



<http://www.icsy.de>

G-Lab

Network Architecture

Approach & Current Work TU Kaiserslautern

B. Reuther, D. Schwerdel, A. Siddiqui

Technische Universität Kaiserslautern
AG Integrierte Kommunikationssysteme



Problem statement

Problem:

*It is hard to integrate new mechanisms
into the current Internet*

Cause:

- tight coupling
- The problem is not limited to specific protocols or mechanisms

➤ **It is an architectural issue!**

Goal: short and long term Flexibility

- Goal: Enable adaptivity according to demands and conditions (short term flexibility)
- Goal: Enable extendability to add, change, and remove protocols (long term flexibility)
 - Scope: everything between transport technologies (typical layer-2) and applications

- Approach: apply principles of SOA
 - Every node consists of loosely coupled services
 - Services are self-contained
 - Services have well-defined interfaces
 - A service is an abstract representation of a micro-protocol or “software-module”



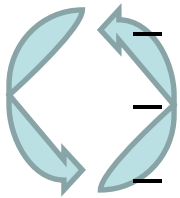
Planned Work

- Protocol Framework (Step 1)
 - In order to manage protocols and their interaction a framework must provide several functionalities
- Protocol Selection and Composition (Step 2)
 - To create language for description of protocols besides selection and composition based on provided description
- Protocol Heterogeneity (Step 3)
 - To handle a network of nodes with heterogeneous protocol's pool



Step 1 (Protocol Framework)

- Step1: Provide a protocol framework (within a single node!)
 - enabling, dynamic use and easy (ex)change of micro-protocols.
 - must control a workflow, i.e. controlling when a protocol is active
 - must provide means for data exchange between different protocols within a node.
 - Currently our main focus
- Work in progress
 - Requirement analysis (What is needed to manage protocols?)
 - Define mechanisms fulfilling requirements
 - Verification: are we able to rebuild TCP/IP workflows
 - Looking ahead at further steps



▶▶▶ Step 2 (Protocols Selection & Composition)

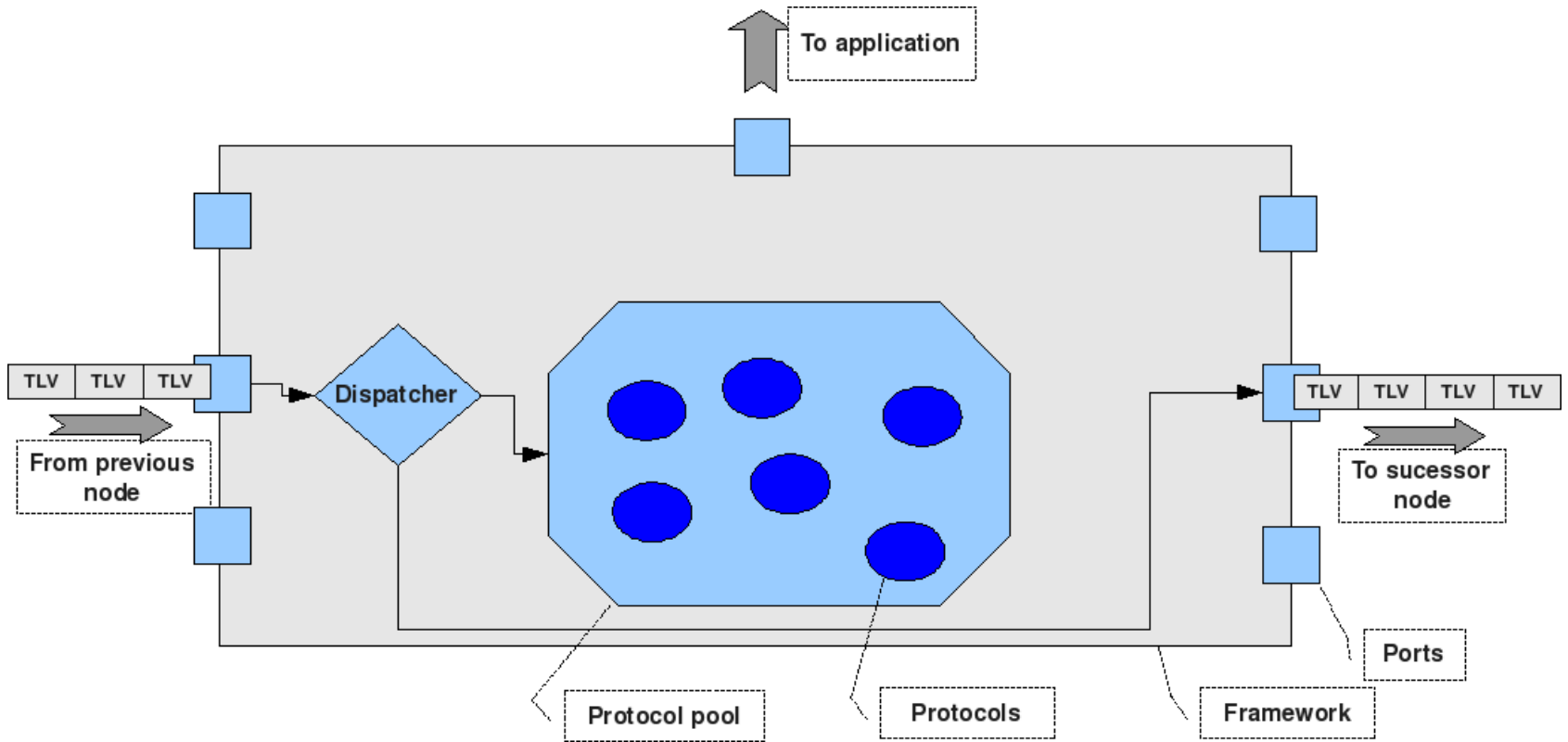
- Step 2: Automatic composition of micro protocols
 - How to fulfill demands under given conditions?
 - The result of such a composition is a workflow
 - Workflows should not be fixed (i.e. not hard coded into some software) but generated on demand.
- Planned
 - Define a micro-protocol description language
 - Verification of DL: To describe TCP/IP functionality
 - To use meta-data to select and compose micro protocols (services)?
 - Resolve conflicts
 - Resolve missing dependencies
 - To provide a SO (Service-Oriented) API to select services based on features

