Real-Time Ethernet on Top of RTAI

University of Hannover
ISE – Real Time Systems Group

Overview

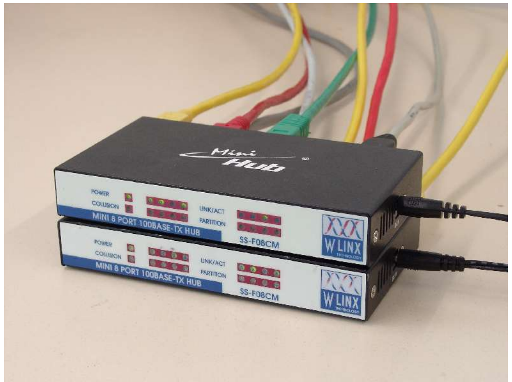
- Motivation
- Concepts and Features
- Recent Improvements
- Applications at the RTS
- Summary and Outlook

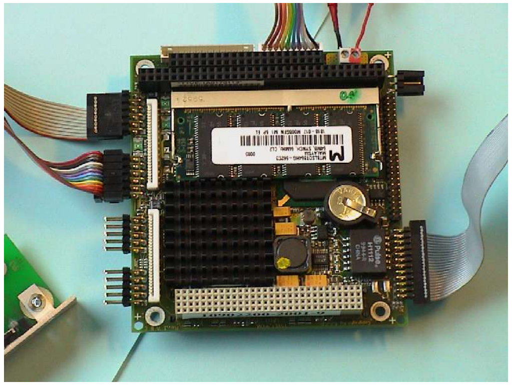
Motivation

Ethernet technology:

- Inexpensive components
  - ➔ Connectors and cables
  - ➔ Network adapters
  - ➔ Hubs or switches
  - ➔ Embedded PCs
- High data rates
  - ➔ 10/100/... MBit/s
- Single-cable solution
  - ➔ Real-time data **and** standard TCP/IP over the same link

➔ **Software Solution**



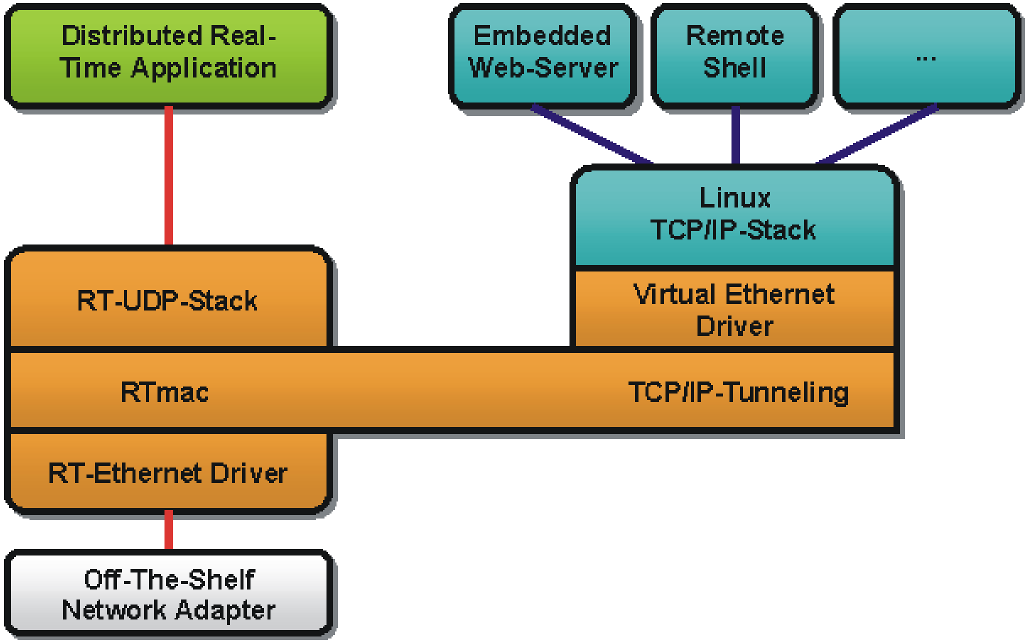


Open Source License

- Define really open protocols
- Remain vendor independent (long-term availability)
- Create flexible platform for science and industry
- Use of existing OSS
  - Original version (David Schleef, 2000)
  - Drivers (Linux kernel)
  - UDP/IP stack (Linux, only in the beginning, now reference)
  - RTAI as real-time OS
- Build up user and developer community
  - **3rd-party feedback**
  - Patches
  - Extensions (drivers, ICMP, etc.)

