

RObust Header Compression WG (ROHC)

**55th IETF
Atlanta, November 18, 2002**

Chairs:

Carsten Bormann <cabo@tzi.org>

Lars-Erik Jonsson <lars-erik.jonsson@ericsson.com>

Mailing List:

rohcn@ietf.org

55th IETF: Pre-Agenda

- **WG chair admonishments**
- **Real agenda**

- ✓ Blue sheets
- ✓ Scribe(s)

Hello! This is an IETF Working Group

- **We are here to make the Internet work (Fred Baker)**
 - Together! (Harald Alvestrand)
- **Rough Consensus and Running Code (Dave Clark)**
- **Working Group is controlled by**
 - IETF Process (RFC2026, RFC2418) – *read it!*
 - Area Directors (ADs): Alison Mankin, Scott Bradner
 - Charter (<http://www.ietf.org/html.charters/rohc-charter.html>)
 - Working Group Chairs: Lars-Erik Jonsson, Carsten Bormann
 - Technical Advisor: Erik Nordmark
- **Work is done on email list: rohc@ietf.org**
 - And on IETF meetings, interim meetings, informal meetings
 - Mailing list is official channel, though

Purpose of WG meetings

- **We assume people have read the drafts**
- **Meetings serve to advance difficult issues by making good use of face-to-face communications**
- **No technical presentations, but lead-ins to stimulate/elicit discussion**
- **Objective: generate technical consensus**

RFC 2026: Internet Standards Process

- **Standards track RFCs:**
 - **WG consensus (as judged by WG chairs)**
 - **WG last call**
 - **IETF last call**
 - **IESG approval (based on AD recommendation)**
 - Quality control!
 - **Publication by the RFC Editor**
- **Informational RFCs**
- **BCP (best current practice) RFCs**

RFC 2026: IPR issues (1)

- **(10.2) No contribution that is subject to any requirement of confidentiality or any restriction on its dissemination may be considered [...]**
- **Where the IESG knows of rights or claimed rights [...] the IETF Executive Director shall attempt to obtain from the claimant [...] a written assurance that upon approval by the IESG of the relevant Internet standards track specification(s), any party will be able to obtain the right to implement, use and distribute the technology [...] based upon the specific specification(s) under **openly specified, reasonable, non-discriminatory** terms.**

RFC 2026: IPR issues (2)

- **Contributions (10.3.1(6)):**
“The contributor represents that he has disclosed the existence of any proprietary or intellectual property rights in the contribution that are reasonably and personally known to the contributor.”
- **I.e., if you know of a patent application for a technology you are contributing, you have to tell.**
Or just shut up entirely!

55th IETF: ROHC WG Agenda, 1(3)

09:00 - Chair admonishments and agenda Bormann (10)

09:10 - WG and document status update Jonsson (10)

09:20 - ROHC RTP, Draft Standard preparations update

09:20 - MIB & ROHC terminology and mapping Jonsson (5)

09:25 - Implementation status and test list Jonsson (5)

09:30 - Implementer's Guide Jonsson (5)

09:35 - "IP-only" profile, purpose and issues Jonsson (10)

55th IETF: ROHC WG Agenda, 2(3)

09:45 - Signaling compression

- | | |
|--|-----------------------|
| 09:45 - User guide, torture-testing and interop | Price (15) |
| 10:00 - SigComp Feedback | Roach/Liu (40) |
| 10:40 - Static Dictionary for SIP/SDP | Bormann (5) |

55th IETF: ROHC WG Agenda, 3(3)

10:45 - ROHC over Wireless Ethernet Media

10:45 - Problem, alternatives and proposal Fleming (10)

10:55 - Issues and discussion Bormann (10)

11:05 - TCP profile

11:05 - The editorial update from 02 to 03 Pelletier (5)

11:10 - Some aspects of the TCP profile Pelletier (5)

11:15 - Next steps Pelletier (5)

11:20 - Formal HC notation West (10)

Document status update, 1(3)

- **Published:**

- **RFC 3095: Framework and four profiles** (was: draft-ietf-rohc-rtp-09.txt)
- **RFC 3096: RTP requirements** (was: draft-ietf-rohc-rtp-requirements-05.txt)
- **RFC 3241: ROHC over PPP** (was: draft-ietf-rohc-over-ppp-04.txt)
- **RFC 3242: LLA RTP** (was: draft-ietf-rohc-rtp-lla-03.txt)
- **RFC 3243: 0-byte RTP req's** (was: draft-ietf-rohc-rtp-0-byte-requirements-02.txt)
- **RFC 3408: LLA R-mode** (was: draft-ietf-rohc-rtp-lla-r-mode-03.txt)
- **RFC 3409: ROHC RTP LLG** (was: draft-ietf-rohc-rtp-lower-guidelines-03.txt)

- = **Currently in AUTH48**

Document status update, 2(3)

- **In RFC editor queue (passed IANA registration)**
 - **draft-ietf-rohc-sigcomp-07.txt**
 - **draft-ietf-rohc-sigcomp-extended-04.txt**
 - **draft-ietf-rohc-signaling-req-assump-06.txt**
- **Submitted to IESG:**
 - **NONE!**

Document status update, 3(3)

- **Current WG documents**
 - RTP/Framework – 5 drafts
 - TCP profile – 4 drafts
 - SCTP profile – 1 draft (requirements) **Work on hold**
 - Notation – 1 draft

WG Status, Goals and Milestones 1(3)

- ✓ I-D on Requirements for IP/UDP/RTP header compression.
- ✓ I-D of layer-2 design guidelines.
- ✓ I-D(s) proposing IP/UDP/RTP header compression schemes.
- ✓ I-D of Requirements for IP/TCP header compression.
- ✓ Requirements for IP/UDP/RTP header compression submitted to IESG for publication as Informational.
- ✓ Requirements for IP/TCP header compression submitted to IESG for publication as Informational.
- ✓ Resolve possibly multiple IP/UDP/RTP compression schemes into a single scheme.
- ✓ Submit I-D on IP/TCP header compression scheme.
- ✓ IP/UDP/RTP header compression scheme submitted to IESG for publication as Proposed Standard.
- ✓ Possible recharter of WG to develop additional compression schemes

✓ DONE

WG Status, Goals and Milestones 2(3)

- ✓ Jan 02 - Requirements and assumptions for signaling compression
- ✓ Jan 02 - Signaling compression scheme submitted to IESG for publication as Proposed Standard, including security approach for SIP compression usage.
- ✓ Jan 02 - Layer-2 design guidelines submitted to IESG for publication as Informational.
- ❑ **Nov 02 - I-Ds of ROHC IP/UDP/RTP bis, framework and profiles separated.**
- ❑ **Nov 02 - ROHC MIB submitted to IESG for publication as Proposed Standard.**
- **Dec 02 - LLA mapping examples submitted to IESG for publication as Informational.**
- **Feb 03 - Requirements for IP/TCP compression submitted to IESG for publication as Informational.**

<p>✓ DONE LATE LATE, ONGOING FROZEN ❑ DS related</p>

WG Status, Goals and Milestones 3(3)

- Feb 03 - IP/TCP compression scheme submitted to IESG for publication as Proposed Standard.
- ❑ Mar 03 - ROHC IP/UDP/RTP schemes submitted to IESG for publication as Draft Standard.
- ❑ Mar 03 - ROHC framework submitted to IESG for publication as Draft Standard.
- Apr 03 - ROHC UDP Lite schemes submitted to IESG for publication as Proposed Standard.
- *Jun 03 - Requirements for IP/SCTP compression submitted to IESG for publication as Informational.*
- *Jun 03 - IP/SCTP compression scheme submitted to IESG for publication as Proposed Standard.*
- Aug 03 - Possible recharter of WG to develop additional compression schemes.

