An Architecture for Concurrent Future Networks

Future Internet 2008
Karlsruhe, 11.11.2008

Lars Völker, Denis Martin, Ibtissam El Khayat,
Christoph Werle, Martina Zitterbart
(Universität Karlsruhe (TH) / Ericsson GmbH)

Motivation

- 4WARD-Project: „Let 1000 networks bloom“
  - Network Virtualization
  - Vast amount of (virtual) networks
  - User might have multiple Networks side-by-side

- How can we do this?
  - How can we run multiple networks?
  - What must a node look like?
  - How do we connect an application to the correct network?
  - What about Security, QoS, and Mobility?
  - Rapid creation of such networks?
Definitions

❖ Just to make life easier for us …

❖ **Network Architecture** – A common understanding within a network. Usually involves common protocols as well as naming and addressing.

❖ **Network Architect** – Designer of networks and/or network architectures.

Structure of this talk

❖ Motivation

❖ Definitions

❖ Node Architecture at a glance

❖ Important Concepts of the Node Architecture

❖ Node Architecture put together

❖ Rapid creation with the Design Process

❖ Conclusion and Outlook
Node Architecture at a glance

- Simplified version
- Netlet as Protocol Container
- Architecture-specific Multiplexer
- Network Access
  - Virtual or physical Networks
  - Network Access Manager

The Netlet – A container for Future Internet protocols

- Netlet (≈ Protocol Stack)
  - Is usually based on one Network Architecture
    - Fits usually just one Network Architecture
    - NAI has to be compatible to Network Architecture
  - Could be build many different ways:
    - Writing Code, Code Generation, 3rd party, Composition, …
  - “Interop Netlet” connects multiple Network Architectures
Application Interface –
Moving forward from the Socket API

- Today’s interfaces to the applications (e.g. Socket API) have drawbacks
  - Peer is described by address and not by name
    - Usually simple name resolution rule (e.g. IPv6 first, IPv4 second)
    - With 1000 networks, choice of the network important!
  - Applications might have requirements on communication!
    - Please protect this connection

- Needed changes:
  - Move name resolver from application to system
  - Allow application to influence choice of network with requirements

Network Access Interface –
Transparently support Network Virtualization

- With 1000 networks, many of them will be virtual networks!
  - That means: Virtual Networks won’t be special
  - Possibly the common case

- The Network Access Interface
  - Hides the differences of physical and virtual networks
  - Supports the description of the underlying network
    - e.g., latency, energy consumption, cost
    - Triggers of network events
    - So the selection of Netlets could also be based on the network properties
Automatic Selection – Choosing the best Netlet

- With “1000 networks” the user cannot just manually choose the network!

- Idea:
  - Let application, user, and administrator describe requirements/goals
  - Description of underlying network
  - “Estimate” the Netlet’s behavior
  - Rank the Netlets based on this

- This is based on Multi Attribute Utility Theory (MAUT) and [1]


Criteria

- To determine what’s best we need to describe Netlets!
- We looked at several criteria, here some examples
Calculating total value

How to aggregate the criteria to overall utility:

\[ v(a_i) = \sum_{j=1}^{m} w_j \cdot v_j(c_j(a_i)) \]

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative ( i )</td>
<td>( a_i )</td>
</tr>
<tr>
<td>Criterion ( j )</td>
<td>( c_j )</td>
</tr>
<tr>
<td>Weight for criterion ( j )</td>
<td>( w_j )</td>
</tr>
<tr>
<td>Attribute value of ( a_i ) with regard to criterion ( c_j )</td>
<td>( c_j(a_i) )</td>
</tr>
<tr>
<td>Value function for criterion ( c_j )</td>
<td>( v_j )</td>
</tr>
<tr>
<td>Total value of alternative ( a_i )</td>
<td>( v(a_i) )</td>
</tr>
</tbody>
</table>

Value Functions

- Function to “translate” the individual criteria’s values to a (generic) utility
- This function has to fit to the criterion
  - Each criterion could have a specific value function
  - It should be adaptable

[Graph: A possible Value Function for Latency]
Determine Effective Bit Strength (EBS) of involved cryptographic primitives:
- Authentication, Key Exchange, Encryption, Message Authentication
- Aggregate those EBS values using the Min-Function
- EBS → Utility Value
  - Value is not linear for most users

Influencing the Decision Process

- Value functions
  - Can be replaced and/or adjusted
- Weights
  - Put criteria into proportion
- Requirements
  - Describe which values are ok for criterion
  - Works also for qualitative criteria
Design Process –
Accelerating the Creation of Netlets

- Iterative process aiding the future network architect to design new networks
- The network architect will start with his own requirements and will refine them during the process
- He will derive a Blue Print for the network architecture’s components and functionalities
- With this Blue Print
  - Components of the network architecture can be implemented (if not using standard components already existing on the market)
  - Network topologies can be designed by network administrators
Conclusion / Benefits

- Create a new network architecture if needed
  - Every virtual network can have their own
  - You could even have applications come with their own
  - Has to be a simple process!
    • The Design Process enables this

- Netlets can be treated as almost arbitrary black boxes
  - Approaches like RNA and SILO can run in a Netlet
  - We try to not introduce too many invariants
    • Allow for future development

- Choosing the Netlet and Network Architecture dynamically
Thank you for your attention

The Project Website: http://www.4ward-project.eu/