Trust-to-Trust Protocol (T2P)

Raimo Kantola
Aalto University
Department of Communications and Networking (Comnet)

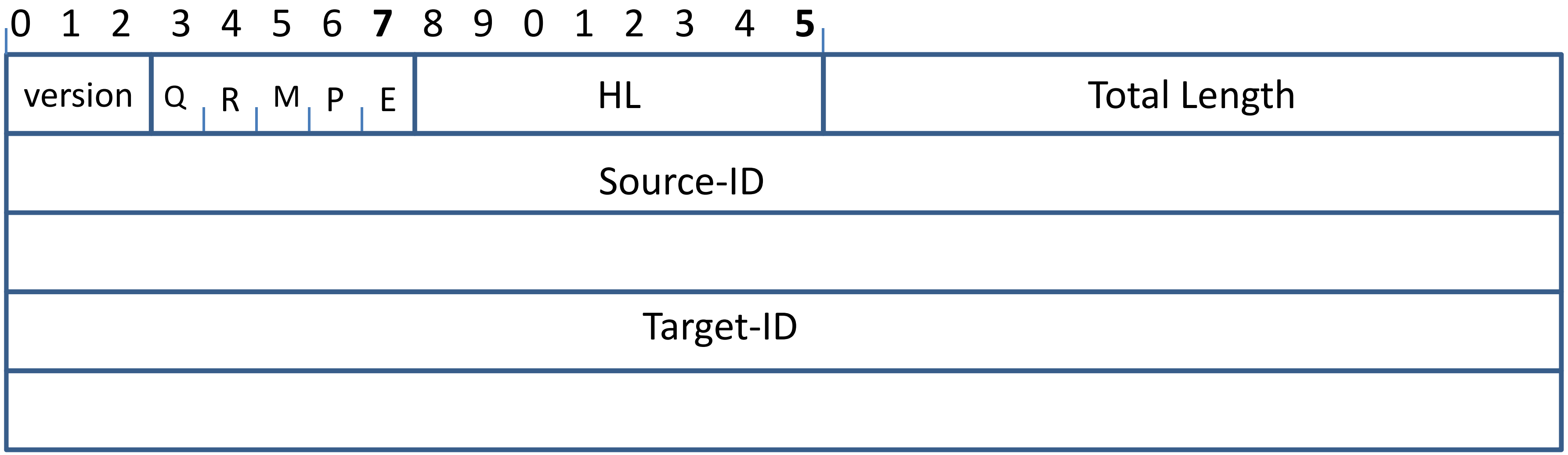
T2P Requirements

- Enhance trust between 2 customer networks and users in the 2 customer networks by facilitating IP trace back and return routability checks and thus help to ensure non-repudiation of communication.
  - T2P lets the inbound edge decide whether it wants to exclude source address spoofing etc before it admits communication
  - Inbound edge node can collect history information about RLOCs and IDs and use that as the basis for packet admission
- Carry identities edge to edge

- Operate multi-homed edge functions by providing on-demand routing through the multi-homed edge

  T2P could be modeled as
  - a protocol on top of UDP or
  - A new protocol codepoint in IP header could be defined (in parallel with UDP, TCP, SCTP etc) or
  - a new ethertype could be defined and T2P would then be carried over Ethernet directly

Protocol Header



0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

version | Q | R | M | P | E | HL | Total Length

Source-ID

Target-ID

Version – Protocol Version, for now = 1

HL – Header Length in octets, here shown as HL = 20 (range: 4…255), HL includes 1st word, IDs and T2P control data formatted as TLV elements.

