IPTEL Working Group   Manjunath Bangalore, Cisco Systems
Internet Draft   Rajneesh Kumar, Cisco Systems
draft-ietf-iptel-trip-gw-00.txt   Jonathan Rosenberg, dynamicsoft
June 2002   Hussein Salama, Cisco Systems
Expiration Date: December 2002   Dhaval N. Shah, Cisco Systems

Usage of TRIP in Gateways for Exporting Phone Routes

Status of this Memo

Abstract

This document describes a new application of the Telephony Routing over IP (TRIP) protocol. TRIP was engineered as a tool for inter-domain exchange of telephone routing information. However, it can also be used as a means for gateways and soft switches to export their routing information to a Location Server (LS), which may be co-resident with a proxy or gatekeeper. This LS can then manage those gateway resources. We discuss the motivations for this application, and then show how a subset of TRIP is needed for this application.

1. Terminology and Definitions

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 RFC 2119 [1].

Some other useful definitions are:

Trunk: In a network, a communication path connecting two switching systems used in the establishment of an end-to-end connection. In selected applications, it may have both its terminations in the same switching system.

TrunkGroup: A set of trunks, traffic engineered as a unit, for the establishment of connections within or between switching systems in which all of the paths are interchangeable except where subgrouped.

Carrier: An organization that provides connections for telephony services between end-users and/or exchanges.

2. Introduction

In typical VoIP networks, Internet Telephony Administrative Domains (ITADs) will deploy numerous gateways for the purposes of geographical diversity, capacity, and redundancy. When a call arrives at the domain, it must be routed to one of those gateways. Frequently, an ITAD will break their network into geographic POPs, with each POP containing some number of gateways, and a proxy server element that fronts those gateways. The proxy server is responsible for managing the access to the POP, and also for determining which of the gateways will receive any given call that arrives at the POP.

This configuration is depicted graphically in Figure 1.

```
                                                   +----------+
                                                   |          |
                                                   |   GW     |
                                                -> +----------+
                                              //
                                            //
                                          //       +----------+
                                        //         |          |
                                      //           |   GW     |
                                    //             +----------+
                                  //
                 +----------+   //                                TO PSTN
                 |          |  /                   +----------+
                 | Routing  |  --------------->    |          |  ----->
        -------->| Proxy    |                      |   GW     |
                 |          |  --                  +----------+
                 |          |    --
                 +----------+      --
                                     ---           +----------+
                                        --         |          |
                                          --       |   GW     |
                                            --     +----------+
                                              -->

                                                   +----------+
                                                   |          |
                                                   |   GW     |
                                                   +----------+
```

Figure 1: Gateway and LS Configuration

The decision about which gateway to use depends on many factors, including their availability, remaining call capacity and call success statistics particular POTS destination. For the proxy to do this adequately, it needs to have access to this information in

real-time, as it changes. This means there must be some kind of communications between the proxy and the gateways to convey this information.

In this document, we specify a usage of TRIP to communicate this information from the gateways to the location servers associated with the proxies that make call routing decisions. Section 3 looks at TRIP [4,5]. We then describe the details of a TRIP solution in Section 4. It is assumed that the reader is familiar with TRIP.

3. TRIP

TRIP was engineered as a tool for interdomain route exchange. At first glance, it may not seem appropriate for application in a gateway. However, TRIP provides exactly the features needed to solve the problem at hand. TRIP allows one entity (in this case, a gateway) to convey to another (in this case, a proxy) a set of telephony routes which terminate through it. These routes are represented by telephony addressing information along with attributes that can express resource availability and preferences. In TRIP, the routing tables are exchanged once. Only when they change are updates sent. This is exactly the capability needed for a gateway to send its routing information to a proxy, and hence allows TRIP to be leveraged for this purpose. Since routing information is sent when it changes, using TRIP does not require communications on the critical path of call setup signaling.

TRIP also supports a keepalive between peers. This keepalive is a short message, sent fairly frequently. It does not contain routing information. The period of the keepalive is negotiated once at startup time. If one of the entities crashes, the other flushes all routes received from it.

TRIP can contain attributes describing a route. New attributes can easily be added. The available capacity of a route is needed by the proxies to properly load balance and route to a set of gateways. This capacity can be expressed as an attribute.

TRIP can be run over IPSec or TLS between two peers, providing

authentication, integrity and privacy.

