KNX IP Simulation

Status Update

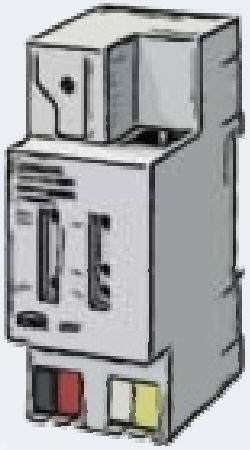
Wolfgang Kastner

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

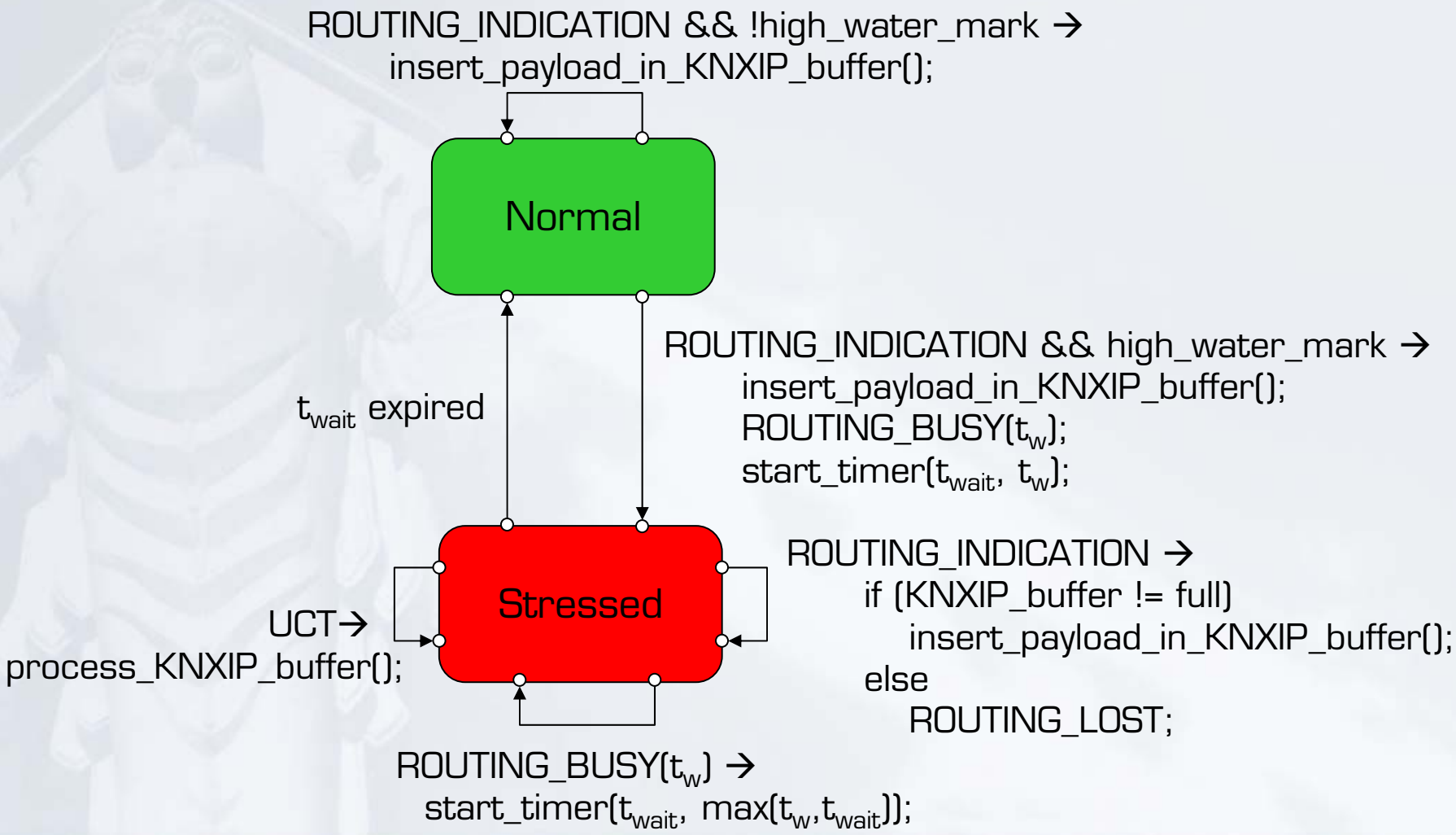
www.auto.tuwien.ac.at

KNX IP Device Modeling

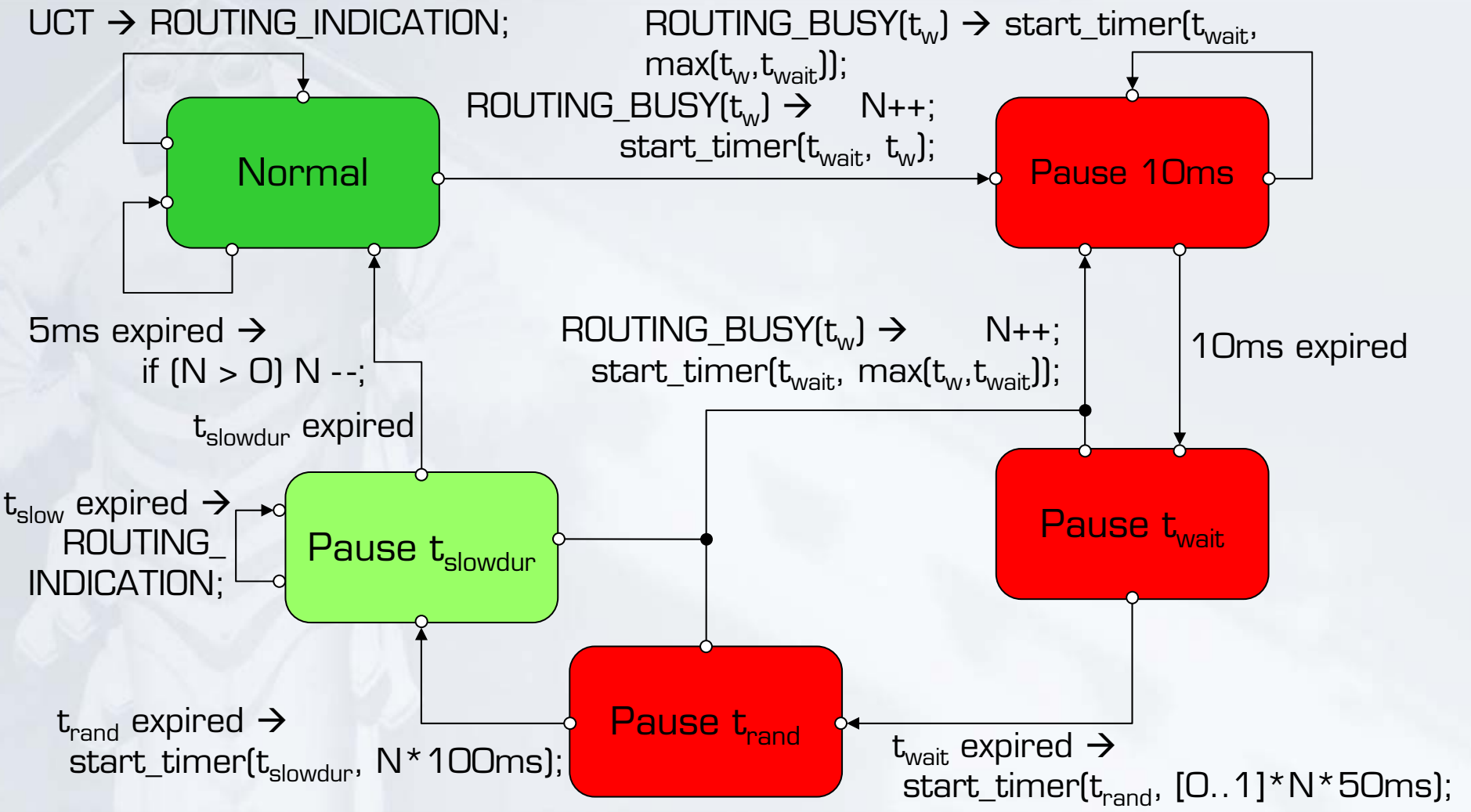
- Assumption:
  - MAC part (Ethernet)
  - µP part (KNX IP)
- Specific parameters:
  - KNX IP flow control
  - Buffer size
  - Data transmission rate
  - Network parameters
  - Ethernet flow control (postponed)



KNX IP Flow Control (Sink)



KNX IP Flow Control (Source)



