DICE Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

S. Kumar
Philips Research
S. Keoh
University of Glasgow Singapore
O. Garcia-Morchon
Philips Research

October 21, 2013

DTLS Relay for Constrained Environments
draft-kumar-dice-dtls-relay-00

Abstract

The 6LoWPAN and CoAP standards defined for resource-constrained devices are fast emerging as the de-facto protocols for enabling the Internet-of-Things (IoTs). Security is an important concern in IoTs and the DTLS protocol has been chosen as the preferred method for securing CoAP messages. DTLS is a point-to-point protocol relying on the IP routing to deliver messages between the client and the server. However in some low-power lossy networks (LLNs) with multi-hop, a new "joining" device may not be initially IP routable. This prevents DTLS from being directly useful as an authentication and confidentiality protocol during this stage, requiring other security protocols to be implemented on the device. These devices being resource-constrained often cannot accommodate more than one security protocol in their code memory. To overcome this problem and reuse DTLS, we present a DTLS Relay solution for the non-IP routable "joining" device to establish a secure DTLS connection with a DTLS server. Further we present a stateful and stateless mode of operation for the DTLS Relay.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 24, 2014.

Table of Contents

1. Introduction

The Internet-of-Things (IoTs) vision is more closer to reality with the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [RFC4944] and Constrained Application Protocol (CoAP) [I-D.ietf-core-coap] standards . The 6LoWPAN adaptation layer allows for transmission of IPv6 Packets over IEEE 802.15.4 networks [ieee802.15.4] and thereby enabling end-to-end IPv6 connectivity of "Things". CoAP is a web protocol based on REST architecture designed to work under the special requirements of the constrained environment. It supports binding to UDP [RFC0768] which is preferred over TCP [RFC0793] in low-power lossy networks (LLNs) such as IEEE 802.15.4.

Security is important concern in such a wireless multi-hop network that could be used in various application domains such as smart energy and building automation. However security protocols are often

heavy-weight both in terms of code and network processing. Due to the constrained nature of most of these devices, multiple security protocols for different purposes and at different networking layers are hard to envision. It is more efficient to use a single security protocol to fulfill multiple security requirements in such constrained environments.

CoAP has chosen Datagram Transport Layer Security (DTLS) [RFC6347] as the preferred security protocol for authenticity and confidentiality of the messages. It is based on Transport Layer Security (TLS) [RFC5246] with modifications to run over UDP. DTLS makes use of additional reliability mechanisms in its handshake due to the lack of TCP reliable transmission mechanisms that are available to TLS.

DTLS is a point-to-point protocol relying on the underlying IP layer to perform the routing between the DTLS client and the DTLS server. However in some LLNs with multi-hop, a new "joining" device may not be initially IP routable. A new "joining" device can only initially use a link-local IPv6 address to communicate with a neighbour node using neighbour discovery [RFC6775] until it receives the necessary network configuration parameters. Before the device can receive these configuration parameters, it may need to authenticate itself or wish to authenticate the network to which it connects. DTLS although a suitable protocol for such authentication and secure transfer of configuration parameters, would not work due to the lack of IP routability of its messages to the intended recipients.

We present a DTLS Relay solution to overcome this problem for the "joining" device to establish a secure DTLS connection with a DTLS server. This draft is inspired by the Protocol for carrying Authentication for Network Access (PANA) Relay Element [RFC6345] which is intended to solve a similar problem when PANA [RFC5191] is used for network access. Further we present a stateful and stateless mode of operation for the DTLS Relay.

This draft is an early description of the solutions and does not provide the complete details yet. This draft is structured as follows: we present a use-case for the DTLS Relay in Section 2, then present the DTLS Relay solution in Section 3 for stateful and stateless mode of operation. Further we present some security considerations in Section 5.