Another advantage of using TRIP here is that it makes the redistribution of local gateway reachability information into inter-domain TRIP relatively simple (refer to Figure 6), because it's the same protocol.

While it is true that TRIP is complex, almost all of this complexity is related to the processing of routes *received* from other peers. An element, such as a gateway, which only *sends* routes to a peer (the proxy), does not need to implement any of those functions. In particular, any processing related to aggregation, TRIB updates, route propagation and advertisement, handling of transitivity and unknown routes, or the decision process need not be implemented. The resulting set of functions are very small. They are composed of only:

- The initial OPEN phase, exchange of keepalive timers, and the process of bringing up the state machine.
- Sending of one or more UPDATE messages containing the routes and parameters of the gateways.
- Sending of a periodic keepalive.

4. Defining TRIP-GW

We call the subset of TRIP needed for this application "TRIP-GW", "TRIP-Lite", or TRIP for gateways. We note that this is a valid subset defined by the specification, so that a TRIP-GW speaker is a conformant TRIP speaker. In this section, we include the various attributes, some of them new and also introduce two new address families. The gateway and proxy behaviors are discussed in more details in sections 4.9 and 4.10 respectively.

4.1. AvailableCircuits Attribute

Mandatory: False.
Required Flags: optional, non-transitive

Potential Flags: None.
TRIP Type Code: To be assigned by IANA.

The AvailableCircuits attribute is used ONLY between a gateway and its peer LS responsible for managing that gateway. It is for this reason that this attribute is non-transitive. If it is received in a route, it MUST NOT be propagated unless the LS is sure that it is relatively static.

The AvailableCircuits identifies the number of PSTN circuits that are currently available on a route to complete calls. The number of additional calls sent to that gateway for that route cannot exceed the circuit capacity. If it does, the signaling protocol will likely generate errors, and calls will be rejected.

AvailableCircuits is measured in integral number of calls. It is very dynamic.

4.1.1. AvailableCircuits Syntax

The AvailableCircuits attribute is a 4-octet unsigned numeric value. It represents the number of circuits remaining for terminating calls to this route.

4.1.2. Route Origination and AvailableCircuits

Routes MAY be originated containing the AvailableCircuits attribute. Since this attribute is highly dynamic, changing with every call, updates MAY be sent as it changes. However, it is RECOMMENDED that a gateway originating routes with this attribute use thresholds, and that routes are re-originated only when the attribute moves above or below a threshold. It is also RECOMMENDED that the thresholds in each direction (going above a threshold and going below a threshold) be different, thus achieving a form of hysterisis. Both of these measures help reduce the messaging load from route origination.

4.1.3. Route Selection and AvailableCircuits

The AvailableCircuits attribute MAY be used for route selection. Since one of its primary applications is load balancing, an LS will wish to choose a potentially different route (amongst a set of routes for the same prefix) on a call by call basis. This can be modeled as re-running the decision process on the arrival of each call. The aggregation and dissemination rules for routes with this attribute ensure that re-running this selection process never results in propagation of a new route to other peers.

4.1.4. Aggregation and AvailableCircuits

Not applicable

4.1.5. Route Dissemination and AvailableCircuits

Routes MUST NOT be disseminated with the AvailableCircuits attribute. The attribute is meant to reflect capacity at the originating gateway only. Its highly dynamic nature makes it inappropriate to disseminate in most cases.

4.2. TotalCircuitCapacity Attribute

Mandatory: False.
Required Flags: optional, transitive
Potential Flags: None.
TRIP Type Code: To be assigned by IANA.

The TotalCircuitCapacity attribute is used to reflect the administratively provisioned capacity as opposed to the availability at a given point in time as provided by the AvailableCircuits

attribute. Because of its relatively static nature, this attribute MAY be propogated beyond the LS that receives it, hence making this attribute transitive.

The TotalCircuitCapacity identifies the total number of PSTN circuits that are available on a route to complete calls. It is used in conjunction with the AvailableCircuits attribute in gateway selection by the LS. The total number of calls sent to the specified route on

the gateway cannot exceed this total circuit capacity under steady state conditions.

TotalCircuitCapacity is measured in integral number of calls. This is not expected to change frequently. This can change, for instance, when certain telephony trunks on the gateway are taken out of service for maintenance purposes.

4.2.1. TotalCircuitCapacity Syntax

The TotalCircuitCapacity attribute is a 4-octet unsigned numeric value. It represents the total number of circuits available for terminating calls through this advertised route. This attribute represents a potentially achievable upper bound on the number of calls which can be terminated on this route in total.

