Trust-to-Trust Protocol (T2P)

Raimo Kantola
Aalto University
School of Science and Technology

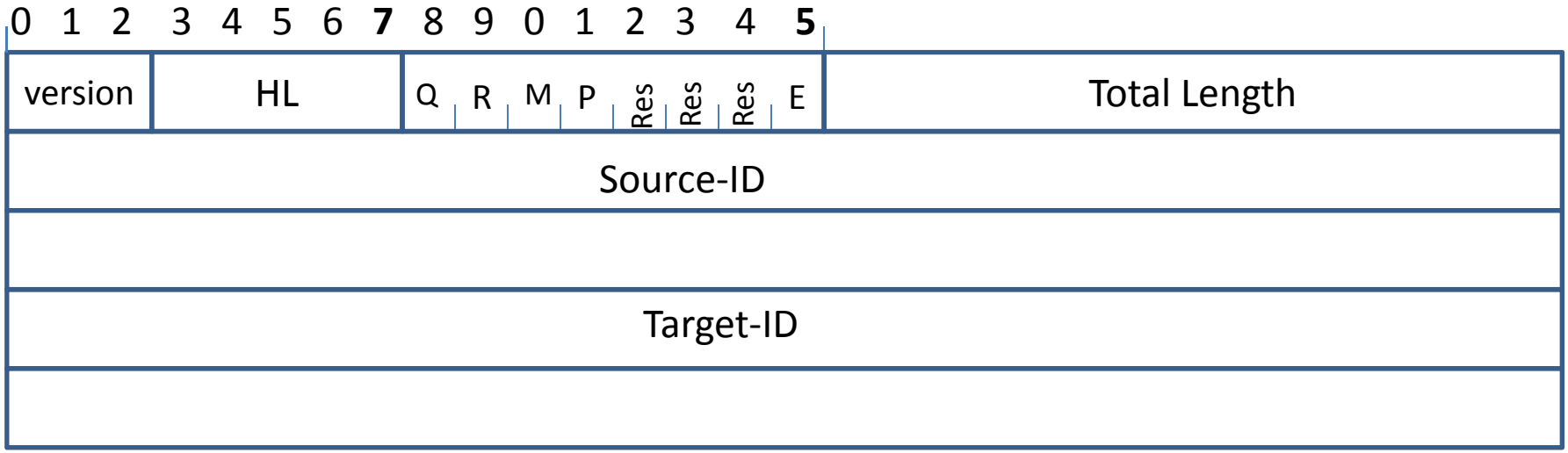
T2P Requirements

- Enhance trust edge to edge by facilitating IP trace back and thus help to ensure non-repudiation of communication.
  - T2P lets the egress decide whether it wants to exclude source address spoofing before it admits communication
- 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 ethertype could be defined and T2P would then be carried over Ethernet directly or
  - A new “protocol” codepoint in IP header could be defined (in parallel with UDP, TCP, SCTP etc)

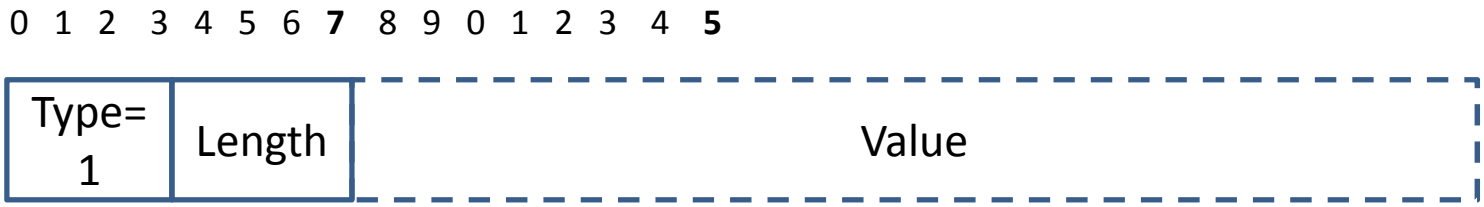
Protocol header



Version – Protocol Version, for now = 1
HL – Header Length in octets, here shown as HL = 20 (range: 4…31)
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