Step 3 (Heterogeneity among Protocols)

- Step 3: Handling of heterogeneity at runtime.
 - In large scale networks it is inevitable that different nodes will support different sets of protocols.
 - Impact on initial definition of workflows
 - Workflows may be adapted by intermediate nodes (e.g. adding, removing or replacing protocols to/from the workflow).
- Other Issues
 - Evolution concepts on a per protocol basis?
 - How to find a path of nodes with similar protocols?
 - Protocol translation in intermediate nodes?
 - Dynamic protocol plug-ins? Security!



Protocol Framework



<http://www.icsy.de>

Communication between Frameworks

- Messages: Protocol on different nodes exchange messages
 - Loosely coupled protocols → data of different protocols must be independent
 - **Mechanism**: TLV, self descriptive because of “TL”
- Group of messages: message may be related to each other
 - **Example**: address, TTL, flow control and payload
 - **Mechanism**: separate groups by markers (e.g. a specific message)
- Flow: a logical stream of messages which are semantically related
 - **Example**: a connection, a video stream
 - **Mechanism**: identified by flow-ids (maybe signaled between two adjacent nodes)
- Connection less behavior: Group of messages == Flow

Framework Requirements I

- Generic requirement:
 - Enable interaction between micro-protocols, so that no protocol needs knowledge about any other protocol!
- Assign a message to a micro-protocol for further processing
 - Each micro-protocol registers for one (or more?) message it is responsible for.
 - The message is identified by its type (T)
 - **Mechanism:** dispatcher and a protocol registry
- Forward messages returned by protocols
 - Each micro-protocol may return one or more messages to forward
 - **Mechanism:** dispatcher (and an outgoing queue)



Framework Requirements II

- Share well-defined data
 - Well-defined structure and semantic
 - **Example:** a routing decision service, Input: a specific address, Output: an outgoing port
 - **Mechanism:** Tuple-Space
- Support management of protocol status
 - Some protocols require initialization and release phase.
 - Micro-Protocols should not implement their own “state management”
 - **Mechanism:** maybe Tuple-Space also?
- Distribute notifications
 - **Example:** signal an error so others may take action
 - **Mechanism:** a publish/subscribe notification handler and dispatcher
- Timers
 - **Mechanism:** delayed notifications