4.2.2. Route Origination and TotalCircuitCapacity

Routes MAY be originated containing the TotalCircuitCapacity attribute.

4.2.3. Route Selection and TotalCircuitCapacity

The TotalCircuitCapacity attribute MAY be used for route selection. Since one of its primary applications is load balancing, an LS will wish to choose a potentially different route (amongst a set of routes for the same destination), on a call by call basis. This can be modeled as re-running the decision process on the arrival of each call. The aggregation and dissemination rules for routes with this attribute ensure that re-running this selection process never results in propagation of a new route to other peers.

4.2.4. Aggregation and TotalCircuitCapacity

An LS MAY aggregate routes to the same prefix which contain a TotalCircuitCapacity attribute. It SHOULD add the values of the individual routes to determine the value for the aggregated route in the same ITAD.

4.2.5. Route Dissemination and TotalCircuitCapacity

Since this attribute reflects the static capacity of the gateway's circuit resources, it is not expected to change frequently. Hence the LS receiving this attribute MAY disseminate it to other peers, both internal and external to the ITAD.

4.3. CallSuccess Attribute

Mandatory: False.
Required Flags: optional, non-transitive
Potential Flags: None.
TRIP Type Code: To be assigned by IANA.

The CallSuccess attribute is a non-transitive attribute used ONLY between a gateway and its peer LS responsible for managing that gateway. If it is received in a route, it MUST NOT be propagated.

The CallSuccess attribute provides information about the number of normally terminated calls out of a total number of attempted calls. CallSuccess is to be determined by the gateway based on the Disconnect cause code at call termination. For example, a call that reaches the Alerting stage but does not get connected because of called party being unavailable is conventionally considered a successful call. Similar is the case when the called party is busy. On the other hand, a call that gets disconnected because of a Circuit or Resource being unavailable is conventionally considered a failed call. The exact mapping of disconnect causes to CallSuccess is at the discretion of the gateway reporting the attribute.

The CallSuccess attribute is used by the LS to keep track of failures in reaching certain telephony destinations through a gateway(s) and use that information in the gateway selection process to enhance the probability of successful call termination.

This information can be used by the LS to consider alternative gateways to terminate calls to those destinations with a better likelihood of success.

4.3.1. CallSuccess Syntax

The CallSuccess attribute is comprised of two component fields - each represented as an unsigned 4-octet numeric value. The first component field represents the total number of calls terminated normally for the advertised destination on a given address family. The second component field represents the total number of attempted calls for the advertised destination.

4.3.2. Route Origination and CallSuccess

Routes MAY be originated containing the CallSuccess attribute. This attribute is expected to get statistically significant with passage of time as more calls are attempted. Therefore, it is RECOMMENDED that the gateway make a choice based on local thresholds to determine when to report this attribute in an UPDATE.

4.3.3. Route Selection and CallSuccess

The CallSuccess attribute MAY be used for route selection. This attribute represents a measure of success of terminating calls to the advertised destination(s). This information MAY be used to select from amongst a set of alternative routes to increase the probability of successful call termination.

4.3.4. Aggregation and CallSuccess

Not applicable

4.3.5. Route Dissemination and CallSuccess

Routes MUST NOT be disseminated with the CallSuccess attribute. Its potential to change dynamically does not make it suitable to disseminate in most cases.

4.4. Prefix Attributes

Mandatory: Conditional Mandatory (if ReachableRoutes is present).
Required Flags: Well-known.
Potential Flags: None.
TRIP Type Codes: To be assigned by IANA.

The Prefix attribute is used to represent the list of prefixes that

the respective route can complete calls to. This attribute is intended to be used with the Carrier or Trunkgroups address family (discussed in a later section).

Though it is possible that all prefix ranges MAY be reachable through a given Carrier, administrative issues could make certain ranges preferable to others.

4.4.1. Prefix Attribute Syntax

The Prefix attribute could be E.164, Decimal or PentaDecimal (refer to TRIP RFC [4]), each of them having it's own type code. The Prefix attribute is encoded as a sequence of prefix values in the attribute value field. The prefixes are listed sequentially with no padding as shown in Figure 2. Each prefix includes a 2-octet length field that represents the length of the address field in octets. The order of prefixes in the attribute is not significant.

```
                          1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 . . . 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
+-------------------------------+-----------+-------------------------------+--
|             Length            | Prefix1...|             Length            | P
+-------------------------------+-----------+-------------------------------+--
```

Figure 2: Prefix Format

4.4.2. Route Origination and Prefix

Routes MAY be originated containing the Prefix attribute.