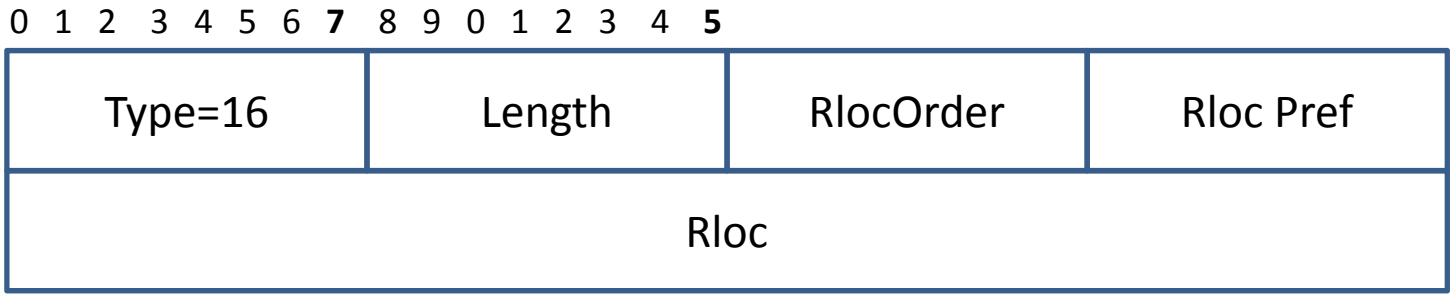
- ID's can be random values generated by CES based on their own algorithms or Mobile Operator assured Ids can be used. The latter could be e.g. MSISDN number that can be checked from HSS/HLR.



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. For example we could use MSISDN number as the ID.
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

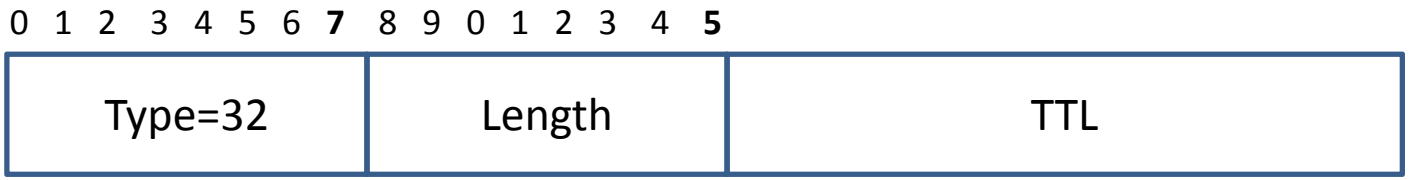
- Query may carry and Response MUST carry



M=1 → either Q or R must be set.
Type=1 = 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)

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

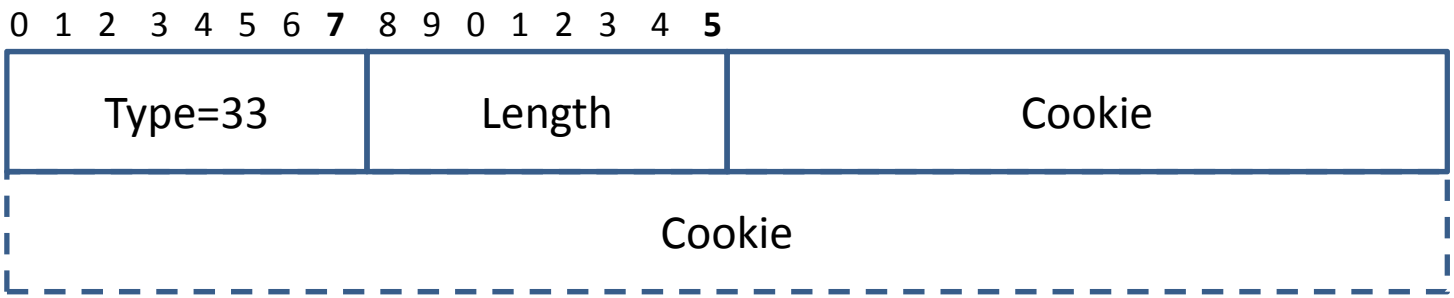


TTL gives the Time-to-Live in Seconds of the sender's state for the communication.
State will be deleted if there are no messages within TTL.
TTL will be restarted upon any message related to the ID in question.

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 M messages will release its state.

If Flag M=1, Cookie TLV may be used



- Length gives the length of Cookie in octets(or 32 bit words?)
- Cookie is variable length up-tp 255 octets
-Is a way of putting-off the need to create state at Egress
- Remote end must return Cookie as such in the next message.