Q – 1 for Query, 0 for data message when response on T2P level not expected

R – 1 , for response, 0 for data message without prior query

M – Monitoring Flag

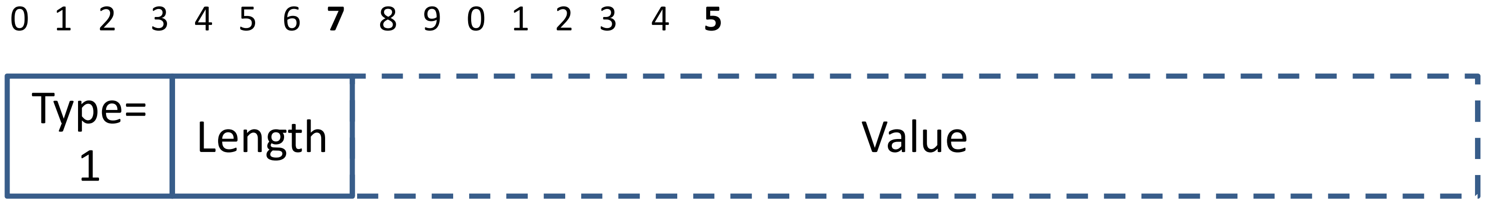
P – Puzzle Flag

E – Extension (for now 0, 1 = Flags extended by 1 octet)

Total Length – message length in octets including this word, IDs, control data and payload data

ID encoding

- ID's can be random values generated by CES based on their own algorithms or Mobile Operator assured IDs can be used. The latter could be e.g. MSISDN number, a derivative of the MSISDN or IMSI number that can be checked from HSS/HLR.



0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

Type= 1	Length	Value

Type=1 → Random ID generated by CES based on its own algorithm

Type=2 → Mobile operator assured ID. CES can query HSS/HLR to check that the ID exists and is valid.

Types: 3…15, 0 reserved for future use.

Value: if BCD encoded, padded to octet boundary from the left.

If Flag M=1, both Q&R carry RLOC TLV(s)

- Query may carry and Response MUST carry

0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

<table>
<tr><td>Type=16</td><td>Length</td><td>RlocOrder</td><td>Rloc Pref</td></tr>
<tr><td colspan="4">Rloc</td></tr>
</table>

M=1 → either Q or R must be set (in this TLV type)
Type=16 = TLV contains info on IPv4 RLOCs
Length = 6* NROF RLOCs
Rloc Order – low values are preferred (over all Rloc types), when suitable found, stop
RlocPref – low values preferred, can use all Rlocs with same Order to share load
=0xFE = prepare flow switchover to preferred Rloc,
=0xFF = do not use Rloc (has probably failed)

NB1: Rlocs are always sender's routing locators.
NB2: Type=17 – reserved for IPv6 Rlocs, Type=18 – MAC Rlocs (48 bit), Type=19….31 other Rloc types (RlocOrder and RlocPref apply to all these types).

On-demand multihoming routing mechanics (1)

- Learning RLOCs:
  - Outbound CES can learn all inbound CES RLOCs and their default state from the DNS query
  - Inbound CES can use T2P to learn RLOCs of the outbound CES
    - T2P Query MAY contain RLOCs: if current state of RLOCs at Inbound edge differs from default as stored in DNS, Query carries the current preferences to outbound edge
    - If there is no ongoing session with the requestor's RLOC and ID, we recommend to ignore the request
    - T2P Response MUST contain one RLOC that appears as source RLOC on the forwarding layer in the inbound CES, T2P response MAY contain other RLOCs
- Monitoring RLOCs
  - T2P can be used to monitor and report the state and state changes of all alternative RLOCs
  - Connection state TTL sets the pace of monitoring
  - CES may accept packets for an ongoing session from all alternative source RLOCs

On-demand multihoming routing mechanics (2)

- Swapping remote RLOC
  - If CES receives a Q/R/M=1 message with sender's RLOCpref=0xFE for which there is ongoing session, CES SHOULD immediately select a new target RLOC and make that the current target RLOC for the session
  - Having requested an RLOC switchover, CES MUST immediately start accepting traffic for the ongoing session using any alternative local RLOC
  - If there are 2 local CES systems, by making the local IP addresses that are allocated to remote hosts virtual, we may be able to hide the RLOC swap from one local CES to another from transport protocols (and applications) on hosts.
  - Hot swap of a session from one CES to another requires session state mirroring from active CES to hot-standby CES: at the beginning of a new flow, state timeouts and at the end of the flow. It is probably best to limit this only to the most important and rather long lasting flows using policy (for performance reasons).
  - If the local IP network does not apply RPF check because of multicast, 2 CES nodes may use the same local IP source address for the packets in the ongoing session without virtual IP address protocols