ONGOING
TO DO
❑ DS related

} Low interest -
future unclear

ROHC RTP, DS preparations update

- **MIB & ROHC Terminology and mapping examples**
- **Implementation status and feature test list (interoperability)**
- **Implementer's guide**
- **A ROHC profile for IP only, purpose and issues**

MIB & terminology and channel mapping

- **The MIB**
 - Should now be rather stable, **PLEASE READ AND COMMENT**
 - Will be updated once more to include an RTP LLA MIB module
 - Refers to the “Terminology and channel mapping examples” draft
- **“Terminology and examples for MIB’s and channel mappings”**
 - The WG update of “*draft-jonsson-rohc-architecture-00.txt*”
 - **Purpose**
 - Simplify ROHC understanding
 - Identify ROHC entities and relationships
 - Establish common terminology
 - Clarify ROHC feedback
 - **Provides reference material for the MIB**

ROHC RTP implementation status, 1(2)

- **Four interoperability test events have been held**
- **“ROHC in the City”, Berlin, November 2002**
 - **Participants**
 - Effnet
 - Ericsson
 - Acticom* (host)
 - Institute for Communications Research (ICR)*, Singapore

* = First time participant

ROHC RTP implementation status, 2(2)

- **Results (preliminary, details will follow)**
 - **First extensive IPv6 testing performed by Ericsson/Effnet**
 - **Initial testing of CSRC-list compression by Ericsson/Acticom**
 - **Red flags raised for ambiguities in the list compression spec.**
 - **Some difficulties identified with extension 3 functionality, e.g. parsing of headers when format interpretation is changed (IP ID)**
- **RFC 3095 test document now available**
 - ***draft-ietf-rohc-rtp-rfc3095-interoperability-00.txt***
 - **Purpose:**
 - Specify feature tests
 - Collect interoperability test status

ROHC RTP implementer's guide, 1(4)

- **Scope of the document**
 - Clarifies RFC 3095
 - Exception: Provides enhanced mode transitions, compatible with the ones described in RFC 3095. **No protocol change!!**
 - Most (all?) content will go into RFC3095bis
 - Will stay as an Internet Draft until the DS work is completed
- **Two additions since last version, based on mail list discussions (next three slides)**

ROHC RTP implementer's guide, 2(4)

Timestamp (TS) interpretation interval

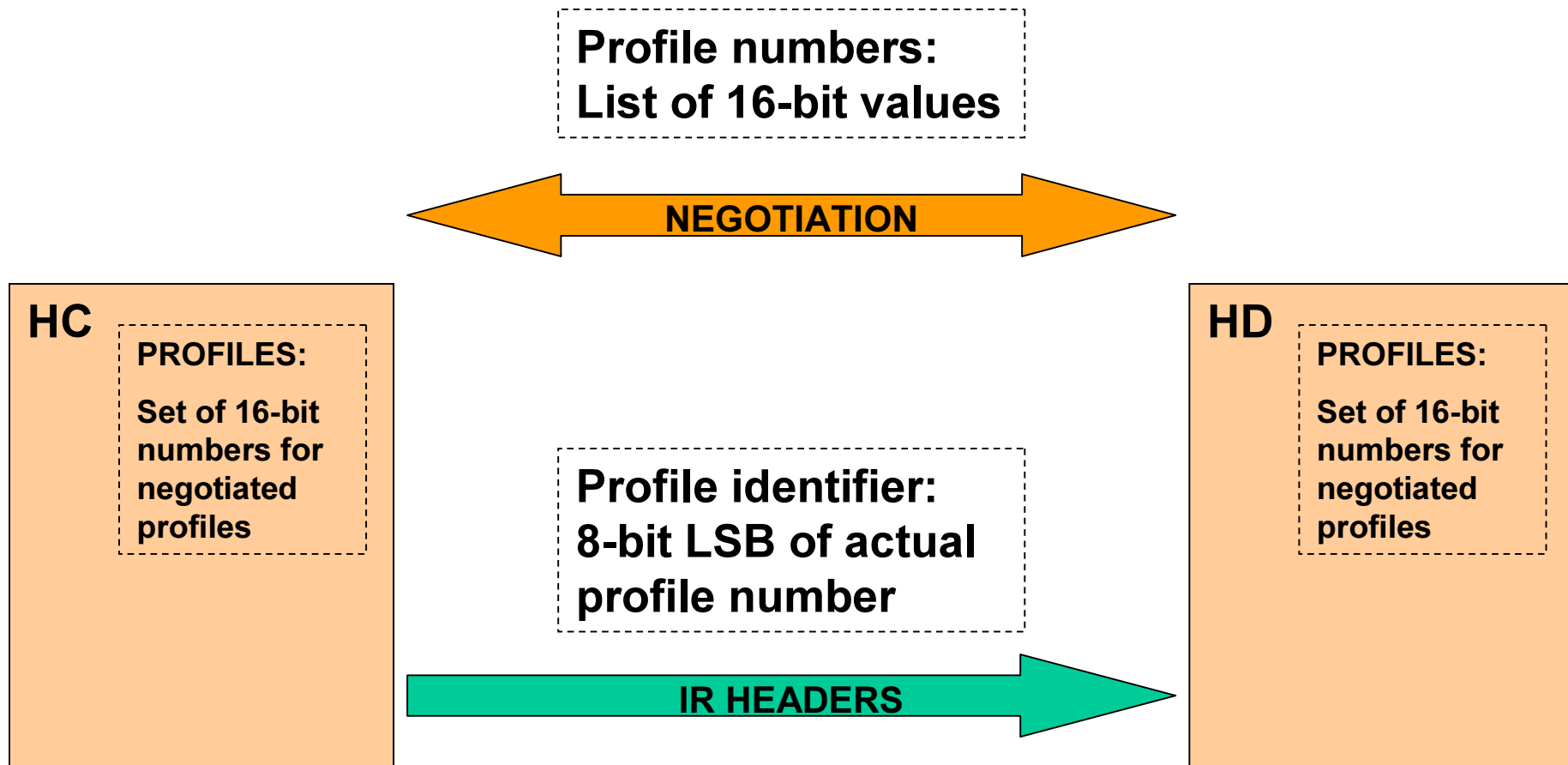
- **RFC 3095 defines two different interpretation intervals for TS**
- **Section 4.5.4 defines one in relation to the timer-based scheme**
- **Section 5.7 defines another one, which should be for all TS**

