SIP Peering Based on Distance Vector Algorithm

Yuanmin Chen, Xiaodong Li, Wei Mao
Computer Network Information Center
Chinese Academy of Sciences
Beijing,100190 P.R.China
{chenyuanmin,lixiaodong,maowei}@cnnic.cn

***Abstract*—SIP[1][2] based VoIP is a key enabling technology for the migration of circuit-switched PSTN architecture to packet-based networks. But the real world did not choose to implement and deploy SIP as initially envisioned by their inventors. The reason is ascribed to SIP spam and accounting problems. The root of this problem is SIP is an Email like protocol, no trust relationship between SIP communication partners. SIP peering is a subject of how to construct trust relationship among SIP Service Providers. And in peering scenario, routing is the most important issues. We introduce distance vector algorithm into SIP routing area, by this approach, SIP Service Providers can dynamically construct trust relationship with each other, can also lower costs, improve call quality and offer new IP-enabled services to their enterprise customers.**

Keywords: SIP; VoIP; Distance Vector Algorighm

I. INTRODUCTION

SIP peering[3][6] refers to relationships between SIP service providers in which they agree to exchange VoIP traffic to keep it on IP backbones instead of the PSTN. By keeping voice traffic on IP networks, VoIP carriers can lower costs, improve call quality and offer new IP-enabled services to their enterprise customers. SIP peering should solve several technical problems, one of the most important issues is routing of SIP message, the routing path may affect the security and QoS of communication.

A number of different approaches for finding a good path are possible. One useful way of categorizing these approaches is on the basis of the type of information the entities need to exchange in order to be able to find the good path. Distance vector algorithms are based on the exchange of only a small amount of information. Each entity (gateway or host) that participates in the routing protocol is assumed to keep information about all of the destinations within the system. The distance vector algorithm[5] [7] [8] is iterative, asynchronous, and distributed. For distance vector algorithm, each node talks to only its directly connected neighbors, but provides its neighbor with least cost estimates from itself to all the nodes.

In this paper, we describe a mechanism that uses the distance vector algorithm for building peering relationship among SIP Service Providers. With this flexible algorithm, SIP Service Providers can renew route table dynamically, this can dramatically reduce the cost of session construction. As each pairs of SIP Service Providers have already set up trust relationship, can mostly reduce the harmful of VoIP spam and they can negotiate about flexible accounting principle.

The paper is organized as follows: section 2 presents an overview of the SIP peering problem and the existing solution have been discussed widely. Section 3 describes the proposed method will be used as well as the SIP peering model we assumed. In section 4, we illustrate the SIP peering based on distance vector algorithm solution and finally section 5 concludes the paper.

II. BACKGROUND

A. Problem Overview

SIP has seen widespread deployment within enterprises and service providers, but current deployments usually do not follow RFC3263[4], but use either hard-coded IP addresses or private DNS to route calls between SIP services providers, the inter-connection part of SIP is still a problem . This phenomena is due to several aspects as below.

In SIP ecosystem anybody is able to connect to anybody, there is no business relationship between communication partners. Therefore, no feasible billing system can be deployed.

Spam over Internet Telephony (SPIT) is another concern. The free for all nature of the email ecosystem has led to a barrage of unsolicited email (SPAM) which poses a serious threat to the usefulness of email.

The email model is not suitable for stringent Quality of Service (QoS) deployments. As there are no pre-arranged relationships with between all communicating SIP servers, there are no mechanisms to guarantee neither network performance on the IP layer for the actual voice transmission, nor can there be comprehensive tests on SIP layer compatibility. As the ingress points need to be open to anybody on the Internet, they are exposed to Denial of Service attacks.

In order to solve these listed problems, there are many approaches have been discussed, and federation approach without fail is the mostly widespread implemented. With federation approach several SIP service providers can construct a central SIP Proxy, all federation members just need to open up their ingress elements to request from that central server. A new SIP service provider just triggers a change in the configuration of this box and not at all other SIP Service Providers. This approach is quite popular today, but there cannot be only one federation for all SIP Service Providers of

This work was supported by China Internet Network Information Center

worldwide, so the next problem we have to solve is how to interconnect the unceasing appearing "central SIP Proxy".

B. Requirements

One of the most issues in solving interconnect problem is routing method. Routing is the task of finding a path from a sender to a desired destination.

Giving the extensibility and maneuverability of SIP peering system, the routing method must be flexible and simple to implementation. As SIP service provider may join/leave the peering system at any time, the routing method should broadcast the latest routing information to all active SIP service providers as soon as possible. Quickly and automatically renew the route table is a great benchmark to the route algorithm.

One advantage of distance vector routing protocols is simplicity. Distance vector routing protocols are easy to configure and administer. With distance vector algorithm messages only broadcast between direct connected nodes, when a SIP service provider join/leave the peering system, its neighbor trigger the process of renew routing table quickly.