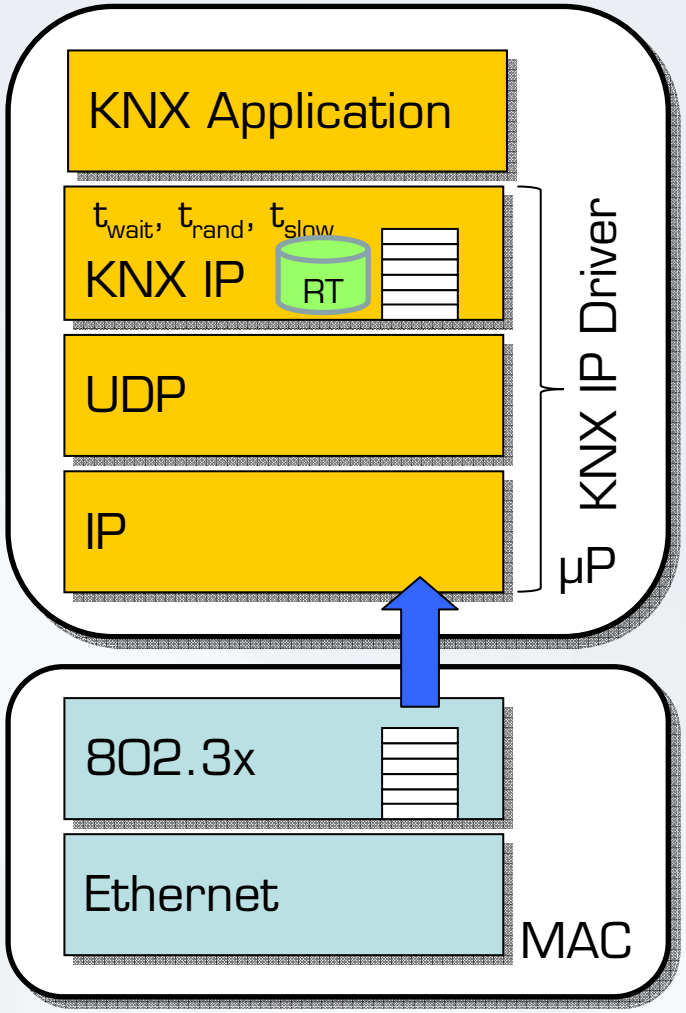
KNX IP Device Model

KNX Application

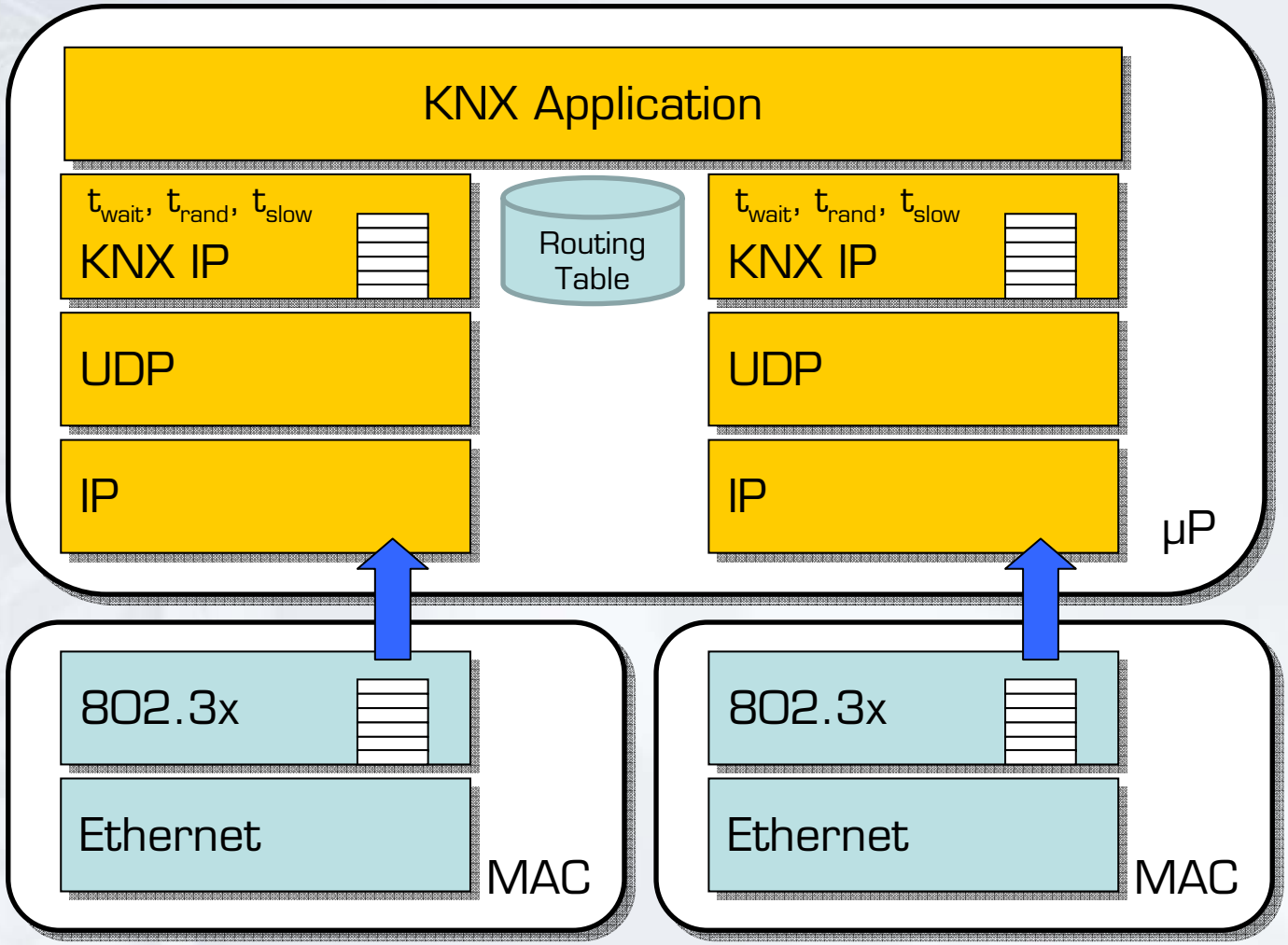
- Traffic generator

KNX IP Driver

- KNX IP Protocol
- Ringbuffer (50 messages)
- Routing table (RT) with KNX group addresses
- Promiscuous mode
- Interface to KNX application: message handling if sending allowed



