Customer Edge Traversal Protocol (CETP)

Raimo Kantola
Aalto University
Department of Communications and Networking (Comnet)
Nicklas Beijar, Zheng Yan, Maryam Pahlavan

Disclaimer

- This slideset is created to describe ongoing research prototyping.
- There are many potential reasons that may justify changes to this definition
  - Some possible extensions are mentioned in a slide
  - Message encoding has changed as a result of the ongoing prototype development and may change further until it settles to the first official version

Wider role of CETP (1)

- 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:
    - Communication Identities and Identity management
    - Policy based management for traffic admission (per host or user and application)
    - Legacy Interworking with hosts and customer networks that do not support CETP
    - Unwanted /malicious traffic source identification and location
    - Trust incident reporting, trust value calculation for ISPs, hosts and applications
    - 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 been renamed. The name, CETP reflects more precisely what is accomplished by the protocol.

Wider role of CETP (2)

- CETP provides both tunneling edge to edge and (embedded/ in-band or piggybacked) signaling edge to edge.
  - The signaling helps to implement the idea of collaborative Firewalls
- CETP creates a shift in the basic communications paradigm used in the Internet
  - The present Best Effort paradigm = network does its best to serve the needs of the sender = transport sender's packets to the receiver
  - Publish/Subscribe turns the table around: The receiver has to subscribe to content – otherwise none of the stuff that is published will be delivered
  - CETP is a step towards Best Effort Communications(BEC) that is a synthesis between the classical Best Effort and Publish/Subscribe.
- The idea of BEC is that the network should apply its best effort to both the sender and the receiver and balance their interests in the act of communication.
  - The subscription of Pub/sub is replaced by policy in BEC
  - The interest of the receiver == receive what it wants to receive and drop the rest
- We expect CETP adoption first for wireless/mobile access networks

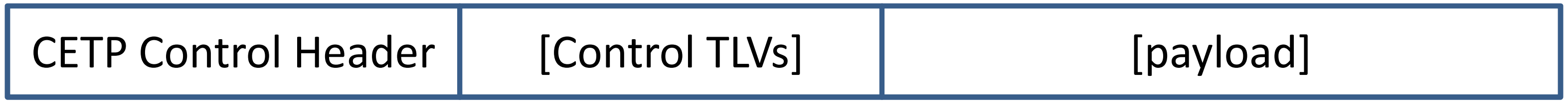
CETP Requirements

- Carry identities and the payload protocol in a dynamically established tunnel 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 trace back 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 authenticity of CETP signaling is ensured by crypto-graphical signature/certificate over CETP headers, 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 helps to solve the following problems

- Isolation of customer and core networks
  - Multihoming of customer networks so that multihoming has no impact on core network routing nor the core network RT or FIB size
  - Isolate technology choices in customer networks from choices in the core network
- Trust enhancement customer network to customer network
  - Gives the receiver’s edge tools to eliminate source address spoofing before flow admission to the receiver
  - Gives the receiver the possibility to define a policy that demands the sender to present a sufficient ID before any communication is admitted to the receiver
  - Helps to narrow down the location of sources that send unwanted or malicious traffic
- Together with server side Private Realm Gateway (PRGW), CETP
  - Improves the methods of reuse of IPv4 addresses for mobile hosts and objects in Internet of Things → IPv4 address exhaustion problem is effectively alleviated because most new users are wireless. Also a mobile device can be reachable without keep-alive signaling by the mobile. From the point of view of the mobile access is interrupt driven.
  - It makes sense to focus first deployment efforts into the wireless case.

CETP packet structure



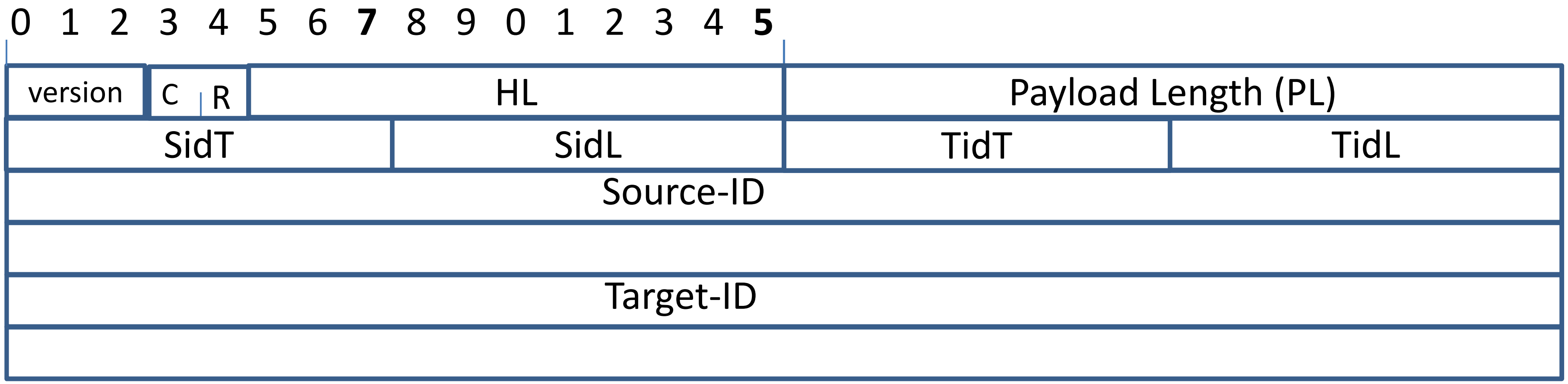
CETP Control Header

[Control TLVs]

[payload]

- CETP Header is made of the compulsory control header and the optional control TLVs
  - Compulsory Cntrl header has a flag saying whether any TLVs are present or not
  - Compulsory Cntrl header = Fixed part + 2 IDs of communicating parties
  - Cntrl TLVs= TLV header = means for edge to edge cntrl signaling allowing the receiver side make the important decisions for flow admission guided by its policy
  - Total length of CETP Header < 2048 octets given by the Header Length (HL) field in the Fixed part of the compulsory control header
- Payload is e.g. an encapsulated IP packet
  - Starts at a 32-bit boundary
  - If no control TLVs are carried, must be present in the packet
  - If one or more cntrl TLVs are carried, may be present

Protocol Header



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

version | C | R | HL | Payload Length (PL)

SidT | SidL | TidT | TidL

Source-ID

Target-ID

Version – Protocol Version, for now = 1

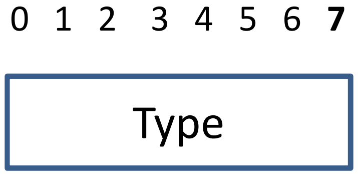
HL – Header Length in octets (here HL = 24) (range: 12…2047), HL includes the Fixed part, IDs and CETP control data formatted as TLV elements.

