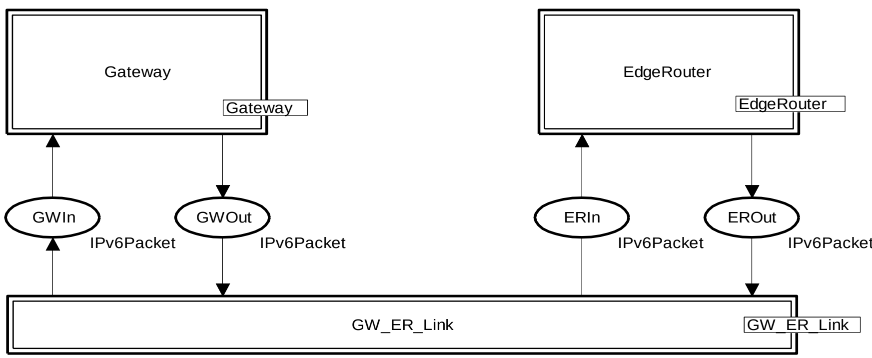
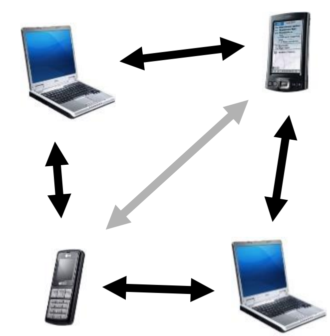
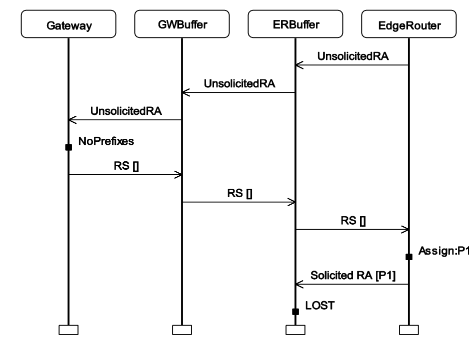
an ERICSSON company

Lecture 7
An Example of Industrial Application







Lars M. Kristensen
Department of Computing, Mathematics, and Physics
Western Norway University of Applied Sciences
Email: lmkr@hvl.no / WWW: home.hib.no/ansatte/lmkr

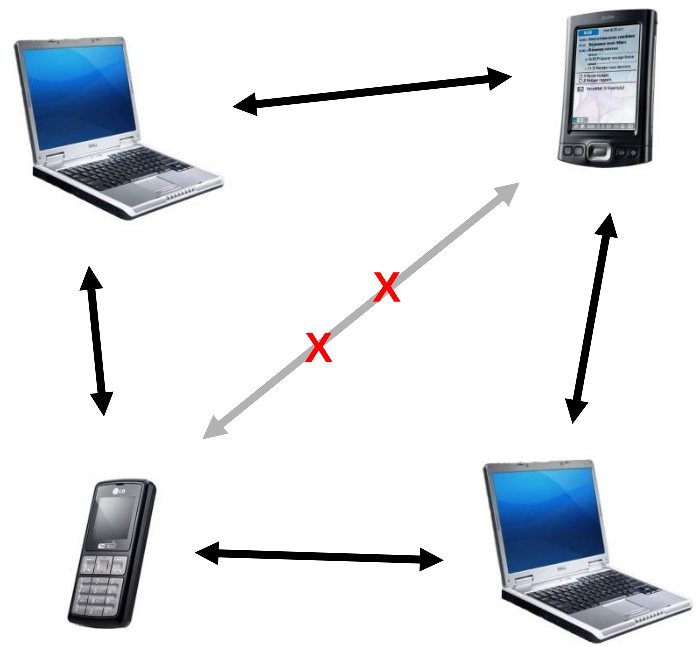
Protocol Design @ Ericsson Telebit

- **Design of an Edge Router Discovery Protocol (ERDP) for mobile ad-hoc networks**
  - a CPN model was constructed constituting a formal executable specification of the ERDP protocol
  - simulation and message sequence charts were used for initial investigations of the protocol behaviour
  - state spaces were applied to conduct a formal verification of key properties of ERDP

- **Modelling, simulation, and verification helped in identifying several design omissions and errors**
  - demonstrates the benefits of using formal modelling techniques in a protocol software design process

