# Customer Edge Traversal Protocol (CETP)

Raimo Kantola

Aalto University

Department of Communications and Networking (Comnet)

Nicklas Beijar, Zheng Yan

# Disclaimer

- This slideset is created to describe ongoing research prototyping.
- There are many potential reasons that may justify changes to this definition
  - Some of them a mentioned in the slides

# Wider role of CETP

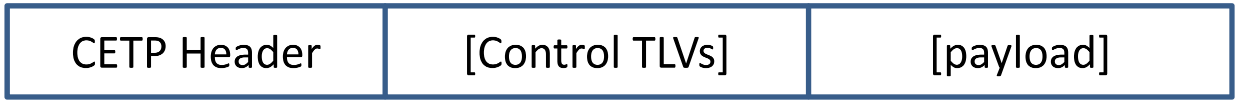
- CETP is an edge to edge protocol for tunneling packets from one customer network to another. Each network has its own private address space.
- CETP is a part of an Internet Trust Framework (ITF).
  - Entities involved in the ITF are
    - Hosts, users/subscribers, agents of communicating parties called Customer Edge devices, ISPs, a Global Trust Operator (GTO) and communicating applications
  - Some of the functions of ITF are:
    - Identities and Identity management
    - Policy based management for traffic admission (per host or user and application)
    - Legacy Interworking with hosts and customer networks that have not changed
    - Unwanted traffic source identification and location
    - Trust incident reporting, trust value calculation for ISPs and hosts
    - Tariff establishment as a function of trust value
- Goal of ITF is to make unwanted traffic business unprofitable.

NB: This protocol was formely called Trust to Trust Protocol (T2P) but has now been renamed. The name, CETP reflects more precisely what is accomplished by the protocol.

# CETP Requirements

- Carry identities edge to edge and tunnel the payload protocol dynamically edge to edge
- Operate multi-homed edge functions by providing *on-demand routing* through the multi-homed edge
- Enhance trust between 2 customer networks and users in the 2 customer networks by facilitating return routability checks, assurance of IDs and RLOCs of communicating parties, IP traceback and thus help to ensure non-repudiation of communication.
  - CETP lets the inbound edge decide whether it wants to exclude source address spoofing, what types of IDs to use before it admits communication and also whether communication is encrypted or non-encrypted edge to edge
  - Inbound edge can report suspect reflector DDoS attack (i.e use of its IP address in spoofed queries to sender)
  - Inbound edge node can collect history information about RLOCs and IDs and use that as the basis for packet admission
  - A CETP node can collect trust evidence and send that to other ITF components for processing
- CETP could be modeled as
  - a protocol on top of UDP or
  - A new protocol code point in IP header could be defined (in parallel with UDP, TCP, SCTP etc) or
  - a new ethertype could be defined and CETP would then be carried over Ethernet directly

