Network Working Group Leo Brown
Internet Draft Netfuse Telecom
Intended status: Standards Track May 14, 2015

Telephony Route Exchange Protocol (TREX)
draft-trex-01

Abstract

This document outlines a protocol for exchanging telephony routing prefixes between voice switches and routing databases both internally and between carriers, the purpose being to provide a means of rapidly advertising available routes for terminating telephony traffic, along with the features and cost of using these routes such that connected systems are able to build and adjust routing tables accordingly. The protocol has been developed in order to solve a number of problems around number databuild in telecommunication networks, which in 2015 is still a manual and time-consuming process. Due to the manual nature of this work, networks often use inefficient and out-of-date least-cost routing policies, a problem with a significant financial cost which this protocol seeks in part to assuage.

1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

2. Protocol Overview

The Telephony Route Exchange Protocol (TREX, pronounced tee-rex) is a session-based server/server route advertisement protocol, similar in nature to the Border Gateway Protocol [RFC4271] in that a permanently connected session both exchanges information on available routes, and the absence of a connected session may relate to the lack of any route availability from the serving network. The protocol can also be compared to the SMPP messaging protocol in that connecting systems can be senders, receivers, or perform both roles. Finally, this protocol is also closely related to TRIP [RFC3219] but this protocol is concerned primarily with voice gateways rather than just prefix availability and cost.

Sessions in TREX can be established with optional authentication, and route advertisements may contain a number optional Information Elements, helping to define textual representations, cost, as well as dates and times the routes are available within.

3. Packet Structure

The TREX protocol is a binary structure composed of a method ID (1 byte) followed by one or more Information Elements (IEs) which are formed of a 1 byte identifier, followed by data, and are terminated by a NUL (0x00) character. An IE identifier of 0x00 indicates there are no (further) following IEs. Each IE has a predefined type which defines the size of the IE data. Some IEs have variable-length data, these IEs will contain a length parameter between the IE type and the IE data. Each Method has it's own unique list of associated IEs, and IE identifier codes may be reused across different methods.

The following 9-byte message is a CONNECT method at sequence 1, with a single information element giving the hostname simply as "A". In hexadecimal the message would read 010000000101014100.

```
+----------+-----------+--------+---------+------+------+
| 0x01     | 0x00000001 | 0x01   | 0x01    | 0x41 | 0x00 |
+----------+-----------+--------+---------+------+------+
| CONNECT  | SEQ = 1    | HOSTID | LEN = 1 | "A"  | NUL  |
+----------+-----------+--------+---------+------+------+
```

4. Session Methods Overview

The Telephony Route Protocol supports a number of different message types referred to as "methods".

Session methods may contain one or more Information Elements (IEs) which provide content to and modify the method's operation. The methods and their IEs are described in detail in the following chapter.

A summary of the request, reply and other methods is shown below.

4. 1. Request Methods

CONNECT	The request to set up a session across the link
DISCONNECT	Announcing the permanent closure of the link This method is not required in TCP/IP
SUBSCRIBE	Request from the connected peer that route advertisements should be announced on the link
UNSUBSCRIBE	Request that no further advertisements should be announced on the link
ROUTE	A route advertisement containing one or more one or more dial prefixes along with ancillary info
KEEPALIVE	A connection-polling method used to maintain connection state over connectionless protocols

4. 2. Reply Methods

OK	Confirmation that the previous request succeeded
ERROR	Advice the a request could not be understood or processed

4. 3. Other Methods

ACK	Acknowledgement of receipt of a packet

5. Session Methods Definition

5. 1. Request Methods

5. 1. 1. CONNECT Method (0x01)

The CONNECT method is a request to connect with a remote peer and used to set up authentication if required. The user MUST supply a HOSTID Information Element (0x01) and, if requested, MAY supply an AUTH variable-length Information Element (0x02) in response to a security challenge (see the ERROR reply method below).