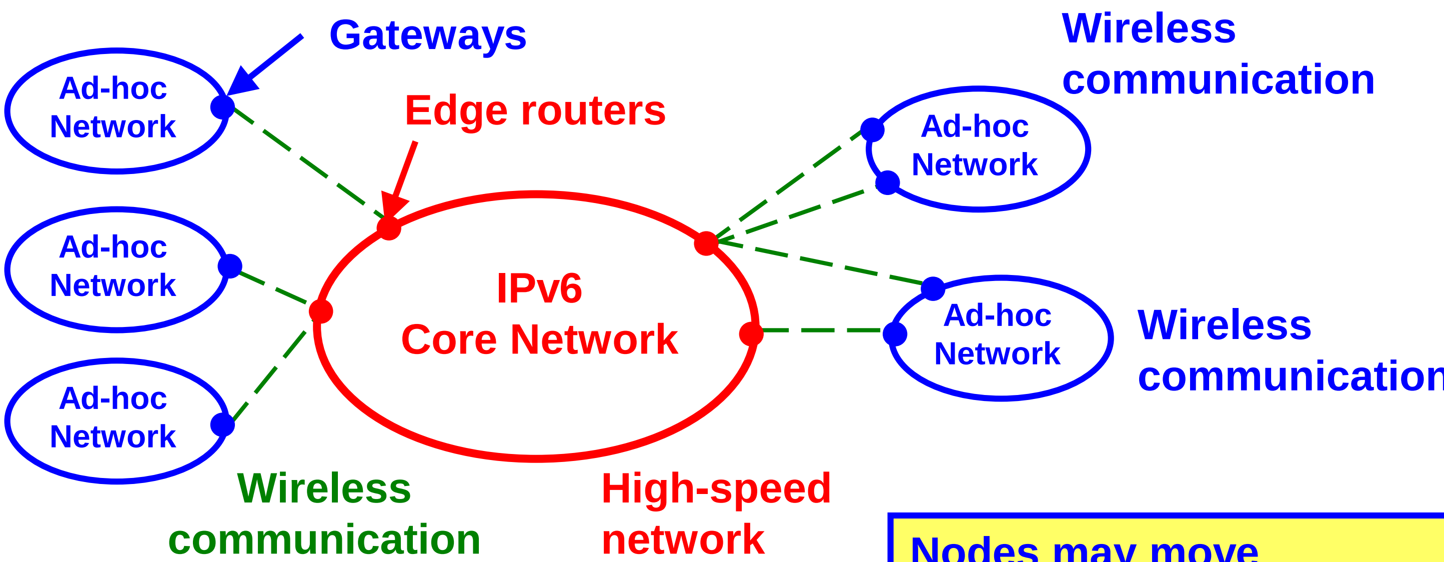
Mobile Ad-hoc Network

- **Collection of mobile nodes (devices)**
  - laptops, tablets, mobile phones, …
  - capable of establishing a communication infrastructure for their common use



- **The nodes operate autonomously**
  - in a fully self-configuring and distributed manner,
  - without any pre-existing communication infrastructure (such as designated base stations and routers).