4.4.3. Route Selection and Prefix

The Prefix attribute MAY be used for route selection.

4.4.4. Aggregation and Prefix

Routes with differing Prefix attribute MUST NOT be aggregated. Routes with the same value in the Prefix attribute MAY be aggregated and the same Prefix attribute attached to the aggregated object.

4.4.5. Route Dissemination and Prefix

The LS receiving this attribute should disseminate it to other peers, both internal and external to the ITAD.

4.5. TrunkGroups Attribute

Mandatory: False.
Required Flags: optional, transitive
Potential Flags: None.
TRIP Type Code: To be assigned by IANA.

The TrunkGroups attribute is used to represent trunkgroups on the gateway that the gateway can complete the calls to. It allows providers to route calls to destinations through preferred trunks. This attribute is relatively static.

4.5.1. TrunkGroups Syntax

The TrunkGroups attribute is a variable length attribute that is composed of a sequence of trunkgroup length-value fields. It indicates that the gateway can complete the call through any trunkgroup in the sequence.

Each trunkgroup is a length-value field (shown in Figure 3 below). The length field is a 1-octet unsigned numeric value. The value field is a variable length alphanumeric field and is also called the trunkgroup label field. The length field represents the size of the trunkgroup label in number of octets. The maximum length is 128 octets.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 ...  7 8 9 0 1 2 3 4 5 ...
+---------------+--------------------+---------------+--------------------
|    Length     | TrunkGroup Label...|   Length      | TrunkGroup Label..
+---------------+--------------------+---------------+--------------------
```

Figure 3: TrunkGroups Syntax

4.5.2. Route Origination and TrunkGroups

Routes MAY be originated containing the TrunkGroups attribute.

4.5.3. Route Selection and TrunkGroups

The TrunkGroups attribute MAY be used for route selection. Certain trunkgroups MAY be preferred over others based on provider policy.

4.5.4. Aggregation and TrunkGroups

Routes with differing TrunkGroups attribute MUST NOT be aggregated. Routes with the same value in the TrunkGroups attribute MAY be aggregated and the same TrunkGroups attribute attached to the aggregated object.

4.5.5. Route Dissemination and TrunkGroups

This attribute is not expected to change frequently. Hence, the LS receiving this attribute MAY disseminate it to other peers, internal to ITAD. Routes MUST not be disseminated external to the ITAD, with TrunkGroups attribute.

4.6. Carrier Attribute

Mandatory: False.

Required Flags: optional, transitive
Potential Flags: None.
TRIP Type Code: To be assigned by IANA.

The Carrier attribute is used to represent the list of Carrier Identification Codes(CIC's) that the gateway can complete the calls to. It enables providers to route calls to destinations through preferred carriers. A CIC represents a unique code assigned to the carrier or telephony service provider offering the service The CIC definition is as defined in the ITU specification referenced below[8,9].

This attribute is relatively static.

4.6.1. Carrier Syntax

The Carrier attribute is a variable length attribute that is composed of a sequence of carrier identification codes. It indicates that the route can complete the call to any of the carriers represented by the carrier identification codes in the sequence.

Each carrier identification code in the sequence is a fixed length

2-octet unsigned numeric value as shown below.