C = 0 = only payload, C = 1 = there is at least one control TLV present in the message
R = 0, Reserved for future use

Payload Length (PL) – Nrof of octets in the payload starting from octet number HL counting from zero.
 (encoded as in IP, not like length in CETP, so max value=2**16 - 1)
SidT – Source ID type, SidL – source ID length, TidT – Target ID type, TidL – Target ID Length

ID Type encoding

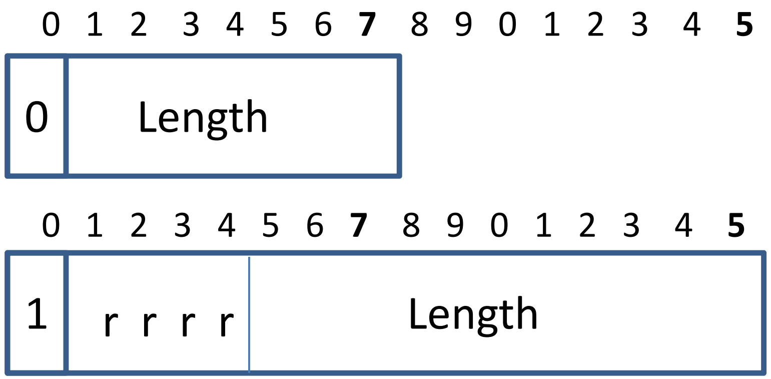
- ID’s can be random values generated by CES based on their own algorithms or Mobile Operator assured IDs can be used. The latter could be e.g. MSISDN number, a derivative of the MSISDN or IMSI number or a certificate based on those that can be checked from HSS/HLR.



Range: 1 …0x7F (7 bits in use!)

Type=1 → Random ID generated by CES based on its own algorithm (no CA)
Type=2 → Local (corporate) network certified ID (corporate net has its own CA)
Type=3 → Mobile operator assured ID (can be used in ”closed” networks, like in IMS)
Type=4 → 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)
Type=5 → FQDN as an ID
Type=6 → Temporary ID allocated by a visited network
Types: 7…0x7F, 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.

Length encoding for IDs and TLVs



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

0 | Length

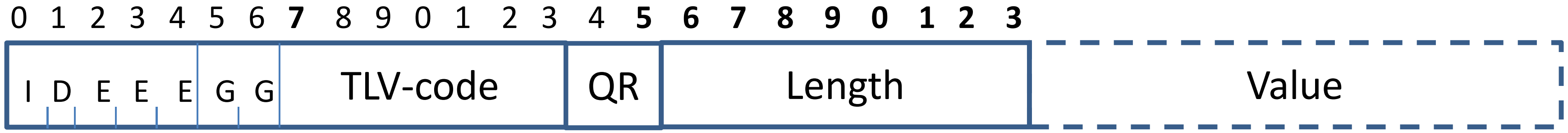
0 1 2 3 4 5 6 **7** 8 9 0 1 2 3 4 **5**

1 | r r r r | Length

r – reserved for future use

- The first bit of the length field indicates the number of bits for specifying the **length of the the value** in the given TLV (or in the ID of the mandatory header)
  - L=0 -> 7-bit length (0-127)
  - L=1 -> 11-bit length (0-2047) (not used for IDs)

Control TLV format



- Length gives the nrof **of octets in Value**, length is encoded like for Ids
  - (max value = 2047 – 8 – SidL – TidL) – when one TLV fills up the max length of the CETP header.
- QR=00, TLV query that can also solicit sender’s value, sender expects a response.
- QR=01, response to previous Query, no ack to response expected, gives sender’s value
- QR=10, (reliable) response, responder expects an ack from querier, gives sender’s value
- QR=11, ack to response, [optionally] gives the receiver’s value (that was just received in QR=10)
- We call a CETP message with at least one TLV’s QR=00 flags a *CETP query*
- I D= 00: If not understood, sender tells the receiver to ignore the TLV
- I D = 01: If not undestood, sender tells the receiver to ignore the TLV and send a backoff code in response
- I D = 10: If not understood, sender tells the receiver to delete all cntrl TLVs in the message
- I D = 11: If not understood, sender tells the receiver to delete all cntrl TLVs in the message and send a backoff code in response
- E – reserved for extension flags, set to 0 for now.
- GG =00 – ID-types , GG=01 – payload types, GG=10 – RLOC types, GG=11 – cntrl information types ;
- TLV-code (range 0x00…0x7F); GG appended with TLV-code = TLV type (range 0x00 … 0x01FF)
- A TLV is always padded after value to a 32-bit word boundary. Length does not include the padding:
  → HL = ∑(3+TLV-length + TLV padding) + 8 + SidL + TidL +(nrof TLVs>127), where the sum is over all TLVs.

TLV Codes: Payload, RLOCs, Cntrl Information

NB: GG=00 is reserved for ID-types!

GG=01

Code=0x01	Compressed IPv4 header encapsulated payload
Code=0x02	Reserved for Compressed IPv6 header encapsulation
Code=0x03	Reserved for Ethernet encapsulated payload
Codes=0x4…0x7F	other encapsulation types (Code=0x0 = all payload types)

GG=10

Code=0x1	IPv4 RLOC and RLOC preferences (IPv4 Reachability info)
Code=0x2	IPv6 RLOCs and RLOC preferences (IPv6 Reachability info)
Code=0x3	Ethernet (MAC address) RLOCs and preferences (MAC Reachability info)
Code=0x4…0x7F	Reserved for other RLOC types and their preferences (Code=0x0=all RLOC types)

GG=11

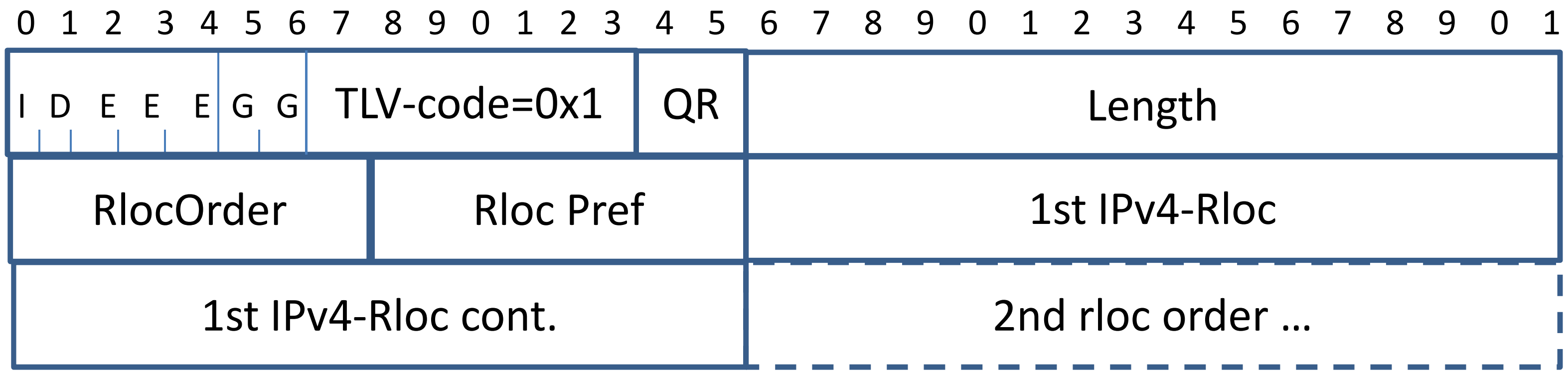
Code=0x1	Timeout (TOUT) of customer edge state
Code=0x2	Cookie
Code=0x3	Address of Certification Authority (such as MO HSS)
Code=0x4	FQDN
Code=0x5	Header signature (over compulsory header + all TLV-types)
Code=0x6	Unexpected message report
Code=0x7…0x1F	Reserved
Code=0x20 & 0x21	Backoff Codes
Code=0x22…0x7F	Reserved for future use