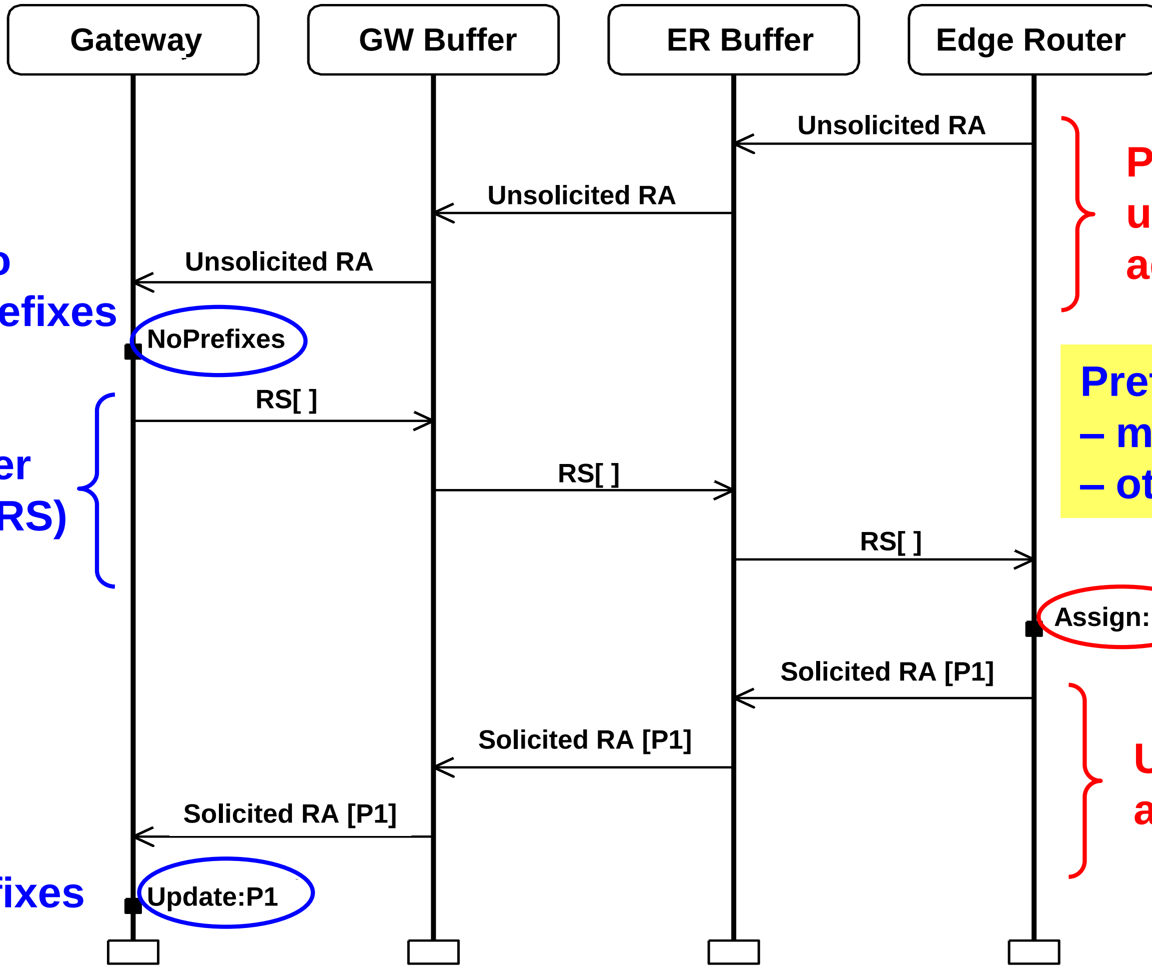
Network Architecture



Nodes may move
– within the ad-hoc networks
– from one ad-hoc network to another

- **ERDP supports**
  - gateways in discovering edge routers, and
  - edge routers in configuring gateways with a globally routable IPv6 address prefix.

Basic Gateway Configuration



Periodical multi-cast of unsolicited router advertisement (RA)

Currently no assigned prefixes

Prefixes have a limited lifetime
– must be refreshed
– otherwise they will expire