# CETP packet structure



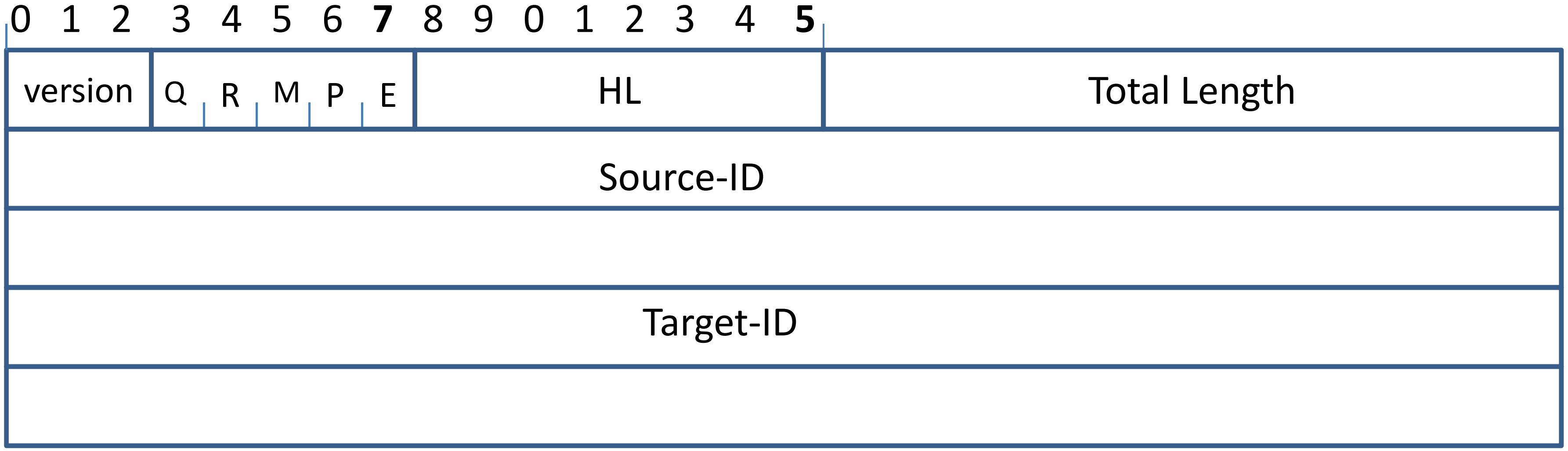
CETP Header

[Control TLVs]

[payload]

- Header carries a *control word* with Flags, etc and the host/user Ids
  - If all flags = 0, packet just carries payload and ”CES control plane” of the receiving CES does probably not need to process the packet
- One or more control TLVs may be present
  - Total length of control TLVs < 1020 octets.
  - CETP header has Header Length (HL) that gives number of 32 bit words in CETP Header+Control TLVs
  - The string of control TLVs is padded to 32 bit boundary, padding is ignored by the receiver
- Payload is e.g. an encapsulated IP packet
  - Starts at a 32-bit boundary
  - If all flags = 0, must be present
  - If one or more flags = 1, may be present

# 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 32-bit words, here shown as HL = 5 (range: 3…255), HL includes the control word, IDs and CETP control data formatted as TLV elements.

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

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

M – Monitoring Flag

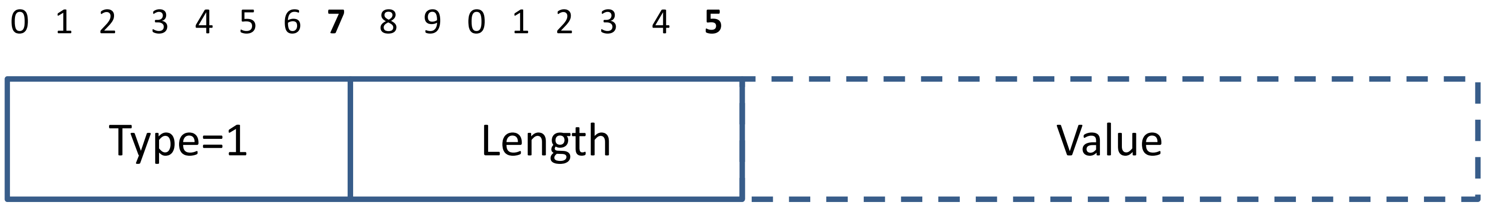
P – Puzzle Flag

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 or a certificate based on those 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

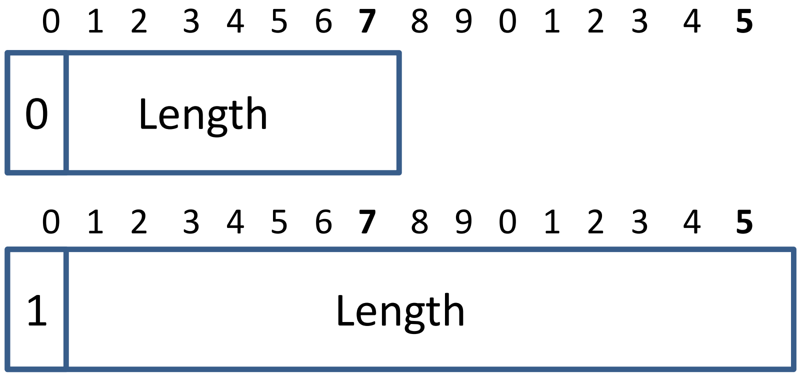
Value

Type=1 → Random ID generated by CES based on its own algorithm
Type=2 → Mobile operator assured ID (can be used in ”closed” networks, like in IMS)
Type=3 → User certificate obtained from Mobile Operator=CA, CES can query HSS/HLR to check that the ID exists and is valid (can be used even when CES are connected to the Internet
Types: 4…15, 0 reserved for future use (e.g. Internet of Things objects have their own ID schemas etc)
Value: if BCD encoded, padded to octet boundary from the left.