A Brief Look Inside...

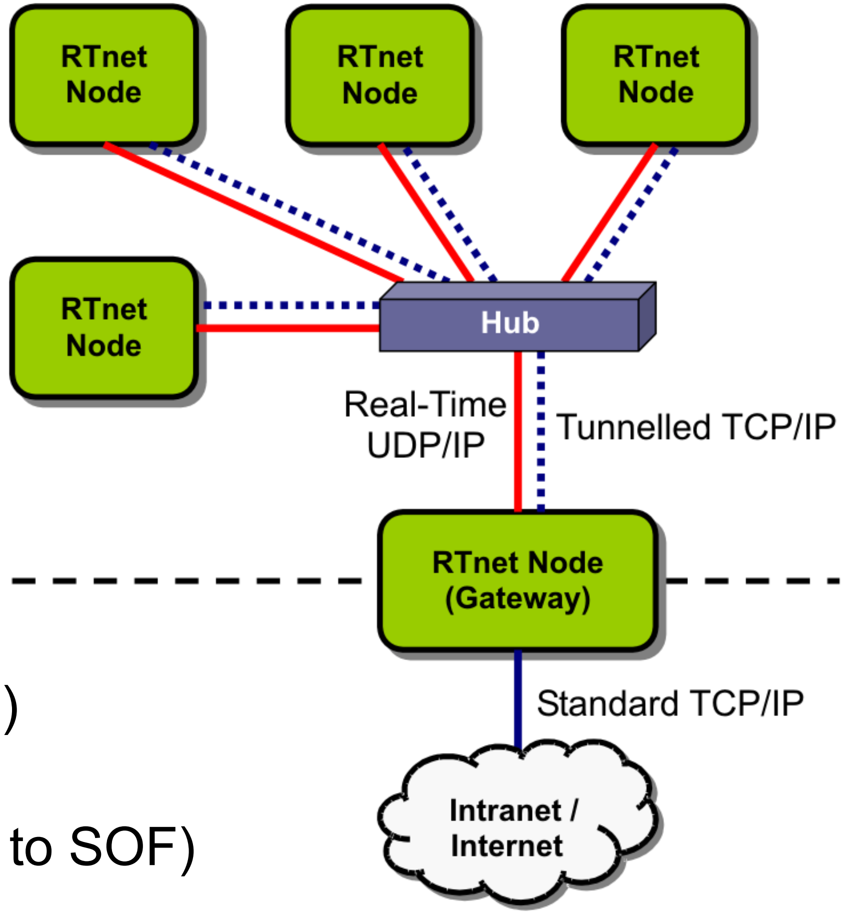
- Linux-like NIC driver layer
- Optional media access control (RTmac)
- VNIC tunnels non real-time traffic



- Extensible stack (Layer 3 and 4)
  - Independent buffer pools (sockets, NICs, VNIC, etc.)
  - IP fragmentation supported with restrictions
- BSD socket API (UDP and Packets)

Real-Time Media Access Control

- Requires dedicated network
- RTmac controls transmission access to NIC
- Multiple access control mechanisms feasible
- Basic TDMA
  - Master transmits periodic synchronization packet (SOF)
  - Clients transmit only within a dedicated slot (offset relative to SOF)
  - Global time stamp service