2. Use Case

We present here a target usecase based on [I-D.jennings-core-transitive-trust-enrollment] describing a rendezvous protocol that allows a constrained IoT device to securely connect into a system or network. The main idea is that the joining

Device has a pre-established trust relationship with a "Transfer Agent" entity, for e.g. Pre-Shared Keys provisioned during manufacturing. This "Transfer Agent" provides the needed trust credentials to the Device and/or a Controller in the system to establish a secured connection to perform further authentication and transfer of system/network configuration parameters. This step is enabled by an "Introducer" entity which informs the "Transfer Agent" about the details of Controller to which the joining Device should connect, and provide to the Controller the identity including one-time credentials for enable secure connection to the Device. The transitive trust trust establishment procedure is explained in detail in [I-D.jennings-core-transitive-trust-enrollment] and we focus here on how to enable this using DTLS.

As depicted in the Figure 1, the joining Device (D) is multi-hop away from the Controller (C) and not yet authenticated into the network. At this stage, it can only communicate one-hop to its nearest neighbour (N) using their link-local IPv6 addresses. However, the Device needs to communicate with end-to-end security with a Transfer Agent (T) or to Controller (C) to authenticate and get the relevant system/network parameters. If the Device (D), initiates a DTLS connection to the Transfer Agent that has been pre-configured, then the packets are dropped at the neighbour (N) since the Device (D) is not yet admitted to the network or there is no IP-routability to Device (D) for any returned messages.

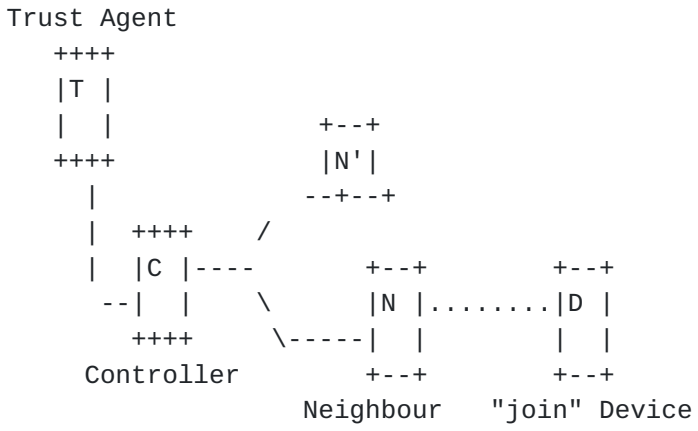


Figure 1: Use case depiction in a multi-hop network

Further the Device (D) may wish to establish a secure connection to the Controller (C) in the network assuming credentials are exchanged out-of-band, for e.g. a hash of the Device (D)'s raw public key could be provided to the Controller (C). However, the Device (D) is

unaware of the IP address of the Controller (C) to initiate a DTLS connection and perform authentication.

To overcome these problems with non-routability of DTLS packets and/or discovery of the destination address of the DTLS server to contact, we define a DTLS Relay solution. This DTLS Relay ability is configured into all authenticated devices in the network which may act as the Neighbour (N) device for newly joining nodes. The DTLS Relay allows for relaying of the packets from the Neighbour (N) using IP-routing to the intended DTLS server. Further, if the DTLS server address is not known to the joining Device (D), then messages are delivered to a pre-configured DTLS server address (mostly the Controller (C)) known to the DTLS Relay.

3. DTLS relay

In this section, we describe how the DTLS Relay functionality can be achieved. When a joining device as a client attempts a DTLS connection (for example to a "Transfer Agent"), it uses its link-local IP address as its IP source address. This message is transmitted one-hop to a neighbour node. Under normal circumstances, this message would be dropped at the neighbour node since the joining device is not yet IP routable, or it is not yet authenticated to send messages through the network. However, if the neighbour device has the DTLS Relay functionality enabled, it forwards DTLS messages to specific servers. Additional security mechanisms need to exist to prevent this forwarding functionality to be used by rogue nodes to bypass any network authentication procedures and are discussed in Section 5.

The DTLS Relay can operate in two different modes: stateful and stateless. We present here both the methods, however for inter-operability, only one of the modes should be mandated. Within each mode, the DTLS Relay can further forward packets based on the client defined DTLS server address or a DTLS server address that has been configured into the Relay.