On-demand multihoming routing mechanics (3)

- Revoking local RLOC
  - CES sends a Q/R/M=1 with its RLOCpref=0xFE for which there is ongoing session
  - If the same CES has alternative RLOCs, having requested an RLOC switchover, CES MUST immediately start accepting traffic for the ongoing session using any alternative local RLOC
  - If there are 2 local CES systems, by making the local IP addresses that are allocated to remote hosts virtual, we may be able to hide the RLOC swap from one local CES to another from transport protocols (and applications) on hosts.
  - All of previous slide’s story on RPF and state mirroring applies here as well
  - Host standby CES MUST immediately start accepting traffic for the session
- Accepting traffic on alternative RLOCS for a session MAY be time limited (e.g. for making DDOS attacks harder)

Discussion on Hot Swap of RLOCs

- If all RLOC to RLOC delays between inbound and outbound edge nodes are about equal, risk of re-ordering of messages in the flow is minimal
- If the delay differ significantly, hot swap becomes more complicated
- Impact of dynamic routing in the core and the customer network on hot swap need to be studied carefully in order to find the best routing configuration

If Flag M=1, message may contain info on Time-to-Live of the Customer Edge state

0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

Type=32	Length	TTL

TTL gives the Time-to-Live in Seconds of the sender’s state of the communication.
State will be deleted if there are no messages within TTL.
TTL will be restarted upon any message related to the ID in question.

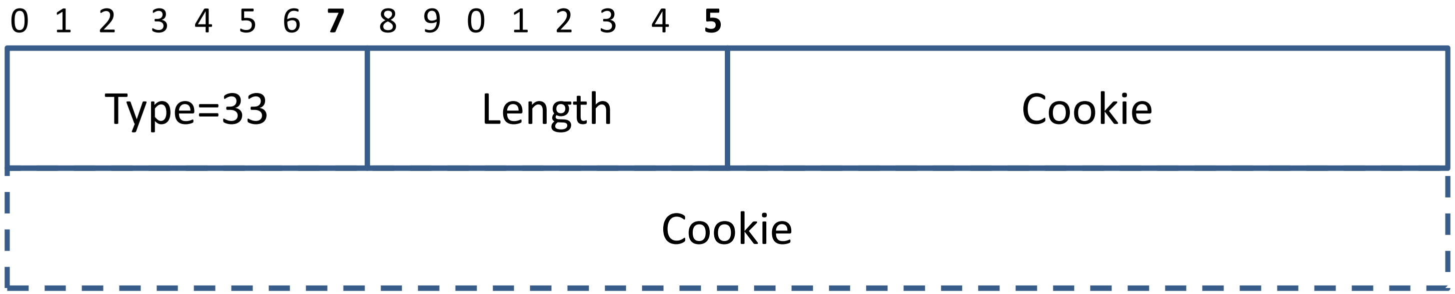
For remote TTL =N, local T2P sets a timelimit of N/2 – 1 and will reset this timelimit upon reception and upon every other sending of message to/from the ID

- on expiry will resend a monitoring message
- sender will count such monitoring messages and after K messages will release its state. Count is set to zero when a response or is seen.

Revoking an ID

- If Q=1 and TTL=0, sender tells the remote edge that it is removing connection state.
- If the remote edge wants to continue communication, it must restart communication from DNS query and accept that the ID of the corresponding host may have changed.
  - By default loss of edge connection state is reported to hosts and e.g. an ongoing TCP session will be deleted.
  - It might be possible to preserve a TCP session while ID is changed using Cookie(?)