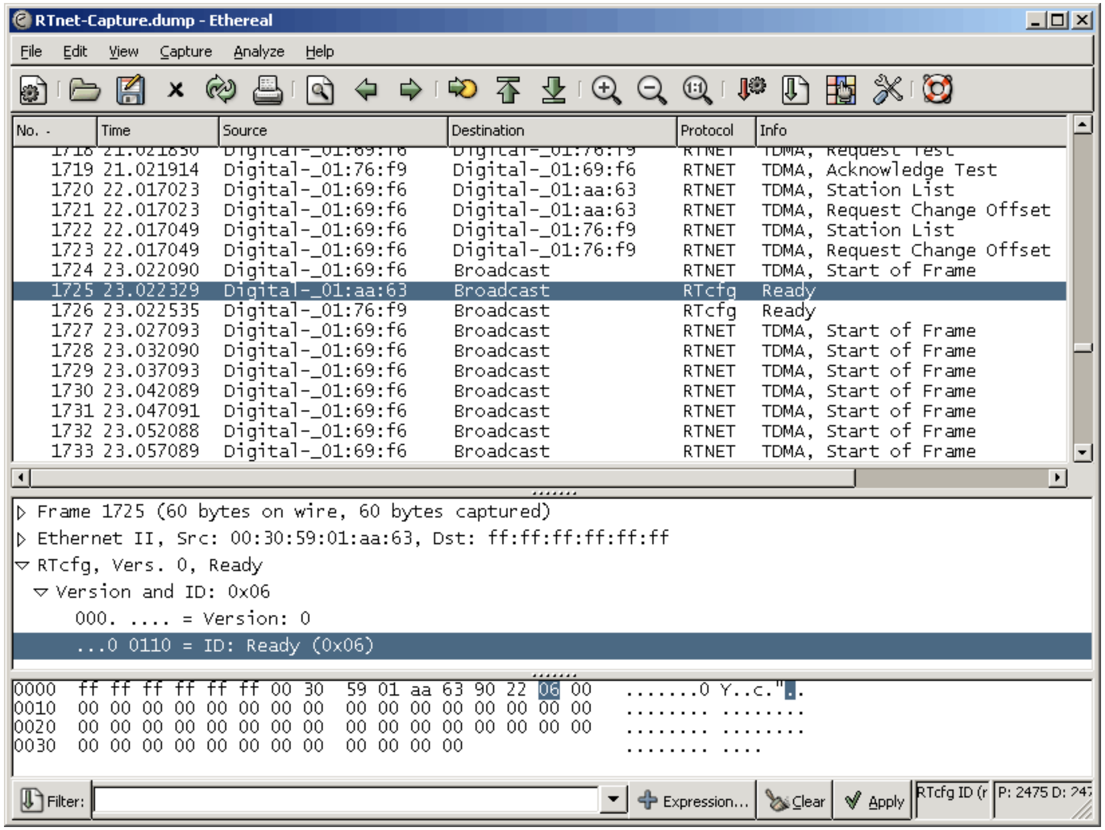


Real-Time Configuration Protocol

- Generic protocol consisting of 3 stages
- Independent of MAC mechanism (RTmac discipline)
- Stage 1
  - Client invitation
  - Distribution of RTmac configuration
- Stage 2
  - Hardware address exchange
  - Distribution of arbitrary configuration data
- Stage 3
  - Final synchronisation after system initialisation

Network Diagnosis

- RTcap: Real-time capturing support
- Ethereal plug-in (RTmac/TDMA, RTcfg)



```
No. -  Time       Source              Destination         Protocol  Info
1718 21.021850  Digital-_01:69:f6   Digital-_01:76:f9   RTNET     TDMA, Request Test
1719 21.021914  Digital-_01:76:f9   Digital-_01:69:f6   RTNET     TDMA, Acknowledge Test
1720 22.017023  Digital-_01:69:f6   Digital-_01:aa:63   RTNET     TDMA, Station List
1721 22.017023  Digital-_01:69:f6   Digital-_01:aa:63   RTNET     TDMA, Request Change Offset
1722 22.017049  Digital-_01:69:f6   Digital-_01:76:f9   RTNET     TDMA, Station List
1723 22.017049  Digital-_01:69:f6   Digital-_01:76:f9   RTNET     TDMA, Request Change Offset
1724 23.022090  Digital-_01:69:f6   Broadcast           RTNET     TDMA, Start of Frame
1725 23.022329  Digital-_01:aa:63   Broadcast           RTcfg     Ready
1726 23.022535  Digital-_01:76:f9   Broadcast           RTcfg     Ready
1727 23.027093  Digital-_01:69:f6   Broadcast           RTNET     TDMA, Start of Frame
1728 23.032090  Digital-_01:69:f6   Broadcast           RTNET     TDMA, Start of Frame
1729 23.037093  Digital-_01:69:f6   Broadcast           RTNET     TDMA, Start of Frame
1730 23.042089  Digital-_01:69:f6   Broadcast           RTNET     TDMA, Start of Frame
1731 23.047091  Digital-_01:69:f6   Broadcast           RTNET     TDMA, Start of Frame
1732 23.052088  Digital-_01:69:f6   Broadcast           RTNET     TDMA, Start of Frame
1733 23.057089  Digital-_01:69:f6   Broadcast           RTNET     TDMA, Start of Frame
```

```
▷ Frame 1725 (60 bytes on wire, 60 bytes captured)
▷ Ethernet II, Src: 00:30:59:01:aa:63, Dst: ff:ff:ff:ff:ff:ff
▽ RTcfg, Vers. 0, Ready
  ▽ Version and ID: 0x06
      000. .... = Version: 0
      ...0 0110 = ID: Ready (0x06)
```

```
0000  ff ff ff ff ff ff 00 30  59 01 aa 63 90 22 06 00   .......0 Y..c.".. 
0010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00              ........ ....
```