Reliability of TLV signaling

TLV	Recommended approach
RLOC	Outbound edge solicits its own RLOCs in 1st message after DNS query with QR=00; Inbound edge responds with QR=01 if RLOC state is the default that is available in DNS; if RLOC state is non-default, inbound edge MAY respond with QR=10 asking the outbound edge to confirm reception of the non-default state of RLOCs
TOUT	Outbound edge solicits its TOUT in 1st msg after DNS query with QR=00; inbound edge responds with QR=01, … the exact flow may depend on the values of TOUT and inbound CES policy
Cookie	Sent in QR=10, if next message (QR=11) does not contain cookie, the connection is deleted
ID type Req	Inbound edge can send as QR=00 or QR=01 (this is like FYI), inbound edge will forget about the connection immediately (other sequences are possible but this takes the min effort from inbound edge). A new flow must start with new ID type from scratch with RLOCS and TOUTs etc.
FQDN	Sent by inbound edge to learn the domain name of the sender host (QR=00); if there is no response, the connection will be deleted; Response may be either QR=01 or QR=10; in the last case ACK QR=11 with the FQDN of the outbound host is returned; if there is not ACK, resp may be repeated
TLV query	If outbound edge does not follow default policy and does not solicit RLOC and TOUT or for monitoring purposes, inbound edge may send a query with several TLVs set to QR=00 without soliciting its own information or with its own information
Header signature	Either party can send QR=00, in practice this TLV goes together with other TLVs and QRs are set as needed.
Unexp M rep	May be sent either QR=01 (no response expected) or QR=10, ACK expected.
Backoff codes	May be sent either QR=10 (ACK expected) or QR=01, no ACK expected.

RLOCs: QR=00 may carry and QR=01 or 10 MUST carry one or more RLOCs

GG=10



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

I D E E E G G | TLV-code=0x1 | QR | Length

RlocOrder | Rloc Pref | 1st IPv4-Rloc

1st IPv4-Rloc cont. | 2nd rloc order …

Length = 6* NROF RLOCs (2 octet encoding for length in order to align the rest with 32-bit boundary)
If QR=10 was used for response; By QR=11, querier may return its decision on which RLOC it will use (querier may refuse to communicate on other RLOCS from this point onwards)
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 sender's routing locators (except when QR=11)
NB2: Code=2 – reserved for IPv6 Rlocs, Code=3 – MAC Rlocs (48 bit), Codes=0x4….0x7F other Rloc types (RlocOrder and RlocPref apply to all these types).
Flags: ID=00 → querier tells the responder that if it does not have additional RLOCs, ignore the TLV.

On-demand multihoming routing mechanics (1)

- Learning RLOCs:
  - Outbound CES can learn all inbound CES RLOCs and their default state from the DNS query
    - Both edges should use CETP to solicit changes to the default preferences (this is default policy)
  - Inbound CES can use CETP to learn all RLOCs of the outbound CES if (oCES did not follow default policy)
    - CETP Query MAY contain RLOCs (QR=00): if current state of RLOCs at Inbound edge differs from default as stored in DNS, Query carries the current preferences to outbound edge
    - If the requirements of admission by the inbound edge are not fullfilled, iCES may ignore the query 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
    - With the sequence QR=00 →QR=10 →QR=11, both edges can always converge into the preferred set of RLOCs allowing both edges to close other RLOCs for the current flow?
- Monitoring RLOCs
  - CETP can be used to monitor and report the state and state changes of all alternative RLOCs
  - Connection state Timeout sets the pace of monitoring
  - CES may accept packets for an ongoing session from all alternative source RLOCs (or it may use a stricter policy and allow only CETP level monitoring on standby RLOCs until RLOC switchover)

On-demand multihoming routing mechanics (2)

- Swapping remote RLOC
  - If CES receives a (QR=00 or QR=01) 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 (RPF goes together with multicast support), 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)

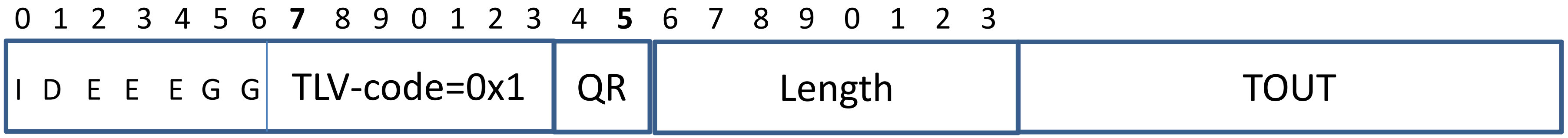
- Revoking local RLOC
  - CES sends an RLOC TLV (QR=00) with its RLOCpref=0xFE for which there is ongoing session (asking for response to confirm reception)
  - If the same CES has alternative RLOCs, having requested an RLOC switchover, CES MUST immediately start accepting traffic for the ongoing session 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.
  - The revoking CES must also send the RLOC switchover command to other CES in the same network.
  - Host standby CES MUST immediately start accepting traffic for the session
  - All of previous slide's story on RPF and state mirroring applies here as well
- 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 is an item for further study.

Time-out of the Customer Edge state

GG=11



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

I D E E E G G | TLV-code=0x1 | QR | Length | TOUT

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

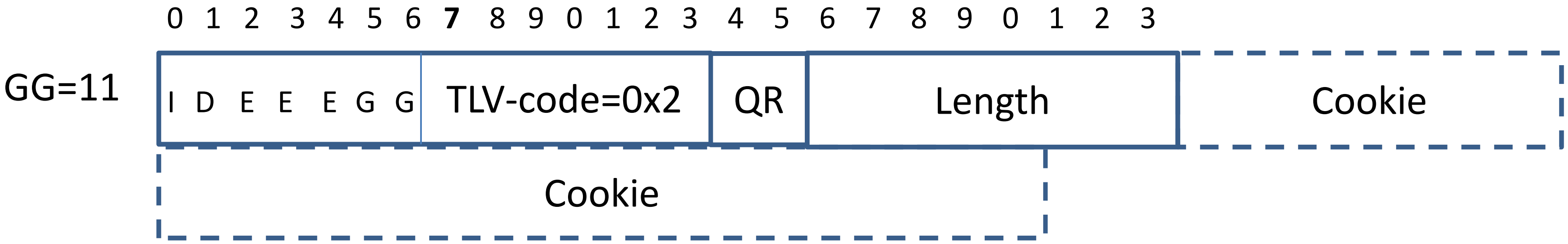
For remote TOUT = N (<local TOUT), local CETP sets a timelimit of N/2 – 1 and
- on expiry will resend a monitoring message (if there has been no traffic)
- the CES with longer timeout will release the connection when its own TOUT expires and will report this to the remote end by sending TOUT-TLV=0.

Flags: ID=00; support is compulsory, so the values of the flags do not matter.

Revoking an ID with TOUT TLV

- If (QR=00 or = 01 or = 10) TOUT=0, sender tells the remote edge that it is removing connection state.
  - with QR=00, sender asks for confirmation, expects that the remote end also tells that it has done the same
- 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.