The CONNECT method is either replied to with an OK (in the case of an authorised connection) or an ERROR (in the case of missing or invalid credentials).

5. 1. 2. DISCONNECT Method (0x02)

The DISCONNECT method is used to notify the connected peer that the connection is about to be closed. No further communication can take place on the link until a CONNECT message is issued. No IEs are defined for this method.

5. 1. 3. SUBSCRIBE Method (0x03)

The SUBSCRIBE method is used to request that a remote peer provide routing information and updates. Optionally, the SUBSCRIBE method may contain the SINCE Information Element (0x01), a date past which any new updates should be sent. This is a 64-bit UNIX integer specifying the number of seconds since Jan 1st, 1970.

5. 1. 4. UNSUBSCRIBE Method (0x04)

The UNSUBSCRIBE method is used to instruct the remote peer that it should not send any future route advertisements on the link. There are no IEs associated with this method. This method has no IEs.

5. 1. 5. KEEPALIVE Method (0x05)

The KEEPALIVE method is used to both test the link status and to, as the method suggests, keep the connection alive, which in UDP NAT environments is be especially important. A KEEPALIVE packet MUST be replied to with an ACK packet (0xf0), or else the peer may assume that the connection is no longer active. This method has no defined IEs.

5. 1. 6. ROUTE Method (0x06)

The ROUTE method is used to alert a peer to an available telephony route and to provide details on this route. ROUTE objects must provide a route ID (0x01), at least one Prefix (0x02) and can provide zero or more other information elements.

Other possible IEs, and their length (in bits) are:

IE	Name	Type	Description
0x01	ID	MED	An integer route ID identifying the routeset, used to update existing info
0x02	Prefix	BIG	An E.164 numbering format number prefix as a large unsigned integer (64 bits)
0x03	Updated	TIM	The date on which this route was most recently modified (64-bit UNIX time)
0x04	Name	VAR	A textual description of the route
0x05	Range Length	NIB	For fixed-length number, the range size of the numbers addressed by the prefix
0x10	Start DOW	DOW	The start day of week for the routeset
0x11	End DOW	DOW	The end day of week for the routeset
0x12	Start TOD	TOD	The start time of day for the routeset
0x13	End TOD	TOD	The start time of day for the routeset
0x20	Currency	CUR	The currency (three-character ISO code) for quoted call costs
0x21	Minute Cost	DEC	The per-minute cost for calls to the routeset
0x22	Connect Cost	DEC	The connection cost for calls to the routeset
0x23	Minimum Cost	DEC	The minimum cost for calls to the routeset
0x24	Time Rounding	BIT	Time rounding applied to the route, where 0 indicates no rounding, and 1 indicates rounding up to the next minute
0x25	Cost Rounding	RND	The currency rounding in use for the routeset. The value represents???
0x30	Hop Count	WRD	The number of hops to reach the destination network

5. 2. Reply Methods

5. 2. 1. OK Method (0x10)

The OK Method is used to instruct the connected peer that their request was completed successfully. An OK packet MUST include the SEQUENCE (0x01) Information Element in order to tell the connected peer which request has succeeded.

5. 2. 2. ERROR Method (0x11)

The OK Method is used to instruct the connected peer that their request was completed successfully. An OK packet MUST include the SEQUENCE Information Element (0x01) in order to tell the connected peer which request has succeeded. An ERROR packet MUST contain a REASON information element (0x02) to instruct the connected peer the nature of the error. Some codes for the REASON IE are below:

```
0x01 Authentication Required (MUST be accompanied by CHALLENGE IE)
0x02 Authentication Rejected
0x03 Invalid Request
0x04 Internal Error - Permanent
0x05 Internal Error - Temporary
```

The ERROR method may also contain a CHALLENGE (0x03) IE which is a 4-octet salt used for prompting the requester to provide an authentication string.