Example: Egress CES captures SYN, EXORs 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, Egress creates state, being sure that ingress RLOC has not been spoofed.

Cookie – use cases

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

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, state is created
  - If Q=1, response is sent but state is not created
- If Q=1 and TTL=0, the remote end will remove connection state for the pair of IDs

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 ingress CES will most likely give it to the source host to solve.

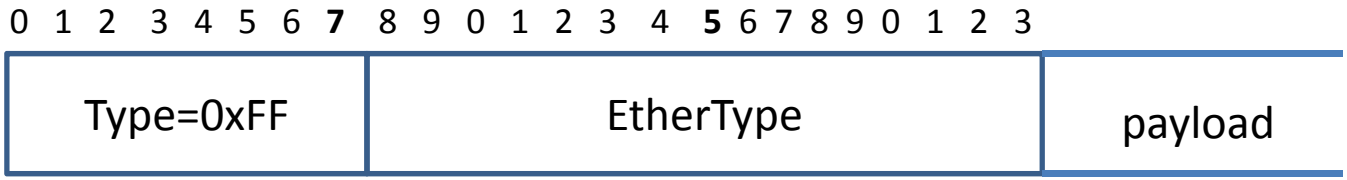
Type=34

Good puzzles are needed here!

NB: most likely legacy hosts do not understand these puzzles, so deployment of this feature is difficult.

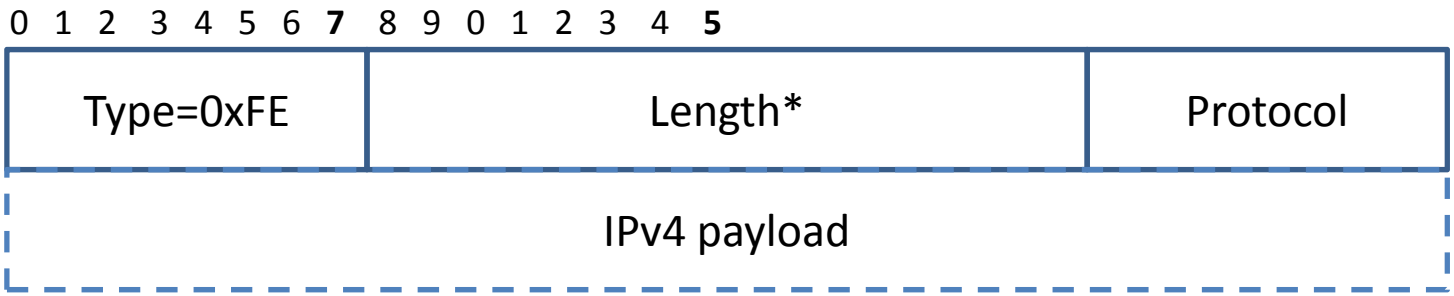
If Q=R=0, Message carries a payload protocol

If Q=1 or R=1, message may carry a payload protocol.



NB1: this ”TLV” can not be followed by other 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 = 35…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



This resembles RFC-2004: Minimal Encapsulation within IP and assumes that the core transport takes place over IPv4. 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
- Option 3: A new well-known port number is defined
  - T2P is carried over UDP

Summary of T2P

- CES can send queries, responses and data messages using T2P to another CES
  - Data may also be embedded in queries and responses for the purpose of reducing the nrof messages (or queries/responses can be embedded in data messages)
- Q/R allow monitoring
  - the state of the RLOCs and
  - the state of the connection and
  - execute a smooth swap of RLOC for a flow without hosts noticing more than a possible temporary slowdown of the flow
- Cookie allows excluding rloc spoofing before creating state at egress
- T2P directly supports minimal encapsulation of payload IPv4 for the case of underlying core IPv4 reducing header overhead

Possible extensions

- Egress could Query the host or user name of the initiator of communication
- Control information length in the header might ease the processing of messages
- 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: underlying core protocol is Ethernet, 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
- 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:

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

Notes 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
- An 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 personnal devices can admit communication only from your mobile
- Advantage of Edge to Edge protocol with MAID against end-to-end protocol with MAID is that mobile destination does not need to see a single message to an application that has a MAID only policy
- A faked MSISDN can be real → probably a token given by the initiator's HSS/HLR is needed to eliminate this attack (this is not defined in this slideset) – Egress must be able to check the token from the HSS/HLR