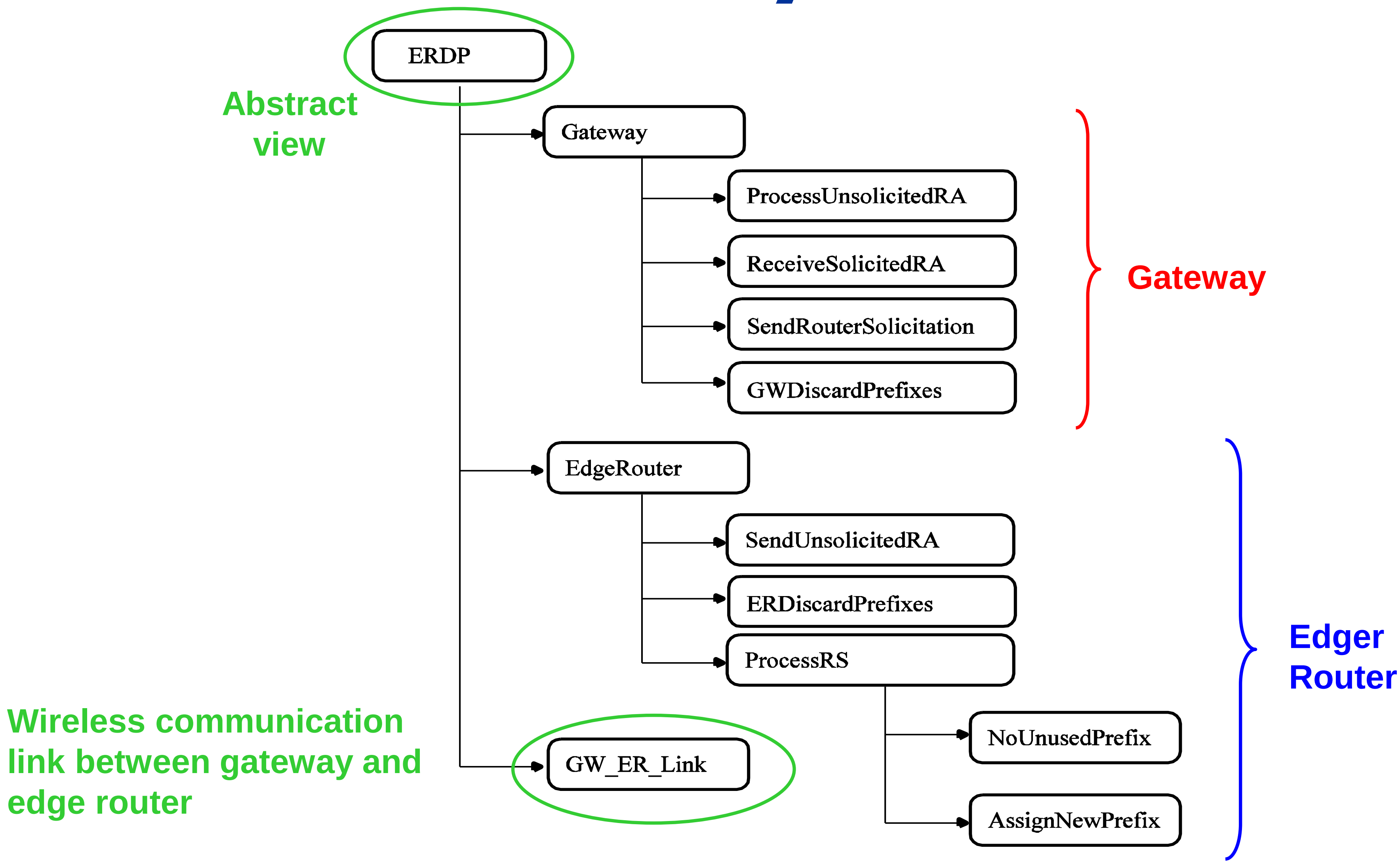
Unicast router solicitation (RS)

New prefix

Unicast solicited router advertisement (RA)

Update prefixes

Module Hierarchy



CPN Tools Demo

- **Walk-through of the ERDP CPN model**
  - Basic configuration scenario

Western Norway
University of
Applied Sciences

Development of CPN model

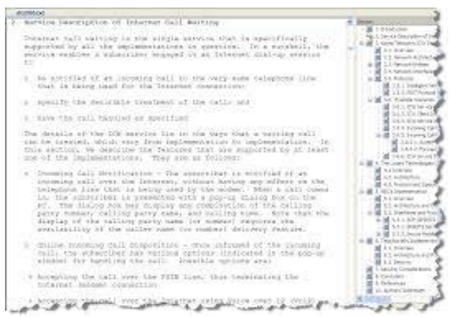
- **The CPN model was developed**
  - in cooperation with protocol engineers at Ericsson Telebit
  - in conjunction with the development of the ERDP specification
  - iteratively in three review rounds

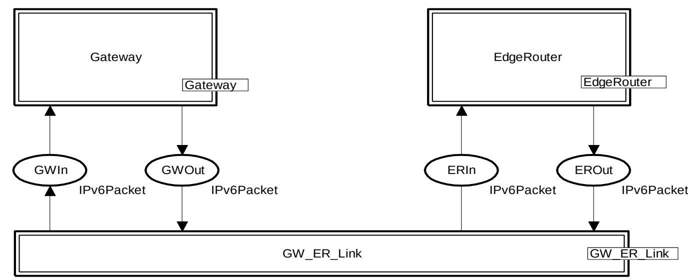
- **70 person-hours were spent on CPN modelling.**

- **Protocol developers were given a 6-hour course on the CPN modelling language**
  - enabled them to read and interpret the CPN models
  - used as basis for discussions of the protocol design

First Modelling Round

- **Development started with the creation of an initial ERDP specification (in natural language).**
- **A first version of the CPN model was created**





- **While creating the initial CPN model and discussing it, the engineers identified**
  - Several design errors
  - incomplete aspects and ambiguities in the specification
  - ideas for simplifications and improvements of the design

Second and Third Rounds

- The ERDP specification and the CPN model were revised and extended based on round 1.
- Round 2 identified a number of new issues to be resolved.