Cookie TLV may be used to postpone creating connection state at inbound edge



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

GG=11

I D E E E G G

TLV-code=0x2

QR

Length

Cookie

Cookie

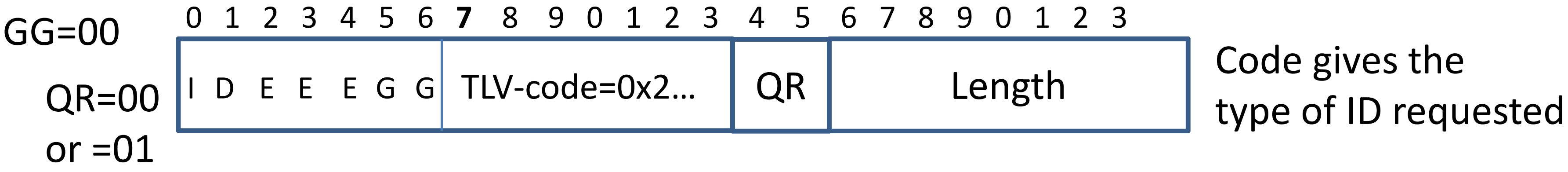
- Querier sets (QR=10) and sends its cookie,
- if policy says that inbound edge will respond with cookie, it expects to receive a message (QR=11) with the same cookie back from outbound edge before it admits the flow.
- Length gives the length of Cookie in octets
- Remote end must return Cookie as such 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. 30 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)
  - With TCP this has its difficulties but we have seen some advanced ”L2-switches”, that support the same idea
  - SCTP has been designed for this
- Inbound edge wants to use mobile operator assured or certified identity → sends (QR=10) response with cookie → ingress has to re-start the flow with mobile operator assured ID (or certificate issued by MO)
  - New message from the initiator 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/ CA address response can be used at the beginning of a new flow



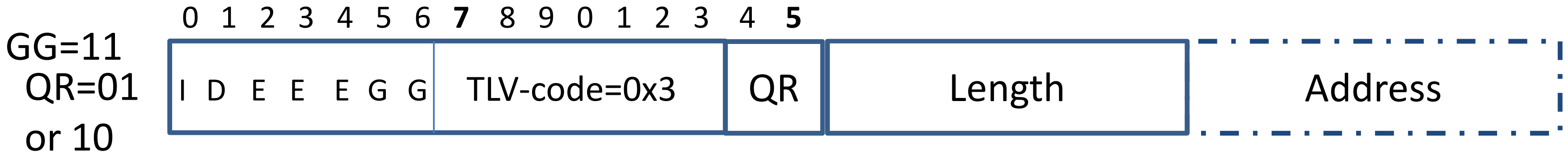
GG=00

QR=00
or =01

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

I D E E E G G	TLV-code=0x2…	QR	Length

Code gives the type of ID requested



GG=11
QR=01
or 10

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

I D E E E G G	TLV-code=0x3	QR	Length	Address

- If GG=00 & QR=00 or =01, this TLV defines a request or advice for a new ID (type)
  - E.g. Inbound edge requires a mobile operator assured ID (MAID) or certificate
  - QR=00, querier may keep state, QR=01 – querier does not keep state
  - Length must be present, because the TLVs appear in the optional TLV header, if length=/=0, the sender solicits its own preferred value in the Value field. For now, in ID request Length == 0!
- CA address TLV (code =0x3), Value gives the routable address for assurance queries (addess of HSS or some other CA), if the new ID is not certified but just assured, no address is required, thus CA addess TLV is not needed, just start with the new ID; Format of the address is TBD.
- 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 Query and Response together
- NB: ID query is described separately from other TLV queriers because connection state handling is different in the 2 cases.

On Mobile Operator assured ID

- The number of mobile broadband subscriptions (in 12/2011) was about 1.2B = half as many as there are Internet users – there is a huge potential in big cities on the emerging markets and a large potential in developed countries.
  - Growth of mobile BB is about at the same pace as growth of Internet users (ITU data 12/2011)
  - 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 has 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
- 4 types of request/answer transactions are needed
  - Edge node MUST be able to request for MOC or MAID allocation 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
  - HSS (to CES): HSS must be able to monitor that a particular ID is still in use in the CES (ID needs a validity timeout) and cancel the ID from an edge node e.g. in handovers
  - It must be possible to revoke an ID at any time (for example when under attack)
  - 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