# TLV Types

| | |
|---|---|
| Type=1…15 | Identity TLVs (1=random, 2-MO assured, 3-ID certified by CA…) |
| Type=16 | IPv4 RLOC and RLOC preferences (IPv4 Reachability info) |
| Type=17 | IPv6 RLOCs and RLOC preferences (IPv6 Reachability info) |
| Type=18 | Ethernet (MAC address) RLOCs and preferences (MAC Reachability info) |
| Type=19…31 | Reserved for other RLOC types and their preferences |
| Type=32 | TTL of customer edge state |
| Type=33 | Cookie |
| Type=34 | ID type request |
| Type=35 | Certification address |
| Type=36 | FQDN |
| Type=37 | Generic TLV query list |
| Type=38 | RLOC record signature (over all RLOC types |
| Type=39 | Unexpected message report |
| Type=40…63 | Reserved |
| Type=64 | Response Code |
| Type=65 | Puzzle (may be not needed?) |
| Type=66…0xFD | Reserved for future use |
| Type=0xFE | Compressed IPv4 header encapsulated payload |
| Type=0xFF | Ethernet encapsulated payload |

# Length encoding in IDs and TLVs



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

0 Length

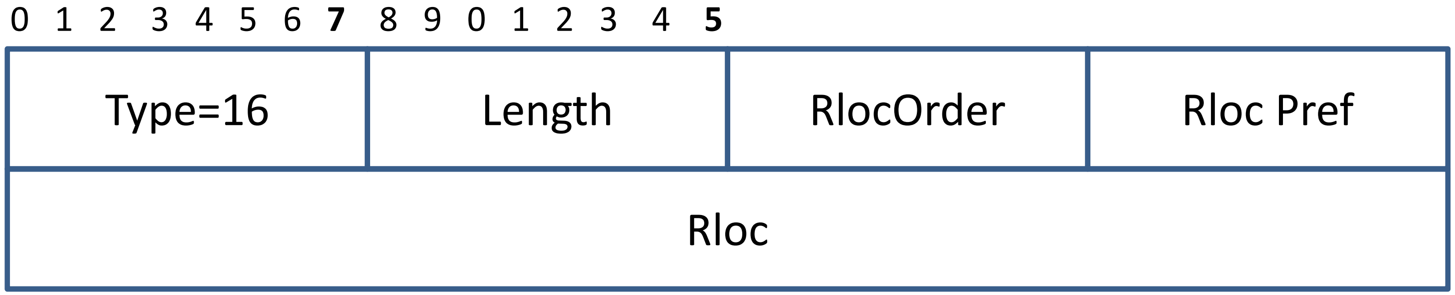
0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

1 Length

- The first bit of the length field indicates the number of bits for specifying the length of the given TLV
  - L=0 -> 7-bit length (0-127)
  - L=1 -> 15-bit length (0-32767)
- The same format of the length field is used in all TLVs for simplified processing

# 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**

| | | | |
|---|---|---|---|
| Type=16 | Length | RlocOrder | Rloc Pref |
| Rloc | | | |

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
    - May use CETP to confirm the current preferences
  - Inbound CES can use CETP to learn all RLOCs of the outbound CES
    - CETP 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 (in order to make network scanning more difficult)
    - CETP Response MUST contain one RLOC that appears as source RLOC on the forwarding layer in the inbound CES, CETP response MAY contain other RLOCs
- Monitoring RLOCs
  - CETP 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 delays 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 – this if for FS.