3.1. Relay in Stateful mode

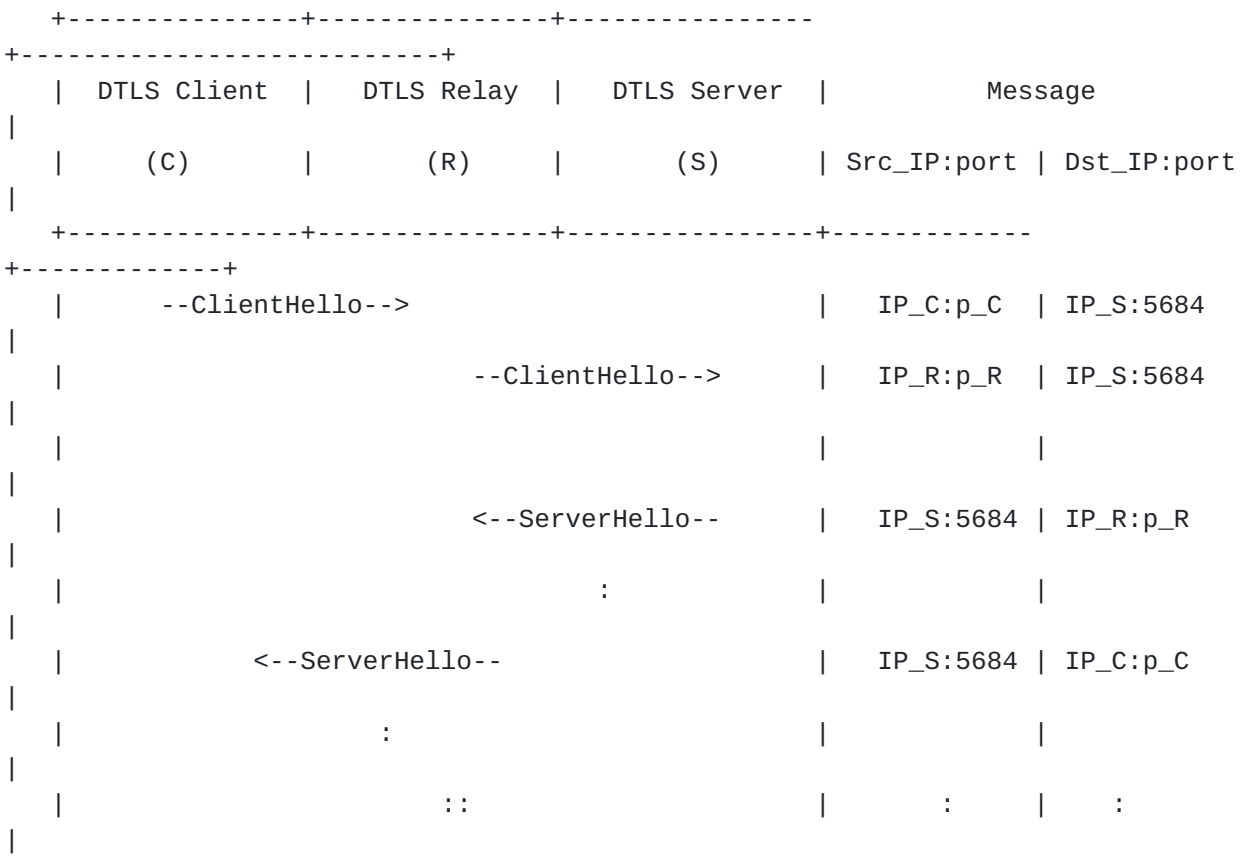
The neighbour node on receiving a DTLS message from a joining device enters into DTLS Relay mode. In this mode, the neighbour node has the additional functionality to send DTLS messages further to the end-point DTLS server the joining device wishes to contact. In the stateful mode of operation, the message is transmitted to the end-point as originating from the DTLS Relay by replacing the IP address and port to DTLS Relay's own IP address and a randomly chosen port. The DTLS message itself is not modified.