```
 0                   1                   2                   3
 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 2 3
+-------------------------------+-------------------------------+-----
|        CarrierIdCode1         |        CarrierIdCode2         | ...
+-------------------------------+-------------------------------+-----
```

Figure 4: Carrier Syntax

4.6.2. Route Origination and Carrier

Routes MAY be originated containing the Carrier attribute.

4.6.3. Route Selection and Carrier

The Carrier attribute MAY be used for route selection. Certain carriers MAY be preferred over others based on provider policy.

4.6.4. Aggregation and Carrier

Routes with differing Carrier attribute MUST NOT be aggregated. Routes with the same value in the Carrier attribute MAY be aggregated and the same Carrier attribute attached to the aggregated object.

4.6.5. Route Dissemination and Carrier

This attribute is not expected to change frequently. Hence, the LS receiving this attribute MAY disseminate it to other peers, both internal and external to the ITAD.

4.7. TrunkGroup and Carrier Address Families

When a set of GWs are to managed at the granularity of carriers or trunkgroups then it may be more appropriate for a GW can advertise routes using the Carrier address family or trunkgroup address family respectively. Also, in a TRIP-GW association between the gateway and the LS responsible for managing that gateway, there are some attributes that more naturally fit in as advertised properties of trunkgroups or carriers rather than that of advertised prefixes, For example: the AvailableCircuit information on a particular trunkgroup or a particular carrier. To express this relationship, the existing TRIP address families are inadequate. We need separate address

families that can associate certain properties like AvailableCircuits information to trunkgroups or carriers.

The primary benefits of this model are as follows:

- It allows a service provider to route calls based strictly on the trunkGroups or carriers.
- it facilitates more accurate reporting of attributes of a dynamic nature like AvailableCircuits by providing the ability to manage resources at the granularity of a trunkgroup or a carrier.
- Gateways can get really large with the ability to provision

hundreds or a few thousand circuits and this can increase the potential for traffic that reports dynamic resource information between the gateway and the LS. The model introduced can potentially alleviate this UPDATE traffic hence increasing efficiency and providing a scalable gateway registration model.

4.7.1. Address Family Syntax

Consider the generic TRIP route format from TRIP[4] shown below.