RTnet Requirements

- Linux 2.4.19 or better (2.6 is work-in-progress)
- RTAI 24.1.11 or better (including 3.x)
- Available for x86 and PowerPC
- Standard NIC with supported chipset
  - Intel 8255x EtherExpress Pro100
  - DEC 21x4x Tulip
  - RealTek RTL8139
  - AMD PCnet32/PCnetPCI
  - VIA Rhine
  - NatSemi DP8381x
  - MPC8xx (SCC and FEC Ethernet)
  - MPC8260 (FCC Ethernet)
  - SMSC LAN91C111

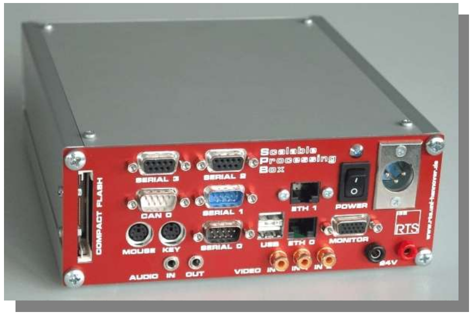
Recent Improvements

- Release 0.7.0
  - API based on Real-Time Driver Model (RTDM)
  - Rewritten routing system
  - Real-time IP forwarding (allows structured RT networks)
  - Revised and new management tools (rtifconfig, rtroute, rtping)
- Real-Time Publish-Subscribe on Top of RTnet
  - OCERA component ORTE runs on RTnet
  - ORTE: GPL implementation of RTI's RTPS protocol
  - Requirements: ORTE CVS check-out, RTAI 3.x, RTnet 0.7.0
  - See www.ocera.org