# If Flag M=1 (or Q/R=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.

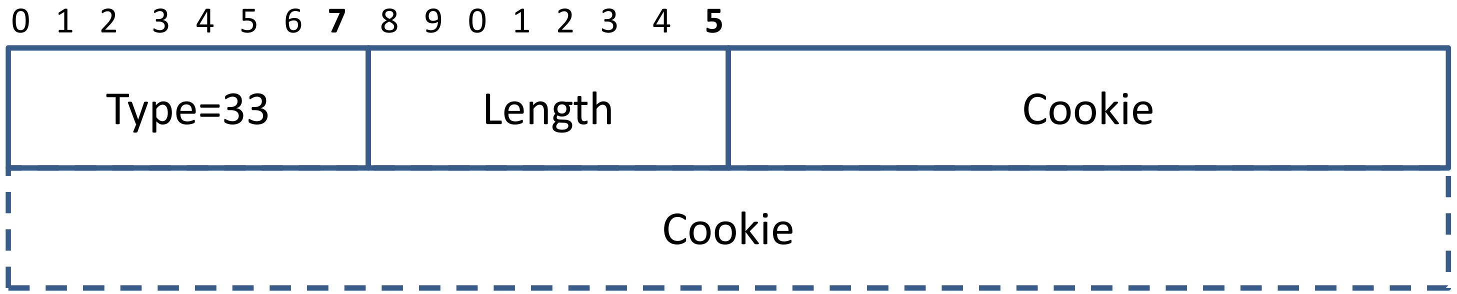
For remote TTL =N, local CETP sets a timelimit of N/2 – 1 and will reset this timelimit upon reception and upon every other sending of message to/from the ID

- 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 a message 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 (is seen by) 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



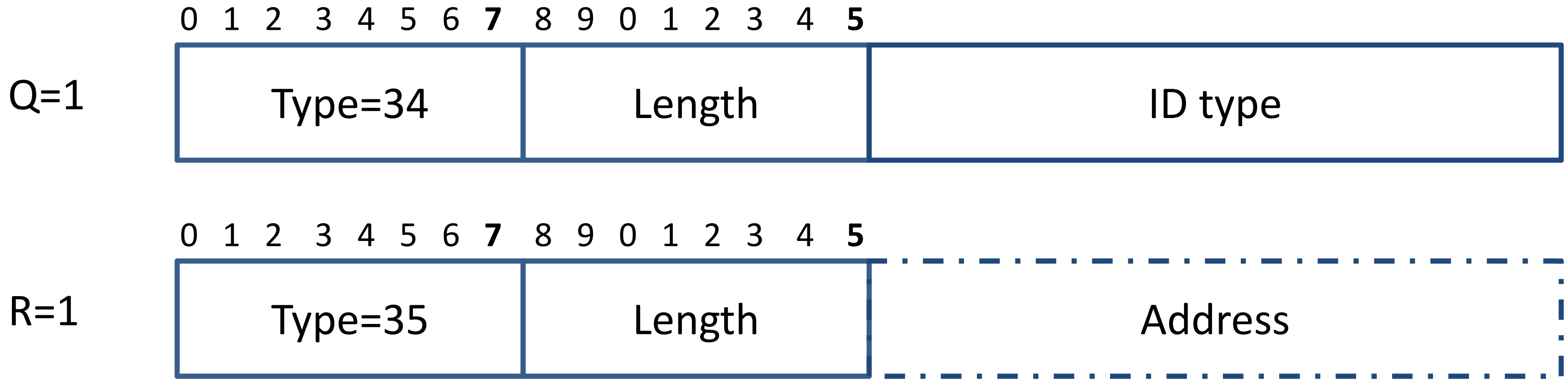
- Length gives the length of Cookie in octets
- Cookie is variable length up-to 255 octets
- Is a way of putting-off the need to create state at inbound edge
- Remote end must return Cookie as such in the next message.
- Cookie is a way of doing forwarding protocol (e.g. IP) level return routability 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 (without fooling the core routing system)

# Cookie – use cases

- Inbound edge wants to postpone creating state for a new flow – sends response with cookie → source RLOC 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 (or certificate issued by MO)
  - New message from ingress contains: new ID or certificate, cookie
- 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=35 | Length | Address |
|---|---|---|

- 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 or some other CA)
- Using the received address, the inbound CES can execute HSS (or CA) 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
- NB: ID query is separate from a generic TLV query because connection state handling is different in the 2 cases.