- Once more, the ERDP specification and the CPN model were revised and extended.
- In round 3, no further problems were discovered.

Design Problems Identified

- **A number of issues were identified during**
  - construction of the CPN model
  - interactive and automatic simulation of the CPN model
  - discussions of the CPN model among the project group members

Category	Round 1	Round 2	Total
Errors in protocol specification/operation	2	7	9
Incompleteness and ambiguity in specification	3	6	9
Simplifications of protocol operation	2	0	2
Additions to the protocol operation	4	0	4
Total	11	13	24

Integrating CPN Technology

- **We used an iterative process involving**
  - a conventional natural language specification
  - a formal and executable CPN model
  - message sequence charts (MSCs) integrated with simulation was to investigate the detailed behaviour of ERDP
  - presenting the operation of the protocol in a form which was well-known to the protocol developers
- **Complementary descriptions are required**
  - the implementers of the protocol are unlikely to be familiar with CPNs
  - important parts of the ERDP specification are not reflected in the CPN model (such as the layout of packets)
- **Construction of CPN models was a thorough and systematic way to review the protocol design.**

State Spaces and Verification

- **State space analysis was pursued after the three iterations of modelling.**
- **The purpose was to conduct**
  - An exhaustive investigation of the ERDP behaviour
  - Verification of its key properties
- **The first step was to obtain a finite state space**
  - The CPN model above can have an arbitrary number of tokens on the packet buffers
  - As an example, the edge router may send an arbitrary number of unsolicited router advertisements

Properties and Approach

- **The key property of ERDP is the proper configuration of the gateway with prefixes**
  1. For a given prefix and state where the gateway has not yet been configured with that prefix, the protocol must be able to configure the gateway with the prefix.
  2. The edge router and the gateway should be consistently configured, i.e., the assignment of a prefix must be recorded in both entities.

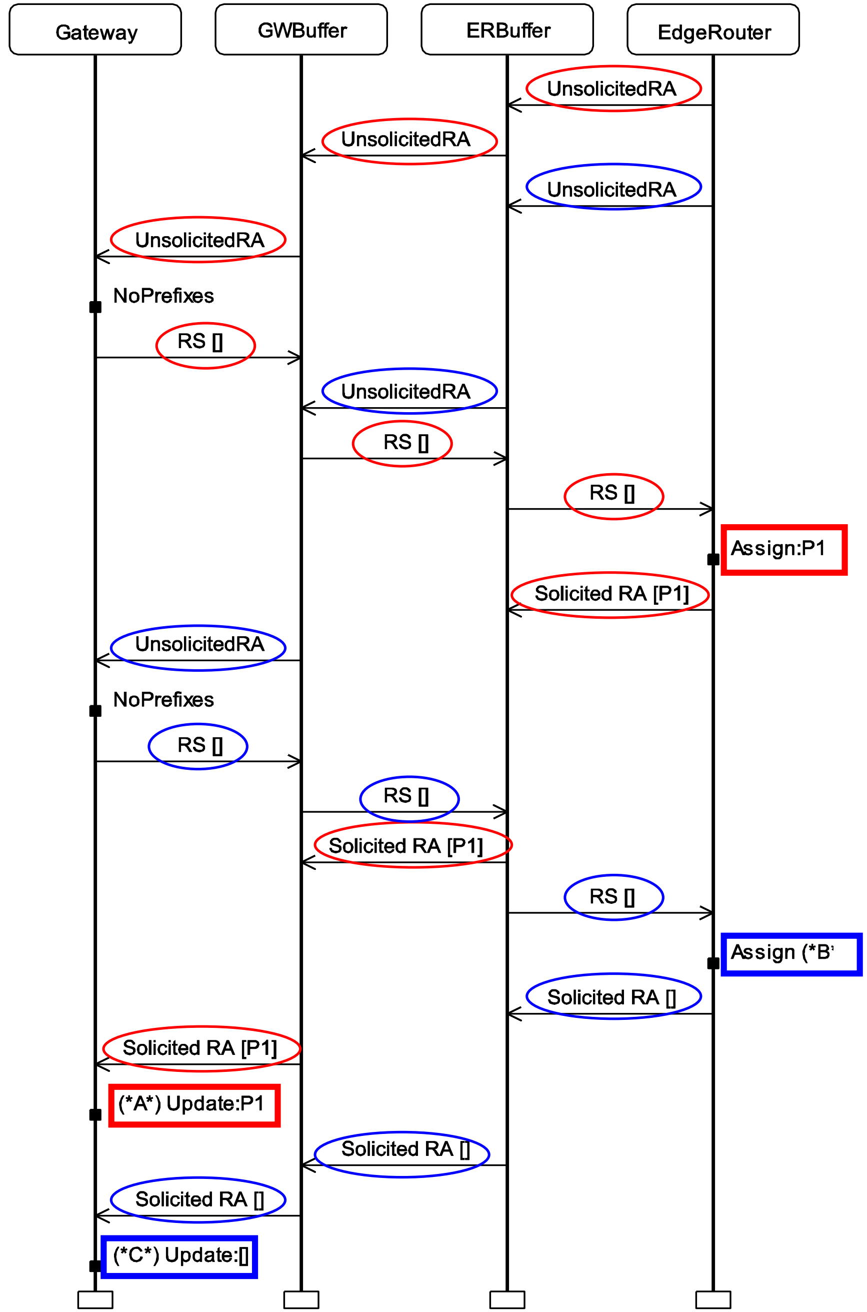
- **Verification approach**
  - Start the state space analysis from the simplest possible configuration and then gradually relax the assumptions.
  - As the assumptions are relaxed, the size of the state spaces grows.