- **Both these are used. The one from 4.5.4 is used when the timer-based scheme is enabled, otherwise the one from 5.7**

- **The user guide explains what must be considered in the process of enabling or disabling timer-based compression, so that the change of interpretation interval will work out**

ROHC RTP implementer's guide, 3(4)

Negotiation of profile support



ROHC RTP implementer's guide, 4(4)

Negotiation of profile support

- **Conclusions**
 - **Two profiles with the same 8-bit LSB value are not supposed to be concurrently used over a ROHC channel**
 - **The negotiation protocol must ensure that the negotiated PROFILES set can not include several profiles with the same LSB value**
 - **This is fine with ROHC-over-PPP**

A ROHC profile for IP only - Purpose

- RFC 3095 defines profiles for Uncompressed, IP/UDP, IP/UDP/RTP and IP/ESP
- It has been agreed to define an IP profile to complement these
- The purpose of this IP profile **IS NOT TO**
 - Define a simplified ROHC profile with a different approach than 3095
- The purpose of this IP profile **IS TO**
 - Make the 3095 profile suite “complete”
 - Allow one who has a 3095 implementation to easily modify that for IP-only compression
 - Provide a profile to use for transports not yet supported by ROHC
- The profile is thus
 - Similar to the IP/UDP profile, basically the same excluding UDP
 - The technical content of the draft is less than one (1) page !!!

An IP profile, static chain termination

- How to terminate the STATIC chain?
 - In the case of IP/UDP and IP/UDP/RTP, the static chain is per profile definition implicitly terminated with a UDP or a combined UDP/RTP header portion, immediately following a static IP header portion with UDP as Protocol/Next Header



- With an IP profile, there is no single Protocol/Next Header value that implicitly terminates the chain based on the profile definition



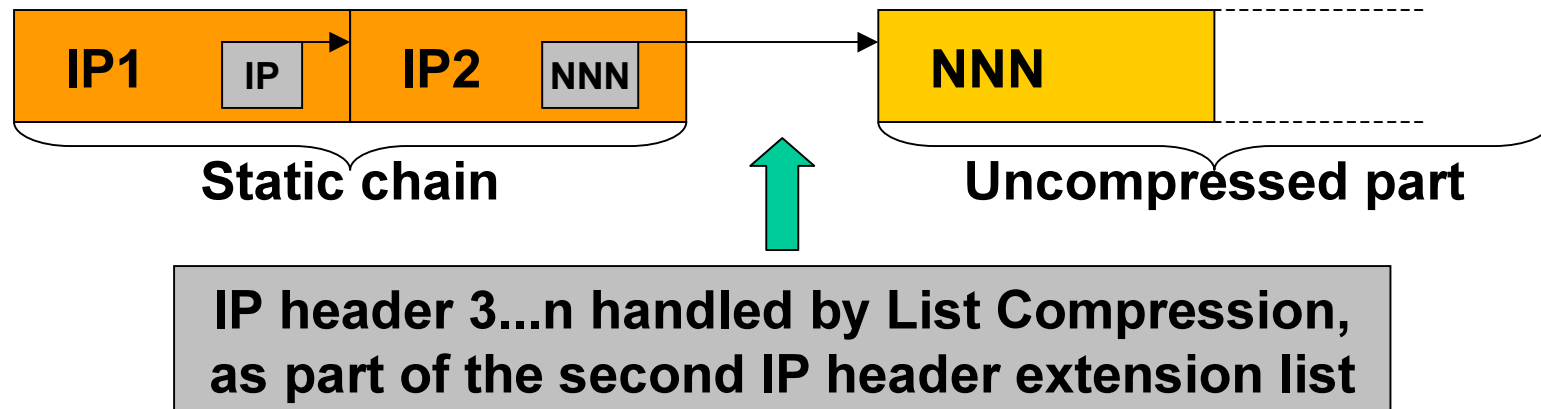
- Termination could be made by explicit means, or implicitly if the Protocol/Next Header field (NNN) indicates anything but IP or IPv6

An IP profile – Multiple IP headers, 1(4)

- **Compression of more than 2 IP headers?**
 1. The profiles of RFC 3095 handles up to 2 IP headers, but can not compress additional IP headers
 2. Since a UDP header is required to terminate the static chain for the 3095 profiles, more than 2 IP headers would mean the header can not be compressed at all
- **This is a non-desired limitation for the IP profile**
 - However, since the IP profile does not require one specific non-IP header to terminate the static chain, 2) could simply be addressed with a rule saying that the static chain would have to be implicitly terminated after the second IP header portion
 - Solving 1) require some additional effort, but might be worth considering, and the following slides outline 2 potential solutions

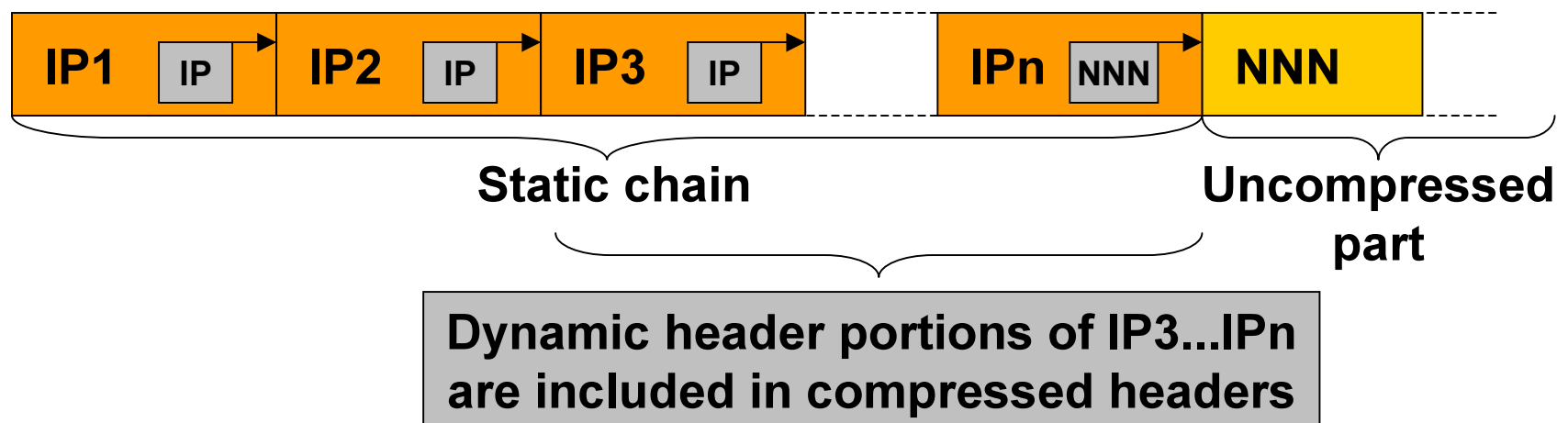
An IP profile – Multiple IP headers, 2(4)