KNX IP Router Device Model



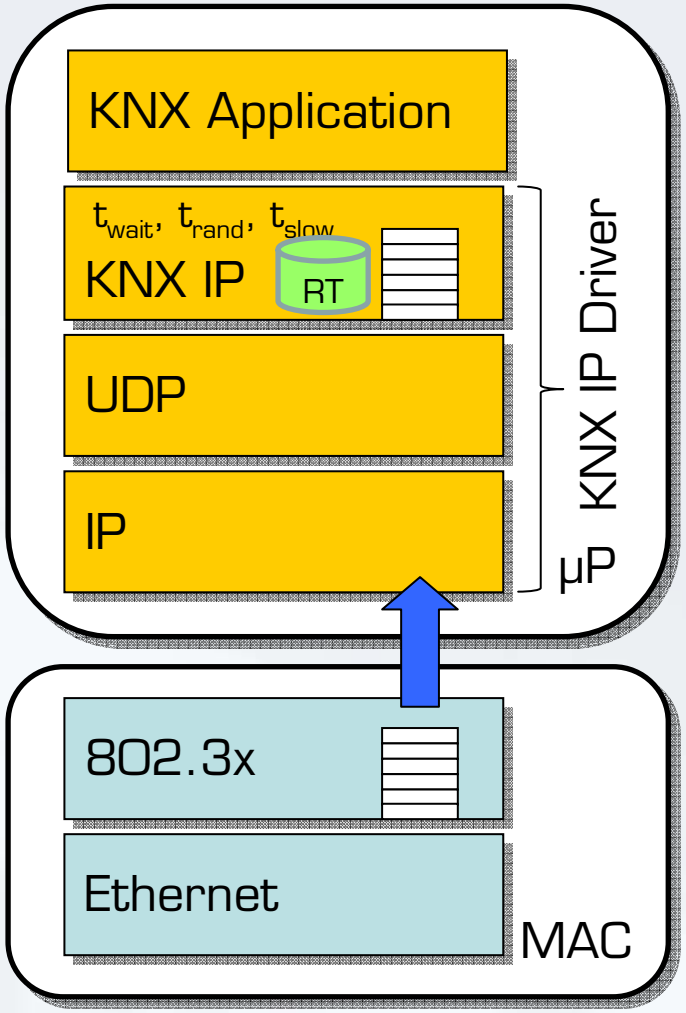
KNXnet IP Router Model

(Special) KNX Application

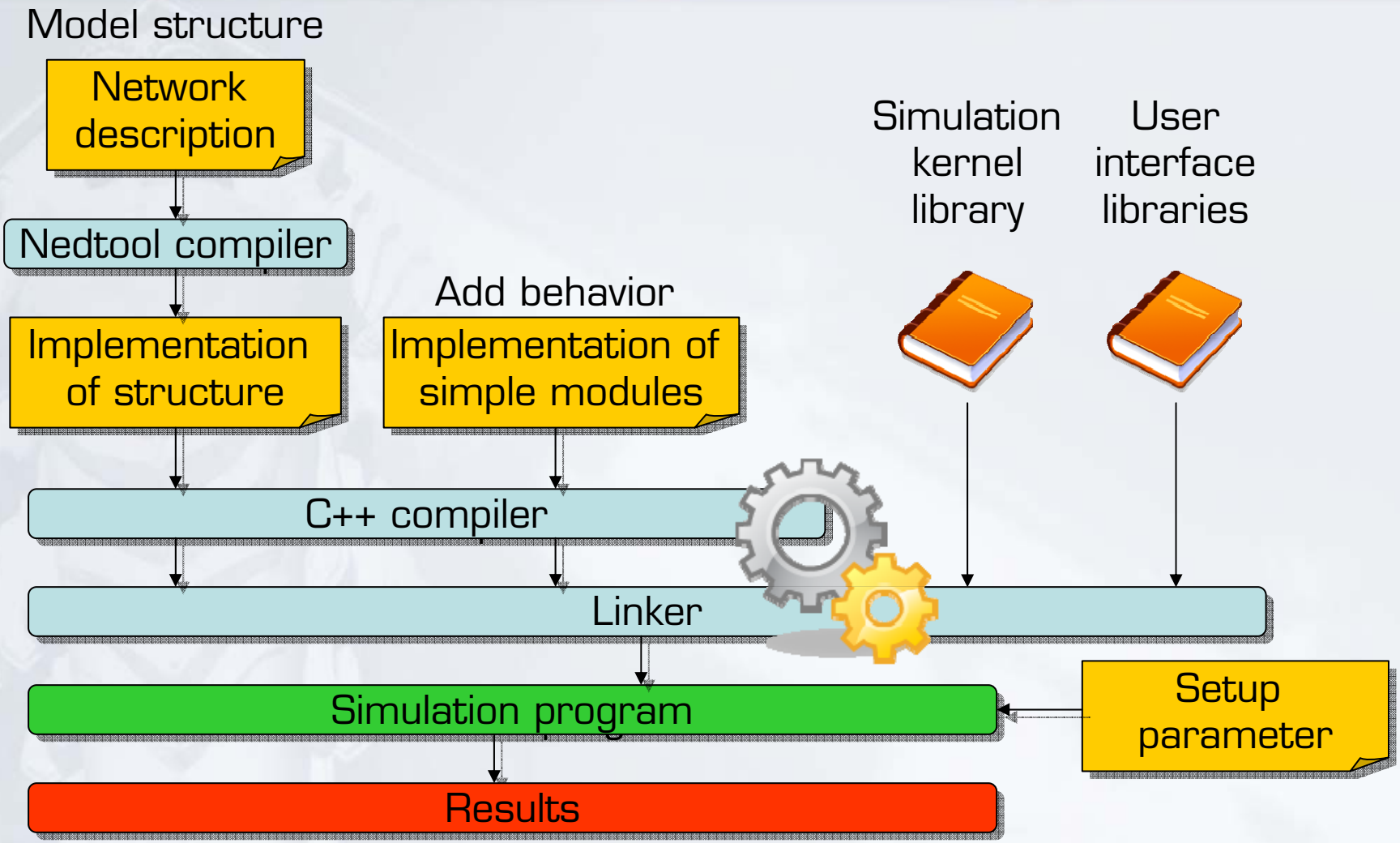
- Traffic generator (half duplex)

KNX IP Driver

- KNX IP Protocol
- Ringbuffer (50 messages)
- Routing table (RT) with KNX group addresses
- Promiscuous mode



Build and Run Simulations



(Some) Implementation Details

- handle_message
  - isSelfmessage() → processTimer
  - from_udp() → processPacketfromBelow
  - otherwise → processPacketfromAbove
- processPacketfromBelow()
  - Routing Indication
    - groupAddressFilter
    - checkQueue → normal/stressed (old state!), drop
  - Routing Busy
    - handleBusy

(Some) Implementation Details

- processPacketfromAbove
  - Generate Routing Busy datagrams
  - Handle messages form KNX Application

- processTimer
  - Update state machine (waittimer)
  - Check ring buffer (send timer)
  - Wait processing time (processing timer)

What can be watched?

