IPv6 over Low power WPAN WG (6lowpan)

Chairs:

Geoff Mulligan <geoff@mulligan.com>

Carsten Bormann <cabo@tzi.org>

Mailing List:

6lowpan@ietf.org

Jabber:

6lowpan@jabber.ietf.org

- **We assume people have read the drafts**
- **Meetings serve to advance difficult issues by making good use of face-to-face communications**
- **Be aware of the IPR principles, according to RFC 3979 and its updates**

✓Blue sheets
✓Scribe(s)

Milestones (from WG charter page)

Document submissions to IESG:

- **Aug 2008 x 2 Improved Header Compression (PS)**
- **Aug 2008 // 6 Security Analysis (Info)**
- **Sep 2008 // 3 Architecture (Info)**
- **Sep 2008 x 4 Routing Requirements (Info)**
- **Nov 2008 x 1 Bootstrapping and ND Optimizns (PS)**
- **Dec 2008 x 5 Use Cases (Info)**

Also: running documents for implementers, interop

78th IETF: 6lowpan WG Agenda

09:00	Introduction, Agenda	Chairs (5)
09:10	2 – HC-07	JH (20)
09:30	1 – ND	
09:30	ND-11	ZS (40)
10:10	thubert--backbone	PT (10)
10:20	RA Suppression	LT (10)
10:30	3 – Security	
10:30	HIP	RGM (20)
10:50	qiu--secure-router	YQ (10)
11:00	0 – TCP HC	LT (10)
11:10	0 – thubert--simple-frag-recovery	PT (10)

78th IETF: 6lowpan WG Agenda

09:00	Introduction, Agenda	Chairs (5)
09:10	2 – HC-07	JH (20)
09:30	1 – ND	
09:30	ND-11	ZS (40)
10:10	thubert--backbone	PT (10)
10:20	RA Suppression	LT (10)
10:30	3 – Security	
10:30	HIP	RGM (20)
10:50	qiu--secure-router	YQ (10)
11:00	0 – TCP HC	LT (10)
11:10	0 – thubert--simple-frag-recovery	PT (10)

Compression Format for IPv6 Datagrams in 6LoWPAN Networks

(draft-ietf-6lowpan-hc-07)

Jonathan Hui
Pascal Thubert

6LoWPAN WG Meeting
78th IETF Meeting
Maastricht, Netherlands

Updates Since Anaheim

- Added section on mapping between link-layer addresses and IIDs
- Added text on restricting compressed headers to first fragment when using RFC 4944 fragmentation
- WGLC expired with no significant comments

Remaining Issues

- Section 4.2 specifies compression for IP-in-IP
- Compression of inner IPv6 header addresses should be based on the outer IPv6 header, not link-layer addresses
- Proposed change:
  - SAC=1/SAM=11 and DAC=1/DAM=11
  - Derive from link-layer header ➜ Derive bits encapsulating header

Next Steps

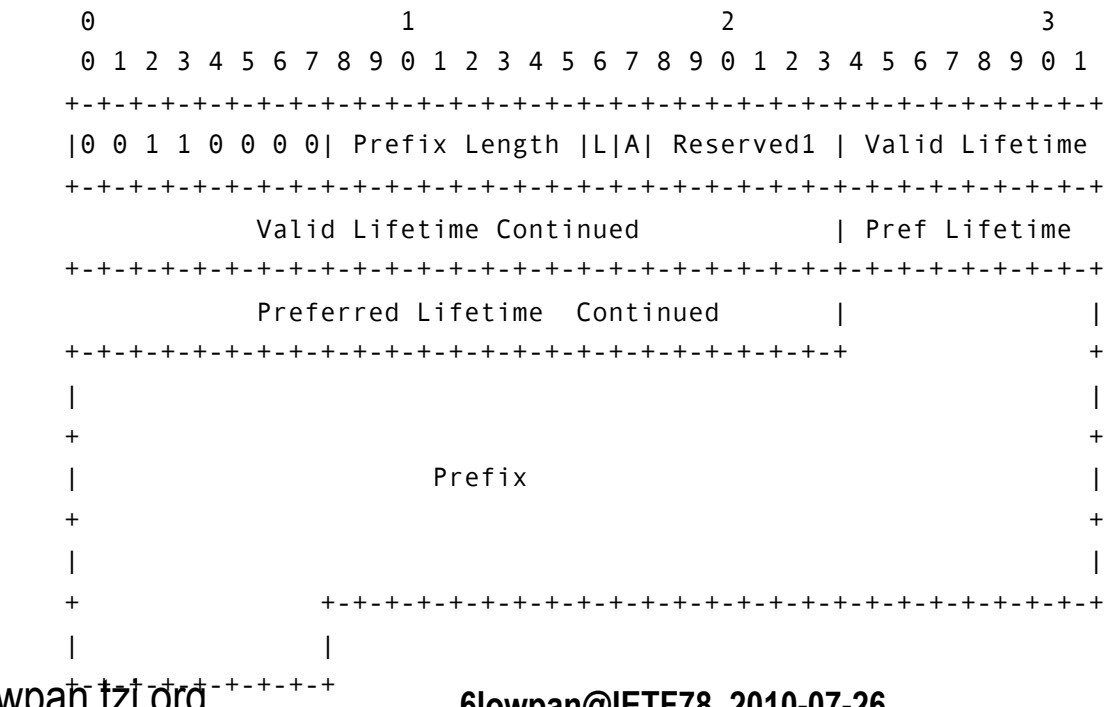
- Update for IP-in-IP, quick WGLC?

Status HC for 6lowpan

- 6lowpan HC is focusing on stateless HC
  - with some help from the 6lowpan context
- HC-07 pretty much nails it for IP and UDP
  - TCP much harder to do stateless
  - but what about ICMPv6 and header-like payloads (RPL)?
- HC-07 will be “baseline” for 6lowpan
- Additions for other headers/header-like payloads will need to be negotiated
  - network wide? (This is a v2 Lowpan)
  - per-neighbor state? (This guy knows XYZHC)
  - ND work needed?

Current pre-drafts on ICMPv6HC and generic header/header-like HC (1)

- Generated during pre-IETF draft break
- 1: “draft-ietf-6lowpan-icmphc-00.txt” (O’Flynn)
  - draft specific to ICMPv6, 4–28 byte saving



Current pre-drafts on ICMPv6HC and generic header/header-like HC (2)

- **2: “draft-bormann-6lowpan-ghc-00pre” (Bormann)**
  - **taking up the ideas from RFC 3320**
    - RFC 3320: SigComp UDVM (Universal Decompressor Virtual Machine)
    - way too complex ➜ simplify by an order of magnitude
  - **generic to any header-like header/payload (RPL?)**
  - **more modest savings (usually 2-5 bytes less than O’Flynn)**
  - **1-page spec, simple bytecodes**

Current pre-drafts on ICMPv6HC and generic header/header-like HC (3)

0kkkkkkk	Copy k+1 bytes of actual data (k < 96)	The k+1 bytes of data
011sssss	s = (sssss * 8)	
10000nnn	reserved	
10001kkk	Insert 8 bytes copied from previous bytes, at k + s bytes distance; s += 8	
1001nnnn	Insert n+2 bytes of zeroes	
1010iiii	Insert all bytes from Context i	
1011iiii	Insert 8 bytes from Context i; i.e., the context value truncated/extended to 8 bytes, and then insert 0000 00FF FE00	
11nnnkkk	Insert n+2 bytes from previous bytes, k + s bytes distance; s = 0	

- **Both are not nearly ready to be standardized**
  - **won't impact HC-07**

78th IETF: 6lowpan WG Agenda

09:00	Introduction, Agenda	Chairs (5)
09:10	2 – HC-07	JH (20)
09:30	1 – ND	
09:30	ND-11	ZS (40)
10:10	thubert--backbone	PT (10)
10:20	RA Suppression	LT (10)
10:30	3 – Security	
10:30	HIP	RGM (20)
10:50	qiu--secure-router	YQ (10)
11:00	0 – TCP HC	LT (10)
11:10	0 – thubert--simple-frag-recovery	PT (10)

“Neighbor Discovery Optimization for Low-power and Lossy Networks”

draft-ietf-6lowpan-nd-11

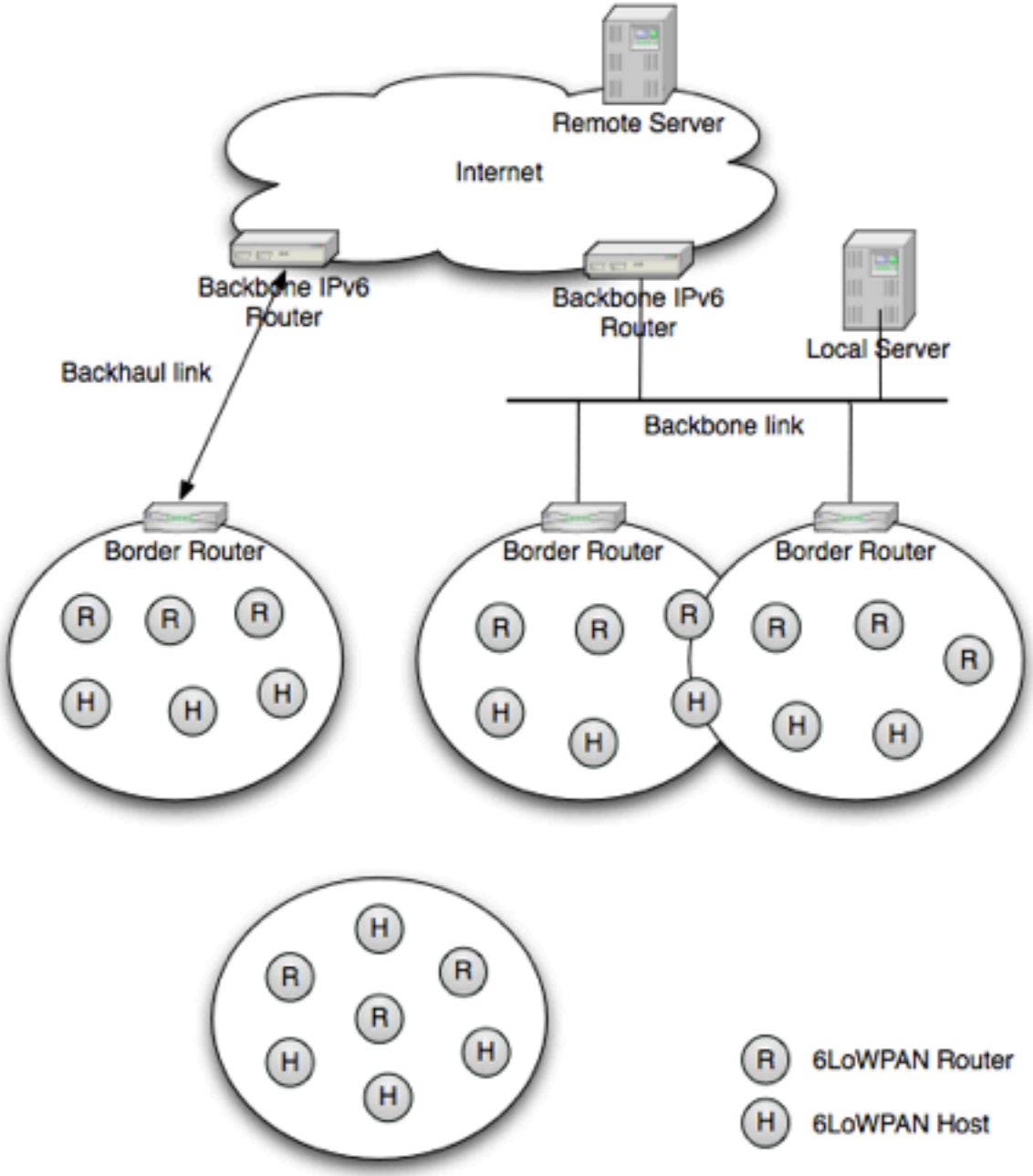
Zach Shelby, Samita Chakrabarti, Erik Nordmark