- **A possible list compression solution**
 - **Add list compression support for IP headers, which would be used for all IP headers above the initial two. The static chain would still have to be implicitly terminated after the second IP header, and the Protocol/Next Header of the second static portion would give the type of the first header not part of the compressed list, i.e. the head of the uncompressed part**



An IP profile – Multiple IP headers, 3(4)

- A solution with dynamic data in compressed headers
 - Add optional dynamic header portions to compressed headers for extra IP headers, and let the static chain indicate the presence of these through an arbitrary number of IP headers in the static chain. For all but the first two IP headers, compressed headers will include a dynamic header portion



An IP profile – Multiple IP headers, 4(4)

- **Advantages with the latter solution**
 - **The static chain would be terminated only by a Protocol/Next Header field with a value different than IP/IPv6**
 - **The dynamic portion for IP3...n would have the same structure as the dynamic portion for IP1 and IP2 in IR and IR-DYN**
 - **The dynamic portion for IP3...n would be exactly the same in IR, IR-DYN and compressed headers**

ROHC RTP DS, next steps

- **Update MIB and issue WG last call for:**
 - **draft-ietf-rohc-mib-rtp-05.txt** (Proposed Standard)
 - **draft-ietf-rohc-terminology-and-examples-00.txt** (Informational)
- **Solve the IP profile issues and last-call in WG**
- **Improve the test list document, and include status**
- **Evaluate findings from Berlin, and update the implementer's guide accordingly**
- **Produce the “RFC3095 surgery plan”**

SigComp

Richard Price
(richard.price@roke.co.uk)

SigComp User Guide

- Informational companion to SigComp RFC
 - UDVM assembly language
 - Bytecode for well-known decompression algorithms
 - SigComp-specific mechanisms
- Updated -01 draft available
 - Enhanced assembly language
 - “Raw” bytecode for each decompressor
 - Example compressed messages

Next Steps

- Proposed algorithm <draft-ziyad-rohc-tccb-01.txt>
 - Add bytecode to User Guide?
 - Describe compressor in separate draft?
- Further explanation of SigComp-specific mechanisms
 - Recovery from errors?
 - Making use of additional resources?
- More detailed test message flows?

SigComp Torture Tests

- Provides tests for each UDVM instruction
 - Instructions are divided into related subsets
 - Each subset tested independently
- Focuses on boundary and error cases
 - Unusual behaviour (e.g. freeing non-existent state)
 - Edge conditions (e.g. inputting 0 bytes, division by 0)
 - Self-modifying code
- Covers issues not typically found in well-behaved code
 - But may be exploited by malicious users

Next Steps

- Current version tests the UDVM only
- Further tests required for other SigComp entities
 - State handler
 - Memory management
 - Interaction between multiple compartments
 - SigComp dispatcher
 - Initialising the UDVM
 - Delimiting SigComp messages over TCP
 - SigComp feedback mechanism
- Any other issues that should be covered by the tests?

SigComp Interop

- Proposal to hold SigComp interop at SIPit
 - Stockholm City Conference center
 - 24 – 28 February 2003
 - Registration begins 20 November!

SigComp Recovery

Adam Roach,
adam@dynamicsoft.com

Zhigang Liu,
zhigang.c.liu@nokia.com

Problem Statement

When a SigComp message cannot be decoded by the recipient (for whatever reason), the sender needs to learn of such a failure.

Scenario 1: Handset Restart

- Handsets are not highly available nodes. When a handset loses state (e.g. battery dies), its SigComp partner in the network won't know of any state loss.
- Without any additional mechanism, we must wait for the node in the network to timeout.

Scenario 2: GPC Apps

- Similarly, if, e.g., a laptop with a GPRS card is running multiple applications, the ability to reset only one compartment – the one corresponding to a failed application – is useful.

Scenario 3: Server Failover

- The use of DNS to form clusters of “equivalent” nodes is a common approach for high-availability, at least in SIP. (That is, a single DNS name has multiple IP addresses with varying q values).
- In the case of a server failure, the new node won’t have appropriate state, even though it still “looks” like the same compartment from the application layer.

Three Approaches

1. Status Quo: application timeouts combined with explicit "state_memory=0" signaling
2. Explicit "Reset" message that completely resets compartments
3. Explicit "NACK" message that indicates specific cause of failure

Solution 1: Status Quo

- Recipient can indicate “state_memory=0” in next message (either piggybacked with an application-level message, or as a standalone indication)
- If application-layer timeout occurs, application layer of sender will reset compartment and re-send message.

Solution 1: Pros and Cons

Pros

- Requires no additional standardization

Cons

- Extra communication between Application and Presentation layers
- “state_memory=0” approach requires application-layer message in reverse direction or additional security
- Retransmission approach adds latency

Solution 2: Reset message

- Recipient can send “reset all compartments associated with this IP address” message (piggybacked or standalone)
- Recipient can send “reset the compartment to which this message corresponds” (piggybacked or with security)

Solution 2: Pros and Cons

Pros

- Simple to implement
- Provides for immediate recovery

Cons

- Applies “sledgehammer” approach: resets entire compartment or compartments
- “

”

Solution 3: NACK Message

- Recipient can send specially-formatted standalone message that identifies specific cause of failure
- Sender can make fine-grained adjustments to offending compartment

Solution 3: Pros and Cons

Pros

- Retransmitted message is as small as possible
- Recovers from failures other than state loss (e.g. state ID collisions)
- Provides details useful for debugging implementations and deployments

Cons

- More complicated implementation
- Cannot be piggybacked with application-layer message
- Slightly larger than message for other two approaches

Interesting Side Effects

- Use of NACK mechanism, once verified, allows use of optimistic compression.
- Use of NACK mechanism allows for diagnostics of network problems.

Solution 1, State Loss

1	Bytecodes, App Msg	800 bytes
2	Bytecodes, App Msg	800 bytes
3	State IDs, App Msg	110 bytes
4	state_memory=0	10 bytes
5	Bytecodes, App Msg	800 bytes
		<hr/>
		2520 bytes total