```
 0                   1                   2                   3
 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
+---------------+---------------+--------------+----------------+
|       Address Family          |     Application Protocol      |
+---------------+---------------+--------------+----------------+----
|             Length            |      Address (variable)      ....
+---------------+---------------+--------------+----------------+----
```

Figure 5: Generic TRIP Route Format

The Address Family field will be used to represent the type of the address associated for this family, which is based on the TrunkGroup or Carrier. The codes for the new address families will be allocated by IANA.

The Application Protocol field is same as the one for the Decimal, PentaDecimal and E.164 address families defined in TRIP[4]. The Length field represents the total size of the Address field in bytes.

The value in the Address field represents either the TrunkGroup or the Carrier address families and the syntax is as follows:

For the TrunkGroup Address Family, the Address field is a variable length alphanumeric field (trunkgroup label), where length is determined by the length field of the route. The maximum value of the length field for this Address Family is 128 bytes.

For the Carrier Address Family, the length field represents the length of the Address field in bytes. The Address field is a fixed length field that consists of the CarrierIdCode (CIC). Again, the CIC is as defined in ITU specification M.400[8]. CIC is a fixed

length 2-octet unsigned numeric value as shown in Figure 4 above.

If a gateway supports any of these address families, it should include that address family as one of the Route types supported in the OPEN message capability negotiation phase.

The following attributes are currently defined to be used with all the address families including the TrunkGroup and Carrier address families.

- AvailableCircuits Attribute
- TotalCircuitCapacity Attribute
- CallSuccess Attribute

  It is recommended that the above three attributes be used by the gateway with the TrunkGroup or Carrier address families, if possible. This will potentially offer a more efficient resource reporting framework, and a scalable model for gateway provisioning.

  However, when the gateway is not using TrunkGroup or Carrier address family, it MAY use the above attributes with the Decimal, PentaDecimal and E.164 address families.

  The Prefix attributes MUST NOT be used with the Prefix address families.

  The Carrier attribute MUST NOT be used with the Carrier and TrunkGroup address families.

  The TrunkGroups attribute MUST NOT be used with the TrunkGroup and Carrier address families.

4.8. Other attribute considerations

4.8.1. Cost/Pricing attribute

In exploring attributes suitable for the GW-LS communications, Pricing is another attribute that was considered. Though at first glance, it seems like a useful piece of information to be advertised by the gateway to express the price offered by carriers to different destinations, in reality, the computation of pricing can get quite complex. For example, the price offered by a provider can vary by

time of day, it can be based on an agreement between two service providers interconnecting with each other, it could be based on one price for the first 'n' minutes and a different price after that, Least Cost routing choices and Grades of service offered by a carrier can affect pricing. There could be other factors as well. Expressing this complex interplay between different factors that determine pricing is non-trivial to represent. It will be a subject of further investigation to determine whether advertising pricing associated with carriers in its simple form without any dependencies adds value to be included as an attribute in the TRIP-GW communications.

4.9. Gateway Operation

A gateway uses TRIP to advertise its reachability to its domain's Location Server(s) (LS, which are closely coupled with proxies). The gateway operates in TRIP Send Only mode since it is only interested in advertising its reachability, but is not interested in learning about the reachability of other gateways and other domains. Also, the gateway will not create its own call routing database. Therefore, the gateway does not need a complete implementation of TRIP. A lightweight version of the protocol is sufficient. In this section we describe the operation of TRIP on a gateway.

4.9.1. Session Establishment

When opening a peering session with a TRIP LS, a TRIP-GW gateway follows exactly the same procedures as any other TRIP speaker. The TRIP-GW gateway sends an OPEN message which includes a Send Receive Capability in the Optional Parameters. The Send Receive Capability is set by the gateway to Send Only. The OPEN message also contains the address families supported by the gateway. When the TRIP LS receives the gateway's OPEN message, it sets its local policy such that no UPDATE messages are sent to the TRIP-GW gateway. The remainder of the peer session establishment is identical to TRIP.

4.9.2. UPDATE Messages

Once the peer session has been established, the gateway sends UPDATE messages to the TRIP LS with the gateway's entire reachability. The Gateway also sends any attributes associated with the routes.

If the gateway's reachability changes at any point in time, the gateway MUST generate UPDATE message(s) with the change. The frequency of successive UPDATE messages MUST follow the same rules

specified for TRIP[4]. The TRIP-GW gateway MUST support all

mandatory TRIP attributes.

If the gateway receives an UPDATE message from the TRIP LS, it MUST silently discard it as specified for TRIP[4]. No further action should be taken.

4.9.3. KEEPALIVE Messages

KEEPALIVE messages are periodically exchanged over the peering session between the TRIP-GW gateway and the TRIP LS as specified in Section 4.4 of TRIP RFC[4].

4.9.4. Error Handling and NOTIFICATION Messages

The same procedures used with TRIP, are used with TRIP-GW for error handling and generating NOTIFICATION messages. The only difference is that a TRIP-GW gateway will never generate a NOTIFICATION message in response to an UPDATE message, irrespective of the contents of the UPDATE message. Any UPDATE message is silently discarded.

4.9.5. TRIP-GW Finite State Machine

When the TRIP-GW finite state machine is in the Established state and an UPDATE message is received, the UPDATE message is silently discarded and the TRIP-GW gateway remains in the Established state. Other than that the TRIP finite state machine and the TRIP-GW finite state machine are identical.

4.9.6. Call Routing Databases

A TRIP-GW gateway may maintain simultaneous sessions with more than one TRIP LSs. A TRIP-GW gateway maintains one call routing database per peer TRIP LS. These databases are equivalent to TRIP's Adj-TRIBs-Out, and hence we will call them Adj-TRIB-GWs-Out. An Adj-TRIB-GW-Out contains the gateway's reachability information advertised to its peer TRIP LS. How an Adj-TRIB-GW-Out database gets

populated is outside the scope of this draft (possibly by manual configuration).

The TRIP-GW gateway does not have databases equivalent to TRIP's Adj-TRIBs-In and Loc-TRIB, because the TRIP-GW gateway does not learn routes from its peer TRIP LSs, and hence it does not run call route selection.

4.9.7. Route Selection and Aggregation

TRIP's route selection and aggregation operations MUST NOT be implemented by TRIP-GW gateways.

4.10. LS/Proxy Behavior

TRIP completely specifies the behavior of the LS as a TRIP speaker. However, the primary question is: is an LS connected to a gateway an internal or external peer of the gateway?

The most obvious choice, internal peer, is not the best choice. If an LS has ten peer GWs, all of them advertising reachability to 1408*, it will flood all ten routes to all other LSs in the same ITAD. This won't scale, because each LS in the ITAD will have to create a separate Adj-TRIB-In for each GW in that ITAD. In addition, it doesn't allow a SIP Proxy server or a H.323 GK to load balance among the GWs of its zone/subdomain.

A similar problem exists when an LS is an external peer to the gateways, and has direct peering relationships with one or more internal peers. However, an ingress LS to an ITAD does not perform aggregation. Only the egress LS aggregates routes.

Therefore, it is RECOMMENDED that the LS actually run two instances of TRIP; one with an external peering relationship to the gateways, and the other with an internal peering relationship with one or more LS's within the ITAD. The interface between these instances is largely a local matter; routes are exported from one and imported to the other. In the process of exporting routes from the GW instance, it may be useful to consolidate the routes before importing them to the other LS instance. Some details and motivation for this operation

are provided below. In addition, the routes may also be aggregated. This architecture is shown in Figure 6.