Progress since Anaheim

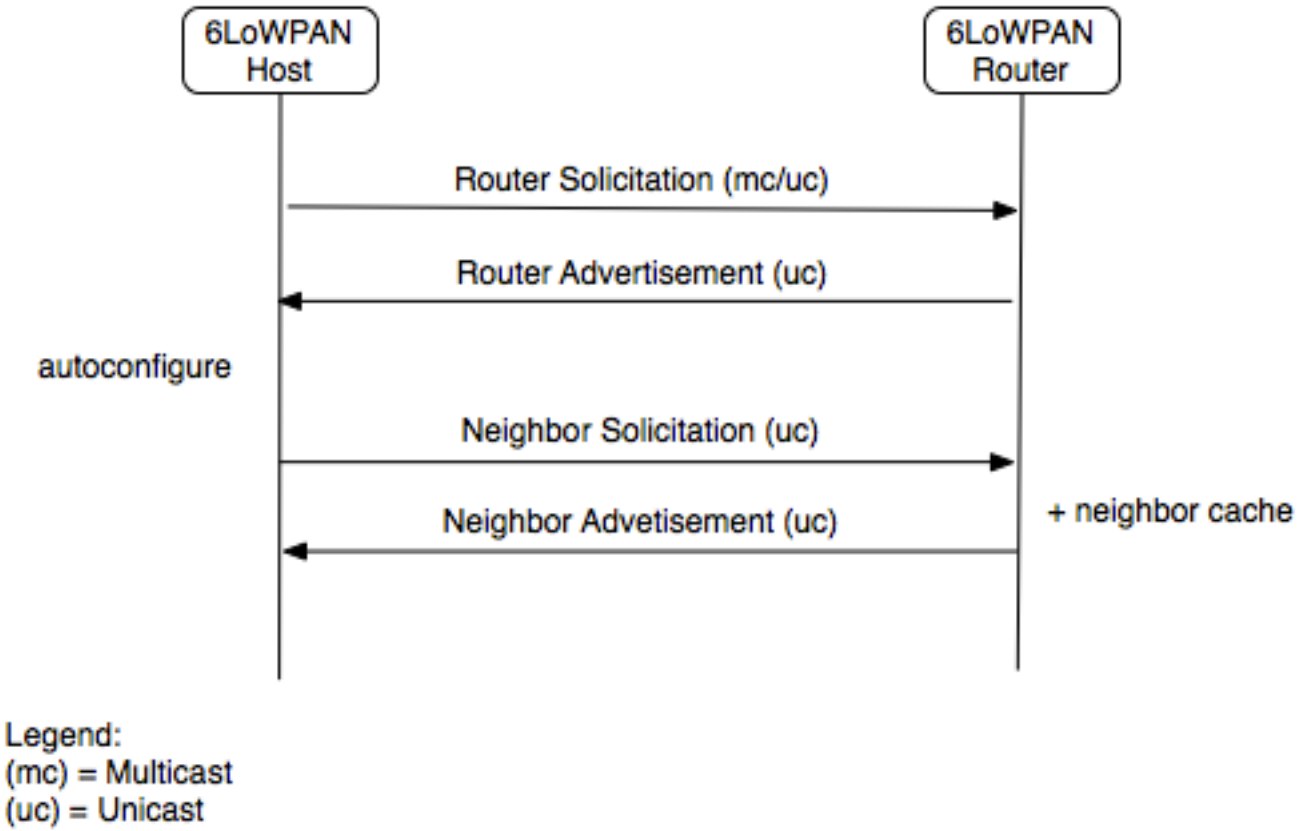
- nd-09
  - Complete re-write based on draft-chakrabarti-6lowpan-ipv6-nd-simple-00
  - Registration using NS/NA
- nd-10
  - Minor clarifications and improvements
  - Closed 9 tickets
- nd-11
  - Further clarifications, field optimizations
  - Integrated feedback from ZigBee interop event
  - Closed 7 tickets

ND optimized for LLNs

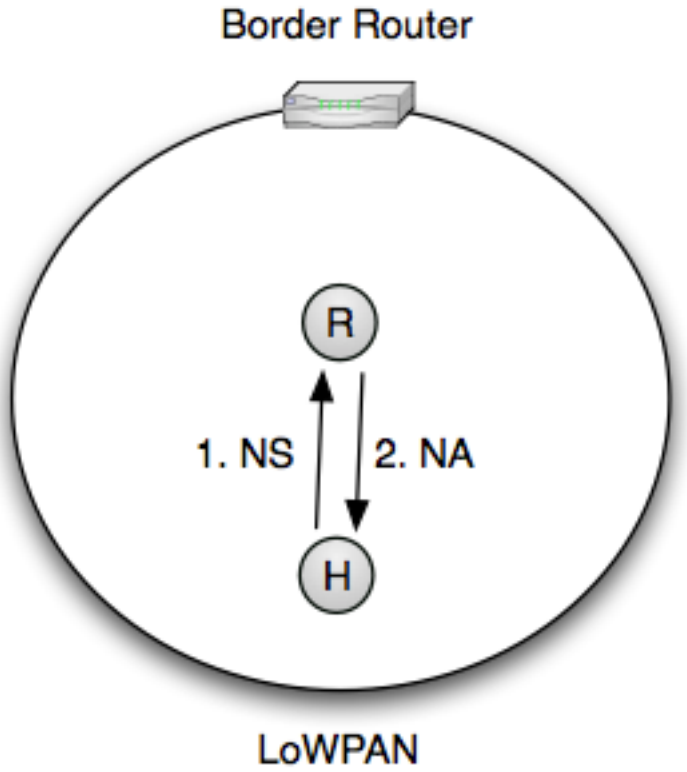
- RFC4861 optimizations and extensions for LLNs
  - Optimizing the host-router interface
  - Address registration mechanism using NS/NA
  - Less traffic (and very little multicast)
  - Host initiated message exchanges (RS, NS)
  - Less memory and code needed
- Optional duplicate address detection
- Optional multihop prefix and context dissemination
- Compatible with link-layer mesh and IP routing


Remote Server
Internet
Backbone IPv6 Router
Backbone IPv6 Router
Local Server
Backhaul link
Backbone link
Border Router
Border Router
Border Router
R
H
6LoWPAN Router
6LoWPAN Host