# 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.
  - Internet of Things: a CES serving your personal devices can admit communication only from your mobile that can your SIM card
  - MAID helps to avoid SPIT in mobile packet voice services
  - Good for MO to MO connected CES (e.g. GRX or VLAN used to separate MO-to-MO traffic from other incoming traffic! NB: MAID is not a certificate, assurance is based on closed network connection between the edge nodes.
- 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(s) of the MAID(s) is TBD

# Implementing MO Certificates (MOC) using Diameter

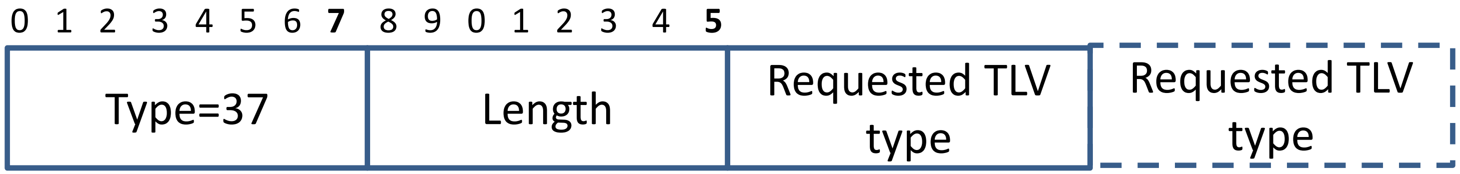
- Mobile operators could agree to respond to queries coming from other Mobile operators (and also all types of CES) providing trust services to their mobile subscribers – a new Diameter Application MAY be needed
- 1 or 2 types of request/answer transactions are needed
  - Edge nodes MUST be able to request for MOC or MAID for a host that is roaming in their network (similar to AA-request/answer or DER/DEA of the EAP)
  - This query would be triggered upon reception of an Attach from the mobile
  - Inbound edge node MAY wish to request validation of source ID (i.e. MOC) 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…

# Domain information

| 0 1 2 3 4 5 6 **7** | 8 9 0 1 2 3 4 **5** | |
|---|---|---|
| Type=36 | Length | FQDN |

- Transports the FQDN associated with the sender ID
- If no FQDN is associated with the sender ID, a TLV with length=0 is sent
- In case several FQDNs exist and the originating CES chooses to provide all, then multiple TLVs are sent
- The Domain information TLV is used for providing replies to reverse DNS lookups as well as for responding to naming level return routability query by an inbound CES
- In the latter case using the received FQDN, a inbound CES can execute DNS query, receive all RLOCs in response, check that communication is using one of them resulting in a return routability check covering naming and the forwarding protocol (e.g. IPv4)
  - Cookie can tie the Q and R together and let the inbound CES postpone creating state

# On-demand TLV request



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

Type=37

Length

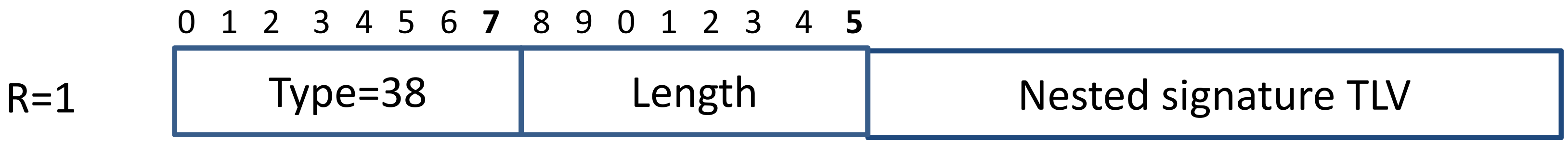
Requested TLV type

Requested TLV type

- Length give the nrof of requested TLVs since TLV types are encoded as octets
- If the receiving CES has connection state for the ID, upon receiving an on-demand TLV request, the CES will reply with the requested TLV(s) or with an error message
  - If there is no connection state for the ID, we recommend to ignore the query in order to make scanning harder
- The choice of sending TLVs by default or on-demand is defined in the policy
- Examples of TLVs that can be requested on-demand:
  - Requested TLV type = 16: Request for IPv4 RLOC information
  - Requested TLV type = 17: Request for IPv6 RLOC information
  - Requested TLV type = 32: Request for TTL information
  - Requested TLV type = 36: Request for Domain information
  - Requested TLV type = 38: Request for RLOC signature

# Signed RLOCs

- Plain text return routability check reveals a host that is trying to spoof an RLOC. The need for this mechanism may be avoided if all ISPs on the planet agree to carry CETP in a separate VLAN from the legacy Internet and/or agree to use ingress filtering in Provider Edge for all RLOCs.
- Plain text return routability check validity can be questioned: if an ISP network routing is compromised, an RLOC can be ”stolen” and the check in plain text will not reveal this
- Such an attack can be overcome by signing cryptocratically the  RLOC TLVs.
- A new TLV is needed for the response



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

R=1

Type=38 | Length | Nested signature TLV

- Each sub-TLV will contain the signature for one type of RLOCs (IPv4/MAC etc)
- To make this possible, the responding CES must have an ID itself and it must be registered into a certification authority (e.g. HSS?)
- FS: it may be sufficient to sign all RLOC TLVs of a CETP message with one signature.