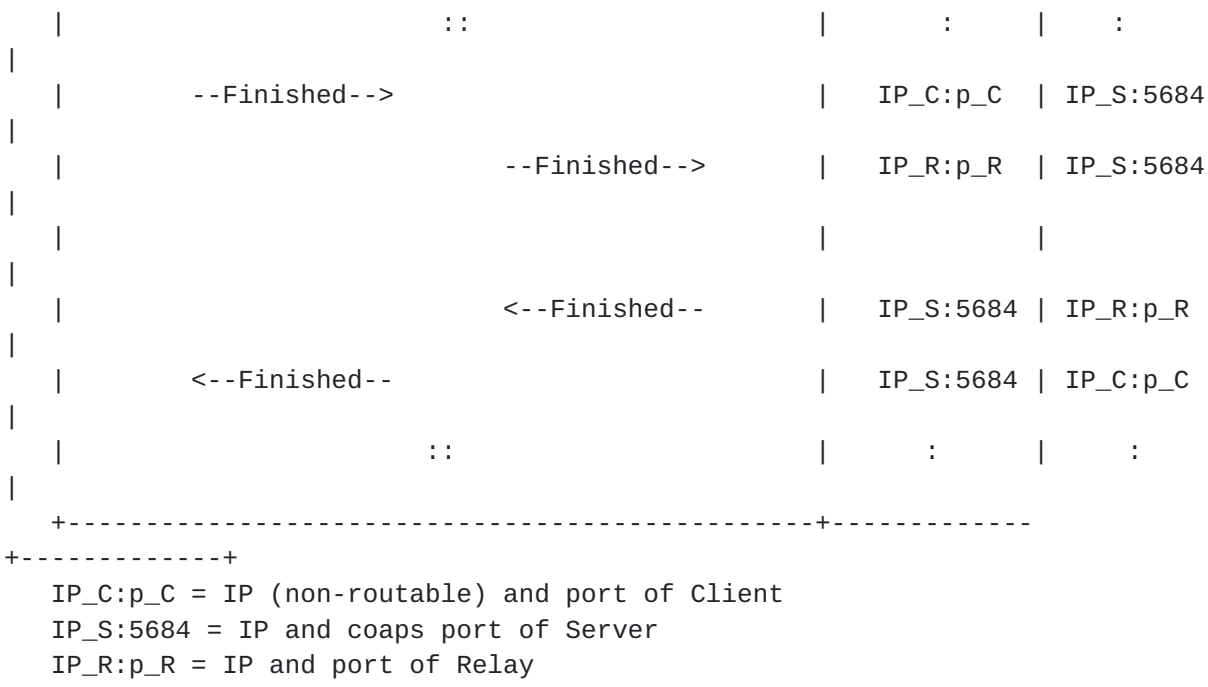
Additionally, the DTLS Relay must track the ongoing DTLS connections based on the following 4-tuple stored locally:

- Client source IP address (IP_C)
- Client source port (p_C)
- DTLS Server IP address (IP_S)
- Relay source port (p_R)

The DTLS server communicates to the Relay as if it were communicating to the end-point Client with no modification required to the DTLS messages. The Relay on receiving this message, looks up its locally stored 4-tuple array to identify to which joining device (if multiple exists) the message belongs. The DTLS message's destination address is replaced with that of the link-local address and port of the joining device from the lookup array and forwarded to it. The Relay does not modify the DTLS packets and therefore the normal processing and security of DTLS is unaffected.

The following message flow diagram indicates the various steps of the process where the DTLS server address in known to the joining device:

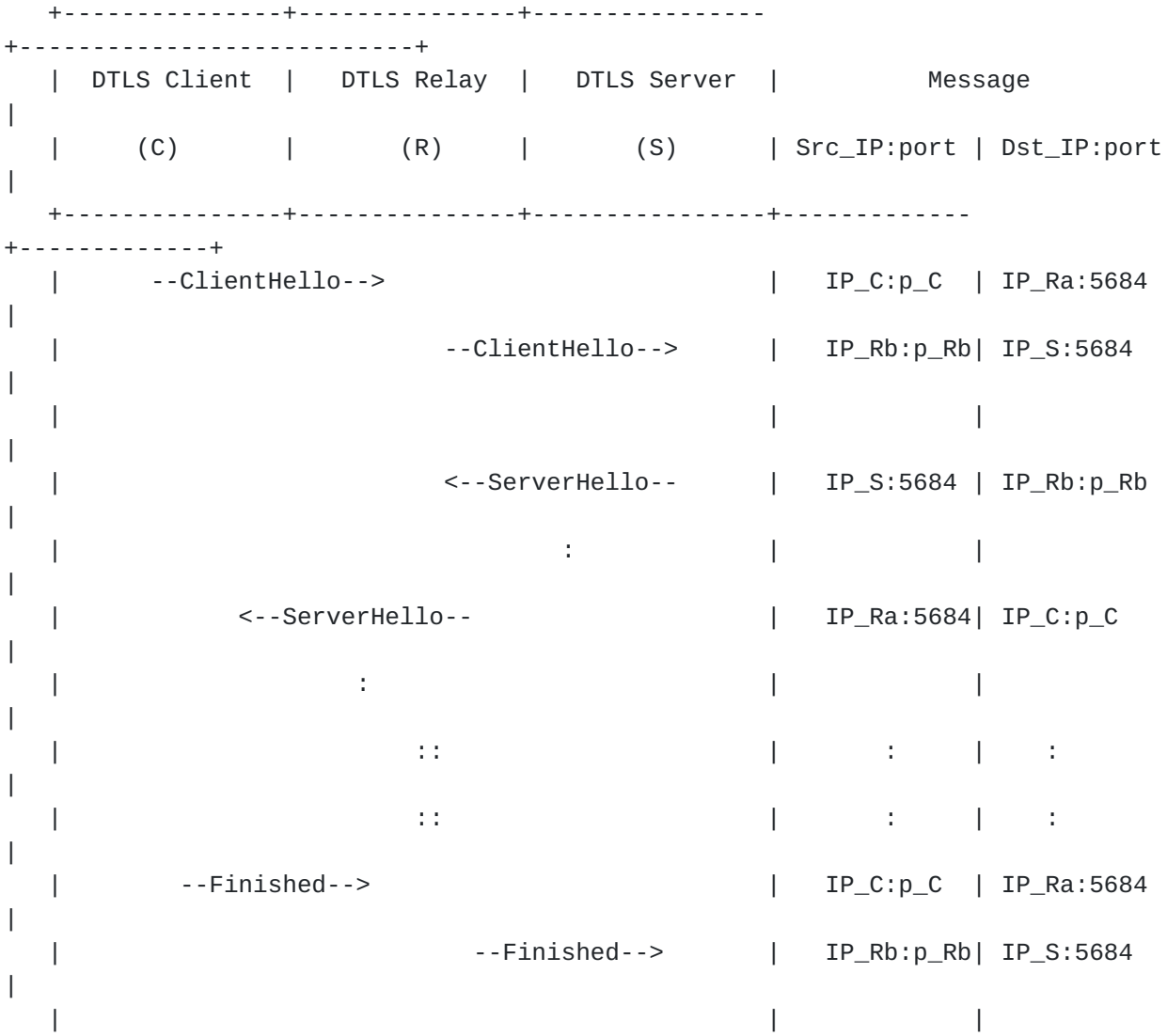




IP_C:p_C = IP (non-routable) and port of Client
IP_S:5684 = IP and coaps port of Server
IP_R:p_R = IP and port of Relay

Figure 2: Message flow in Stateful mode with DTLS Server defined by Client

In the situation where the joining device is unaware of the IP address of DTLS server it needs to contact, for e.g. the Controller of the network, the DTLS Relay can be configured with IP destination of the default DTLS server that a joining device needs to contact. The joining device initiates its DTLS request as if the DTLS Relay is the intended end-point DTLS server. The DTLS relay translates the DTLS message as in the previous case by modifying both the source and destination IP address to forward the message to the intended DTLS server. The Relay keeps a similar 4-tuple array to enable translation of the DTLS messages received from the server and forward it to the DTLS Client. The following message flow indicates this process:



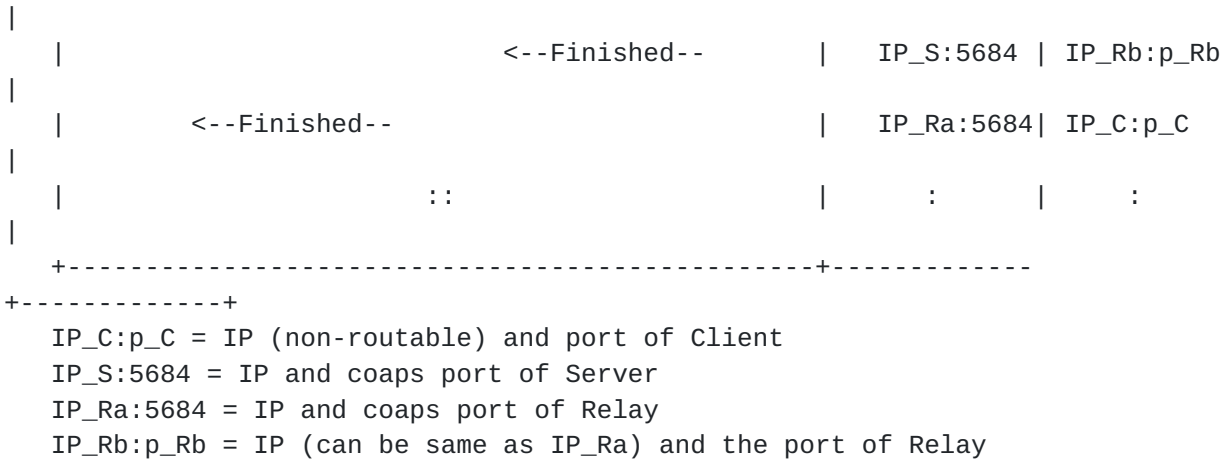


Figure 3: Message flow in Stateful mode with DTLS Server defined by Relay

3.2. Relay in Stateless mode

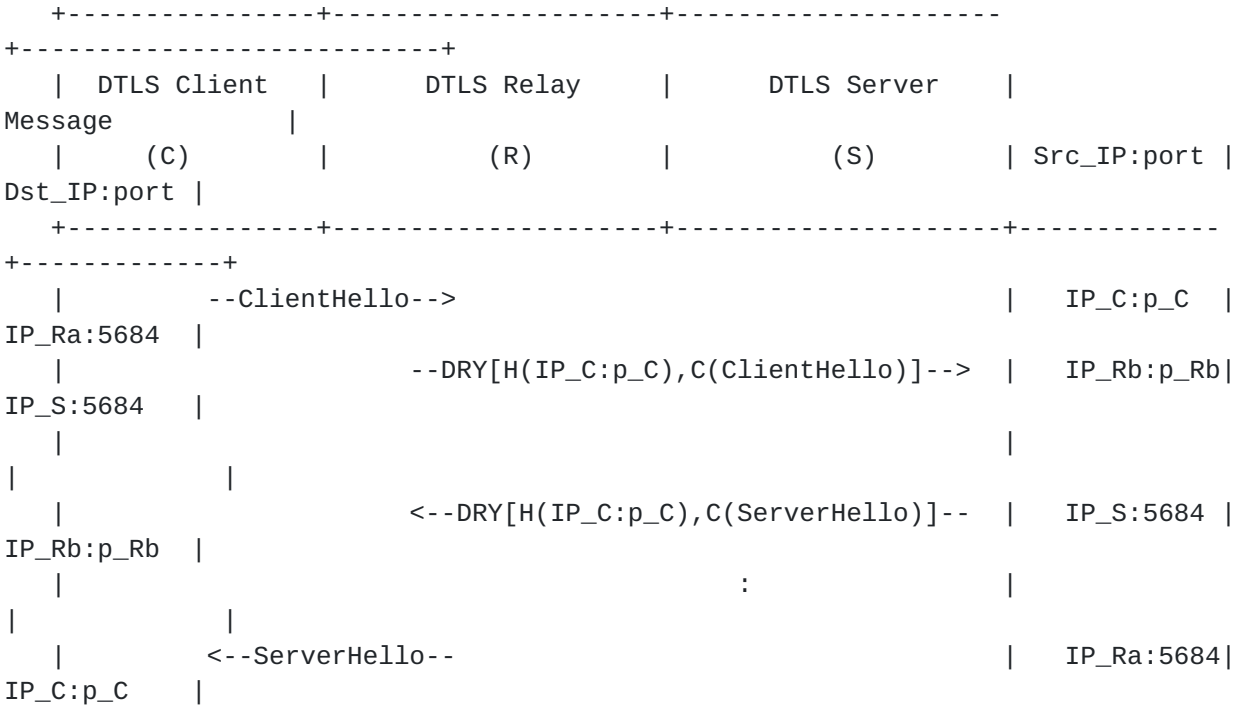
In the alternative mode of operation for the DTLS Relay, a stateless approach is applied where th Relay does not need to store a local 4-tuple array. Just as in the previous case, if a DTLS client with only link local addressing wants to contact a trusted end-point DTLS server, it send the DTLS message to the Relay. The Relay instead of translating, encapsulates this message into a new type of message called DTLS Relay (DRY) message. The DRY consists of two parts:

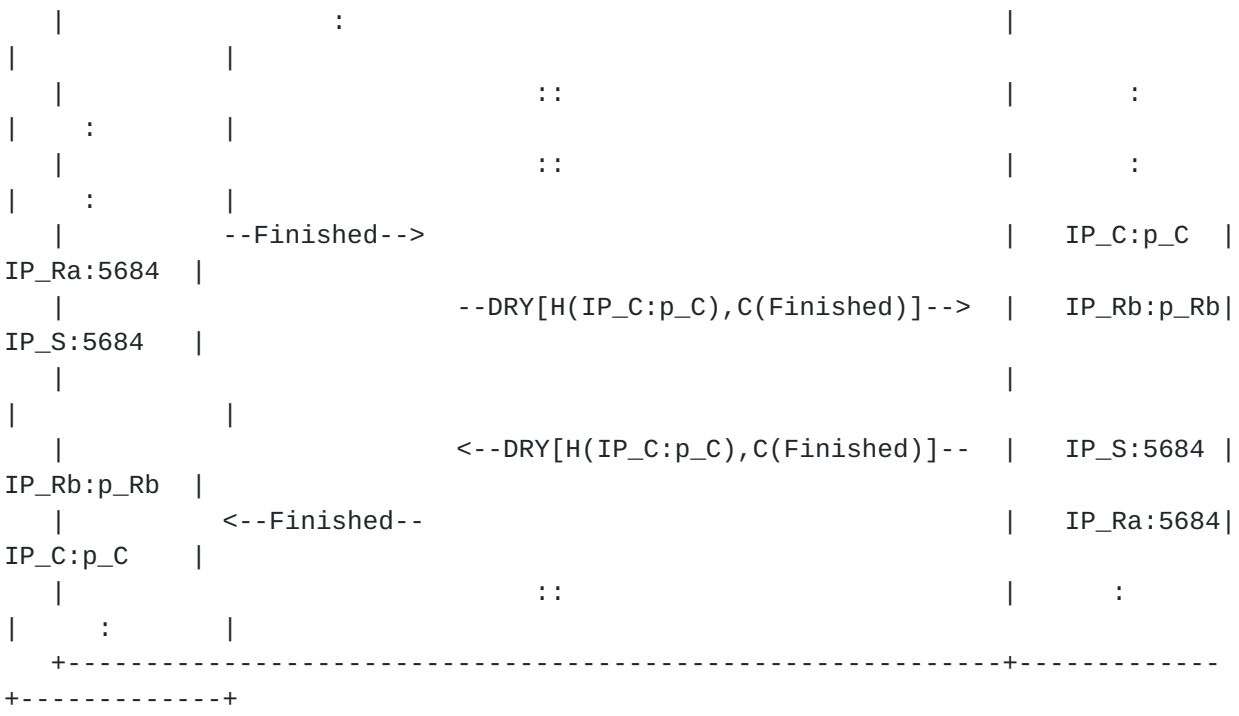
- Header (H) field: consisting of the source link-local address and port of the DTLS Client device, and

- Contents (C) field: containing the original DTLS message.