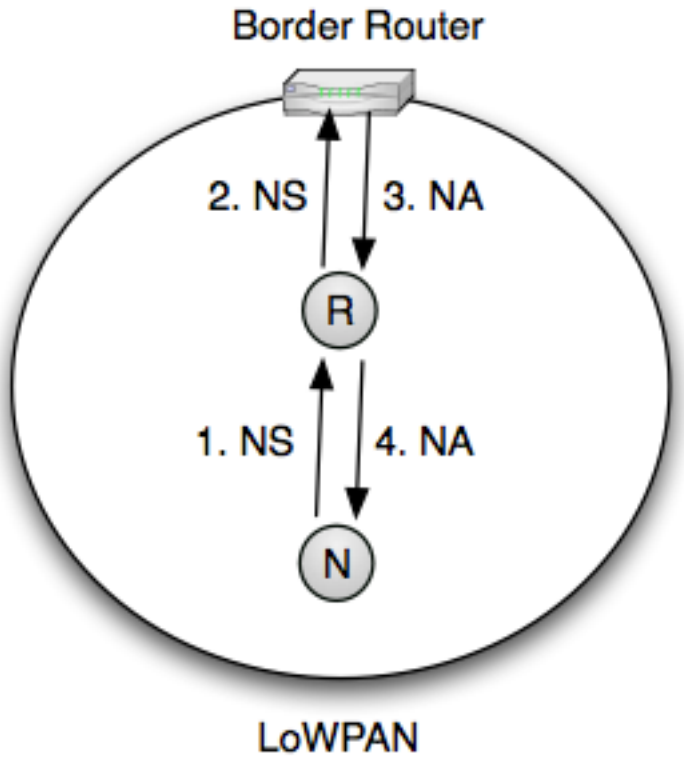
Basic operation



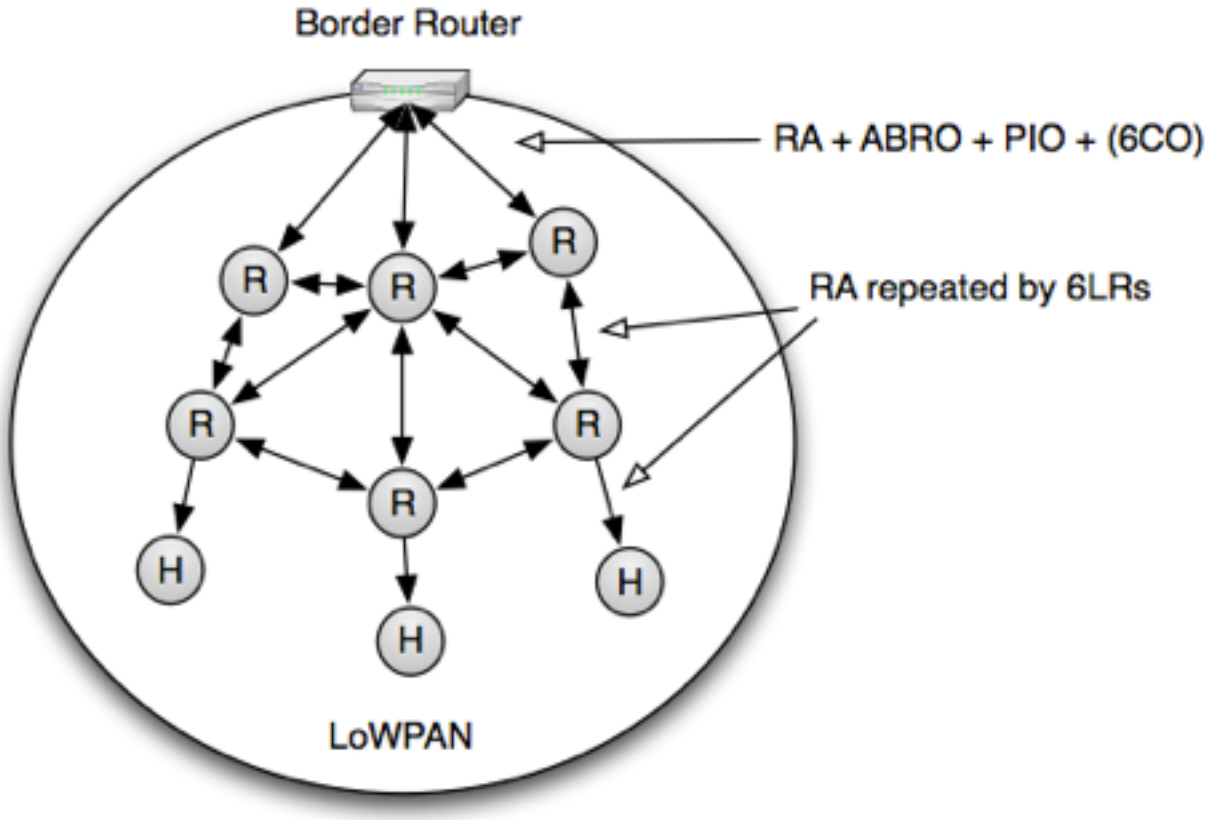
Host-Router interface



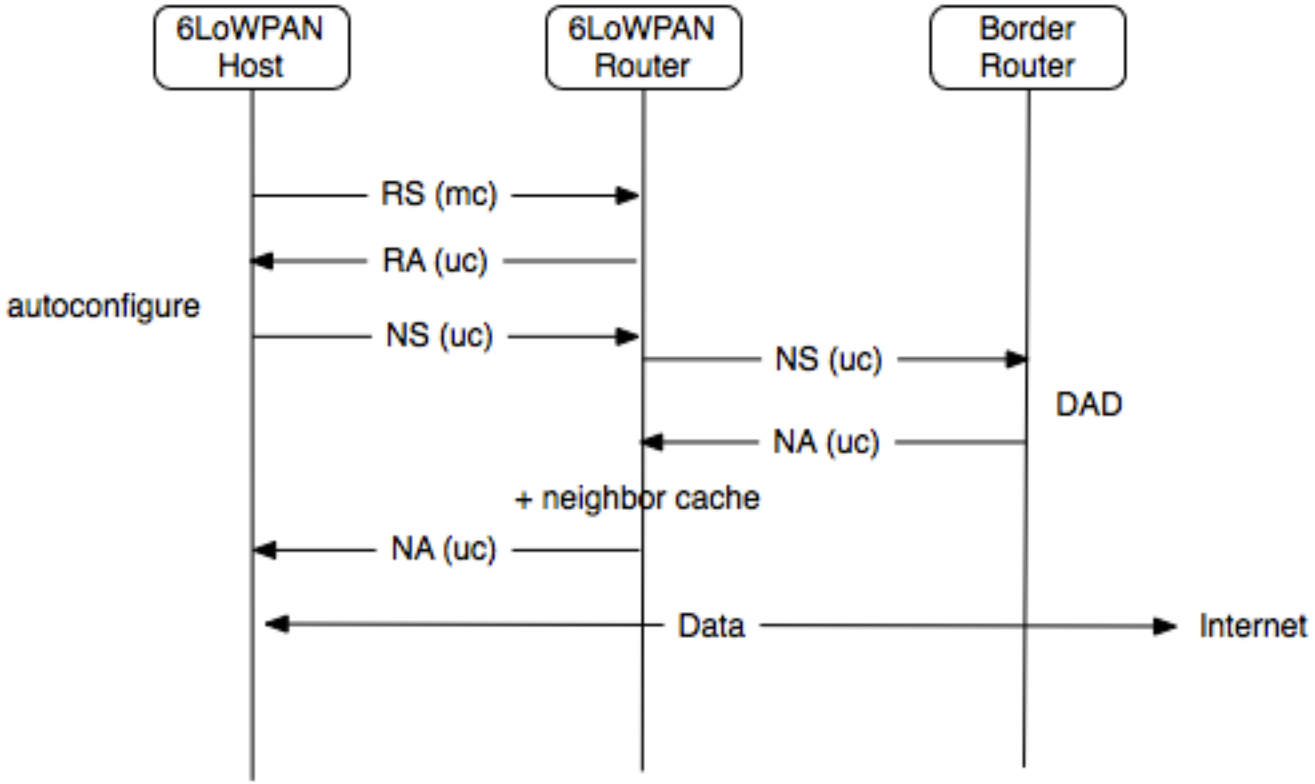
Duplicate address detection



Multihop prefix distribution



Put it all together...



Legend:
(mc) = Multicast
(uc) = Unicast

Current status

- Draft is stable (minor changes since -09)
- Useful vendor interop feedback integrated in -10
- Open issues
  - Minor nits to be fixed (#89, 90)
  - Close tickets on message size optimization (#82,88)
  - Tentative GP16 as NS src on initial registration (#87)
  - Clarification on routing protocol interaction (#91)
- Next step
  - Release nd-12 this week

78th IETF: 6lowpan WG Agenda

09:00	Introduction, Agenda	Chairs (5)
09:10	2 – HC-07	JH (20)
09:30	1 – ND	
09:30	ND-11	ZS (40)
10:10	thubert--backbone	PT (10)
10:20	RA Suppression	LT (10)
10:30	3 – Security	
10:30	HIP	RGM (20)
10:50	qiu--secure-router	YQ (10)
11:00	0 – TCP HC	LT (10)
11:10	0 – thubert--simple-frag-recovery	PT (10)

6LoWPAN
Backbone Router

(draft-thubert-6lowpan-backbone-router-02)

Pascal Thubert

6LoWPAN WG Meeting
78th IETF Meeting
Maastricht

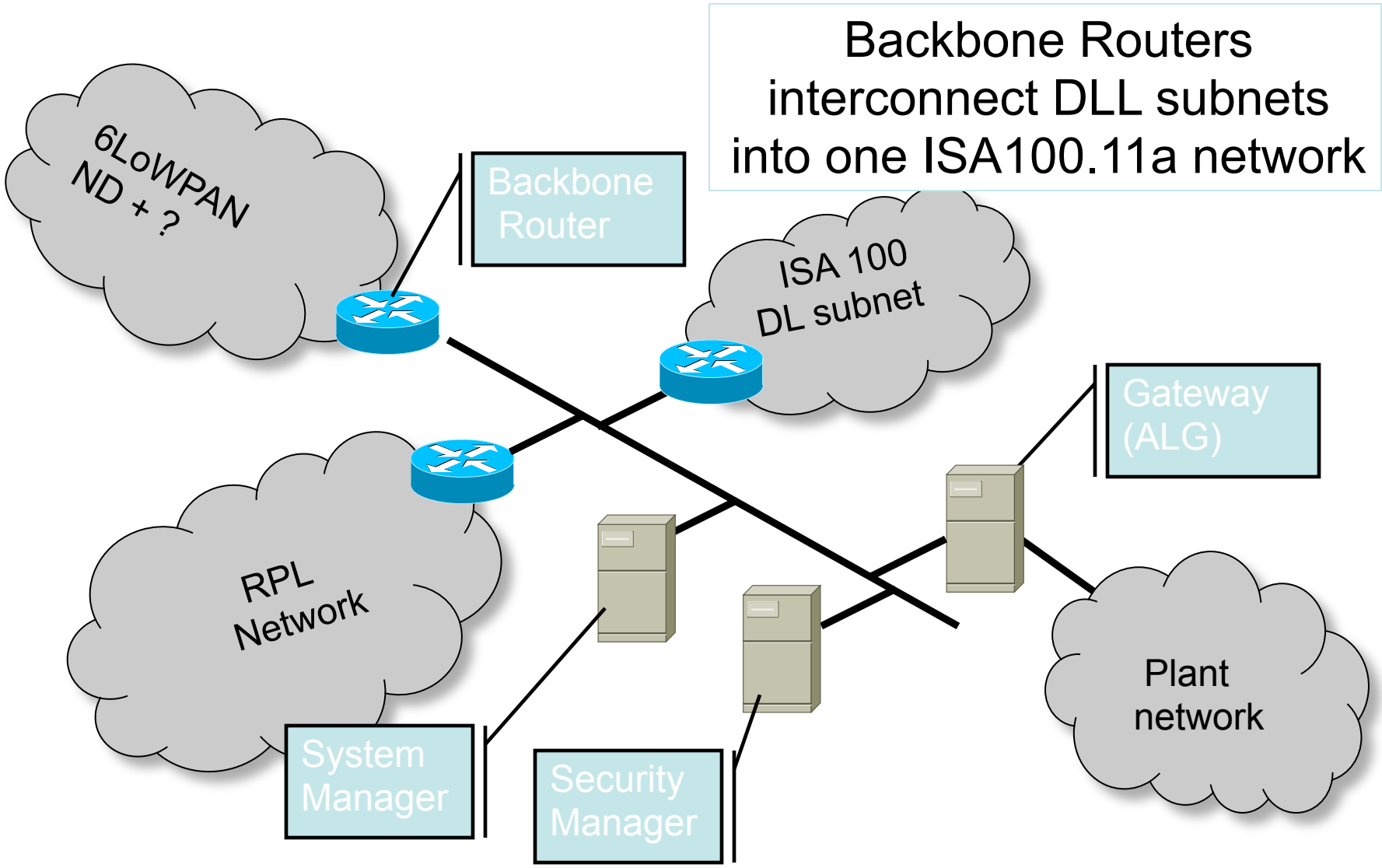
What's new?