# Reporting unexpected messages

- One of the DDoS attack types is the reflector attack in which usually a compromised host under the attacker’s control sends legal queries with the spoofed source IP-address of the victim. Reflector is not compromised but will follow the protocol and respond to the victim that is not expecting the response
- Inbound CES receives an unexpected ”response”, based on policy it may count such messages from the source
- Upon N unexpected received messages, inbound CES generates a CETP message where M=1 and Q = 0/1 (i.e. CES may request a response):

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

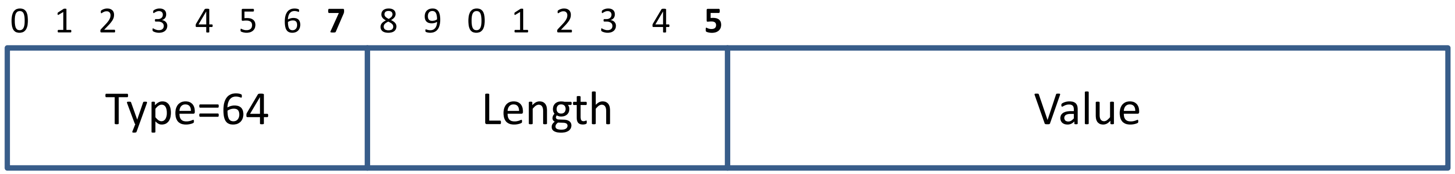
| Type=39 | Length | Value |
|---|---|---|

- Counting to N avoids amplification effect
- Value is the copy of the first M bytes of the unexpected ”response”.
- Question: would reporting the count and time over the count be of any help?

# Processing unexpected message reports

- Upon reception of M/Q with TLV=39, outbound CES SHOULD tighten its policy of using CETP to check its incoming (query) messages
  - CES can use return routability checks on forwarding and naming levels to validate queries and ignore queries with spoofed addresses
  - CES should have a way of reporting the spoofing to a higher level trust management system?