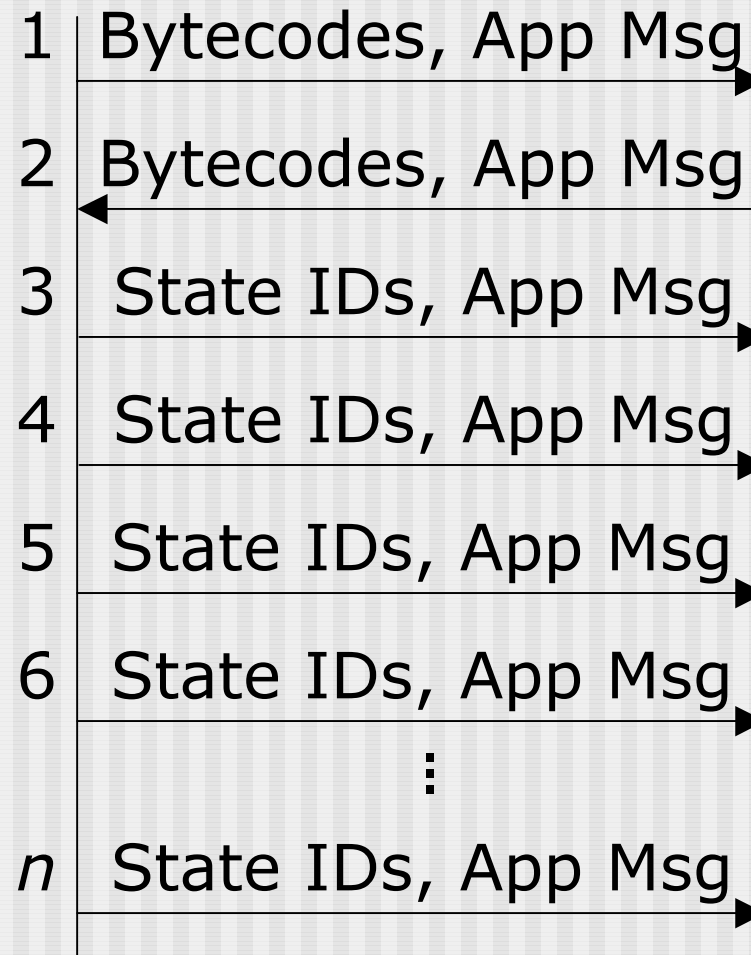
Solution 2, State Loss

1	Bytecodes, App Msg	800 bytes
2	Bytecodes, App Msg	800 bytes
3	State IDs, App Msg	110 bytes
4	IP-Level Reset	10 bytes
5	Bytecodes, App Msg	800 bytes
		<hr/>
		2520 bytes total

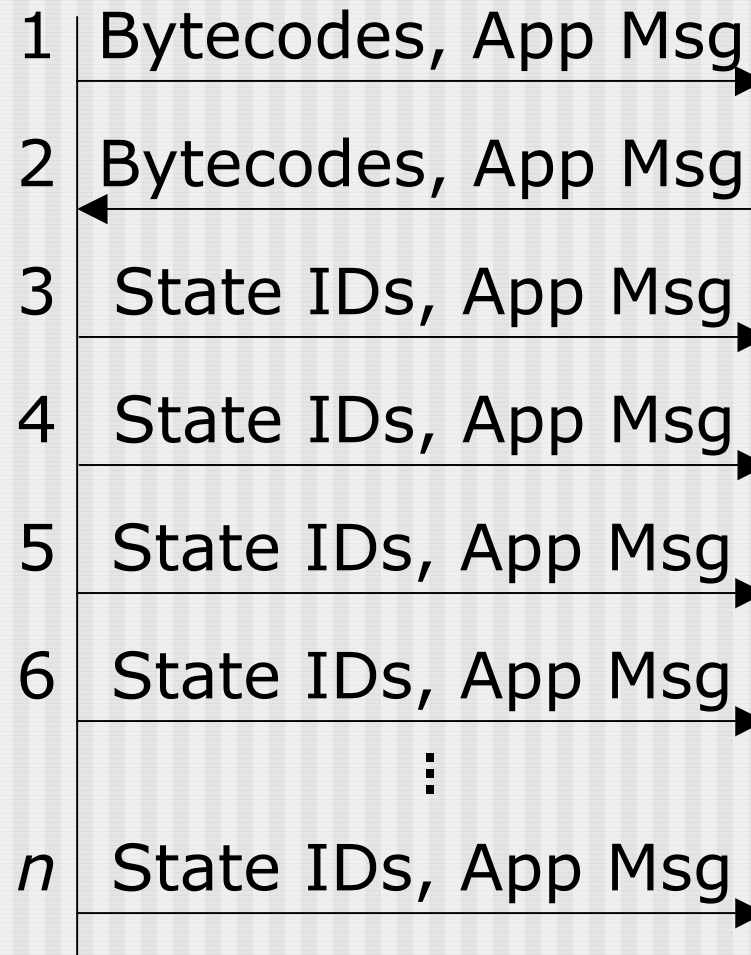
Solution 3, State Loss

1	Bytecodes, App Msg	800 bytes
2	Bytecodes, App Msg	800 bytes
3	State IDs, App Msg	110 bytes
4	NACK	26 bytes
5	State ID, App Msg	400 bytes
		<hr/>
		2136 bytes total

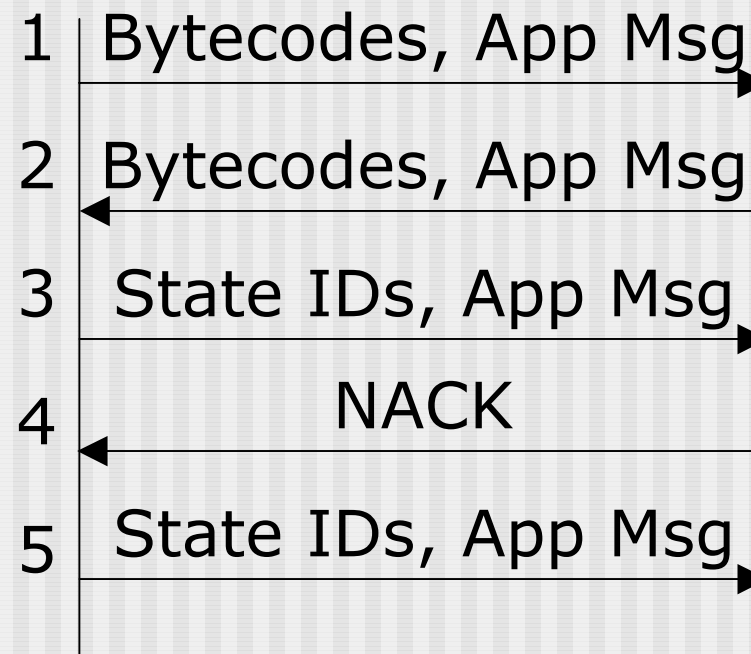
Solution 1, ID Collision



Solution 2, ID Collision



Solution 3, ID Collision



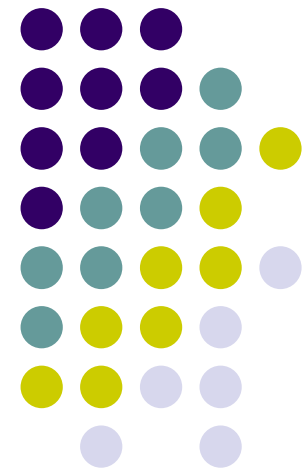
Questions

- Do the drawbacks of the status quo warrant this additional work?
- Is retransmission of a failed message over a reliable transport the responsibility of the application or the SigComp stack?
- Is the additional complexity of NACK worth the nominal improvement in efficiency and resiliency to non-state-loss situations?
- Do we need to be able to piggyback on failure indicator messages?