Framework Requirements III

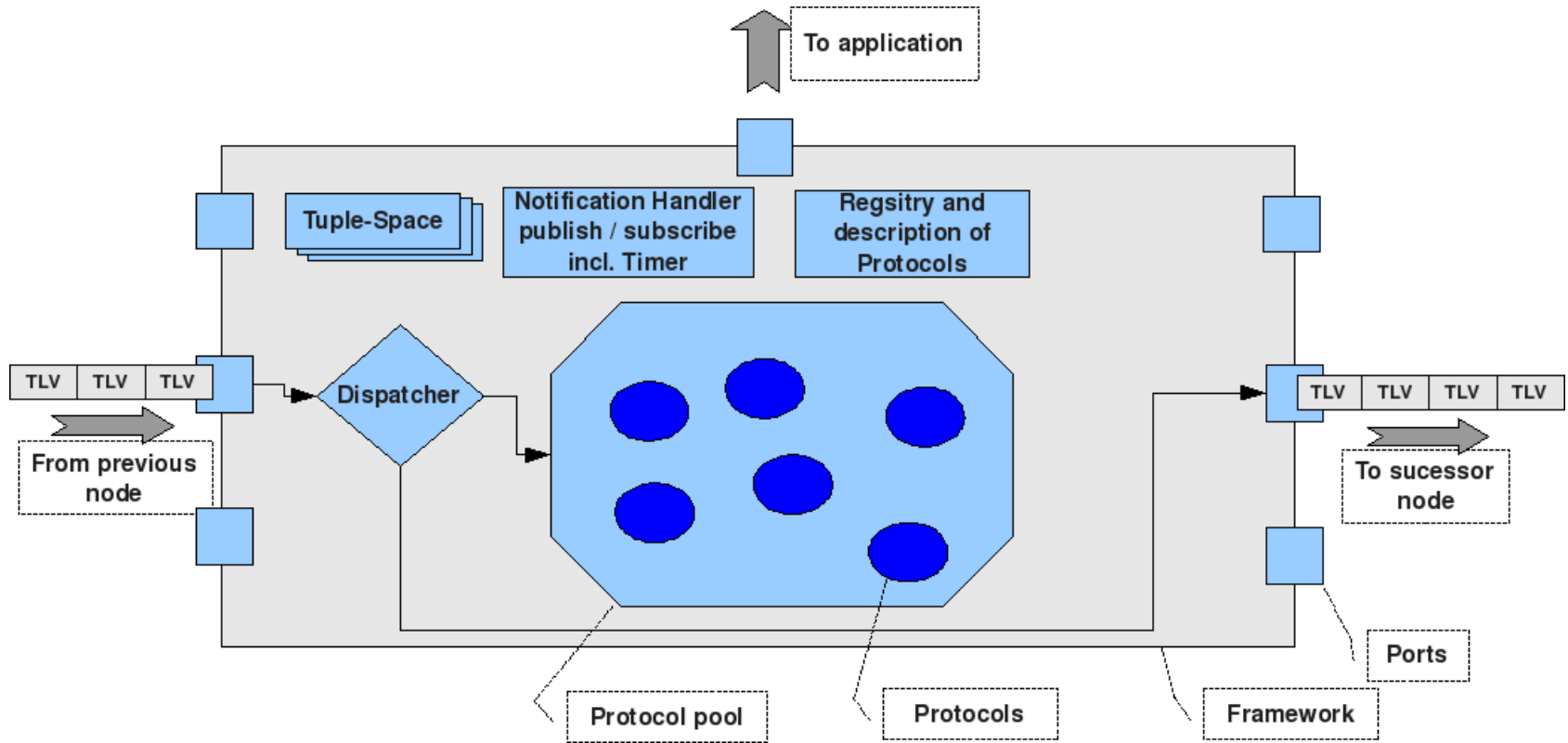
- Provide access to other messages
 - Each service specifies which other messages it want to access
 - **Example**: CRC or Compression may read or read&write other messages
 - **Mechanism**: requires support of dispatcher and references maybe encoded individually (several options discussed)
- Process messages provided by protocol instances
 - **Example**: decryption may deliver several messages for further processing
 - **Mechanism**: requires support of the dispatcher

Framework Requirements IV

- Handle role of a node
 - **Examples:** sender, receiver, intermediate node, gateway, ...
 - Configured or determined by protocols
 - Impact on which protocol will be processed
 - **Mechanisms:** node configuration and value in Tuple-Space
- Handle processing mode
 - **Example:** distinguish “error” and “no error”
 - Determined by protocols, e.g. CRC
 - Impact on which protocol will be processed
 - **Mechanism:** mode of dispatcher
- Handle constraints
 - **Example:** inhibit resource reservation without authentication
 - Configured
 - Impact on which protocol will be processed or if any process of a group will be processes
 - **Mechanisms:** consistency check by dispatcher?