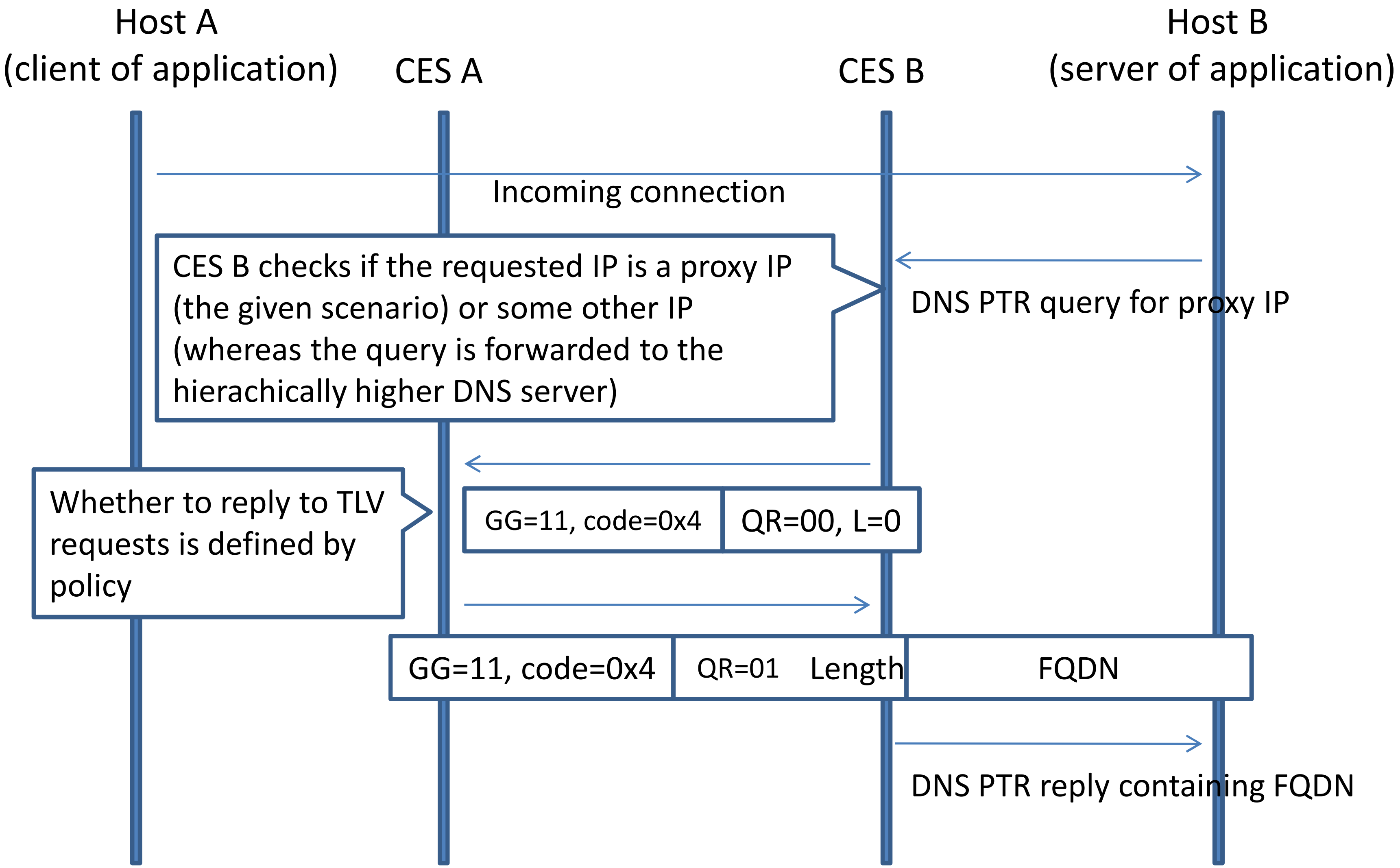
GG=11

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

I D E E E G G	TLV-code=0x4	QR	Length	FQDN

- Query (QR=00) may solicit own FQDN but this is usually not needed
- Response (QR=01)transports the FQDN associated with the sender ID
- If no FQDN is associated with the sender ID, a TLV with length=0 is sent (Or should the responder use backoff code BR1 etc…?)
- 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
- TBD: Instead of responding with length=0, could the destination in some cases decide to generate a domain name dynamically?

Example reverse DNS lookup



Host A (client of application)

CES A

CES B

Host B (server of application)

Incoming connection

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)

DNS PTR query for proxy IP

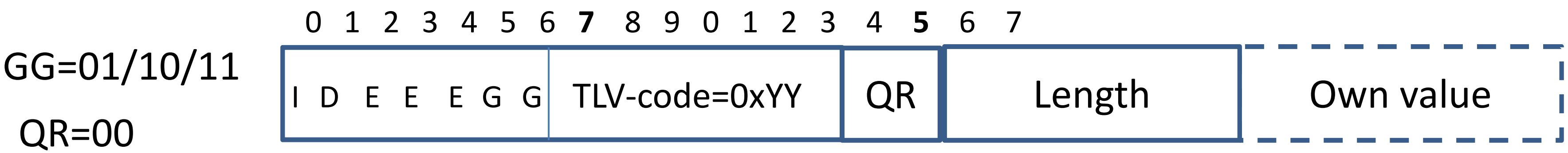
Whether to reply to TLV requests is defined by policy

GG=11, code=0x4	QR=00, L=0

GG=11, code=0x4	QR=01 Length	FQDN

DNS PTR reply containing FQDN

On-demand TLV request



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

GG=01/10/11

QR=00

I D E E E G G | TLV-code=0xYY | QR | Length | Own value

- Lengths are set = 0, if the querier does not wish to solicit its own values
- 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 first make sure that all our own requirements are met before reponding.
- 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 = (GG=01) a certain encapsulation type indicated by code (length=0) (Note this is different from the dataplane payload that appears as last element in the packet this TLV appears within TLV header while the dataplane payload is after TLV header)
  - Requested TLV type = (GG=10) code=0x1: Request for IPv4 RLOC information
  - Requested TLV type = (GG=10) code=0x2: Request for IPv6 RLOC information
  - Requested TLV type = (GG=11) code=0x1: Request for TOUT information
  - Requested TLV type = (GG=11) code=0x3: Request for CA address (by default solicited)
  - Requested TLV type = (GG=11) code=0x4: Request for Domain information
  - Requested TLV type = (GG=11) code=0x5: Request for Header signature

Signing CETP headers

- Plain text return routability check reveals a flow that is trying to spoof an RLOC. The need for signed RLOCs 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.
- As we sign RLOCs, why not the whole CETP header (compulsory+TLVs) ? A new TLV is needed!

GG=11

QR=00

& QR=01 or 10 or 11

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

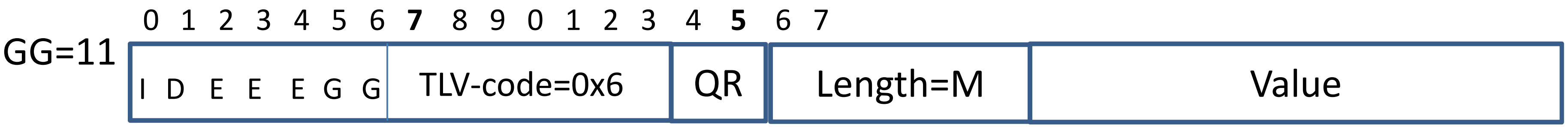
I D E E E G G	TLV-code=0x5	QR	Length	Nested signature TLV

- The format for the signature is TBD (names, IDs, signature algorithms, PKI information)
- To make this possible, the responding CES must have an ID itself and it must be registered into a certification authority (e.g. MO HSS?)
- It seems logical that the signature TLV appears as the last TLV in the TLV header…

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
- When 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

QR=01 or QR = 10 or 11



GG=11

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

I D E E E G G	TLV-code=0x6	QR	Length=M	Value

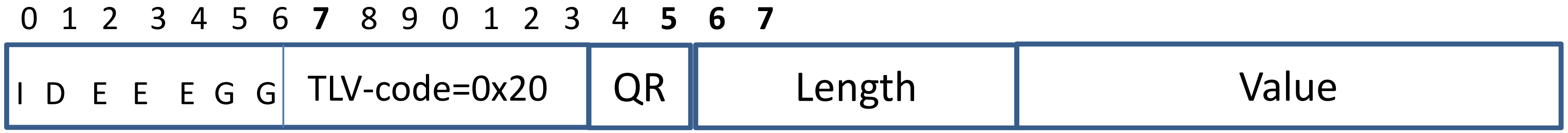
- Counting to N avoids amplification effect
- Value is the copy of the first M bytes of the unexpected ”response”.
- Question: would reporting the count and time over the count be of any help?
- If QR=10, ack (QR=11) will echo the TLV value that was sent.

Processing unexpected message reports

- Upon reception of TLV (GG=11; Code=0x6), 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?

Backoff Codes (1)

GG=11



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

I D E E E G G | TLV-code=0x20 | QR | Length | Value

- In TLV=0x20, with QR=01, One-octet back-off codes MAY be reported
- Example values:
  - (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)

Backoff Codes (2)

GG=11

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

I D E E E G G	TLV-code=0x21	QR	Length	Value

- In TLV=0x21, with QR=01 or 10,
  - 1st octet of value=error code = 0x1
    - Unknown TLV – added for backwards compatibility for future versions, in this case Length = 1+ length of the unknown TLV and
    - after the error code, the unknown TLV is sent back
    - (a different codepoint is used because the value structure differs from the normal backoff code value structure)
  - 1st octet of value= error code = 0x2
    - Not supported TLV (e.g. not supported ID type, not supported encapsulation, not suppported RLOC type or not supported optional future control TLV) – ”not supported” is because of policy or lack of some physical resources.
    - After the error code the not supported TLV type or the full TLV is sent back