- Split the from the ND spec
  - WG decision (Hiroshima)
- Added registration from RPL
- Removed duplicate unique ID detection
  - As discussed on the list, too complex

What's BR?

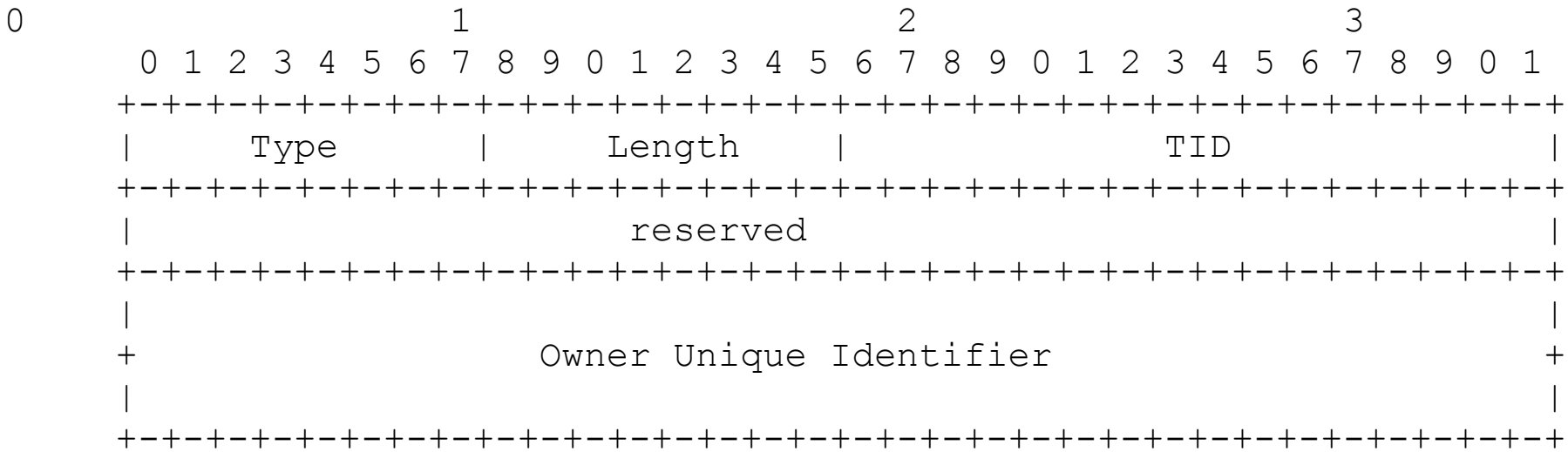
- Common ND based abstraction over a backbone
- Scales DAD operations (distributes LBR)
- Scales the subnetwork (high speed backbone)
- Allows interaction with nodes on the backbone or in other subnets running different operations

ISA100.11 reference model



Binding Tracking Option

- Used to resolve confilicts
- Need In ND: TID to detect movement
- Need In RPL: Object Unique ID for DAD



????? Questions ?????

78th IETF: 6lowpan WG Agenda

09:00	Introduction, Agenda	Chairs (5)
09:10	2 – HC-07	JH (20)
09:30	1 – ND	
09:30	ND-11	ZS (40)
10:10	thubert--backbone	PT (10)
10:20	RA Suppression	LT (10)
10:30	3 – Security	
10:30	HIP	RGM (20)
10:50	qiu--secure-router	YQ (10)
11:00	0 – TCP HC	LT (10)
11:10	0 – thubert--simple-frag-recovery	PT (10)

Suppress RA

draft-toutain-6lowpan-ra-suppression-01.txt

Laurent Toutain

Nicolas Montavont

Dominique Barthel

Why?

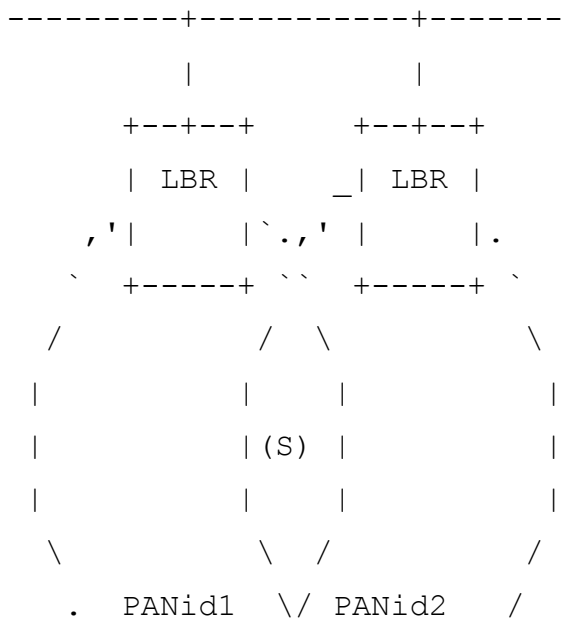
- NDP is complex to implement on LoWPAN.
- NDP consumes energy with periodic RA
- NDP adds delays before configuration

- Prefix is sometime complex to manage
  - Multi-homed network

How

- Use 6LP compression to define an **implicit prefix**:
  - Draft takes context value `OxF`
  - Either as source or destination
- For e-gress packets, the LBR uncompress 6LP header into IPv6 header
  - Add prefix
  - Adjust L4 checksum
- For in-gress packets, LBR either
  - if destination is known: uses context `0xF` to support RA suppression (checksum adjusted)

Star topology



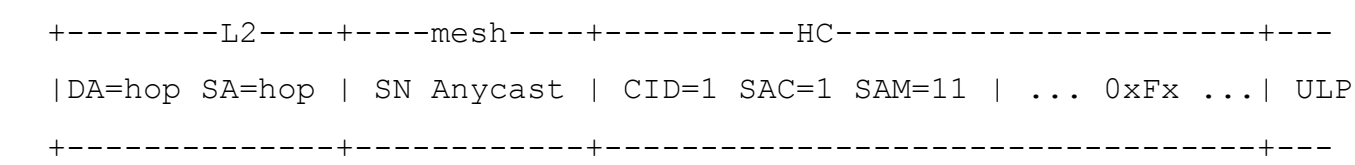
Default router:

- Use L2 anycast address
  - May create duplicates
- If LBR MAC is known; used it

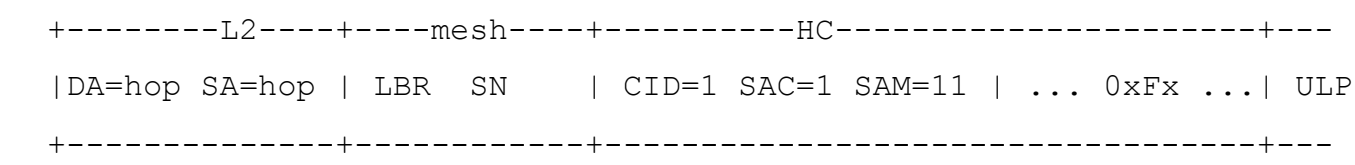
```
From Sensor Node to LBR:
  +-----------L2---------+----------6LP----------------------+---
  |DA=L2Anycast SA=SN     | CID=1 SAC=1 SAM=11 | ... 0xFx ...| ULP
  +----------------------+-----------------------------------+---

Form LBR to Sensor Node:
  +-----------L2---------+----------6LP----------------------+---
  |DA=SN SA=LBR           | CID=1 DAC=1 DAM=11 | ... 0xxF ... | ULP
  +----------------------+-----------------------------------+---
```

Mesh topology

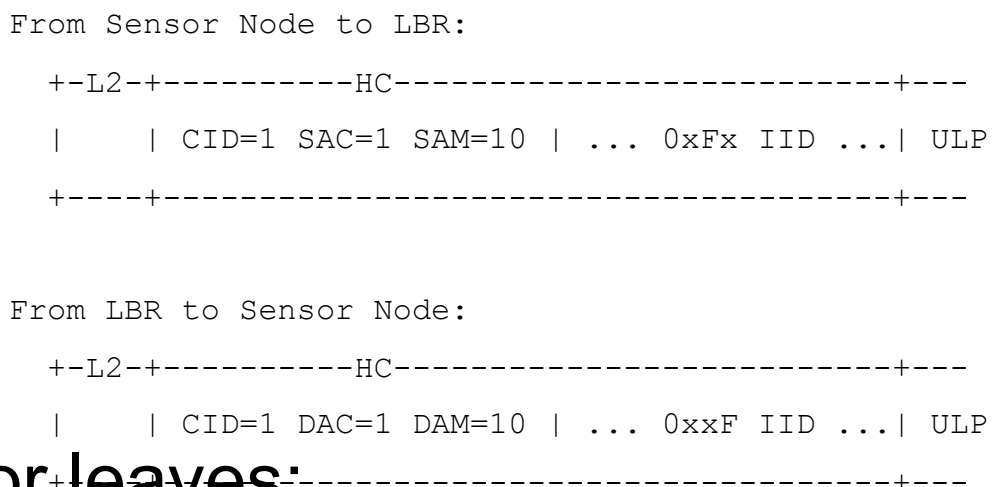
From Sensor Node to LBR:



