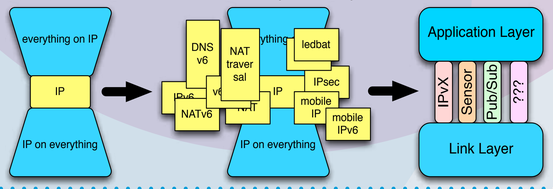
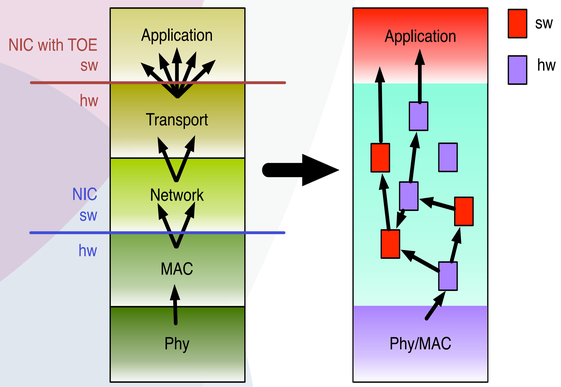
Hardware Support for Dynamic Protocol Stacks

ePICS
Engineering Proprioception in Computing Systems

From Static Systems to Dynamic Systems

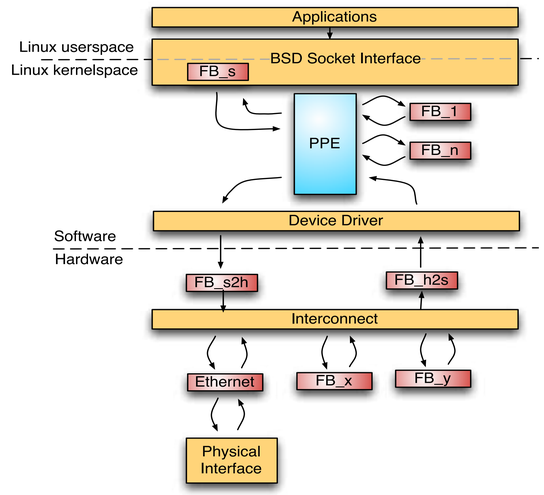
The tremendous success of the Internet can be attributed to the diversity of supported physical transport media, which allows users to be connected always and everywhere; and its plethora of applications that offers something for anyone. With the addition of more and more features to the original Internet architecture (such as firewalls, VPNs, NATs, P2P, etc.) the question arises whether the Internet architecture ought to be redesigned from scratch. Several research initiatives supported work in the area of clean slate architectures. In the context of such an initiative, we suggested in earlier work to split network functionality into individual functional blocks (FBs) that can be assembled into optimized protocol stacks at runtime.





Architecture

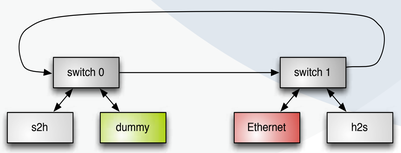
The overall EmbedNet architecture is depicted in the figure. Packets are sent from an application through a BSD socket to the Linux kernel. Here, packets are forwarded between the individual FBs by a packet processing engine. Each FB has an associated flag that determines whether the FB is currently executed in hardware or in software. If the FB should be executed in hardware the PPE copies it to the memory region that is shared with the hardware and notifies the hardware of the new packet. The hardware then reads the packet from the shared memory and forwards it to the next FB. From there it can be either sent to another FB in hardware, software or to the Ethernet interface. In hardware packets are forwarded in a network on chip (NoC) consisting of switches arranged in a unidirectional ring. Each switch is also connected to a configurable number of functional blocks. A software controller configures the FBs with the addresses of the other FBs so that packets can be forwarded correctly.

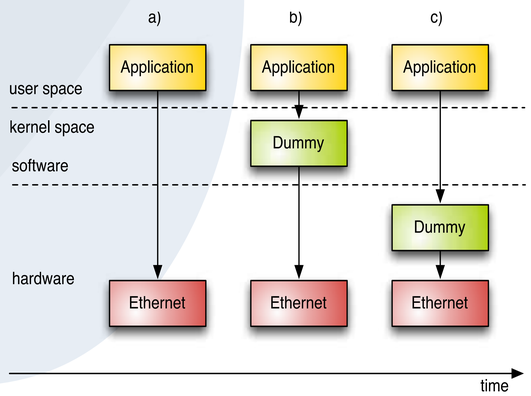


Evaluation

Platform
- Xilinx Virtex-6 ML605 evaluation board
- Microblaze soft-core CPU

Hardware Setup
- Four FBs: Ethernet, hw2sw, sw2hw, dummy
- Two switches



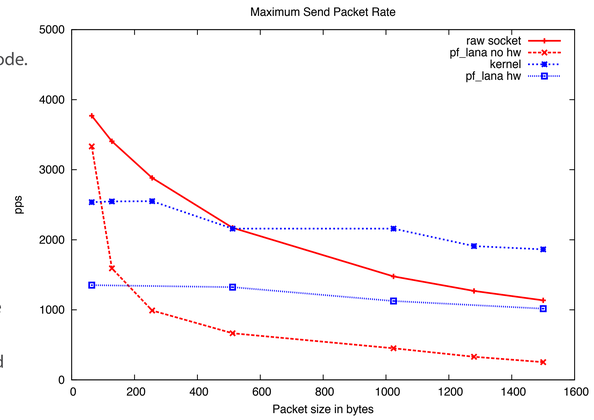


Results

Functional evaluation (Figure on the left):
a) An application sends packets through the Ethernet FB to another node.
b) We add a software FB while the application continues to send data.
c) We move this FB to the hardware.

Performance evaluation (maximum packet sending rate):
- Baseline: 4'000 min. sized packets/s from a Linux raw socket
- 3'000 min.sized packets/s from our own Linux socket (no hw)
- 2'500 packets/s from kernel space to our hw
- 1'500 packets/s from users pace to our hw
- 0.8 Gbit/s in hw only

The limiting factor is the handshake required to transmit data over the software/hardware boundary. Therefore, we plan to implement a ring buffer in which several packets can be transmitted at once. This should increase the performance by a factor corresponding to the number of packets in this buffer.



Authors and Acknowledgements

Ariane Keller: ariane.keller@tik.ee.ethz.ch
Daniel Borkmann: dborkma@tik.ee.ethz.ch
Stephan Neuhaus: stephan.neuhaus@tik.ee.ethz.ch

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement no. 257906.
www.epics-project.eu

SEVENTH FRAMEWORK PROGRAMME

ANCS 2012 - ACM/IEEE Symposium on Architectures for Networking and Communications Systems