Reporting Unwanted Traffic and Malware

- CETP does not have means to report unwanted traffic or malware to the sender
  - If RLOC spoofing by hosts is eliminated, the sender's CES would be not compromised, so sending such reports may make sense
  - Malware and unwanted traffic detection may take some time, thus connection state on both edges may have been deleted before the detection of malware
  - Such reporting would open a possible attack: bad-mouthing innocent hosts which leads to denial of service.
- Because of the latter 2 reasons, such reporting is separated from CETP into other functions of the Internet Trust Framework: processing of such evidence is tempered at all stages by the assessment of trustworthiness of the sender

Thoughts on 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
- If the remote end recognises the ID, but there is no state for the pair of Ids
  - If C=0: state MAY be created (e.g. if CES has banned the source RLOC, it will ignore the message) [default is that first msg of a flow has C=1]
  - C=1 (incl. e.g. RLOC and TOUT TLVs with QR=00 or 01): 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 (QR=01) messages
- If CES is waiting for a response and it has state for the pair of IDs, 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

Admission Policy Examples

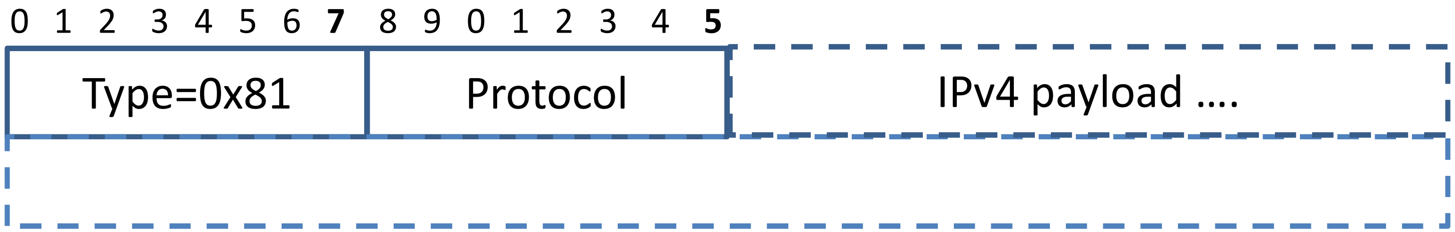
- WWW server:
  - admit max N inbound flows (ID pairs) at a time (N might depend on device type), can be managed by the Policy Control for mobile users.
  - Option: DDOS detection counting active/dormant
- Limited WWW access to a target ID
  - if (source RLOC XOR Mask=nn), Execute full return routability 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: MAID required, count
  - If count >N, ignore (stop responding for time T)

Principles for Payloads over CETP

Payload is not included in HL. Payload length is given in the Fixed part of the header

- Payload belongs to the ”dataplane”
  - Payload is transferred from Edge to Edge using the classical Best Effort IP-service or similar packet transport service (Ethernet etc…)
  - A Payload must appear after the compulsory control header if C=0, if not, the message is ignored (i.e. TOUT keeps running out if state exists).
  - A payload may appear after TLV-headers if C=1
- In the payload, payload-type is one octet. This includes the last G-flag + the payload TLV-code. No other flags are present – they would imply some negotiation and that belongs to the control plane.
- If a negotiation of which payload encapsulation to use is needed, that takes place with the TLV-headers.

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=0x81

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 = Payload length + 16 (PL encoding is like in IP header)

+ Fragmentation can not be used

+ TTL = core IP TTL - 1

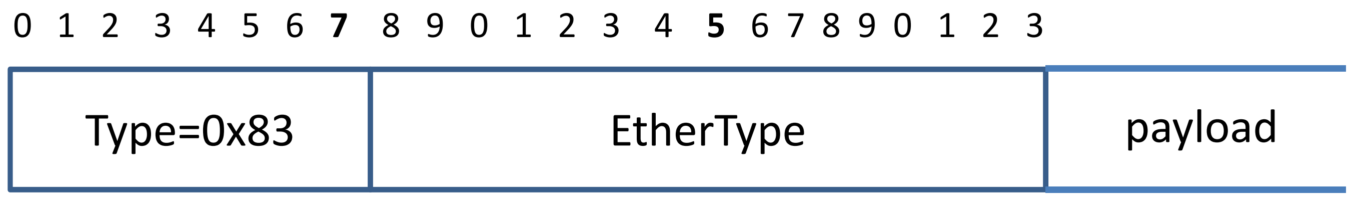
+ Protocol = copied from the above element

+ Header Checksum – calculated locally

+ Source IP address: allocated by CES locally

+ Destination IP address – mapped by CES locally

CETP may carry any payload protocol that runs over Ethernet



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=0x83 | EtherType | payload

NB1: If payload is IPv4, source and destination address fields are set = 0, and reset to appropriate values by the receiving CES
Alternatively, values of IP addresses =/= 0, both customer networks must be in the same (private) IP address space

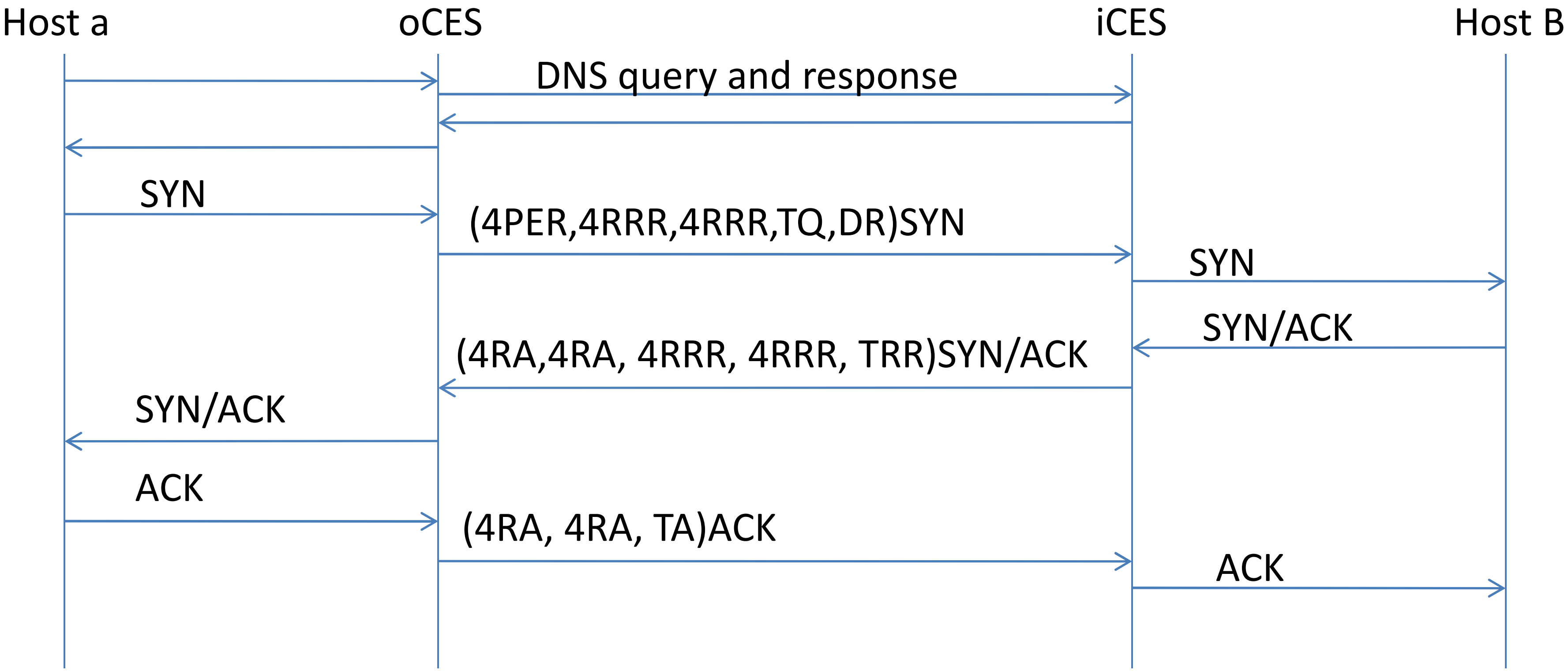
NB2: if EtherType for CETP is defined, one CETP message can carry another CETP message making it possible to monitor many Ids with a single message between 2 Customer Edge Nodes.