```
    .                          .
    .                          .
    .         +----------------.---------------------------+
    .         |                .+-+  +-+                   |
    .         |                .|A|  |C|                   |         +-----+
    .         |                .|g|  |o|                   |         |     |
    .         |+-------------+ .|g|  |n|  +-------------+  |      -- | GW  |
    .         ||             | .|r|  |s|  |             |  |    --   +-----+
    .         ||  TRIP       | .|e|  |o|  |  TRIP       |  | ---
    .         ||  Instance   <-.|g<--|l<---  Instance   |--+-        +-----+
    .         ||             | .|a|  |i|  |             |  |         |     |
    .         ||  (I-TRIP/   | .|t|  |d|  | (TRIP-Lite  |--+---------| GW  |
    .         ||   E-TRIP)   | .|i|  |a|  |  Receiver)  |  |         +-----+
    .         ||             | .|o|  |t|  |             |--+---
    .         |+-/-----------+ .|n|  |i|  +-------------+  |   ---   +-----+
    .         | /              .| |  |o|                   |      -- |     |
    .         |/               .| |  |n|                   |         | GW  |
    .         /                .+-+  +-+                   |         +-----+
    .        /|             LS .                           |
    .       / +----------------.---------------------------+
    .      /                   .
    .     /                    .
    .    /                     .
    .   /                      .
    .  /                       .
    .+/------------+           .
```

```
    .|             |           .
    .|             |           .
    .|             |           .
    .|     LS      |           .
    .|             |           .
    .|             |           .
    .|             |           .
    .|             |           .
    .+\-----------+            .
    .  \                       .
    .   \                      .
    .    \                     .
    .     \   +-----------------.---------------------------+
    .      \ |                 .+-+  +-+                   |
    .       \|                 .|A|  |C|                   |         +-----+
    .        \                 .|g|  |o|                   |         |     |
    .         \-------------+  .|g|  |n|  +-------------+  |      -- | GW  |
    .         ||            | .|r|  |s|  |             |  |    --   +-----+
    .         ||   TRIP     | .|e|  |o|  |  TRIP       |  | ---
    .         ||   Instance <-.|g<--|l<---  Instance   |--+--      +-----+
    .         ||            | .|a|  |i|  |             |  |         |     |
    .         ||   (I-TRIP/ | .|t|  |d|  | (TRIP-Lite  |--+---------| GW  |
    .         ||    E-TRIP) | .|i|  |a|  |  Receiver)  |  |         +-----+
```