- KNX sent/received:
  - KNX Routing Indication datagrams sent/received
- KNX Busy sent/received:
  - KNX Routing Busy datagrams sent/received
- KNX dropped:
  - KNX Routing Indication datagrams dropped
- KNX State
  - State of KNX IP device (normal, ignore, busy, …)

Simulation Parameters

knxIpFlow.t_busyWaitTime = 50ms;

knxIpFlow.t_slow = 20;

knxIpFlow.t_decrementBusyCounter = 5;

knxIpFlow.t_ignoreTime = 10;

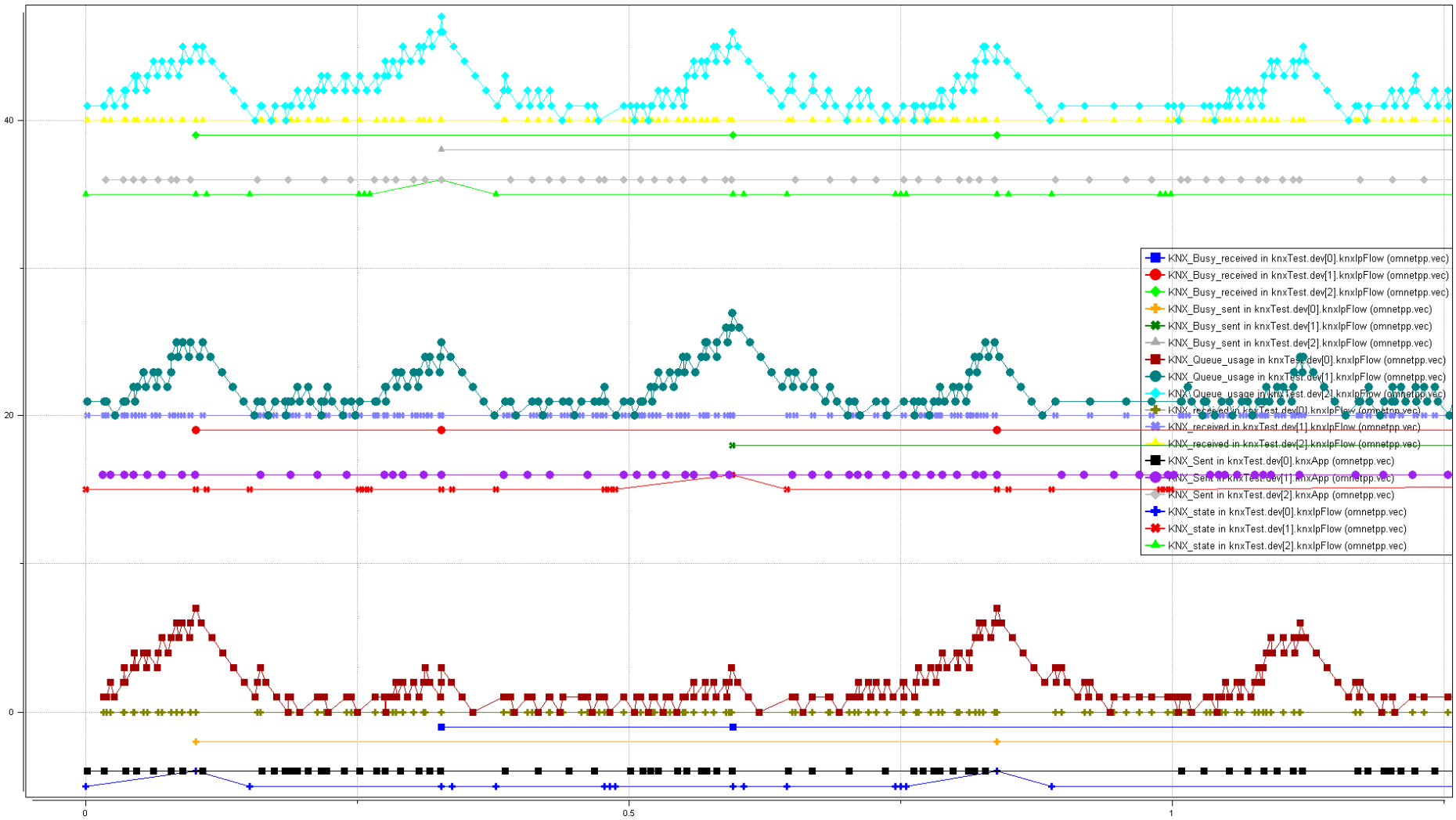
knxIpFlow.t_normalTime = 5;

knxIpFlow.buffersize = 8;

knxIpFlow.highwatermark = 6;