If Flag M=1, Cookie TLV may be used


0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

Type=33	Length	Cookie

Cookie

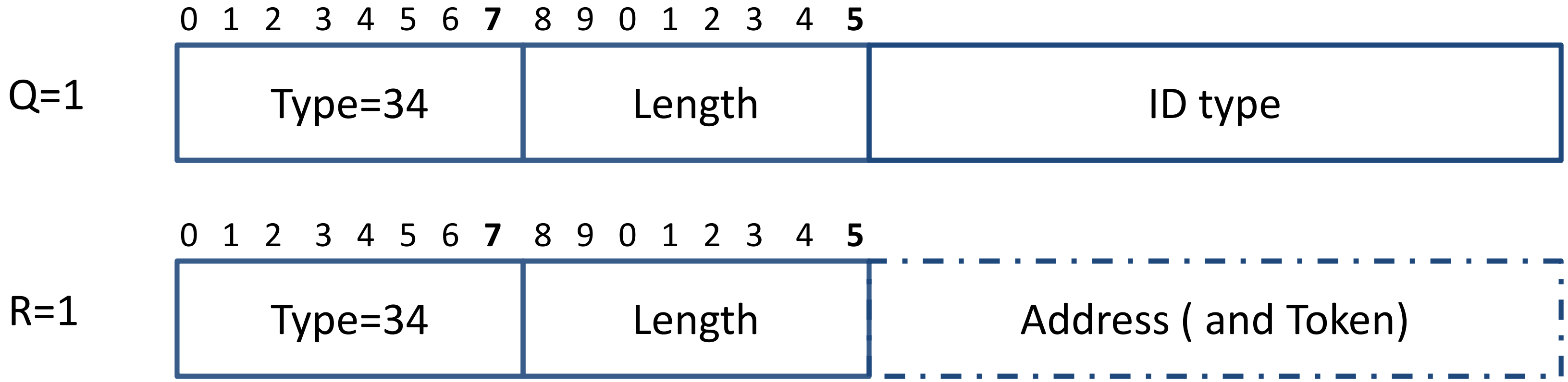
- Length gives the length of Cookie in octets
- Cookie is variable length up-to 255 octets (- rest of T2P header)
-Is a way of putting-off the need to create state at inbound edge
- Remote end must return Cookie as such in the next message.
- Cookie is a way of doing forwarding protocol (e.g. IP) level return routability check

Example: Inbound CES captures SYN, XORs that with a secret string and a timestamp that is stored once in e.g. 10 seconds to create the Cookie.
Upon the next message, Inbound CES creates state, being sure that outbound RLOC has not been spoofed.

Cookie – use cases

- Inbound edge wants to postpone creating state for a new flow – sends response with cookie → source address spoofing is eliminated → outbound edge responds with cookie+next payload (from the initiator of communication)
- Inbound edge wants to use mobile operator assured identities → sends response with cookie → ingress has to re-start the flow with mobile operator assured ID (and token obtained from HSS?)
  - New message from ingress contains: new ID, cookie (and token?)
- Cookie might be helpful for managing state when the inbound edge pushes a puzzle to the outbound edge/initiator of communication?

New ID type request/response



0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

Q=1

Type=34 | Length | ID type

0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

R=1

Type=34 | Length | Address (and Token)

- If Q=1, this TLV defines a request for a new ID (type)
  - E.g. Inbound edge requires a mobile operator assured ID (MAID)
- R=1, Value gives the routable address for assurance queries (addess of HSS) and a token that helps eliminating MAID spoofing (optional)
- Using the received address, a inbound CES can execute HSS query for Mobile Operator assurance using e.g. the Diameter protocol(?)
- Once first message with new ID is received, new state is created, Optional Cookie can tie the Q and R together