ROHC over Wireless Ethernet Media (ROHCoWEM)

Kris Fleming
Intel Corporation
ROHC WG

55th IETF Meeting Atlanta, GA





Motivation

- WLAN/WPAN have similar BER and link layer latency issues which can benefit from the ROHC
- WLAN & WPAN are also limited in bandwidth
- Increase the density of WLAN & WPAN users for access points
 - Data payload overhead can be reduced up to 75% (VoIP with G.729 encoding) allowing additional users
- VoIP is and will become popular for these networks.
- This work should be done in the IETF
 - Independent of the PHY layer

Other Solutions and why that they were not used



- PPPoE

- Cons

- Additional unnecessary overhead

- PPPoE (6 octets)
 - PPP header and FCS (6 octets)

- Unnecessary complexity

- PPP would only be used for parameter negotiation and framing

- Pros

- Parameter negotiation for ROHC already defined

Other Solutions and why that they were not used

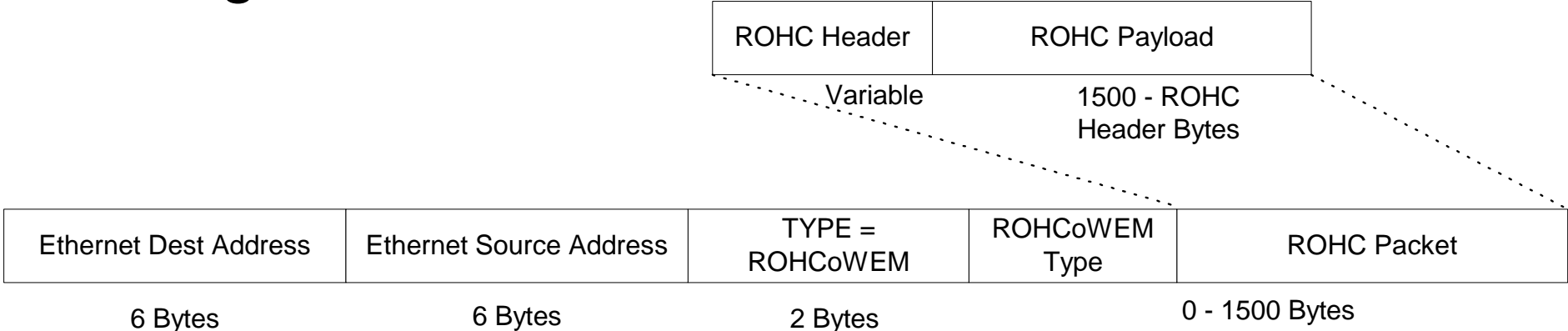


- Use ROHCoWEM, but reuse existing protocol for negotiation parameter
- IPv6 Neighbor Discovery
 - Cons
 - Not intended for discovering neighbor protocol capabilities
 - For ROHCoWEM, this would be link layer capabilities
 - Pros
 - Protocol already defined and would need to be extended to support ROHCoWEM

ROHC over Wireless Ethernet Media



- Allows for ROHC Support for current and future wireless networks
 - IEEE 802.11
 - Bluetooth
- Defines a simple ROHC parameter negotiation





Questions

- Should this be a ROHC WG work item?
- If so, what are opinions about current draft?

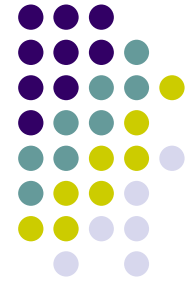


- Kris.D.Fleming@Intel.com

Other names and brands may be claimed as the property of others.

Copyright © 2002 Intel Corporation.

Backups

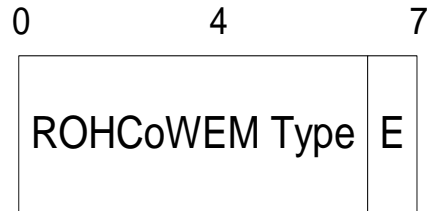
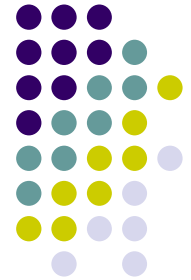


ROHCoWEM Type Definition

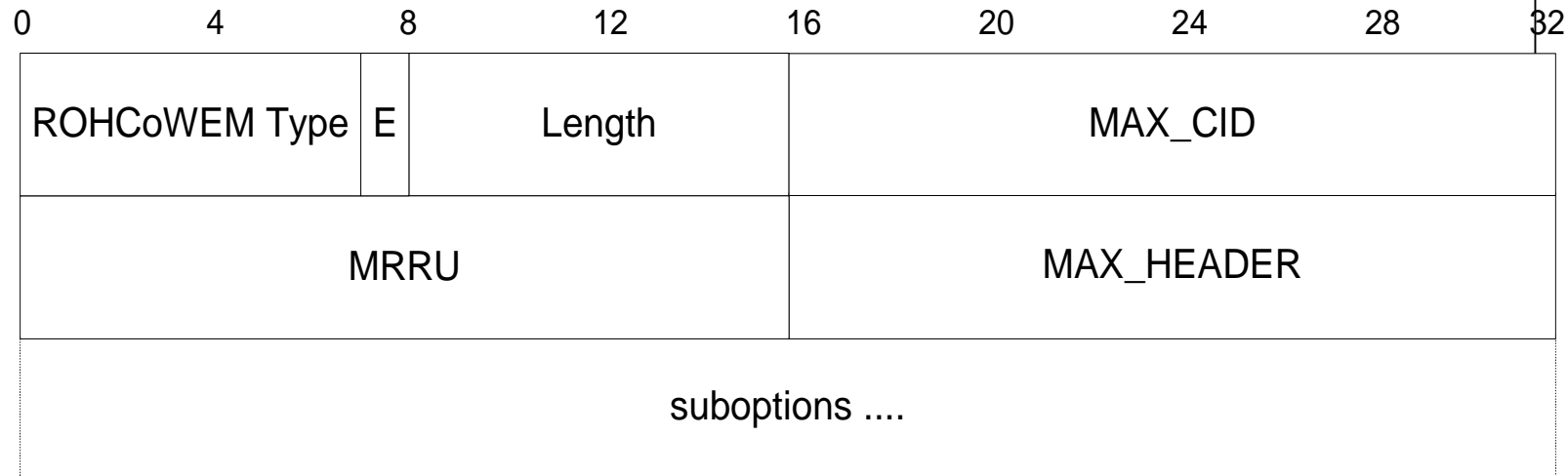


ROHCoWEM Type	Value
ROHCoWEM Configuration Request	0x00
ROHCoWEM Configuration Response	0x01
ROHC	0x02
Reserved	0x03-0xFF