The DTLS end server on receiving the DRY message, decapsulates it to retrieve the two parts. It then uses the Header field information to associate the new state created on the server for the DTLS connection to the DTLS client's address and port. The DTLS server then performs the normal DTLS operations on the DTLS message contents. However when the DTLS server replies, it also encapsulates its message in a DRY message back to the Relay with the Header containing the original source link-local address and port of the DTLS Client. The Relay can decapsulate the DRY message, retrieves the Header information to forward this message to the right DTLS Client device.

The following figure depicts the message flow diagram when the DTLS server end-point address is known only to the Relay:




--Finished-->
IP_C:p_C
IP_Ra:5684
--DRY[H(IP_C:p_C),C(Finished)]-->
IP_Rb:p_Rb
IP_S:5684
<--DRY[H(IP_C:p_C),C(Finished)]--
IP_S:5684
IP_Rb:p_Rb
<--Finished--
IP_Ra:5684
IP_C:p_C

IP_C:p_C = IP (non-routable) and port of Client
IP_S:5684 = IP and coaps port of Server
IP_Ra:5684 = IP and coaps port of Relay
IP_Rb:p_Rb = IP (can be same as IP_Ra) and the port of Relay

DRY[H(),C()] = DTLS Relay message with header H and content C

Figure 4: Message flow in Stateless mode with DTLS Server defined by Relay

The message flow for the case in which the DTLS Client is aware of the end-point DTLS server's address is similar and not described further. It can be derived based on Figure 2 and Figure 4.

3.3. Comparison between the two modes

A comparison between the two modes will be done in an updated version of this draft.

4. IANA Considerations

tbd

Note to RFC Editor: this section may be removed on publication as an RFC.

5. Security Considerations

Additional security considerations need to be taken into account about forwarding of messages from devices through a network to which it has not yet been admitted. This can lead to denial-of-service attacks or misuse of network resources without proper authentication. One way to overcome any large scale misuse of the network is to have a management message from the Controller that initiates already authenticated devices in the network to enter into a DTLS Relay mode. The devices can stay such a Relay mode for a fixed period of time or until the Controller sends a new management message blocking the DTLS Relay mode in all devices in the network. This is often possible since the administrator of the network can be aware when new devices join the network either because of the "Introduction" phase or commissioning phase.

Other mechanisms based on IP destination filtering can be applied by the controller to all Relay nodes to avoid misuse of the network resources.

6. Acknowledgements

The authors would like to thank Sahil Sharma, Ernest Ma, Dee Denteneer and Peter Lenoir for the valuable discussions and suggestions,

7. References

7.1. Normative References

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

7.2. Informative References

[I-D.ietf-core-coap] Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

[I-D.jennings-core-transitive-trust-enrollment] Jennings, C., "Transitive Trust Enrollment for Constrained Devices", draft-jennings-core-transitive-trust-enrollment-01 (work in progress), October 2012.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

[RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC6345] Duffy, P., Chakrabarti, S., Cragie, R., Ohba, Y., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA) Relay Element", RFC 6345, August 2011.

[RFC6775] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power

Wireless Personal Area Networks (6LoWPANs)", RFC 6775, November 2012.

[ieee802.15.4]
IEEE Computer Society, ., "IEEE Std. 802.15.4-2003", October 2003.

Authors' Addresses

Sandeep S. Kumar
Philips Research
High Tech Campus 34
Eindhoven 5656 AE
NL

Email: sandeep.kumar@philips.com

Sye Loong Keoh
University of Glasgow Singapore
Republic PolyTechnic, 9 Woodlands Ave 9
Singapore 838964
SG

Email: SyeLoong.Keoh@glasgow.ac.uk

Oscar Garcia-Morchon
Philips Research
High Tech Campus 34
Eindhoven 5656 AE
NL

Email: oscar.garcia@philips.com