From LBR to Sensor Node:



Routed topology



For leaves:

- Use default route
  - Listen to DIO
- Intermediate router injects IID in RPL

Conclusion

- Compatible with 6lowpan-nd
- Implicit prefix context cannot be given by NDP
  - Manual setup
  - Reserve a value in the context table?
- Implementation will be soon released
  - Star and RPL

78th IETF: 6lowpan WG Agenda

09:00	Introduction, Agenda	Chairs (5)
09:10	2 – HC-07	JH (20)
09:30	1 – ND	
09:30	ND-11	ZS (40)
10:10	thubert--backbone	PT (10)
10:20	RA Suppression	LT (10)
10:30	3 – Security	
10:30	HIP	RGM (20)
10:50	qiu--secure-router	YQ (10)
11:00	0 – TCP HC	LT (10)
11:10	0 – thubert--simple-frag-recovery	PT (10)

Using HIP DEX for 802.15.4 Key Management

Robert Moskowitz
ICSA labs
an Independent Division of
Verizon Business

July 25, 2010

rgm@labs.htt-consult.com

Purpose of this presentation

- Present work on a new HIP Exchange specifically architected for resource limited devices by
  - Explain what HIP is and does and why 802.15.4 and 6lowpan should consider using it
  - Explain the new HIP Diet Exchange (HIP DEX)
  - Explain how HIP DEX would work in a PAN
  - A call for action

What is HIP?

- RFC 4423 introduces the Host Identity Namespace. When the Host Identity (HI) is a Cryptographic key (RSA, DSA, or ECC)
  - 128 bit Host Identity Tag (HIT) is derived from the HI (hashed) and functions as an IPv6 address (/28 prefix) for applications

What is HIP?

- A 4 packet Peer-to-Peer Host Identity Protocol Base EXchange (HIP BEX) establishes a security association (SA, similar to IKE), indexed by the HITs, but independent of the IP address
  - HIP's notion of an End Point Identifier (the HITs) disassociates the current tight binding between the Internetwork and Transport layers
  - Can even function directly on layer 2
- The SA is used to key ESP (RFC 4304) in transport mode
  - Or could key IEEE 802.15.4 MAC layer security

Why Consider HIP

- Although HIP is an IP layer KMS, it is independent of IP
  - The same KMS can function at the MAC layer
- HIP is constructed with long-used and well understood crypto components
  - It is 'easy' to analyze
- HIP does not need backend validation systems
  - It works well with ACLs

What is the role of HITs?

- In HIP the End Point Identifier is
  - Host Identity Tag (HIT) in IPv6
  - Local Scope Identifier (LSI) in IPv4
  - HITs and LSIs are typically only known to the applications and do not transit the network
  - Applications tend to be ignorant of underlying IP addresses, if any
    - Secure mobility WORKs (RFC 5206)
    - IPv4 applications on IPv6 networks
    - No IP, datagrams transported directly over MAC

More on HIP

- HIP is architecturally ideally suited to be a Key Management System (KMS) for both IP and MAC layers
- Current status
  - RFC 4423, 5201-5206
  - Three implementations
    - Boeing, Ericsson, HIPL
    - Boeing uses HIP on 777 line, SMA plans in place
  - Going through revisions, -bis Internet Drafts available

Putting HIP on a Diet
Basic premise

- The HIP Diet EXchange – HIP DEX
- Use static ECDH as Host Identities
- With ECDH derived key only used for session key protection
  - Master Key in 802.11 terminology
  - Randomly generated a key and encrypted with DH derived key
    - CMAC function now defined for Diffie-Hellman key as the the Master Key
    - Key derivation from random key can use CMAC
- We do not need a hash function!
- We can 'manage' without Digital Signatures

Putting HIP on a Diet
Proof of Identity

- Nonce encrypted with Diffie-Hellman key 1st proof
- Diffie-Hellman key used in MAC of HIP payload 2nd proof
- Thus sender of packet must have private key matching HI
- The WHO of the HI is outside of HIP
  - Various methods used, but for DEX can't use a hash!
  - ACLs
  - Anonymous with password authentication

Putting HIP on a Diet
What security assertions lost?

- Use of static DH means loss of Perfect Forward Secrecy (PFS)
  - Static DH (NIST SP 800-56A sec 6.3.2) used as device identities
  - If Private key is compromised, all prior secrets encrypted with it are compromised

Putting HIP on a Diet
Summary of Crypto Components

- A 'Dietetic' HIP exchange CAN be achieved with
  - AES-CBC (and CMAC)
    - AES-CCM used by ESP or MACsec
  - Static ECDH
    - Exchange proves private key ownership
    - Can be installed by manufacturer
      - ECDH key derivation typically only occurs for initial join

HIP Diet Exchange (DEX)
Dealing with a lossful network

- HIP BEX can be slow with packet loss
  - DEX MUST deal with high packet loss
- Implement a repeated send until ACK
  - Alternative to 802.15 immediate ACK
    - Which is not effective on multihop or off PAN
  - I aggressively sends I1 and continues send it until it receives R1
  - R sends R1 for every I1 received
  - I aggressively sends I2 and continues send it until it receives R2, then it transitions to connected state
  - R sends R2 for every I2 received, it transitions to connected state when it starts receiving datagrams

HIP Diet Exchange (DEX)
Adding Password Authentication

- Password Augmented Authentication
  - Provides bootstrap mechanism to add a node to a controller
  - Supports emergency AdHoc access
    - EMT access to a Pacemaker
    - Utility field technician to a substation controller
- Controller implicitly invites password Auth
  - R1 ALWAYS contains a challenge
    - The Puzzle values
  - Initiator MACs challenge with password and encrypts that in the DH derived key

HIP Diet Exchange (DEX)
Adding Password Authentication

- Challenge Encryption
  - Use password as CMAC key
    - MAC nonce from R1 puzzle
      - RFC 4615 (AES-CMAC-PRF-128) is starting point
    - Encrypting a challenge from R1 prevents replay attacks
      - R1 cannot be reused if password response is accepted
  - 'Rogue' Responder attack
    - Initiator cannot tell if R1 came from Responder or attacker unless PKr from another source
  - Need zero knowledge alternative
    - As in IEEE 802.11s SAE
    - And draft-harkins-ipsecme-spsk-auth

The Importance of Randomness

- HIP DEX is HIGHLY dependent on good Random numbers
  - No Hash function typically used in pseudo random number generators
  - Many underlying assumptions on randomness
- An analog approach is in 802.11 Annex H
- RFC 4615 starting with a REAL random seed
- Most keys built from random number from both parties
  - But this is not an excuse to NOT use something decent

Using HIP DEX for MACsec

- Use 6lowpan for HIP directly over MAC layer
  - HIP I2 packet is at least 180 bytes
  - Sec 5 for fragmentation
- Develop pair-wise and broadcast/multicast key distribution
  - HIP DEX has implicit concept of Master and Pair-wise keys
  - Use 802.11 Group key model
    - Or 802.1AE?
- ICMP error messages
  - Remove IP header and run directly over 6lowpan

HIP DEX exchanges

- DEX provides Master and Pair-wise Keys
  - On initial joining of PAN and whenever new MK needed (eg lost state)
  - Accelerated Group key setup within exchange
    - Only if Responder is owner of key

HIP DEX exchanges

- Pair-wise Key Updates
  - Via HIP UPDATE exchange
  - Frequency determined by local policy
    - Lost state or key exhausted
  - Only AES-CBC and CMAC functions needed
- Group Key
  - Via HIP UPDATE exchange
  - Sent by key owner
  - Frequency determined by local policy
    - Lost state, membership change, key exhausted
  - Only AES-CBC and CMAC functions needed

Next Steps

- Form working team
  - IETF and IEEE 802.15 HIP Interest Group
- Refine processes
  - HIP DEX
  - MACsec key hierarchies management
- Present progress at IEEE 802.15 Interim in September

78th IETF: 6lowpan WG Agenda

09:00	Introduction, Agenda	Chairs (5)
09:10	2 – HC-07	JH (20)
09:30	1 – ND	
09:30	ND-11	ZS (40)
10:10	thubert--backbone	PT (10)
10:20	RA Suppression	LT (10)
10:30	3 – Security	
10:30	HIP	RGM (20)
10:50	qiu--secure-router	YQ (10)
11:00	0 – TCP HC	LT (10)
11:10	0 – thubert--simple-frag-recovery	PT (10)

Lightweight Secure Router Protocol in Dynamic Sensor Networks

Ying QIU, Jianying ZHOU, Feng BAO

A*STAR I²R

Motivation

- The demand of wireless sensor networks is growing exponentially.
- A moving sensor mote needs to change its attached routers (or cluster heads) frequently.
- The router (or cluster head) needs to ensure the joining mote is not a malicious sensor.
- The moving mote needs to establish a security tunnel with the new route (or cluster head).