NB3: This encapsulation allows CES to support fragmentation using IP

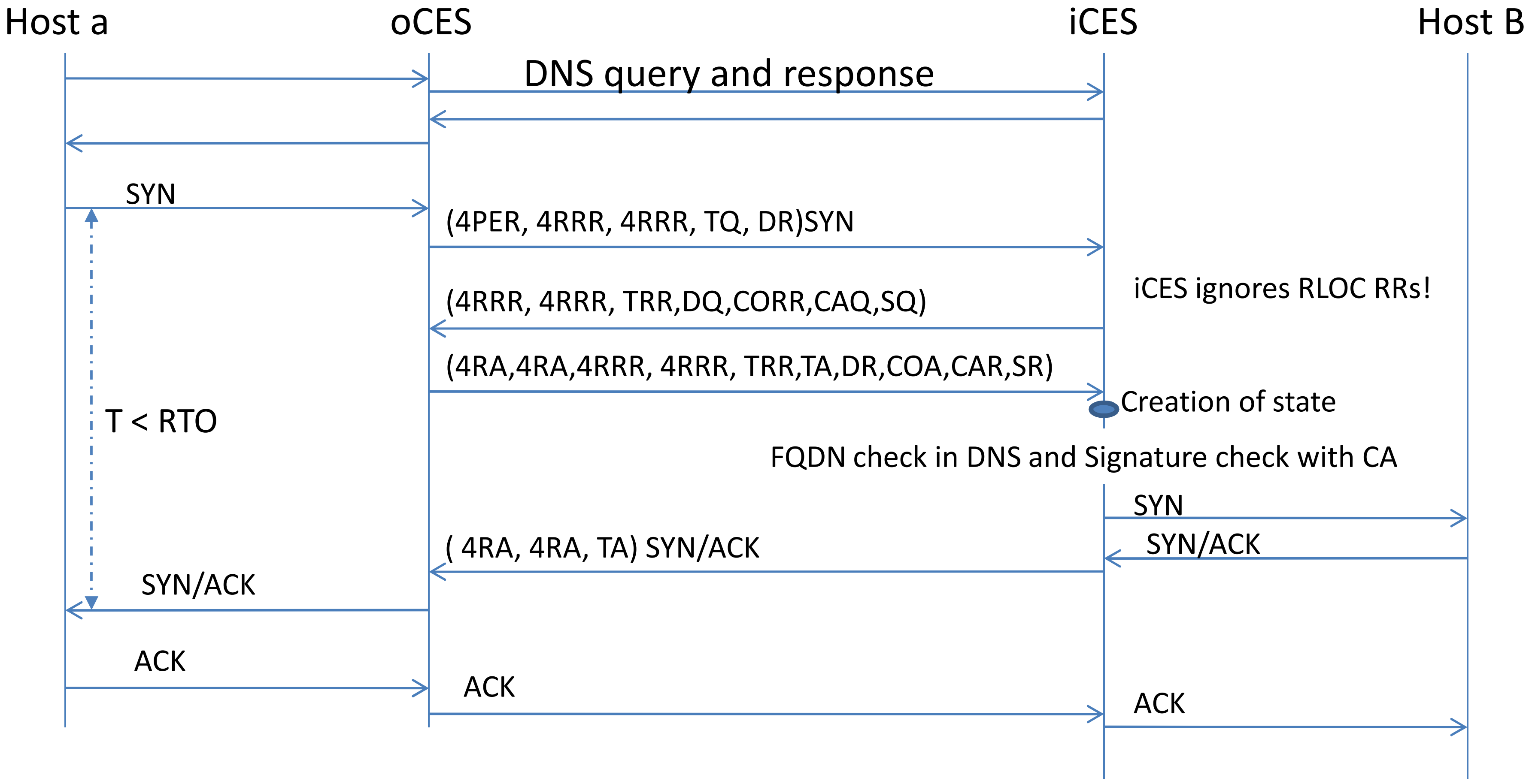
TLV mnemonics

QR → 00=Q, 01=R, 10=RR, 11=A		Hex (GG+Code+QR)
Identity Query	IDQ	0x4…0x1FC = 0x4 × (0x1 … 0x7F)
Identity response	IDR	0x4 × (0x1 … 0x7F) + 1
all Payload Encap Query	aPEQ	0x200
IPv4 Payload Encap	4PEQ, 4PER, 4PERR, 4PEA	0x204…0x207
IPv6 Payload Encap	6PEQ, 6PER, 6PERR, 6PEA	0x208…0x20B
Eth Payload Encap	EPEQ, EPER, EPERR, EPEA	0x20C…0x20F
all RLOC Query	aRQ	0x400
IPv4 RLOC	4RQ, 4RR, 4RRR, 4RA	0x404…0x407
IPv6 RLOC	6RQ, 6RR, 6RRR, 6RA	0x408…0x40B
Eth RLOC	ERQ, ERR, ERRR, ERA	0x40C…0x40F
TOUT	TQ, TR, TRR, TA	0x604…0x607
Cookie	CORR, COA	0x60A…0x60B
CA Address	CAQ, CAR, CARR, CAA	0x60C…0x60F
FQDN	DQ, DR, DRR, DA	0x610…0x613
Signature	SQ, SR, SRR, SA	0x614…0x617
Unexpected M rep	UR, URR, UA	0x619…0x61B
Backoff Code	BR0, BRR0, BA0	0x681…0x683
Backoff Code	BR1, BRR1, BA1	0x685…0x687