1 prefix/no loss/no expiration

- **State space report**
  - State space: 46 nodes and 65 arcs
  - A single dead marking
- **Inspection showed that the dead marking represents an inconsistently configured state**
  - The edge router has assigned a prefix to the gateway
  - BUT, the gateway is not configured with the prefix
- **Query functions were used to obtain a shortest counter example (error-trace).**
- **The error-trace was visualised by means of a message sequence chart.**

MSC error-trace

- The **edge router** sends **two unsolicited RAs**.
- The first one gets through and we obtain a consistent configuration with prefix P1.
- When the second one reaches the **edge router** there are no unassigned prefixes available.
- A **Solicited RA** with an empty list of prefixes is sent.
- The **gateway** updates its prefixes to be the empty list.



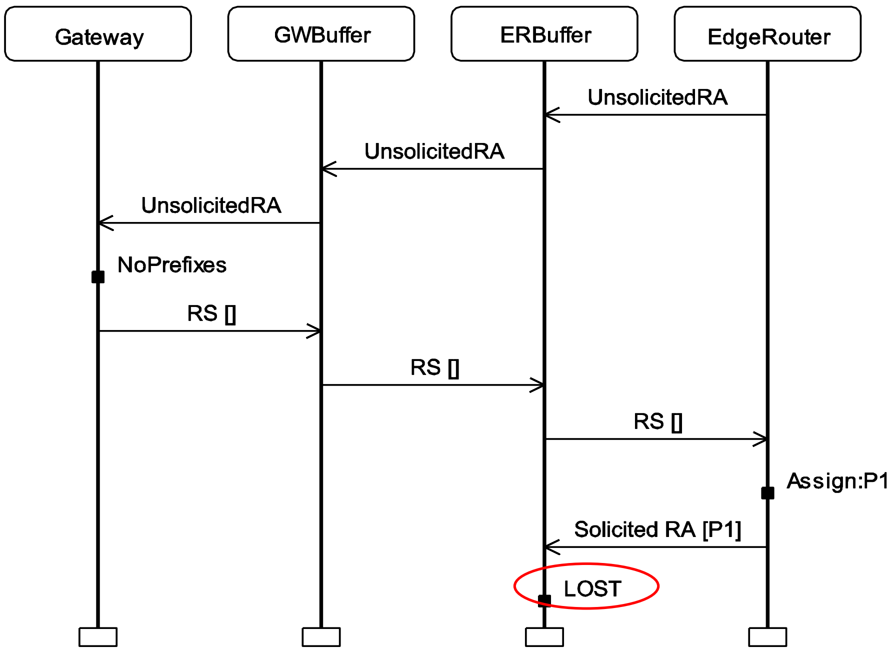
1 prefix/no loss/no expiration

- **Modification: the edge router replies with the list of all prefixes currently assigned to the gateway.**
- **State space report**
  - State space: 34 nodes and 49 arcs
  - No dead markings and 11 home markings
- **All 11 home markings represent consistently configured states**
  - it is always possible to reach a consistently configured state for the prefix
  - when such a state has been reached, the protocol entities will remain consistently configured (one terminal SCC)
- **The terminal SCC was the only non-trivial SCC**
  - A consistently configured state will eventually be reached

1 prefix/loss/no expiration

- **The next step was to allow packet loss on the wireless link between edge routers and gateway.**
- **State space report**
  - State space: 40 nodes and 81 arcs
  - SCC-graph: 36 nodes and 48 arcs
  - A single dead marking
- **The dead marking represented an undesired terminal state with inconsistent configuration.**
- **To locate the problem, an error trace was visualised by means of a message sequence chart.**

MSC error trace



- The solicited RA containing the prefix is lost.
- The edge router has assigned its last prefix and is no longer sending any unsolicited RAs.
- There are no timeouts to trigger retransmission of the prefix to the gateway.

- **The problem was fixed by ensuring that**
  - the edge router will resend an solicited RA to the gateway as long as it has prefixes assigned to the gateway

1 prefix/loss/no expiration

- **State space report**
  - State space: 68 nodes and 160 arcs
  - No dead markings and no home markings
- **Two terminal SCCs each containing 20 markings**
  - in one of them, all markings are consistently configured
  - in the other, all markings are inconsistently configured
- **An error trace was obtained, the protocol design was revised, and a new state space produced**