```
    .         ||            | .|o|  |t|  |             |--+---
    .         |+-------------+ .|n|  |i|  +-------------+  |   ---   +-----+
    .         |               .| |  |o|                   |      -- |     |
    .         |               .| |  |n|                   |         | GW  |
    .         |               .+-+  +-+                   |         +-----+
    .         |             LS .                           |
    .         +----------------.---------------------------+
    .                          .
    .           ITAD           .
    ............................
```

Figure 6: LS Architecture for TRIP-GW

4.10.1. Route consolidation

A signaling server/Location Server(LS) fronting a set of gateways, receives routing information on several TRIP-Lite sessions. Subsequently, this routing information is presented as candidate

routes (possibly as local routes) to the TRIP Decision Process, for every destination for which a route is available. The LS may potentially receive two or more TRIP-Lite routes for the same destination. These alternative routes may be received from the same gateway, or from multiple gateways. It is recommended that the LS consolidate the set of TRIP-Lite routes for every destination, before presenting a candidate route, to the TRIP Decision Process. The purpose of this operation should be to represent the collective routing capabilities of the set of gateways, managed by this LS, and subsequently propagate this information into the core of the TRIP network. An example scenario is shown below. Consider an LS that maintains TRIP-Lite peering sessions with gateways A and B.

o Gateway A advertises a route for destination "SIP 408" on the E.164 address family with the Carrier attribute being C1

o Gateway B advertises a route for destination "SIP 408" on the E.164 address family with Carrier attribute C2

The LS that receives these TRIP-Lite routes can consolidate these routes into a single route for destination "SIP 408" with its Carrier attribute being a union of the the Carrier attribute values of the individual routes, namely, "C1 C2" This operation is refered to as "Consolidation." In the above example, it is possible that a route to the destination "SIP 408" through one or more carriers may have been lost if the individual routes were not consolidated. In general, there is a potential for loss of gateway routing information, when TRIP-Lite routes from a set of gateways are not consolidated, when a

candidate route is presented to the TRIP Decision process. The specifics of applying the consolidation operation to different attributes and routes from different address families, is left to the individual TRIP-Lite receiver implementations. In addition, the route selection procedures as documented in the TRIP RFC [4], do not apply to processing of TRIP-Lite routes received from the gateways.

4.10.2. Aggregation

The set of gateway routes, that are in a consolidated form or otherwise, MAY be aggregated before importing it to the LS instance that is responsible for I-TRIP/E-TRIP processing. This operation

follows the standard aggregation procedures described in the TRIP RFC[4], while adhering to the aggregation rules for each route attribute.

5. IANA Considerations

- The Attribute Type Codes for the new attributes: AvailableCircuits, TotalCircuitCapacity, CallSuccess, Prefix, TrunkGroups and Carrier described in Sections 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6 above, respectively, are to be assigned by IANA.
- The Address Family Codes for the new address families: TrunkGroup and Carrier described in Section 4.7, are to be assigned by IANA.

6. Changes since -03

- Removed Carrier-Trunkgroup attribute and address family and references to it.
- Added Terminology and Definitions section.
- Updated CallSuccess attribute.
- Added Prefix attribute.
- Added Carrier attribute.
- Added TrunkGroups attribute.
- Added TrunkGroup Address Family.
- Added Carrier Address Family.
- Added some more references.

7. Changes since -02

- Removed the requirements section.
- Discussed the motivation for introducing Carrier information into TRIP.
- Defined a new attribute for the E.164 address family.

- Defined a new address family for CarrierCode-TrunkGroup combination .
- Defined new attributes to advertise dynamic gateway characteristics like resource availability, and call success rate.
- Added as section to validate the TRIP-GW solution against the requirements in [7].

8. Changes since -01

- Added requirements.
- Added more formal analysis of REGISTER and added analysis of SLP.
- Removed circuit capacity attribute.

9. Changes since -00

- Added text to stress the value of this proposal for managing a gateway cluster.
- Added attributes for circuit capacity and DSP capacity.
- Added section on LS operation, discussing aggregation issue.

Authors' Addresses

Manjunath Bangalore
Cisco Systems
Mail Stop SJC-21/2/2
771 Alder Drive
Milpitas, CA 95035
email: manjax@cisco.com

Rajneesh Kumar
Cisco Systems
Mail Stop SJC-21/2/2
771 Alder Drive
Milpitas, CA 95035
email: rajneesh@cisco.com

Jonathan Rosenberg
dynamicsoft
72 Eagle Rock Avenue
First Floor
East Hanover, NJ 07936
email: jdrosen@dynamicsoft.com

Hussein F. Salama
Cisco Systems
Mail Stop SJC-24/3/2
170 W. Tasman Drive
San Jose, CA 95134
email: hsalama@cisco.com

Dhaval N. Shah
Cisco Systems
Mail Stop SJC-21/2/2
771 Alder Drive
Milpitas, CA 95035
email: dhaval@cisco.com

Acknowledgments

We wish to thank David Oran, Anirudh Sahoo, Kevin McDermott, Randy Baird, Cullen Jennings, Jon Peterson and Li Li for their insightful comments and suggestions.

References

[1] Bradner, S., "Keywords for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: session initiation protocol," Request for Comments 2543, Internet Engineering Task Force, Mar. 1999.

[3] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service location protocol, version 2," Request for Comments 2608, Internet Engineering Task Force, June 1999.

[4] J. Rosenberg, H. Salama, and M. Squire, "Telephony routing over IP (TRIP)," Request for Comments 3219, Internet Engineering Task

Force, January 2002.

[5] J. Rosenberg and H. Schulzrinne, "A framework for telephony routing over IP," Request for Comments 2871, Internet Engineering Task Force, June 2000.

[6] International Telecommunication Union, "Packet based multimedia communication systems," Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[7] J. Rosenberg, "Requirements for Gateway Registration," Internet Draft, Internet Engineering Task Force, Nov. 2001. Work in progress.

[8] ITU-T M.1400 Specification titled "Designations for interconnections among network operators," published October 10, 2001.

[9] ITU-T List of ITU Carrier Codes (published periodically in the ITU-T Operational Bulletin).

[10] J. Peterson, "An Architecture for Gateway Registration Based on Trunk Groups," Internet Draft, Internet Engineering Task Force, Feb. 2002. Work in progress.

Full Copyright Statement