Problem Statement (RFC4919)

- Resource limitation
  - power, computation, communication, memory …
- Wireless nature of communication;
  - open wireless channel, everyone can catch the traffic packets
- Very large and dense WSN;
  - ID and key management
- Location not predefined and moving
  - impossible to pre-configure
- Unreliable devices
  - high risk of physical attacks to unattended sensors;
- Small packet size
- Support 16 and 64 bit addresses

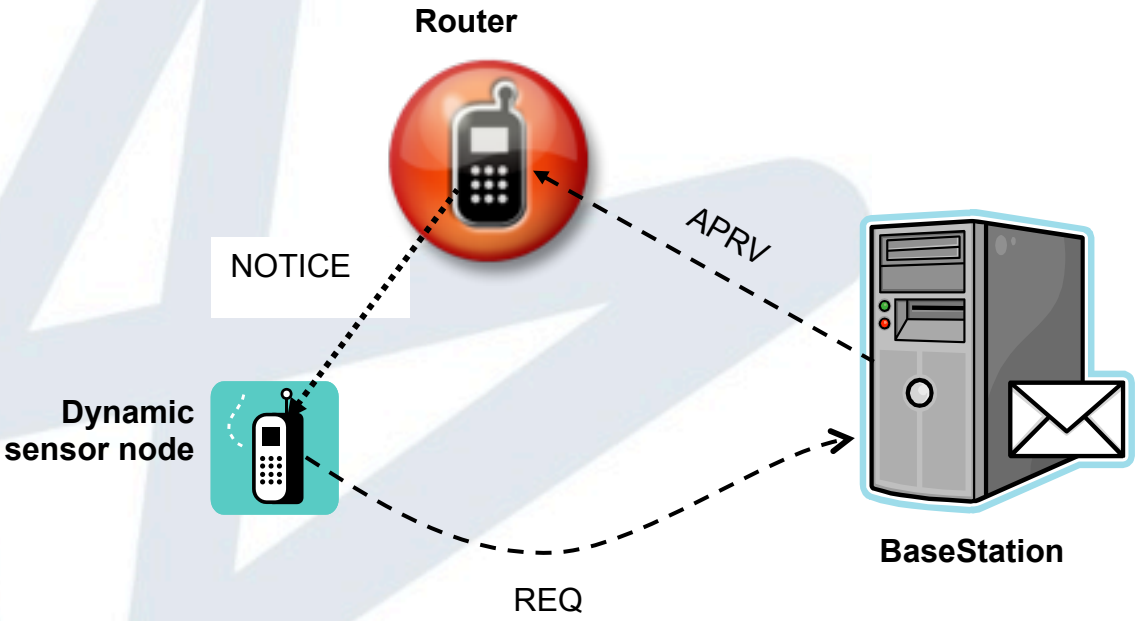
Protocol

- Shared key discovery:
  - each sensor only store a small set of keys randomly selected from a key pool at the deployment. Two nodes may use the key discovery protocol to find a common key from their own sets.
  - saving communication
- Key establishment and update:
  - an efficient and scalable scheme to establish and update the keys among nodes.
- Authentication and encryption:
  - describe how to use node's ID information to authenticate and encrypt the traffic packets.
- Distribution Mode:

Shared Key Discovery

- reduce communication
  - data transmission costs much energy than the computing
- a sensor may only store a small set of keys randomly selected from a key pool at the deployment
  - a sensor's memory is very limited
  - neighbors are impossible to pre-configure
- two nodes may use any existed key discovery protocol to find a common key from their own sets.

Key Establishment



$$req = \{src=ID,\ Dst=\ BS,\ \ RT \,||\, R_0 \,||\, MAC(K_{BN},\ ID||RT||R_0)\ \} \quad (1)$$

$$K_{NR} = H(K_{BN},\ ID||\ R_0\ ||\ R_1\ ) \quad (2)$$

$$aprv = \{src=BS,\ dst=RT,\ \ E(K_{BT},\ ID||R_0||R_1||\ K_{NR}\ )\} \quad (3)$$

$$notice = \{src=RT,\ Dst=ID,\ R_0\ ||\ R_1\ ||\ MAC(K_{NR},\ RT||ID||\ R_0||R_1\ )\} \quad (4)$$

Key Management

Key Cache in Sensor Node N		
Correspondence Node ID	**Key**	**Key Lifetime**
BS	K_{BN}	T_{BN}
$node_i$	K_{Ni}	T_{Ni}
… …	… …	… …
$node_j$	K_{Nj}	T_{Nj}
$SharedKey_x$	K_x	T_x
… …	… …	… …
$SharedKey_y$	K_y	T_y

Correspondence Node ID	Key	Key Lifetime
$node_R$	R_0	*0*

I²R
A*STAR

Key Management

Key Table in Base Station		
Node ID	**Key Stuff**	**Key Lifetime**
$node_i$	K_{Bi}	T_{Bi}
… …	… …	… …
$node_j$	K_{Bj}	T_{Bj}

Distribution Mode

1) Each cluster head manages to establish the shared key with its neighbour cluster heads after deployment.
2) Each sensor node keeps two base station IDs: real base station ID and sub-base-station ID.
3) After deployment, the first round for a mobile node to establish the shared key with the nearest cluster head uses the basic protocol.
4) When the mobile node moves, use the basic protocol to establish the shared key with the new cluster head, via the sold cluster head rather than the real base station.
5) After successfully establishing the keys, the sensor node updates the ID of sub-base-station with the current cluster head.
6) For security reasons, each sensor node must reset its sub-base-station ID to the real base station at a specified interval (say a few hours or days, depending on the various applications) and re-establish keys with its near cluster heads via the real base station. If the base station does not receive any request from a sensor node, it considers the sensor node has been compromised.

Features

- Suitable for both static and dynamic WSN. Any pair of nodes can establish a key for secure communication.
  - Easily scalable
- A roaming note only deals with its closest router for security. No need to change the rest routing path to the base station.
  - Less signalling, hence less power cost
- Base station can manage the revocation list for lost or compromised roaming motes.
  - Stronger security
- System is scalable and resilient against node compromise.
  - Stronger security

Satisfy the Routing Requirements (1)

draft-ietf-6lowpan-routing-requirements

[R01] 6LoWPAN routing protocols SHOULD allow implementation with small code size and require low routing state to fit the typical 6LoWPAN node capacity.

Yes. The cache of key management is variable.

[R02] 6LoWPAN routing protocols SHOULD cause minimal power consumption by the efficient use of control packets and by the efficient routing of data packets.

Yes. Reduce the number and size of signalling messages.

[R03] 6LoWPAN routing protocol control messages SHOULD NOT exceed a single IEEE 802.15.4 frame size in order to avoid packet fragmentation and the overhead for reassembly.

Yes. Every signalling message is included in 1 packet.

Satisfy the Routing Requirements (2)

[R05] The design of routing protocols for LoWPANs must consider the latency requirements of applications and IEEE 802.15.4 link latency characteristics.

Yes. Distribution mode.

[R06] 6LoWPAN routing protocols SHOULD be robust to dynamic loss caused by link failure or device unavailability either in the short term or in the long term.

Yes. The use of nonce R0 & R1 as well as key revoketion.

[R07] 6LoWPAN routing protocols SHOULD be designed to correctly operate in the presence of link asymmetry.

Yes.

[R08] 6LoWPAN routing protocols SHOULD be reliable despite unresponsive nodes due to periodic hibernation.

Yes. The use of nonce R0 & R1.

[R09] The metric used by 6LoWPAN routing protocols MAY utilize a

A*STAR I²R

Satisfy the Routing Requirements (3)

[R10] 6LoWPAN routing protocols SHOULD be designed to achieve both scalability from a few nodes to maybe millions of nodes and minimality in terms of used system resources.

Yes. The protocol guarantees that two sensor nodes share at least one key with probability 1 (100%)

[R11] The procedure of route repair and related control messages should not harm overall energy consumption from the routing protocols.

N/A

[R12] 6LoWPAN routing protocols SHOULD allow for dynamically adaptive topologies and mobile nodes.

Yes. It's the motivation of designing the protocol

[R13] A 6LoWPAN routing protocol SHOULD support various traffic patterns.

N/A

Satisfy the Routing Requirements (4)

[R15] When a routing protocol operates in 6LoWPAN's adaptation layer, routing tables and neighbor lists MUST support 16-bit short and 64-bit extended addresses..

Packet format will be defined in future.

[R16] In order to perform discovery and maintenance of neighbors, LoWPAN Nodes SHOULD avoid sending separate "Hello".

N/A

[R17] In case there are one or more nodes allocated for the specific role of local management, such a management node MAY take the role of keeping track of nodes within the area of the LoWPAN it takes responsibility for.

Yes. The function of sub-basestation

[R18] If the routing protocol functionality includes enabling IP multicast, then it may want to employ structure in the network for efficient distribution, or relay points

A*STAR I²R

Future Works

- Define the transmission format.
- Feedback and improve.

A*STAR I²R

78th IETF: 6lowpan WG Agenda

09:00	Introduction, Agenda	Chairs (5)
09:10	2 – HC-07	JH (20)
09:30	1 – ND	
09:30	ND-11	ZS (40)
10:10	thubert--backbone	PT (10)
10:20	RA Suppression	LT (10)
10:30	3 – Security	
10:30	HIP	RGM (20)
10:50	qiu--secure-router	YQ (10)
11:00	0 – TCP HC	LT (10)
11:10	0 – thubert--simple-frag-recovery	PT (10)

TCP Header Compression for 6LoWPAN

draft-aayadi-6lowpan-tcphc-00.txt

Ahmed Ayadi, David Ros and Laurent Toutain

6LoWPAN WG Meeting

IETF 78, Maastricht, Netherlands, July 25-30, 2010

LOWPAN_TCPHC

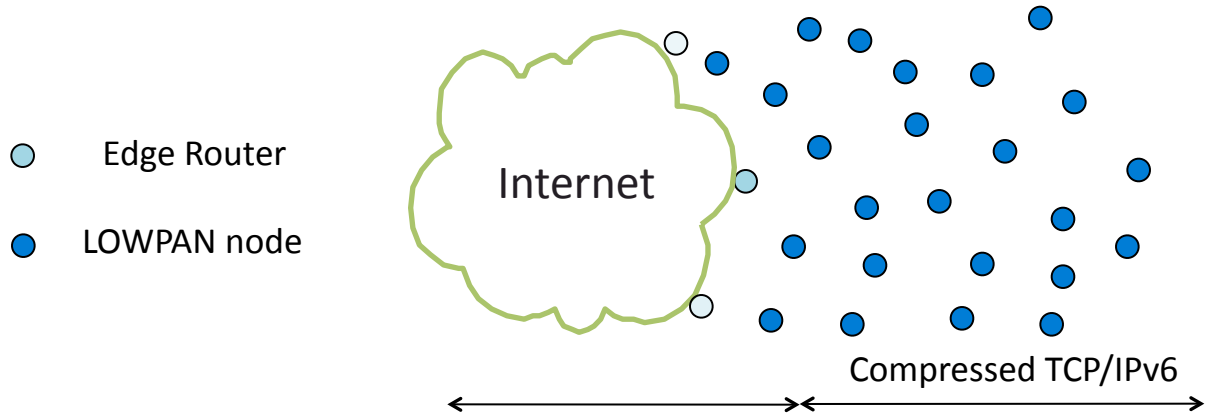
Motivation

- Transmission Control Protocol (TCP) is the most used transport protocol in the Internet.
- TCP can provide useful services for Low power and Lossy Networks such as SSH, TELNET, HTTP.
- Currently, LOWPAN_IPHC defines only a compression scheme for UDP (LOWPAN_NHC).
- **Define a TCP scheme compatible with 6LoWPAN and adapted to LOWPAN**.
- Outside to LoWPAN, LoWPAN to outside, LoWPAN to LoWPAN