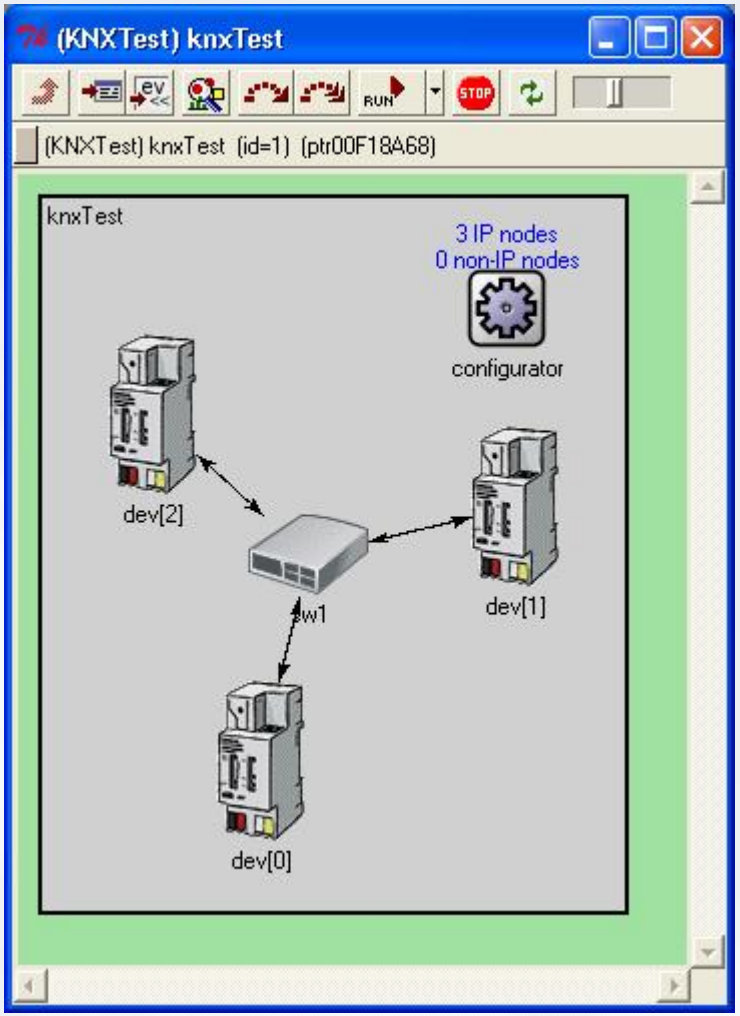
t_processingFlow = 10ms;

Simulation (3 KNX IP Devices)



Network Configuration File (NED)

```
import
    „FlatNetworkConfigurator";
channel ethernetline
    delay 0.1us; datarate 10*1000000;
endchannel
module KNXTest
    parameters:
        n: numeric const;
    submodules:
        configurator: FlatNetworkConfigurator;
        sw1: EtherSwitch2;
        dev: KnxIpHost[n]
    connections nocheck:
            for i = 0..n-1 do
                dev[i].out++ –> ethernetline –> sw1.in++;
                dev[i].in++ <– ethernetline <– sw1.out++;
            endfor;
endmodule
```