Experimental Robots at the RTS

MoRob, SPB, LiRE

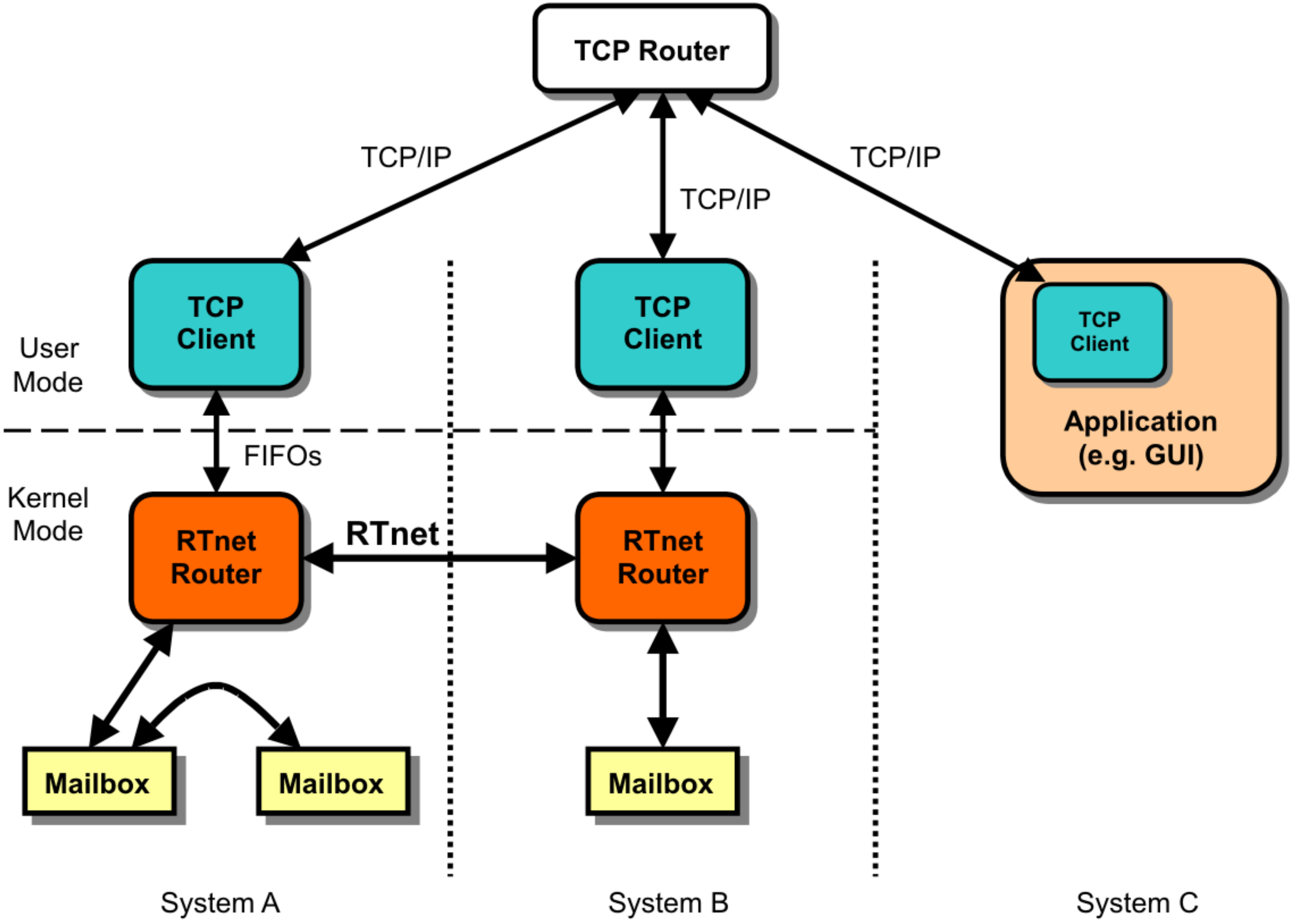
- MoRob – Modular Educational Robotic Toolbox
  - International project to develop a robotics framework for education and research
  - Covers hardware and software
- SPB – Scalable Processing Box
  - Embedded x86 boards in a box
  - Mass storage: Flash disk
  - CAN, RS-232, RS-485, 1-2 LAN



