t_{rand} = [0..1]*N*50ms
  - $t_{slowdur}$ = N*100ms
  - t_{slow} = 5ms
  - t_{dec_N} = 5ms



KNX IP Flow Control (Sink)

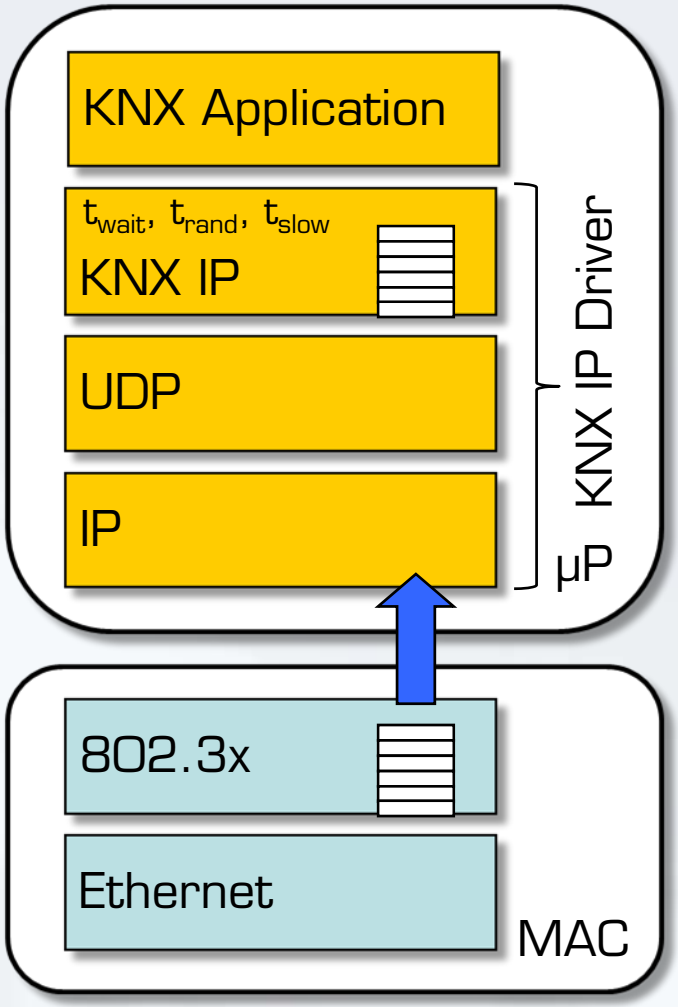


KNX IP Flow Control (Source)



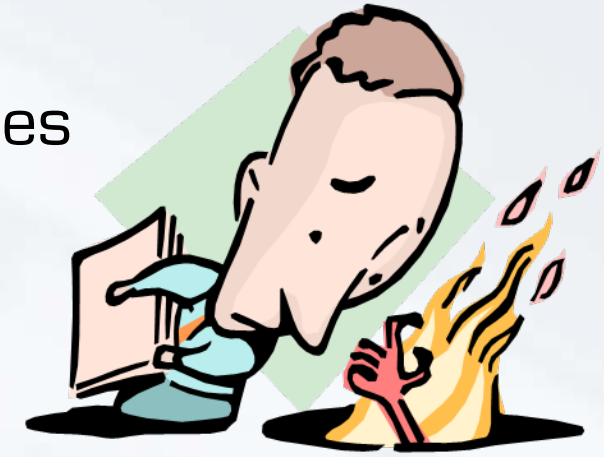
KNX IP Device Model

- KNX IP conformance class
- Variable data rate of routing datagrams (max. 50 per second)
- t_{wait}, t_{rand}, t_{slow}
- Processing time and KNX IP buffer size (water mark)
- Buffer size between MAC and µP
- Data transmission rate between MAC and µP
- IEEE 802.3x buffer size
- Network parameters



Roadmap

- Network reserved for KNX IP
  - Limited to one multicast address (255 devices)
  - More than one multicast addresses (up to 65.000 devices)
- Network shared with office equipment
  - Limited to one multicast address
  - More than one multicast addresses
  - Broadcast traffic load
  - Additional multicast load



Conclusion and Outlook

- Clarify protocol (state machines)
- Intensive code review
- Run simulations (traffic load analysis)
- KNX IP Router and KNXnet/IP Router
- Office Traffic Load Generator

- Later on…
  - Connection to real world devices?
  - Analytical approach?