ROHCoWEM Configuration Request

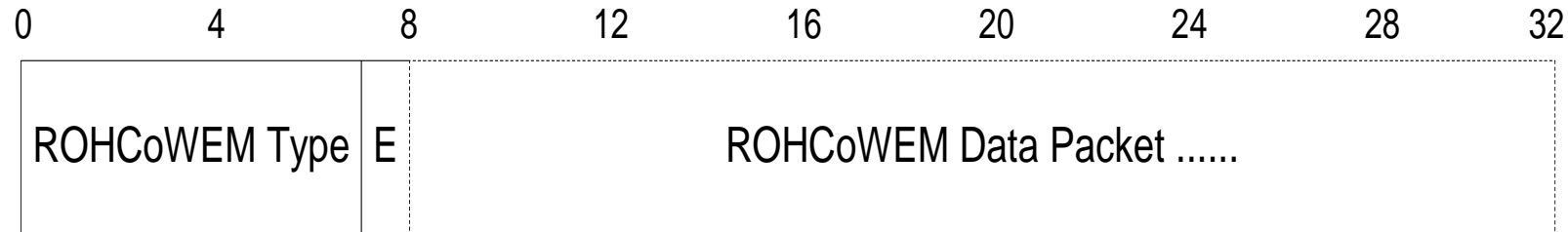


ROHCoWEM Configuration Response



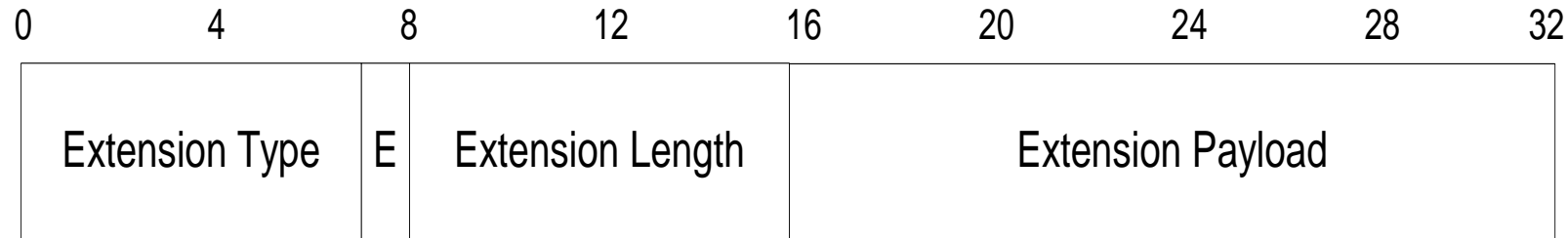
Field	Description
Length	1 octets field for the length measured in octets of the entire ROHCoWEM packet
MAX_CID	2 octets field to indicate the maximum value of a context identifier (see [RFC3095])
MRRU	2 octets field to indicate the maximum reconstructed reception unit (see [RFC3095])
MAX_HEADER	2 octets field to indicate the largest header size in octets that can be compressed (see [RFC3095])

ROHCoWEM Data Packet



- ROHCoWEM Data Packet
 - Normal ROHC packets as defined in RFC3095 [2], section 5.2.

ROHCoWEM Extension Header



- ROHCoWEM Extension Type
 - Defines the contents of the ROHCoWEM Extension Payload
- Extension Flag
- Extension Length
 - length measured in octets of the entire ROHCoWEM packet
- Extension Payload

ROHC over Ethernet -- Issues (1)

Invent another negotiation protocol?

- **For IPv6, ND (neighbor discovery) is obvious choice**

IPv4:

- **Could use ARP (sounds difficult)**
- **Could use PPPoE + RFC3241**
 - **PPPoE discovery is not really needed**
 - **Still need new encapsulation for data packets (see below)**
- **Could use ND (!)**
 - **Any parameters negotiated for IPv6 would apply to IPv4, too**

ROHC over Ethernet -- Issues (2)

Encapsulation issue: Minimum packet length

- Ethernet requires padding for small packets
- Padding scheme needs to be known at bridges:
- **Comp ➔ Ethernet ➔ Bridge ➔ Wireless ➔ Decomp**
 - How to get the bridge not to send the padding?
- **Comp ➔ Wireless ➔ Bridge ➔ Ethernet ➔ Decomp**
 - Encapsulation must be robust against inserted padding

ROHC over Ethernet -- Issues (2 cont.)

- **Could use raw ROHC**
 - Always have to add enough padding bytes at the front
 - need to count padding bytes at receiver
- **Could invent new encapsulation with length field**
- **Could use PPP-style self-describing-padding**
 - MPV = 45 (ouch)
 - Not robust against inserted padding
- **Could use 802 LLC**
 - Three bytes overhead, need to allocate SAP ID

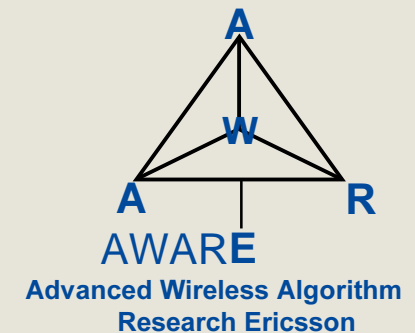
IETF 55 - ROHC WG Meeting

Changes to the ROHC-TCP Profile draft-ietf-rohc-tcp-03.txt



Ericsson AB, Luleå
Ghyslain.Pelletier@epl.ericsson.se
+46 920 20 24 32

November 18th, 2002



ROHC-TCP: Summary of Changes in 03

Editorial changes:

- **New editor** (Ghyslain Pelletier)
- **New structure**, aligned with RFC 3095
- Some redundancies have been merged or removed

Technical changes:

- **ROHC-TCP does not use the “mode” concept anymore**
 - e.g. no explicit “U-mode” or “B-mode” defined;
 - HC begins in unidirectional operation;
 - HC moves to bidirectional operation:
once a first feedback is received,
effectively establishing a feedback channel
- **Clarifications on states and state transitions**

ROHC-TCP: What needs to be done

● Context Replication:

- Concept is understood, but now needs to be specified clearly
- Efficiency - complexity tradeoff
 - IP source/destination addresses only?
 - Other compressed fields as well?
- Base contexts – what does it include?

● Packet formats:

- IR, IR-Static and IR-Replicate packets
- CO packets

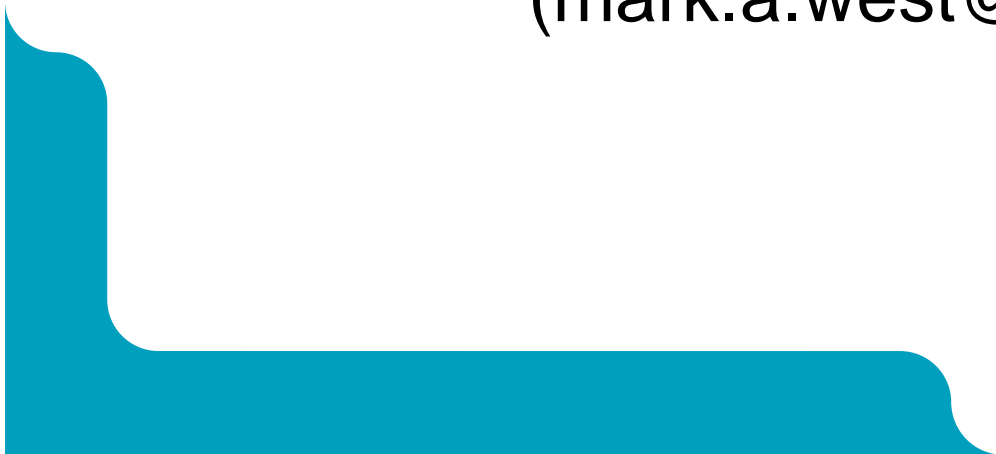
● Finalize compressor/decompressor logic

● Others?