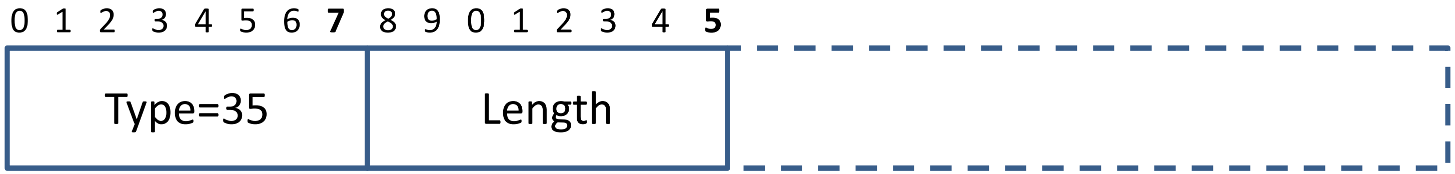
On Mobile Operator assured ID

- The number of mobile broadband subscriptions has overtaken the number of fixed broadband subscriptions and grows faster than fixed BB – there is a huge potential in big cities on the emerging markets and a large potential in developed countries.
  - Most of next Billion Internet users will be mobile
  - Also, more and more Laptops have a SIM card
- Use cases
  - A (Mobile Operator) assured ID (MAID) is good for conducting business between users – commercial commitments (within reasonable limits) can be made based on the ID.
  - Real world Internet: a CES serving your personal devices can admit communication only from your mobile
  - MAID helps to avoid SPIT in mobile packet voice services
- Advantage of Edge to Edge protocol with MAID against end-to-end protocol with MAID is that mobile destination does not need to see unwanted initial messages to an application that has a MAID only policy, also protects battery powered devices from DDOS
- Exact format of the MAID is TBD

Implementing MAID using Diameter

- Mobile operators could agree to respond to queries coming from other Mobile operators providing trust services to their mobile subscribers – a new Diameter Application MAY be needed
- 1 or 2 types of request/answer transactions are needed
  - Edge nodes MUST be able to request for MAID for a host that is roaming in their network (similar to AA-request/answer or DER/DEA of the EAP) or MAID is generated from data returned by DEA locally by the local Edge node
  - Inbound Mobile Operator hosted edge node may decide to trust source MAID without validation like in IMS sessions (because Mobile operator owned CES nodes communicate only through GRX and have sufficient physical security) OR
  - Inbound edge node MAY wish to request validation of source ID from HMS of the initiator of communication (like LIR/LIA in the SIP application of Diameter, this request/answer pair would cross Operator boundary) – provided edge nodes are connected to the open Internet this may be a wise move…

Return Routabilitity check including naming



0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

Type=35

Length

- If Q=1, this TLV defines a request for a host name
  - Value is optional → by default request for regular DNS –like host name. Value could define other name types (SIP URI, tel-URI, E.164 numbers etc.)
- R=1, Value gives the (host etc.) name
  - Several TLVs of this type could give optional names.
- Using the received name, a inbound CES can execute DNS query, receive all RLOCs in response, check that communication is using one of them resulting is a return routability check covering naming and the forwarding protocol (e.g. IPv4)
  - Cookie can tie the Q and R together

Response Codes

0 1 2 3 4 5 6 **7**	8 9 0 1 2 3 4 **5**	
Type=40	Length	

- In messages with R=1, errors MAY be reported
- Examples:
  - (Inbound) CES busy
  - CES congested
  - Target application busy
  - Target host busy
  - Target host not available
- MAY also be sent and SHOULD be accepted without prior Query (Q=1)

Some special cases

- If the remote end does not recognize the target ID
  - It MAY (and is recommened to) silently ignore the message
- If the remote end recognises the ID, but there is no state for the pair of Ids
  - If Q=0, R=0, M=0: state MAY be created (e.g. if CES has banned the source RLOC, it will ignore the message)
  - Q=1: inbound CES MAY serve the flow minimally until it sees that communication seems to flow normally (e.g. it has made a return routability check itself)
  - When local RLOC configuration is non-default, CES MAY serve RLOC queries giving current preferences in Response messages
- If CES is waiting for a response, it MAY delete all messages carrying the ID pair that do not contain the expected response.
- If remote CES = local CES, traffic is looped back locally and using a simpler admission policy (concerning RLOCs) is appropriate

If Flag P=1, TLV describes puzzle

- If P=1, either Q or R are set.
- Query contains description, Response contains answer
  - Makes sense if the puzzle is sufficiently hard, so that outbound CES will most likely give it to the source host to solve.