LOWPAN_TCPHC

Overview

- LOWPAN_TCPHC is implemented on the Edge Router and on the LOWPAN nodes which save the context of the connections.
- LOWPAN_TCPHC does not compress TCP segment in the “connection establishment” phase (SYN), removes the unused fields (Reserved), and replaces the source port and destination port by a Context Identifier (CID),
- LOWPAN_TCPHC sends only changed bytes of dynamic fields (Seq. Num, Ack Num and Window)
- LOWPAN_TCPHC compresses SACK and Timestamp options



LOWPAN_TCPHC

Compressed TCP header types

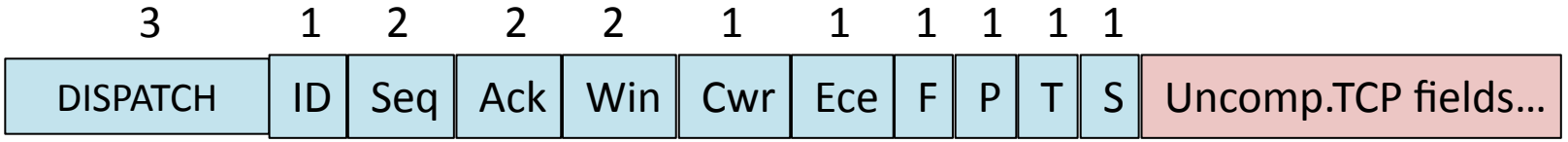
- Regular header (sent out-side LLNs)

- Full header (sent at the Con. Estab. Phase)

- Compressed header

LOWPAN_TCPHC

Header Format



- DISPATCH: types of the TCP header
- ID: the CID field size
- Seq, Ack, Win: fields size

Flags:

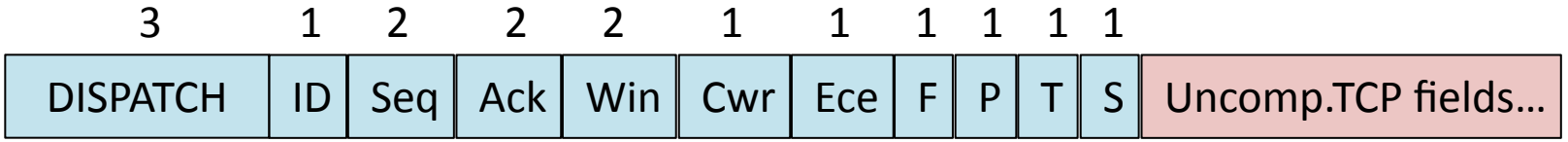
- Cwr: Congestion window reduce
- Ece: Explicit congestion notification
- F: FIN flag
- P: Push flag

Options:

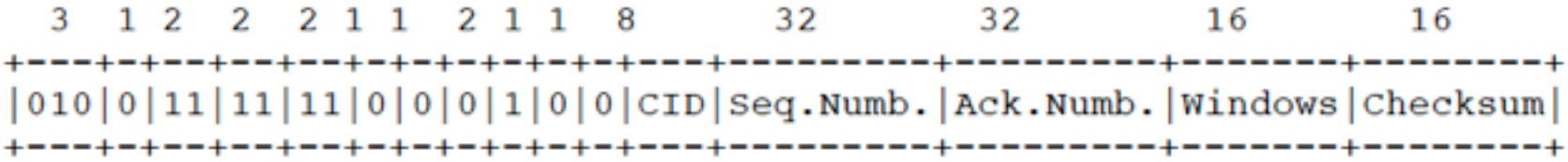
- T: Timestamp option
- S: SACK option

LOWPAN_TCPHC

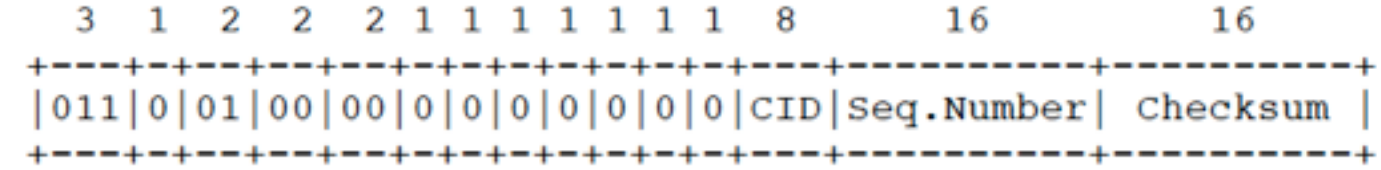
Examples



- Mostly compressed TCP header



- Compressed TCP header



LOWPAN_TCPHC

TCP Options

- MSS and SACK-permitted options are sent uncompressed in SYN segments,
- We assume that the other TCP options such as Window Scale Option (WSO) are useless in LLNs due to the memory constraint of the embedded devices.
- LOWPAN_TCPHC defines Compression for SACK and Timestamp options,
- SACK:
  - Only 1 SACK block is allowed with LOWPAN_TCPHC,
  - Left Edge and right Edge of Sack block are replaced by the offset to the acknowledgment number,
- Time Stamp:
  - Tsval are sent only if the TCP sends data.
  - Tsecr are sent by the the TCP sink
  - Otherwise, the eight bytes should be sent

LOWPAN_TCPHC

Preliminary results

- LOWPAN_TCPHC reduces the TCP header to 6 bytes in more than 95% of cases.
- LOWPAN_TCPHC reduces the transmit by about 14% in a one-hop scenario.

LOWPAN_TCPHC
Conclusion

- LOWPAN_TCPHC is already implemented in Contiki OS.
- The experimental performance evaluation of TCPHC on Telecom Bretagne,

- LOWPAN_TCPHC WG item ?

78th IETF: 6lowpan WG Agenda

09:00	Introduction, Agenda	Chairs (5)
09:10	2 – HC-07	JH (20)
09:30	1 – ND	
09:30	ND-11	ZS (40)
10:10	thubert--backbone	PT (10)
10:20	RA Suppression	LT (10)
10:30	3 – Security	
10:30	HIP	RGM (20)
10:50	qiu--secure-router	YQ (10)
11:00	0 – TCP HC	LT (10)
11:10	0 – thubert--simple-frag-recovery	PT (10)

6LoWPAN
Simple Fragment Recovery

(draft-thubert-6lowpan-simple-fragment-recovery-07)

- Pascal Thubert/ Jonathan Hui

6LoWPAN WG Meeting
78th IETF Meeting
Maastricht

What's new

- Fragment forwarding
  - Using the datagram tag as a switchable label
  - Acks are used to clean intermediate states
- Compressed ack bitmap
- Offsets expressed in the compressed packet
  - Clean layering
  - Removes the first fragment issue

Need for fragment recovery

- Considering
  - that 6LoWPAN packets can be as large as 1280 bytes
  - that Source routing requires space for routing headers
  - that a 802.15.4 frame with security will carry in the order of 80 bytes of effective payload,

=> An IPv6 packet might be fragmented into > 16 fragments at the 6LoWPAN shim layer.

- This level of fragmentation is much higher than that traditionally experienced over the Internet with IPv4 fragments, already known as harmful.
- At the same time, the use of radios increases the probability of transmission loss but retry only 1 hop

Other problems related to frags

- Hop by Hop recomposition
  - Should be avoided: latency and memory hit
- Multipath
  - Forwarding fragments over multipath multiplies the impact of an anomaly
- Recovery buffers Lifetime
  - Terminating device with limited capacity may have trouble maintaining buffers. How long?
  - Intermediate routers congestion

Fragment Recovery proposal

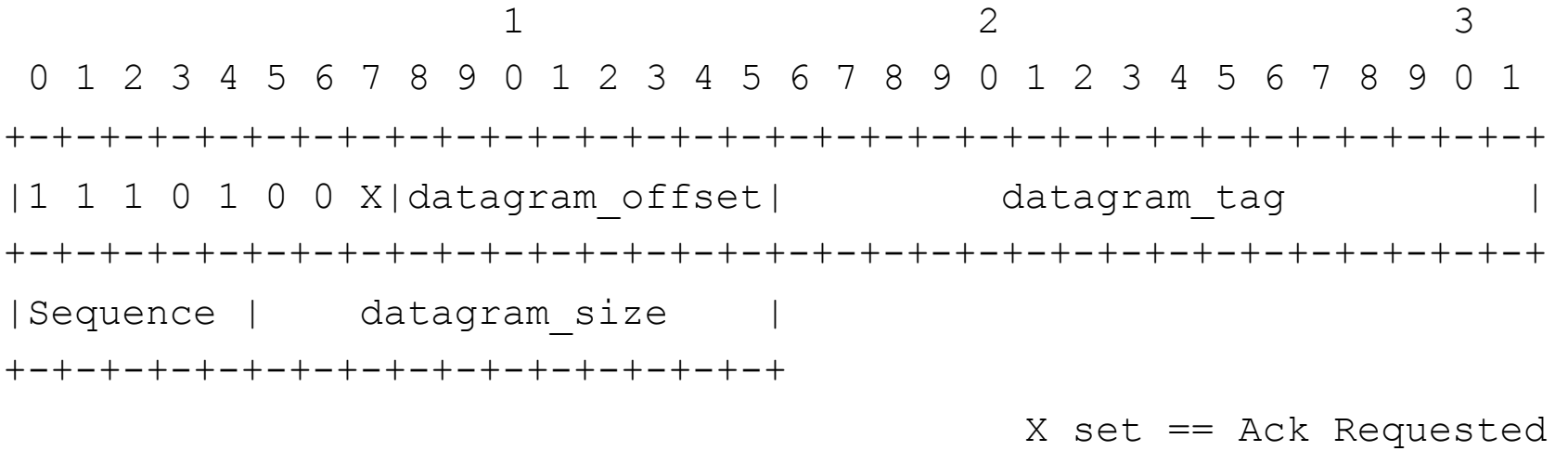
- 32 bits SAck Bitmap
- Variable window size for congestion control
- Round Robin for multipath
- 4 new dispatch types