Example of Successful Flow (TCP) with lax admission policy



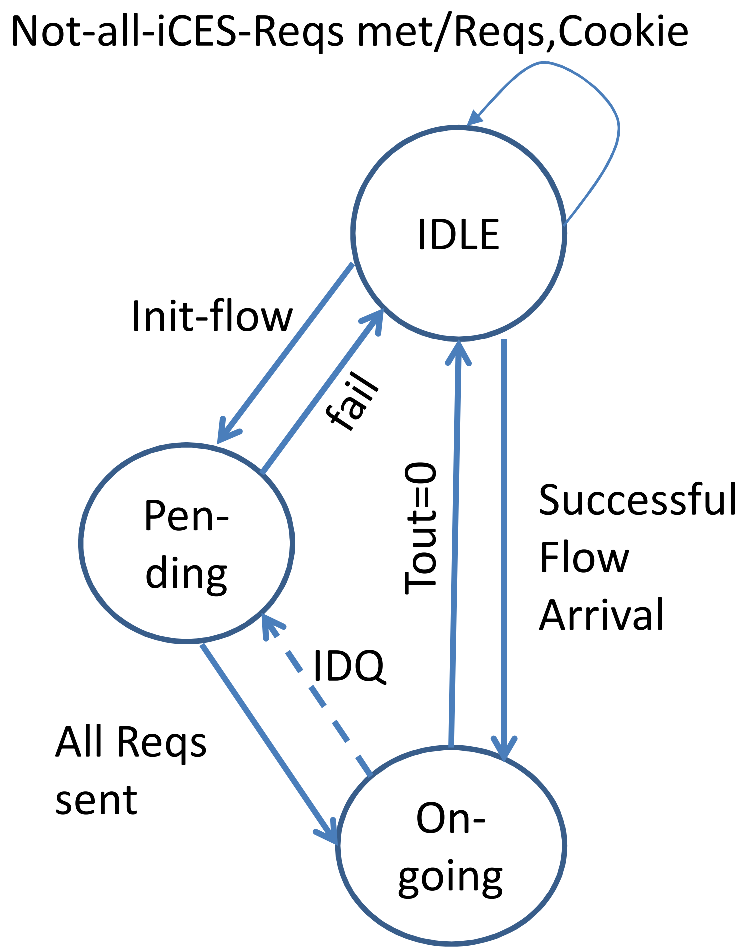
Example of Successful Flow (TCP) with strict admission policy


Host a
oCES
iCES
Host B
DNS query and response
SYN
(4PER, 4RRR, 4RRR, TQ, DR)SYN
iCES ignores RLOC RRs!
(4RRR, 4RRR, TRR,DQ,CORR,CAQ,SQ)
(4RA,4RA,4RRR, 4RRR, TRR,TA,DR,COA,CAR,SR)
Creation of state
T < RTO
FQDN check in DNS and Signature check with CA
SYN
SYN/ACK
(4RA, 4RA, TA) SYN/ACK
SYN/ACK
ACK
ACK
ACK

Policy control of CETP

- If initiating a new flow
  - List of TLVs with QR=00 to embed + yes or no for soliciting own values
  - List of TLVs with QR=01 or 10 to embed with own values
- If receiving a new flow attempt
  - Per received TLV type (QR=00): store solicited value plus not respond, respond with QR=01, or QR=10
  - List of TLVs with QR=00
- If ongoing outgoing flow
  - Per received TLV with QR=01, process and store value
  - Per received TLV with QR=10, preocess and store value and either send ACK or ignore
  - Per received TLV with QR=00, store solicited value, plus either ignore, or respond with QR=01 or QR=10
  - Per received TLV with QR=11, check value, if wrong repeat QR=10 (max N times, default N=3)
- If ongoing incoming flow
  - Per TLV with QR=01, process and store value
  - Per TLV with QR=10, preocess and store value and either ignore ignore ack or send ack
  - Per TLV with QR=11, check value, if wrong repeat QR=10 (max N times, default N=3)

States in Policy Engine



- IDLE = there is no flow. iCES will stay in Idle until all its requirements are met by oCES.
- Pending= CES must queue packets from a served host until it is able to move to ongoing. Having sent the first CETP packet oCES always goes to Pending. When oCES sends all required TLVs, it goes to ongoing and empty its host side queue. iCES may go to Pending in order to swap Ids for an ongoing flow. TOUT in Pending is short.
- Ongoing=switch packets normally from CETP to host and from host to CETP. TOUT in Ongoing is long (at the beginning may be short).
  - In CETP packet, if C-flag set, process TLVs, store response, set Stored C-flag
  - Upon a packet from a served host, dataplane will check C-flag, if set, it will embed the TLV string to the packet (if there is room) or send the string first.

Policy definition by manager

- Required TLVs in the role of iCES
- Required TLVs in the role of oCES
- Offered TLVs in the role of oCES
- Offered TLVs in the role of iCES
- Reliability policy for TLVs (use R or RR)
- ID policy (application, etc…)

Policy Engine returns a decision

- Admit (optionally log)
- Deny (optionally log)
- Require certain ID type (optionally log)
- Require certain TLV types (if not present) (optionally log)

NB: policy may also be dynamic…

How to carry CETP over Internet

- Option 1: A new EtherType is defined
  - CETP is carried over Ethernet core network
- 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 IP/CEPT provides Best Effort Communications
- It is assumed that packet access control in the inbound edge node is based on policy
  - This policy is managed by the network administrator (like Firewall policies today), (or could be controlled by the user device – has security challenges…)
  - The application developer should publish a policy template (or templates) for admission together with a new application
  - 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, [reliable]responses, acks, monitoring and data messages using CETP to another CES (required level of reliability of signaling is determined by policy)
  - Data may also be embedded in queries, responses and acks for the purpose of reducing the number of messages (or queries/responses can be embedded in data messages – depends on how you want to look at this)
- CETP directly supports minimal encapsulation of payload IPv4 for the case of underlying core IPv4 reducing header overhead created by tunneling
- QR=00/01 allow monitoring e.g. the state of RLOCs and state of the connection →
  - Implementation of 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
  - QR=00 /01 with TOUT TLV → monitoring of the state of the connection
- Execute return routability checks either on forwarding or forwarding and naming levels and even sign RLOCs and other control info cryptographically (authenticity)
  - 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 part of the header and the Ids or also the other control information in TLVs
  - Since there is the crypto signature, simple checksum may not be needed.
  - Checksum does not require PKI infra…
- Support for fragmentation (e.g. for the case when underlying core protocol is Ethernet)
  - A new encapsulation with a fragmentation word equal to what is present in IP header
  - Need for support of fragmentation is not clear because the present implementation leaves that to the IP. Even if packets are routed using Ethernet rather than IP, a host is likely communicating over IP and in all these cases fragmentation is taken care of by IP (at least at the moment)
- Other RLOC types: MAC address + BVLAN (for 802.1ah networks), MPLS-TP MAC address + label etc…
  - Other encapsulations (e.g. IP/MPLS or MPLS-TP or variants of carrier grade Ethernet)?

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 is 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 by hosts becomes impossible (except by compromising the operatator's routing system)
  - 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
  - By asking the responder to sign the RLOCs, stealing RLOCS by compromising the routing system becomes impossible
- But how can we locate the host that is spoofing a source address?
  - CES or the serving PE node may have forwarding table or label mapping table level information of the real source pointing to an interconnected source or transit network – may require logging of traffic which is expensive…
  - 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 outbound CES that provides a public service for any hosts…

Open questions?

- The choice of QR value that we adopted here is based on several considerations, e.g.
  - If the value is the sender’s value (QR=00, 01, 10) or the receivers value (only QR=11)
- There are probably special cases that are not mentioned in the slides
- Move to encryption is now not described (leave to application layer?)
- Formats for signature etc are TBD.
- See no FQDN reporting on slide 26?

Justification of some design choices

- Fixed Part: in earlier version Fixed part was just the first 32 bits of the current structure. We moved the ID type and length information into the fixed part of the header for ease of processing.
- We assume that it is possible to get the total length of the message from the underlying protocol. Alternatively it can be calculated by rounding up the HL to the next value divisible by 4 and adding payload length.
- All compatibility flags are in the forward direction, compatibility responses and acks are given by the Backoff codes.