Type=41

Good puzzles are needed here!

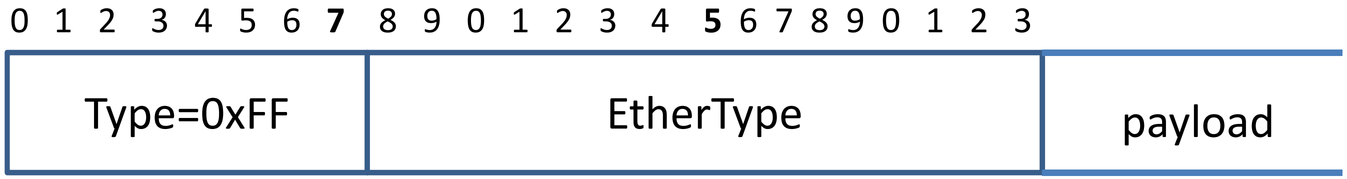
NB: most likely legacy hosts do not understand these puzzles, so deployment of this feature requires updates of host software.

Admission Policy Examples

- WWW server:
  - admit max N inbound flows (ID pairs) at a time (N might depend on device type, nice if the application could manage this number)
  - Option: DDOS detection counting active/dormant
- Limited WWW access to a target ID
  - if (source RLOC XOR Mask=nn), Execute full return routability check, if (source name=yy), admit, else deny
- VOIP, no call waiting
  - Admit max 1 inbound flow at a time
  - Upon 2nd inbound flow, send response with ”target application busy”
  - Redirect to mailbox should be handled on call signaling level
  - All other flows during the call must be initiated by the host or CES must be able to differentiate signaling from media and data for example based on IP port numbers
- VOIP, with max 1 call waiting
  - Admit max 2 inbound (signaling) flows
  - Upon 3rd inbound flow, send response: ”target application busy”
- MAID only policy
  - If ID = MAID, admit
  - Else respond R=1: MAID required, count
  - If count >N, ignore (stop responding for time T)

Message May Carry a payload protocol

- Payload in not included in HL but is included in Total Length.
- Q/R/M bits may or may not be set.
- if all Q&R&M=0, message MUST carry a payload



0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5** 6 7 8 9 0 1 2 3

Type=0xFF	EtherType	payload

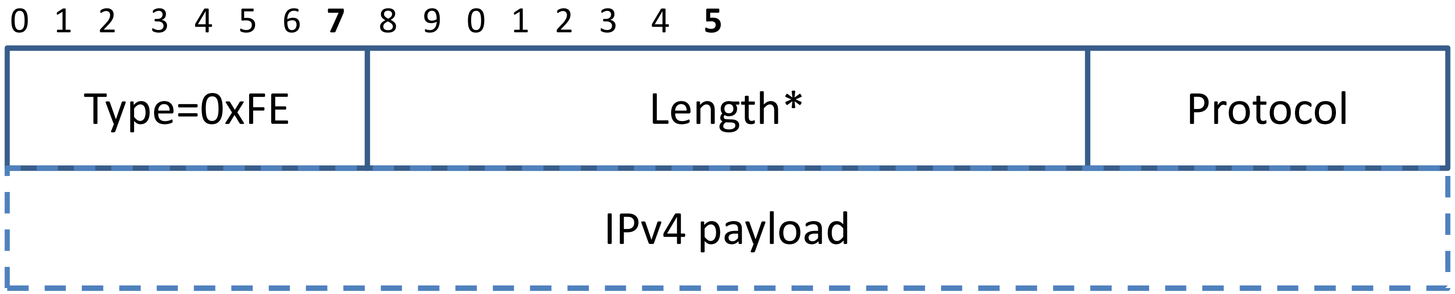
NB1: this ”TLV” can not be followed T2P control information TLVs

NB2: If payload is IPv4, source and destination address fields are set = 0, and reset to appropriate values by the receiving CES

NB3: Type = 36…3F, 42…0xFD are reserved.

