An Approach Towards a Flexible Network Architecture

Bernd Reuther, Abbas Siddiqui, Dennis Schwerdel, Paul Müller
University of Kaiserslautern
{reuther, siddiqui, schwerdel, pmueller}@cs.uni-kl.de

The development of the Internet during the last years has shown that it is difficult to integrate new functionality. Especially the core mechanisms (TCP/IP) are hard to change. This is not a problem of a specific protocol, but of the architecture itself. We consider that tight coupling between several functionalities is the main hindrance for introducing new functionality. To overcome these problems we propose a new flexible network architecture.

Like several other approaches (e.g. RBA, RNA, SILO, ANA, 4WARD) we propose that functionality should be provided by *building blocks* (BB), which implement a micro-protocol (e.g. flow-control or retransmission) or any other service (e.g. monitoring or lookup services). In order to achieve a high degree of flexibility building blocks should be independent of each other. This means a BB must not use direct references to any other BB, but must have a well defined interface. If two BBs are micro-protocols, then their control data should be independent messages, so that both BBs can be used individually. This independence simplifies to add, exchange and remove BBs and fosters arbitrary combinations of BBs, limited only by semantic constraints.

Building blocks must interact with each other to provide more complex functionality. This interaction is described by a protocol graph (or workflow) for each data flow (e.g an end-to-end connection). Such a workflow contains information about: used building blocks, their processing sequence, possible processing branches (e.g. processing may change after an error), data exchange between BBs (a routing BB will require addresses for input and will provide outgoing ports as output) and possible relations among BBs (e.g. a BB may calculate a CRC on data of other BBs). Figure 1 illustrates the principles of workflow processing.

The first goal of our approach is to build a framework for managing BB and processing workflows within each node. The functionality of the framework covers: a repository for BB and their description, handling links for message exchange with other frameworks or applications, support session management of micro-protocols and to provide common functionalities like timers and shared data structures.

The second goal is to generate workflows automatically. This way BBs can be used as soon as they are available, independent of predefined specifications. The automatic generation of workflows also supports handling the huge numbers of possible protocols combinations (e.g. combinations of several video codecs, audio codecs, flow-control mechanisms and address types). At the sender side a workflow will be defined on the basis of application requirements, network constraints and descriptions of available BBs. In order to determine which BBs are available a sender may take into account which BBs are available within the network and at the receiver side, for example by performing negotiations. The workflow description will be signaled from the sender to the receiver side and if necessary also to intermediate nodes. This signaling may be encoded within a stream of messages (in-band) or may be signaled separately during connection setup (out-of-band). Connection less communication requires the in-band approach.

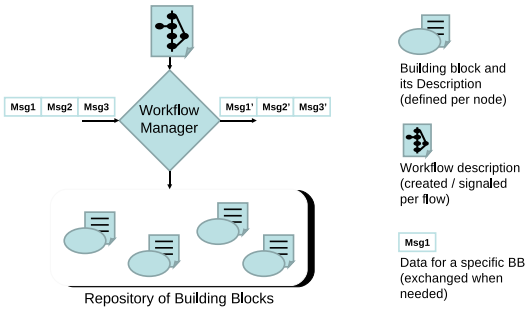

Figure 1. Workflow processing.

In order to achieve these goals we currently investigate interactions of BBs, because these must be specified by workflow descriptions. Further mechanisms for efficient signaling of workflows must be determined, especially for in-band signaling. These results will be evaluated by prototypes utilizing the G-Lab testbed. In order to evaluate the effort for managing such a flexible architecture, complex test scenarios will be defined and tested.