5. 3. Other Methods

5. 3. 2. ACK Method (0xf0)

The ACK method is used to acknowledge a packet from connected peer so that the connected peer does not attempt to retransmit the same packet again. An ACK packet MUST include the SEQUENCE Information Element (0x01) in order to tell the connected peer which packet that the ACK relates to.

6. Information Element Types

6. 1. Integer Types (BIT, NIB, OCT, WRD, MED, and BIG)

There are 6 Integer IE types representing unsigned integers of various sizes, these are: BIT (1 bit), NIB (4 bits), OCT (1 byte), WRD (2 bytes), MED (4 bytes) and BIG (8 bytes).

6. 2. Decimal Type (DEC)

The decimal type is a signed IEEE 754 double-precision binary floating-point number representing a currency value.

6. 3. Variable-length Type (VAR)

The VAR IE type is a variable-length octet string, for this type the byte that follows the IE type will give the length of the following IE data in octets, allowing a maximum IE data length of 255 characters.

6. 4. Currency Type (CUR)

The CUR IE type is a 24 bit value composed of three ASCII octets representing the capital-letter variant of ISO 4217 currency codes, for instance, "GBP" would be represented by octets 0x47 0x42 0x50.

6. 5. Time Type (TIM)

The Timestamp (TIM) IE type is a 64-bit UNIX timestamp, which is an unsigned integer providing the number of seconds since Jan 1st, 1970.

6. 6. Time-of-day Type (TOD)

Time of Day (TOD) IE type provides a 24-hour timestamp, stored as decimal numbers, using 12-bit storage. Therefore 18:00 (6pm) would be represented by 1800 decimal, binary 0111 0000 1000 or 0x708 hexadecimal.

6. 7. Day-of-week Type (DOW)

The Day Of Week (DOY) IE type is a 4-bit storage element (1 nibble) and defines a weekday from 0 (Monday) through to 6 (Sunday). There is no null value for the day of week as the Information Element can simply be omitted if no Day of Week is to be implemented.

6. 7. Rounding Type (RND)

The RND type indicates currency rounding. It is a signed 8-bit integer representing the decimal point at which rounding occurs, where negative values indicate rounding to the left of the decimal point and positive values represent rounding to the right. Therefore a value of 10000001 would mean calls are rounded to the nearest currency unit, and a value of 00000101 would relate to a rounding to the nearest 0.00001 of a currency unit.

7. Session Method Summary

The following table summarises available methods, their IEs and the IE types associated, categorised by type (REQ, REP, and OTH being Request, Reply and Other Method Types, respectively):

Type	Method	Code	IE Name	IE	Size	
REQ	CONNECT	0x01	HOSTID	0x01	VAR	*
			AUTH	0x02	VAR	
REQ	DISCONNECT	0x02				
REQ	SUBSCRIBE	0x03	SINCE	0x01	TIM	
REQ	UNSUBSCRIBE	0x04				
REQ	KEEPALIVE	0x05				
REQ	ROUTE	0x06	ID	0x01	VAR	*
			PREFIX	0x02	BIG	* #
			UPDATED	0x03	TIM	
			NAME	0x04	VAR	
			LENGTH	0x05	NIB	
			STARTDOW	0x10	DOW	
			ENDDOW	0x11	DOW	
			STARTTOD	0x10	TOD	
			ENDTOD	0x11	TOD	
			CURRENCY	0x20	CUR	
			RATE	0x21	DEC	
			CONNECT	0x22	DEC	
			MINIMUM	0x23	DEC	
			MINBILL	0x24	BIT	
			ROUND	0x25	RND	
			HOPS	0x26	WRD	
REP	OK	0x10	SEQ	0x01	MED	*
REP	ERROR	0x11	SEQ	0x01	MED	*
			REASON	0x02	WRD	
			CHALLENGE	0x03	WRD	
OTH	ACK	0xf0	SEQID	0x01	MED	*