  - this time there was only one terminal SCC (containing 20 consistently configured states)
  - if only finitely many packets are lost, a consistently configured state will eventually be reached

1 prefix/loss/expiration

- **State space report**
  - State space: 173 nodes and 513 arcs
  - A single dead marking and a single home marking
- **In the dead marking**
  - the edge router has no further prefixes to distribute and no prefixes recorded for the gateway
  - the gateway is not configured with any prefix
  - expected as prefixes will eventually expire in the edge router
- **The dead marking was also a home marking**
  - The protocol can always enter the expected terminal state
- **If a prefix still is available, it is possible to reach a consistently configured state for the prefix.**

State Space Statistics

|P|	No loss/No expire		Loss/No expire		Loss/Expire	
1	34	49	68	160	173	531
2	72	121	172	425	714	2,404
3	110	193	337	851	2,147	7,562
4	148	265	582	1,489	5,390	19,516
5	186	337	926	2,390	11,907	43,976
6	224	409	1,388	3,605	23,905	89,654
7	262	481	1,987	5,185	44,450	169,169
8	300	553	2,742	7,181	78,211	300,072
9	338	625	3,672	9,644	130,732	505,992
10	376	697	4,796	12,625	209,732	817,903

State Spaces Cover All Cases

- **The inconsistent configurations would probably not have been discovered until a first implementation of ERDP was operational**
  - to discover these problems you need to consider subtle execution sequences of the protocol
  - there are too many of these to do it manually
- **The state space analysis covers all execution sequences in a systematic way**
  - For the ERDP protocol we did <u>not</u> encounter state explosion
  - The key properties could be verified for the number of prefixes that are envisioned to appear in practice

Main Conclusions

- **The application of CPN technology in the development of ERDP was successful**

  1. The CPN modelling language and computer tools were powerful enough to handle a real-world communication protocol and could easily be integrated in the conventional protocol development process
  2. Modelling, simulation and state space analysis identified several non-trivial design problems which otherwise might not have been discovered until implementation, testing, and possibly deployment
  3. Only 100 person-hours were used for CPN modelling and analysis. This is a relatively small investment compared to the many problems that were identified and resolved early in the development

Hands-on Overview

- **Hands-on 1: Navigate and simulate CPN models**
- **Hands-on 2: Editing of CPN models**
- **Hands-on 3: Building a controller PT-net model**
- **Hands-on 4: Building a controller CPN model**

Petri Nets Course Assignment – Coloured Petri Nets Component:
https://github.com/lmkr/cpncourse/blob/master/assignment/assignment.md