# Response Codes



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

Type=64

Length

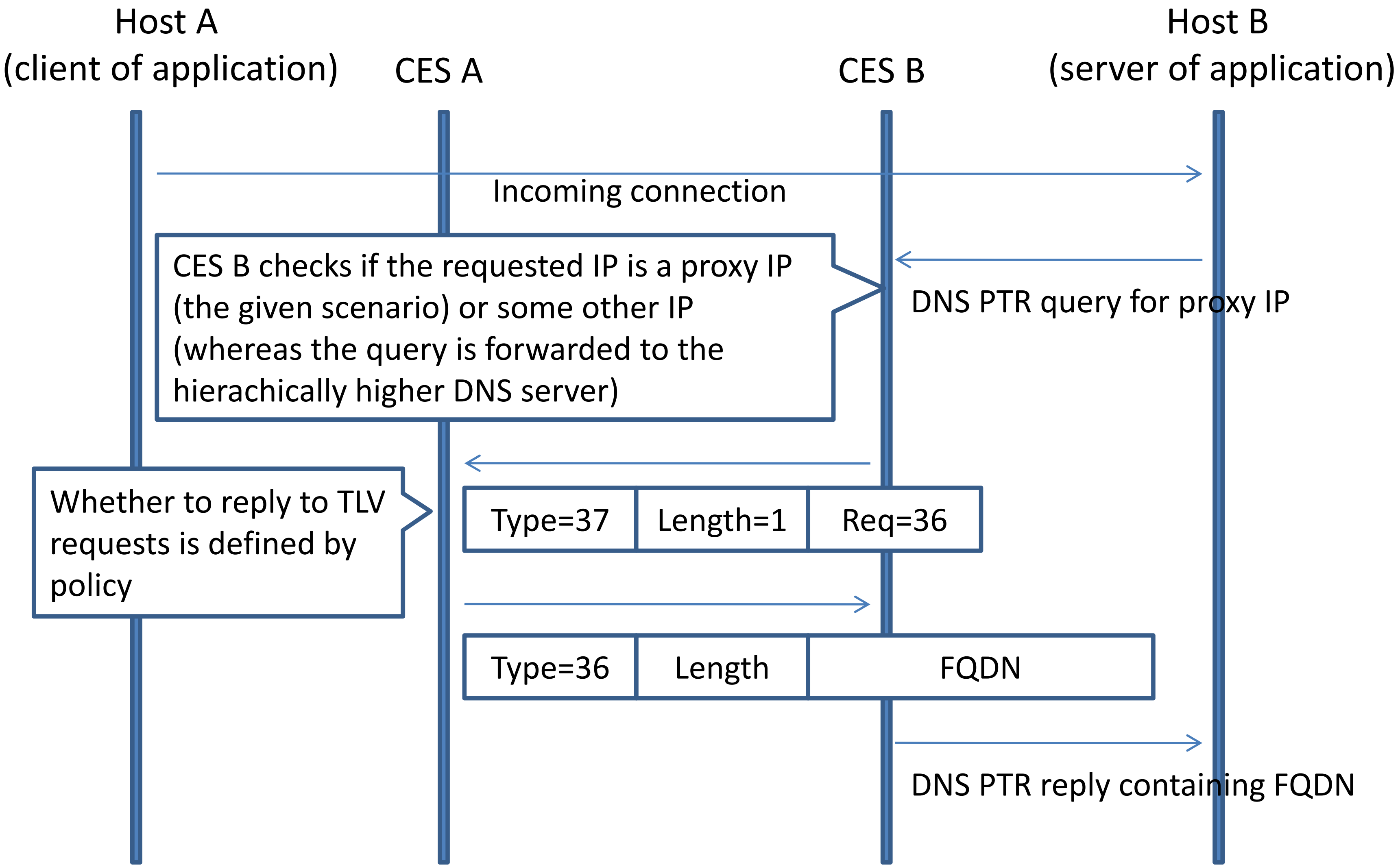
Value

- 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
  - Unknown connection (may be sent but default policy is not to respond to queries for which no connection state can be found in order to make network scanning harder)
  - Unknown TLV – added for backwards compatibility for future versions
- MAY also be sent and SHOULD be accepted without prior Query (Q=1)
- Before accepting a response when there is no connection state, CES may validate the response by executing a return routability check on the sender.

# Some special cases

- If the remote end does not recognize the target ID
  - It MAY (and is recommened to) silently ignore the message to make scanning harder (except when M=1 and the content has a TLV=64)
- 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

# Example reverse DNS lookup



CES B checks if the requested IP is a proxy IP (the given scenario) or some other IP (whereas the query is forwarded to the hierachically higher DNS server)

Whether to reply to TLV requests is defined by policy

| | | |
|---|---|---|
| Type=37 | Length=1 | Req=36 |

| | | |
|---|---|---|
| Type=36 | Length | FQDN |

# 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.

Good puzzles are needed here!

Type=65

NB1: most likely legacy hosts do not understand these puzzles, so deployment of this feature requires updates of host software. For this reason we will put no more effort into this now.

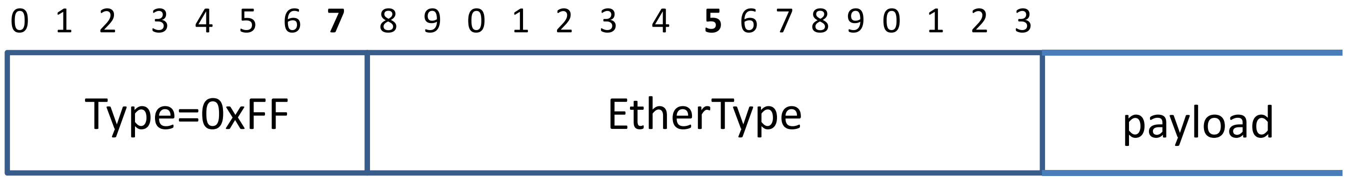
NB2: It may be wiser to replace this feature with some application of HIP (host ID protocol) either CES to CES or host to host.