* = Mandatory # = Element can occur multiple times

8. Session Lifecycle

8. 1. Session Establishment and Progress

TREX sessions are connected between from one system to another typically over a UDP/IP connection on port 8739. Each Request Method MUST contain a packet sequence number which is incremented for each new packet sent on the link. Each Reply Method packet MUST include the packet sequence number to which it is replying. A typical flow of a TREX connection is shown below:

```
SEQ DIR Message    IE Name   IE Value       Notes
-----------------------------------------------------------------
001 --> CONNECT    HOSTID    "p1"           Host p1 connects
001 <-- ERROR      REASON    0x01           Auth challenge
                   CHALLENGE ab83fe13       Challenge salt sent
                   SEQID     0x01           Replying to Seq 01
002 --> CONNECT    HOSTID    p1             Host p1 connects
                   AUTH      abe9736f9(...) Auth provided
002 <-- OK         SEQID     0x02           Connection accepted
003 --> SUBSCRIBE  SINCE     2015-01-01     Subscribe to routes
003 <-- OK         SEQID     0x03           Subscribe accepted
004 <-- ROUTE      PREFIX    155501         Route announcement
                   MINRATE   0.00           Route cost
                   CURRENCY  USD            Route currency
    --> ACK        SEQID     0x04           Acknowledge route
005 <-- ROUTE      PREFIX    155502         (etc)
                   MINRATE   0.02
                   CURRENCY  USD
    --> ACK        SEQID     0x05
```

The above sequence shows that the initial CONNECT method was responded to with an ERROR method containing a CHALLENGE Information Element. The client must now produce an authentication string by producing an MD5 hash of the composite based on a username and password plus the salt provided in the challenge, separated by Colons. This can be represented as MD5(username:password:salt).

Once the session is established a SUBSCRIBE Request Method is sent, requesting route updates since 2015-01-01. At this point a number of ROUTE messages are received from the far end, each of which is responded to with an ACK.

8. 2. Session Timing and Reliability

The TREX protocol is designed to be supported on connectionless protocols such as UDP, and as such timing mechanisms must be employed to check the link status and successful receipt of data. Any TREX peer sending a METHOD should expect a reply within 100 milliseconds.

If no reply is received the sending peer may wish to send the method to the receiving peer again. It is recommended that if no ACK, OK, or ERROR packet is received to the method on 4 retransmissions (5 transmissions in total), each with twice the inter-transmission delay, then the link should be considered dead. Such a connection closure may play out as follows:

```
Time(s) Seq  Direction  Message     Notes
-------------------------------------------------------
6.345   183     -->     SUBSCRIBE   (First attempt)
6.445   183     -->     SUBSCRIBE   (Second attempt)
6.645   183     -->     SUBSCRIBE   (Third attempt)
7.045   183     -->     SUBSCRIBE   (Fourth attempt)
7.845   183     -->     SUBSCRIBE   (Final attempt)
9.445   184     -->     DISCONNECT
```

If a peer handling a request method believes that servicing the request is likely to take more than 100 milliseconds (thus triggering a retransmission) it is advised that the peer sends an ACK in place of an OK or ERROR while the request is processed. This can be considered an analogue of the "100 Trying" provisional response of the SIP2.0 protocol [RFC2543].

9. Implementation Considerations, Challenges and Risks

9. 1. Rate Revisions

From time to time it may be necessary for an operator to amend the cost or other details of a given route. If known at the outset, these changes can effectively be scheduled by setting appropriate start and end dates for the route advertisement, but if a route with an absent end time (a permanent route) has been changed it may be necessary to send another ROUTE update with details of the new route.