Pattern	Header Type
11 101000	RFRAG - Recoverable Fragment
11 101001	RFRAG-AR - RFRAG with Ack Request
11 10101y	RFRAG-ACK - RFRAG Acknowledgement
	(y reserved for ECN)

Fragment Forwarding proposal

- Frags & Acks have a datagram tag
- Unique for the source if the tag
- Proposal uses the datagram tag as a label
- First fragment sets up a bidir label path
- Final ack & errors clean it up
- Next fragments are label swapped along the same path

Recoverable Fragment Dispatch type and Header



X (check) bit

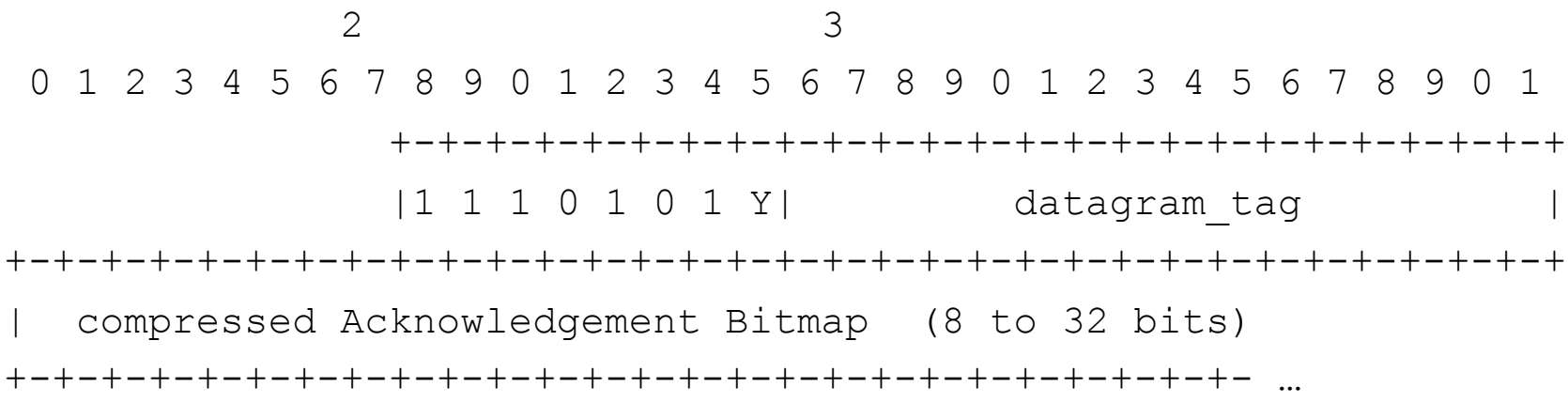
When set, the sender requires an Acknowledgement from the receiver

Sequence

The sequence number of the fragment.
Fragments are numbered [0..N] where N is in [0..31].

Fragment Acknowledgement Dispatch type and Header

The ack bitmap is now compressed:



Bitmap expansion pattern

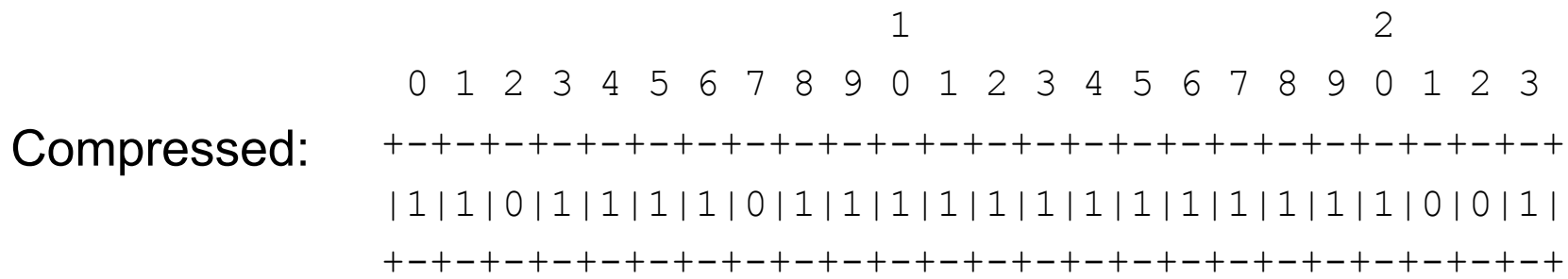
A 32 bits uncompressed bitmap is obtained by prepending zeroes to the XXX in the pattern below.

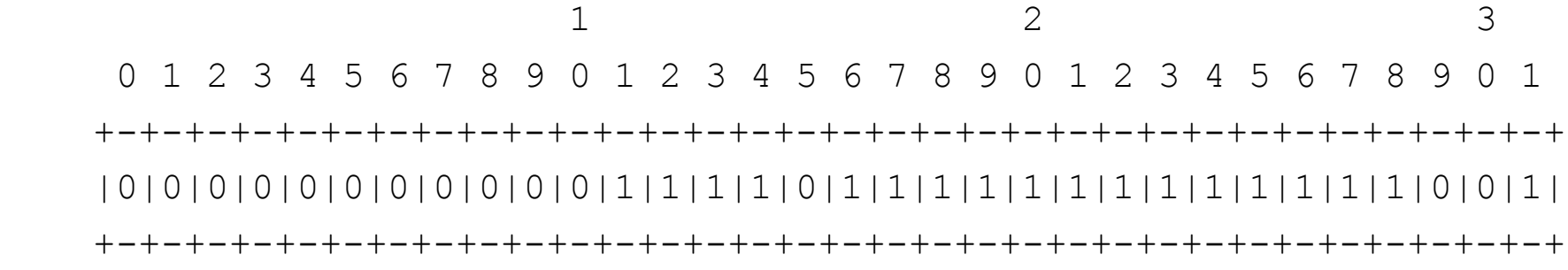
Pattern	Size	Ack
0XXXXXXX	1 octet	1 -> 7
10XXXXXX XXXXXXXX	2 octets	1 -> 14
110XXXXX XXXXXXXX XXXXXXXX	3 octets	1 -> 21
1110XXXX XXXXXXXX XXXXXXXX XXXXXXXX	4 octets	1 -> 28

Bitmap expansion example

Example for a 3 octets pattern (110) :

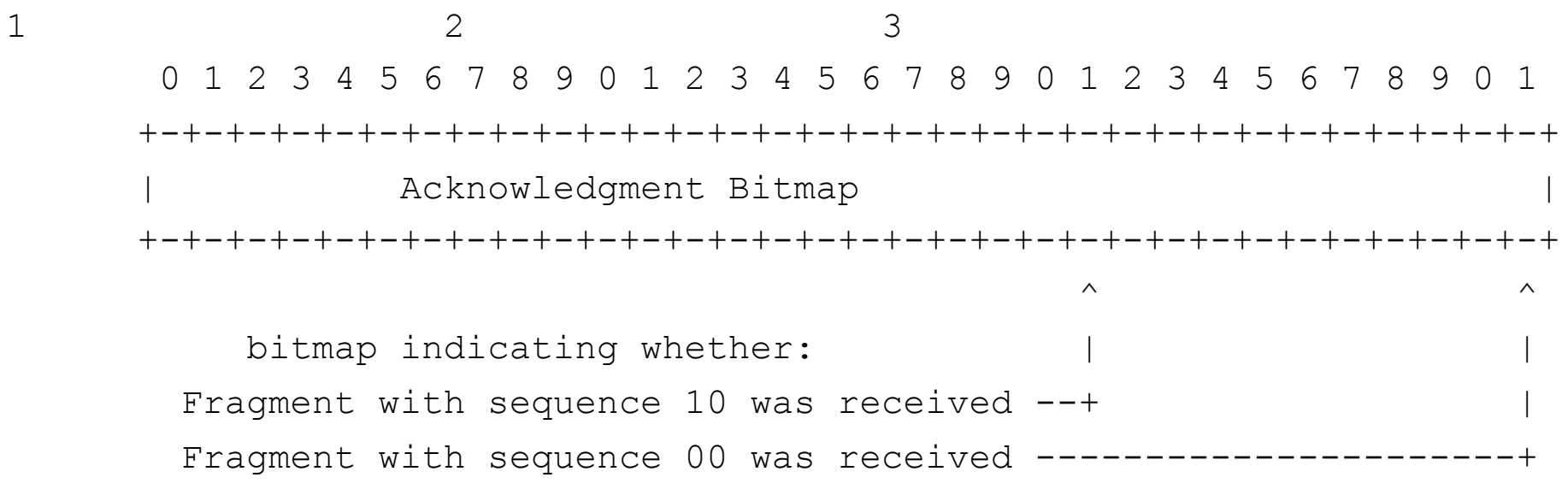
Compressed:





Expanded bitmap

The resulting bitmap reads as follows:



????? Questions ?????

ECN use

- Indicate Congestion in the LoWPAN
  - End to End effect on Transport
  - Required by ISA100.11a
  - Local Effect on Fragment flow control
- Early detection
  - Avoid Wasteful discard of packets
  - Conditions equivalent to RED

Explicit Congestion Notification

- ECN in IPv6: Traffic Class bits 6-7

Binary	Keyword	References
00	Not-ECT (Not ECN-Capable Transport)	[RFC 3168]
01	ECT(1) (ECN-Capable Transport(1))	[RFC 3168]
10	ECT(0) (ECN-Capable Transport(0))	[RFC 3168]
11	CE (Congestion Experienced)	[RFC 3168]

  - Not compressed separately by 4944
  - Added to draft-ieft-6lowpan-hc

- ECN Echo
  - Not an IP function (usually transport)
  - Thus provided by this draft between fragmentation endpoints