# 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), can be manged by the Policy Control for mobile users.
  - 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
- Commanding your own (IoT or other) devices
  - If non-MOC ID, send MOC required
  - If MOC=your certificate, send check query to CA (=HSS), else deny
  - If CA response=OK, admit
  - Else deny
- VOIP, no call waiting
  - Admit max 1 inbound service 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 is 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

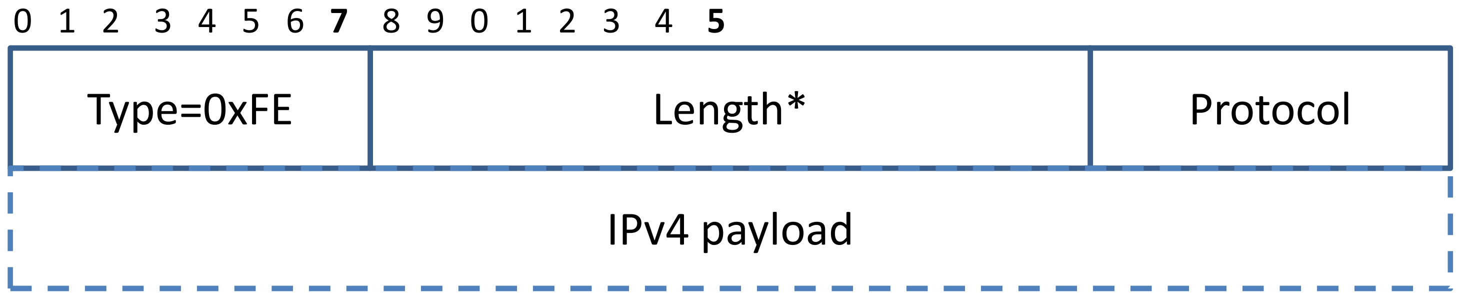
| Type=0xFF | EtherType | payload |
|---|---|---|

NB1: this ”TLV” can not be followed by CETP 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: if EtherType for CETP is defined, one CETP message can carry another CETP 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 CETP



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 CETP over Internet

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

# Summary of CETP (1)

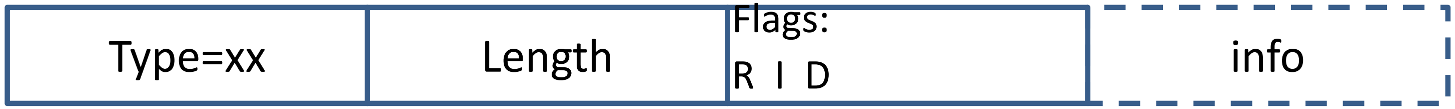
- CETP gives control of packet admission to the inbound CES: if Best effort IP service takes care of the sender’s needs, CETP serves the needs of the receiver
- It is assumed that packet access control in the inbound edge node is based on policy
  - This policy could be controlled by the user device 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.
- CETP 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 CETP (2)

- CES can send queries, responses, monitoring and data messages using CETP 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)
- CETP 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
- CETP supports many types of (Communication) Identities that an application may want to use. Type of ID is policy controlled.

# 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 or other TLVs make 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 CETP 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

# What to do when messages with spoofed source IP addresses are received by CES?

- It is possible to use the network engineering principle that all CETP traffic be carried in a different VLAN than the legacy Internet traffic and agree on tighter policy for all traffic on the CETP VLAN
  - NB: from the transport network point of view: Internet = one VLAN among others
  - If the ISPs agree to apply RPF checks in PE nodes on the CETP VLAN, RLOC spoofing becomes impossible
  - RLOC spoofing by (compromised) legacy hosts can be eliminated because incoming ”Internet VLAN” traffic can not have an RLOC as a source address
  - Use of CETP VLAN can also be agreed by a pair of ISPs
- The alternative is to use return routability checks CES to CES
  - Allows detecting/verifying spoofing and dropping the messages before they reach the target host
- But how can we locate the host that is spoofing a source address?
  - CES or the serving PE node may have routing table or label mapping table level information of the real source pointing to an interconnected source or transit network
  - The network served by a CES is only a subset of the whole Internet. Thus CETP narrows down the search for the spoofer. The difficult case is an inbound CES that provides a public service for any hosts.