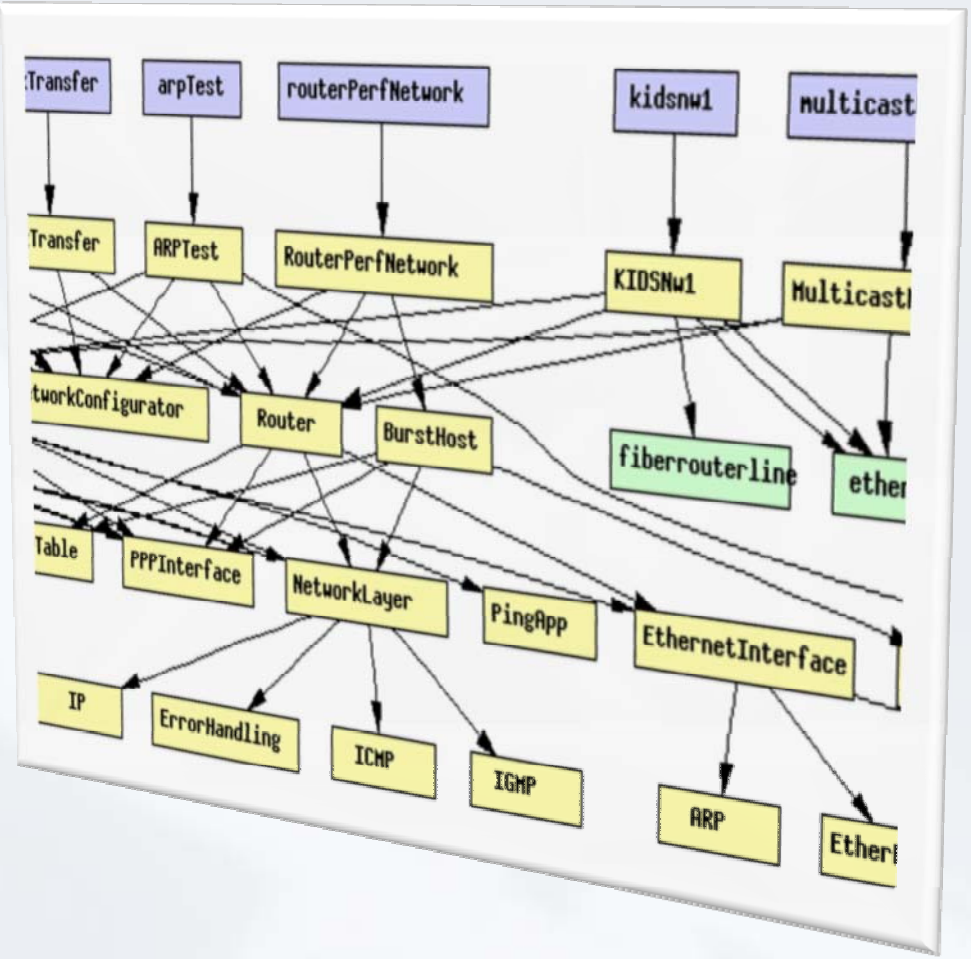
Simulation Configuration File

```
<config>
  <linelist>
        <line id="1">239.24.24.13</line>
        <line id="2">239.24.24.24</line>
  </linelist>
  <routerlist>
        <router id="1">
              <belongsto>1</belongsto>
              <belongsto>2</belongsto>
              <in>1/1/21</in>
              <out>1/1/41</out>
              <inout>1/1/31</inout>
        </router>
        <router id="2">
              ...
        </router>
  </routerlist>
```

```
  <devicelist>
        <device id="1">
              <belongsto>1</belongsto>
              <inout>1/2/2</inout>
        </device>
        <device id="2">
              <belongsto>1</belongsto>
              <inout>1/2/3</inout>
        </device>
        <device id="3">
              <belongsto>2</belongsto>
              <inout>1/2/5</inout>
        </device>
  </devicelist>
</config>
```

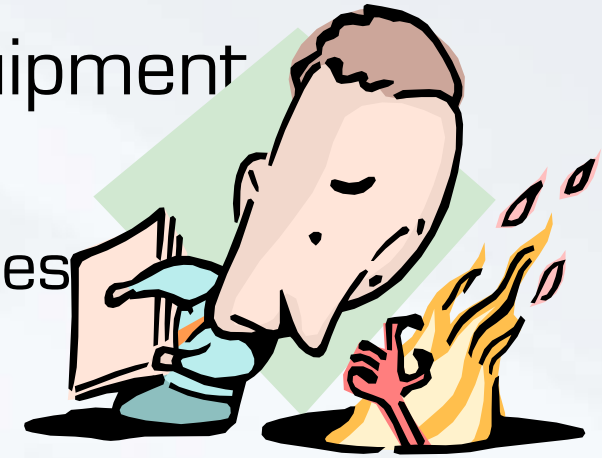
INET Framework

- 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)



Roadmap

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



Next Steps

- Again clarify protocol (state machines)
- Intensive code review
- Split processing time (check and process)
- KNXnet/IP Router
- 802.3x (+ buffer)
- Office Traffic Load Generator
- Run simulations (traffic load analysis)
  - Simulation configuration file, NED file
  - Simulation environment (VMWare)