9. 2. Prefix Aggregation

As alluded to in 5.2.6 of [RFC3219] prefix aggregation techiques are the main determinate of the final routeset size, and optimisations should be made wherever possible. For instance, if the prefixes +155501 +155502 are known to exist and are routeable, but it is known by the advertising peer that other prefixes under +15550 do not exist, the prefix +15550 can be considered an optimisation of the two available routes. On the other hand, advertising routes that are both not owned, and not routable by the advertising peer can cause obvious and serious problems if an alternative network would have been able to deliver the traffic. In order to prevent building incorrect routes, the implementing network should always have a secondary route in place in order to deliver the call to other networks. Alternatively, a pruning system may be used to filter out any advertised routes that are believed to be invalid, which may have other benefits should there be variations in quality as well as comparative cost of available routes.

9. 3. Intentional Fraud

With ever-increasing levels of telecoms fraud, security is a key consideration when implementing TREX. As with all interconnections with partner networks you should only receive routes from networks you trust, and wherever possible you should put checks and balances in place to help ensure that advertised routes are correct. If your implementation of the TREX network utilises advertised rate costs to perform least-cost routing (LCR) and a malign peer advertises rates lower than other networks, they may have the ability to shut down routing or hijack traffic for other purposes.

9. 4. Rate Awareness

In today's telecoms networks the vast majority of route additions (sometimes called databuilding or datafilling) is undertaken as a manual process with rates being agreed contractually and updated from time to time by direct communication between partner networks. As the TREX protocol allows for advertising rates, these may or may not match contractual rates agreed elsewhere with the partner network. Strict adherence to the protocol by technical staff may create a false confidence in the cost of calling certain prefixes, and accordingly lead to routing decisions that use higher rates than those offered elsewhere. To mitigate this risk, the TREX protocol does not make rate announcements mandatory, such that carriers who can not provide up-to-date rates are not forced to advertise temporary or incorrect rates, nor does the protocol or it's authors take responsibility for how these rates are used by the user's switches.

9. 5. Contradictory or Invalid Routeset Updates

As mentioned in this section, it is possible to update previously advertised routesets by sending a new ROUTE message with the same ROUTEID as used previously. However, as the prefixes for a given routeset must be specified in every ROUTE message, it is possible that those prefixes have changed or appear in multiple ROUTE advertisements. It is the responsibility of the peer accepting route updates to store and optimise route data accordingly, which is beyond the scope of this specification.

9. 6. Link Flooding

Due to the vast number of telephone carriers and prefixes operating today, a full set of detailed prefixes which are labelled and marked up with a number of information elements or time-of-day pricing may constitute a significant amount of data, in the order of several megabytes. Where carriers are interconnecting in a LAN environment this should not pose a problem, but synchronising route sets over congested or low-speed links could cause problems during a full refresh. In order to provide a best-practise approach, the TREX protocol a "SINCE" information element as part of the SUBSCRIBE method, which requests the peer only send routes which have changed since this date. It can not be assumed however that the remote peer has implement these checks, and may still send the entire routeset.

11. Future Updates and Improvements

Future releases of the TREX protocol are likely to add further Information Elements, for instance, the ROUTE method provides no way to advertise different gateways or technologies, for instance IP addresses and hostnames for VoIP technologies, or Switch Point Codes for SS7/TDM technology. The ability for a route advertiser to also specify capabilities of the route, including supported Caller ID formats (such as Type of Network, Type of Number etc), link latency and bandwidth, etc.

11. Acknowledgement

Thanks is given to Netfuse Telecom for their resources in developing the protocol and promoting it's use within the field.

12. Normative References

[RFC4271] Y. Rekhter, T. Li, S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC2543] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP: Session Initiation Protocol" RFC 2543, March 1999.

[RFC3219] J. Rosenberg, H. Salama, M. Squire, "Telephony Routing over IP (TRIP)", January 2002.

Authors' Addresses

Leo Brown (Author)
Netfuse Telecom
1 Sydney Street
Brighton
United Kingdom

Email: leo@netfuse.org

Peter Eyres (Editor)
Netfuse Telecom
1 Sydney Street
Brighton
United Kingdom

Email: peter@netfuse.org