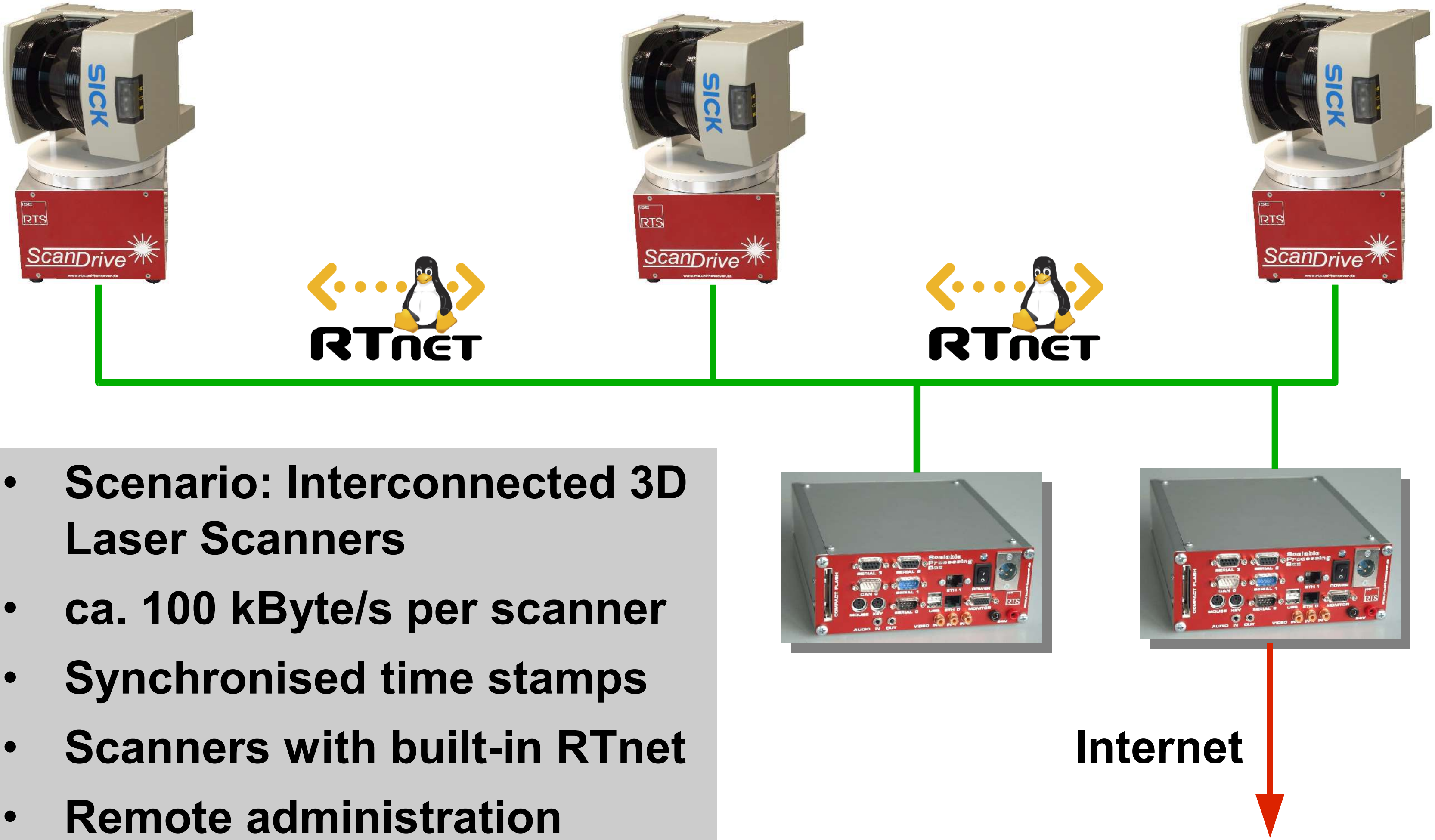
- LiRE – Linux Real-Time Environment
  - Precompiled embedded Linux/RTAI distribution
  - Runs on SPB and any RTAI-capable x86 box
  - Includes RTnet packages => simple access to RT Ethernet!

www.morob.org

Yet Another RT-Middleware...



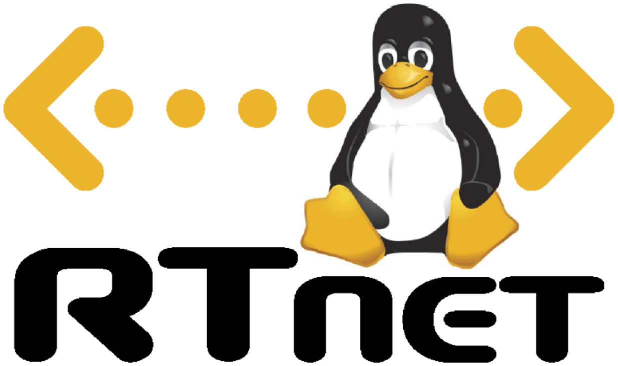
Distributed Real-Time Computing



- Scenario: Interconnected 3D Laser Scanners
- ca. 100 kByte/s per scanner
- Synchronised time stamps
- Scanners with built-in RTnet
- Remote administration

Summary and Outlook

- RTnet: Software-based hard real-time Ethernet
- Community project maintained by the RTS
- Provides foundation for both direct communication and various real-time middlewares
- Highly flexible, adaptable to project needs (network topology, unicast/broadcast, configuration, etc.)
- TDMA Version 2
  - More flexible slot assignment
  - Hot-plugging
  - Fall-back master
- Support for ARM platforms (depends a bit on RTAI...)



rtnet.sourceforge.net

www.rts.uni-hannover.de

kiszka@rts.uni-hannover.de