NB4: if EtherType for T2P is defined, one T2P message can carry another T2P message making it possible to monitor many Ids with a single message between 2 Customer Edge Nodes.

Header compression for IPv4 payload is integrated in T2P


0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

Type=0xFE

Length*

Protocol

IPv4 payload

This resembles RFC-2004: Minimal Encapsulation within IP and assumes that the core transport takes place over IPv4. Unlike in RFC-2004, not even the destination IP address is preserved because it must be mapped by the receiving CES based on target ID.

Protocol = protocol field in the original payload IP header (what is carried: TCP etc…)

Receiver generates the target network IP header as follows:

+Version = 4

+IHL =20

+Type of service – based on local policy
(default= copy from core IP

+Total length = Length* + 16

+ Fragmentation can not be used

+ TTL = core IP TTL -1

+ Protocol = copied from the above element

+ Header Checksum – calculated locally

+ Source IP address: allocated by CES locally

+ Destination IP address – mapped by CES locally

How to carry T2P over Internet

- Option 1: A new EtherType is defined
  - T2P is carried over Ethernet core
- Option 2: A new transport protocol is defined in IPv4 header in parallel to UDP, TCP, SCTP etc.
  - T2P is carried directly over IPv4 for IPv4 core network
- Option 3: A new well-known port number is defined
  - T2P is carried over UDP

Summary of T2P (1)

- T2P gives control of packet admission to the inbound CES: if Best effort IP service takes care of the sender's needs, T2P serves the needs of the receiver
- It is assumed that packet access control in the inbound edge node is based on policy
  - This policy could be controlled by the user device or managed by the network administrator (like Firewalls today)
  - Policy dictates which type of ID is required (for the application), which checks are applied before admitting a new flow, which history information is stored and used in admission etc.
  - policy can even be dynamic, i.e. change as a function of hostile activity – there is room for differentiation in products in this area.
- T2P manages (soft) connection state in CES (i.e. on the Trust layer). State is established and removed dynamically as a side effect of normal communication pattern.

Summary of T2P (2)

- CES can send queries, responses, monitoring and data messages using T2P to another CES
  - Data may also be embedded in queries and responses for the purpose of reducing the number of messages (or queries/responses can be embedded in data messages)
- T2P directly supports minimal encapsulation of payload IPv4 for the case of underlying core IPv4 reducing header overhead
- Q/R allow monitoring
  - the state of the RLOCs implementing on-demand routing over a multihomed edge and
  - execute a smooth swap of RLOC for a flow without hosts noticing more than a possible temporary slowdown of the flow and
  - the state of the connection
- Execute return routability checks either on forwarding or forwarding and naming levels
  - Cookie allows excluding rloc spoofing and helps the return routability checks before creating state at inbound edge

Possible extensions

- Header checksum (like in IPv4)
  - Might be defined to cover either just the fixed 1st word and the Ids or also the other control information
- Support for fragmentation (e.g. for the cases: (a) underlying core protocol is Ethernet, (b) cookie makes a message too long for MTU)
  - A new encapsulation with a fragmentation word equal to what is present in IP header
- Other encapsulations (probably not needed)?
  - Keep all else in payload IPv4 packet but remove source and destination IP address
- Other RLOC types: MAC address + BVLAN (for 802.1ah networks)
- Compatibility bits
  - We take the position that there are no options in the first version of T2P and that all additional TLV information elements will be of the form:

Type=xx	Length	Flags: R I D	info

R – report non-compliance (= receiver does not understand the object)

I – Ignore data element if not understood, process the rest of message

D – delete message silently if data element not understood