Formal Notation for ROHC Update...

Mark West

(mark.a.west@roke.co.uk)



What's changed?

- Not a lot...
 - Alternative way of defining the mappings
 - Made the definition more implementation neutral
- The key goal
 - Define reversible mappings between uncompressed and compressed field values



Removal of pseudo-code

- Too implementation specific
- Now use 'pre-' and 'post-' conditions to capture behaviour of a mapping
- Allow mappings to be composed from other mappings
 - Makes documentation more compact
 - Easier to define new mappings



Example of mapping

- e.g. RANGE(n, v, k):
 - $tmp = NULL ? field_value$
 - $field_value = tmp ? \text{mod}(tmp, k)$
 - $tmp = \text{mod}(\text{mod}(field_value - v, k) + v, 2^n) ? NULL$
- Where:
 - $field_value$ is the well known label for the current field
- The LHS gives the label that is updated by the mapping
- The RHS gives the pre- and post-conditions
 - These swap roles for decompression

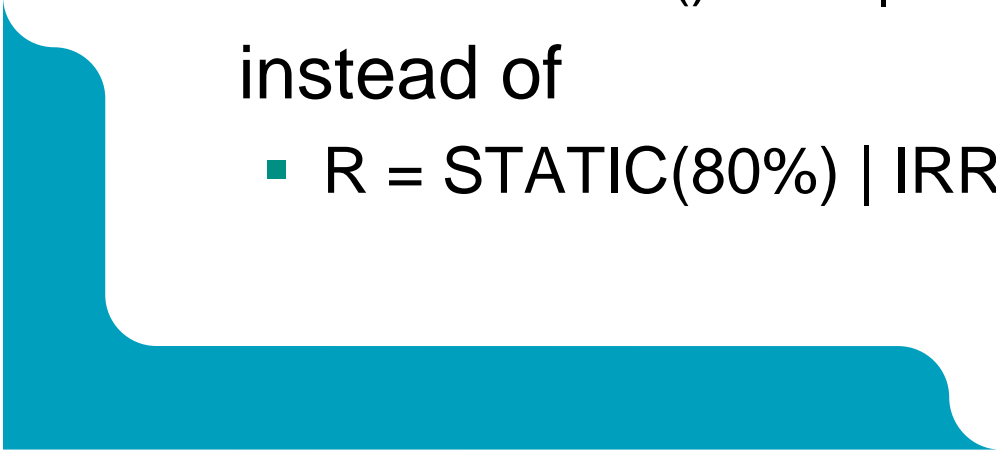
Compression

- Rule 1
 - Pre: $tmp = NULL$
 - Post: $tmp = field_value$
- Rule 2
 - Pre: $field_value = tmp$
 - Post: $field_value = \text{mod}(tmp, k)$
- Rule 3
 - Pre: $tmp = \text{mod}(\text{mod}(field_value - v, k) + v, 2^n)$
 - Post: $tmp = NULL$
- Note that rule 3 precondition is important to test that mapping is reversible at decompressor

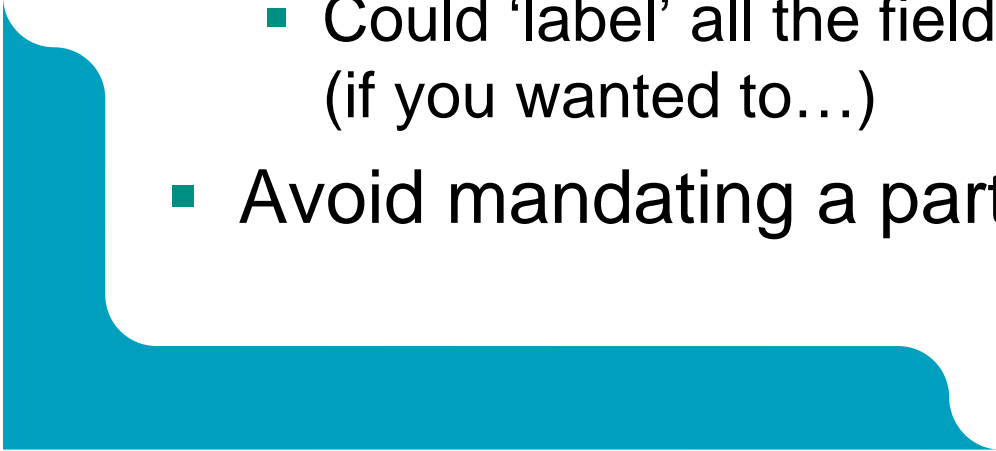
Decompression

- Rule 3
 - Pre: $tmp = NULL$
 - Post: $tmp = \text{mod}(\text{mod}(\text{field_value} - v, k) + v, 2^n)$
- Rule 2
 - Pre: $\text{field_value} = \text{mod}(tmp, k)$
 - Post: $\text{field_value} = tmp$
- Rule 1
 - Pre: $tmp = \text{field_value}$
 - Post: $tmp = NULL$
- Note that pre-conditions should be trivially true for decompression

Choice of mappings

- Probabilities used to be associated with mappings
 - Pointed out previously, this is slightly strange
 - Probabilities now associated with 'choice' operator
 - For notation, effect is largely cosmetic:
 - `R = STATIC() 80% | IRREGULAR(8) 20%`instead of
 - `R = STATIC(80%) | IRREGULAR(8,20%)`
- 

Those pesky bitstacks

- Replaced by 'labels'
 - Use LABEL (16, ip_id) to label a field
 - Use NEXT-FIELD (ip_id) to process it
 - Some rules take labels as parameters
 - e.g. INFERRED-OFFSET base field
 - Does this eliminate the question about separation of field extraction and mappings?
 - Could 'label' all the fields, then process them (if you wanted to...)
 - Avoid mandating a particular implementation
- 

Back to the questions

- Many of the same questions as last time remain...
 - Header deconstruction and compression mappings
 - Believe that we are happy with combined notation?
 - Identification and generation of meta-data
 - Is this part of the 'grammar'?
 - Precise set and form of mappings
 - Much easier to define new mappings
 - So this should not be a problem?
 - Interface to the (unspecified) encoding rules
 - Is there an issue here?

The way forward...

- Try to define minimum required for interoperability
 - What is missing?
(We have an existence proof that compression by this method can work interoperably)
- How should we make use of this to capture TCP behavior?
 - Answers on a postcard to...