Protocol Framework



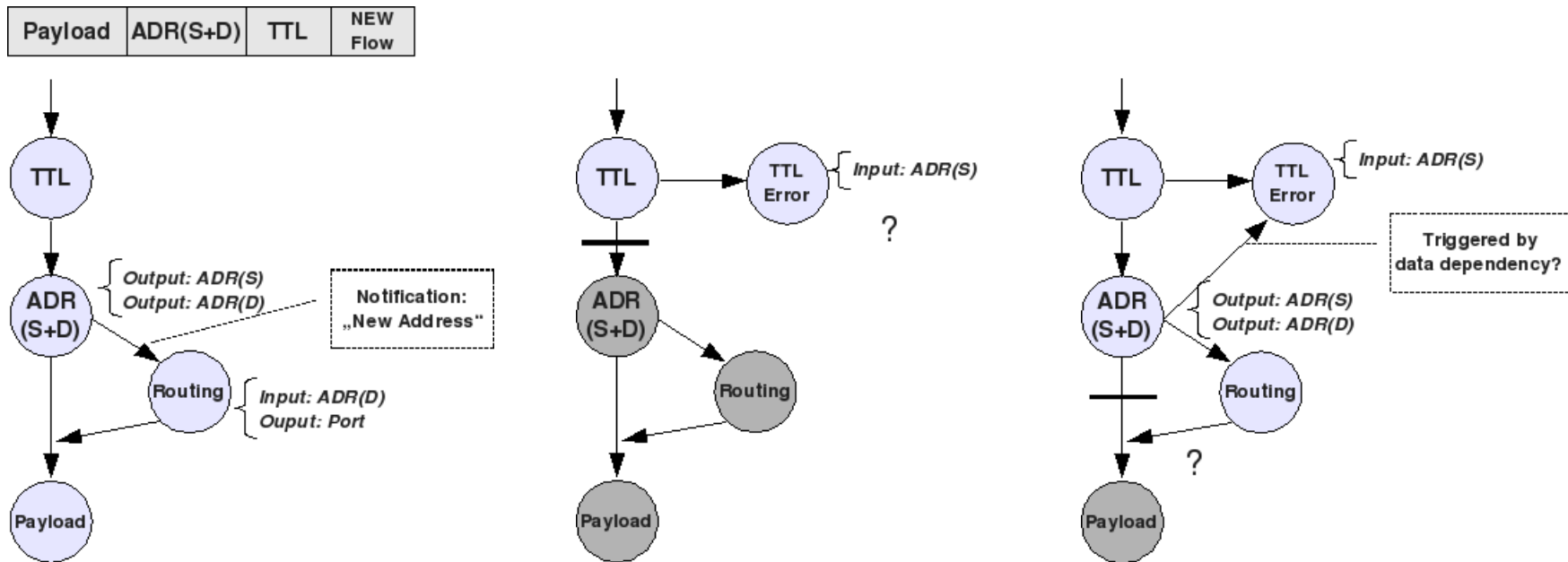
<http://www.icsy.de>



<http://www.icsy.de>



Problem with workflows



Task: find all types of dependencies between micro-protocols considering at least the TCP/IP suite, in order to verify if our approach is sufficient



Interface of services

Input:

- TLV (message or notification), that has triggered the service
- Tuple Space of current flow
- data read or to be read from Tuple Space
- other TLVs to be processed (read only or write)
(after receiving TLV a service may inform the dispatcher about other TLVs needed)

Output:

- data written or to be written Tuple Space
- possibly modified or created TLVs for forwarding
- possibly modified TLVs for local processing (e.g. after decompression)
- success or error indication



Impact on Workflow

<http://www.icsy.de>