III. METHODOLOGY

A. Graphs Structure Overview

An undirected graph is an ordered pair G=(V,E) with the following properties:

The first component, V, is a finite, non-empty set. The elements of V are called the vertices of G.

The second component, E, is a finite set of sets. Each element of E is a set that is comprised of exactly two (distinct) vertices. The elements of E are called the edges of G.

Edge in undirected graph is presented by pairs of vertices, (v_i, v_j) and (v_j, v_i) presented the same edge, we say v_i and v_j is connected if there is an edge between v_i and v_j.

In the scenario of SIP peering, suppose each SIP service provider as a node, we use SSP stand for SIP Service Provider for short. If two SIP service providers have established direct peering relationship, there is an edge between them. Let's assume that any SSP is managing a table to keep track of other SSP that they are allowed to direct communicated with him.

The weight/metric between SIP service providers can be measured by many elements, bandwidth/billing/concurrent sessions and so on, we use Cost as the weight/metric, Cost is a function of the elements that have influence on SIP communication, and it has direct ratio with the communication cost.

When two SSP construct direct trust relationship, they also exchange other communication properties such as certificate information, communication protocol and so on.

Every SSP node maintains a 3-tuplet Route Table $(dest, D, N)$, with each element means destination, distance to destination and next hop to destination.

These SSP nodes exchange information periodically with trusted neighbors, the information exchanged is a 2-tuplet Broadcast Table $(dest, D)$, with each element means destination and distance to destination.

Once a node received Broadcast Table from its neighbor nodes, it renews the local Route Table as soon as possible, and broadcast the latest Broadcast Table to its neighbor nodes.

We construct a peering module within four independent SIP service providers, and use SSP_a SSP_b SSP_c SSP_d to denote them, the direct peering relationship between them as the topology graph shows in Figure 1. Within the graph theory definition, V= (SSP_a, SSP_b, SSP_c, SSP_d), E={ (SSP_a, SSP_b), (SSP_a, SSP_c), (SSP_b, SSP_d), (SSP_c, SSP_d) },the Cost of each edge is depicted on the topology graph.

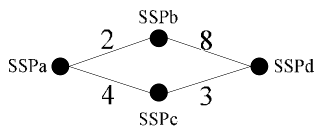

Figure 1. SIP Peering Model-1

At the initial phase, every SSP hold a Route Table only have records about direct peering relationship SSP.

B. Distance Vector Algorithm

In General, if it is possible to get from entity i to entity j directly (i.e., without passing through another gateway between), then a cost Cost(i,j), is associated with the hop between i and j. In the normal case where all entities on a given network are considered to be the same, Cost(i,j) is the same for all destinations on a given network, and represents the cost of using that network. To get the metric of a complete route, one just adds up the costs of the individual hops that make up the route. For the purposes of this memo, we assume that the costs are positive integers.

Let D(i,j) represent the metric of the best route path from entity i to entity j. It should be defined for every pair of entities. Cost(i,j) represents the costs of the individual steps. Formally, let Cost(i,j) represent the cost of going directly from entity i to entity j. It is infinite if i and j are not immediate neighbors. (Note that Cost(i,i) is infinite. That is, we don't consider there to be a direct connection from a node to itself.) Since costs are additive, it is easy to show that the best metric must be described by (1)

$$\begin{aligned} &D(i,i) = 0, && \text{, all } i \\ &D(i,j) = \min_k \,[Cost(i,k) + D(k,j)], && \text{otherwise} \end{aligned} \tag{1}$$

and that the best routes start by going from i to those neighbors k for which Cost(i,k) + D(k,j) has the minimum value. (These things can be shown by induction on the number of steps in the routes.) Note that we can limit the second equation to k's that are immediate neighbors of i. For the others, Cost(i,k) is infinite, so the term involving them can never be the minimum.

IV. SOLUTION

A. Distance Vector Algorithm Approach

In our model, there are 4 SIP service providers. At this point, all SSP's Route Tables have new "shortest-paths" for their DV (the list of distances that are from them to another SSP via a neighbor). They each broadcast this new Broadcast Table to all their neighbors: SSP_a to SSP_b and SSP_c, SSP_b to SSP_d and SSP_a, SSP_c to SSP_d and SSP_a, SSP_d to SSP_b and SSP_c. As each of these neighbors receives this information, they now recalculate the shortest path to other SSP.

For example: SSP_a receives a Broadcast Table from SSP_b and SSP_c, the Broadcast Tables are depicted in Table I, and that tells SSP_a there is a path via SSP_b to SSP_d, with a distance (or cost) of 8, and another path via SSP_c to SSP_d with a distance of 3. Since the current "shortest-path" to SSP_b is 2, to SSP_c is 4, then SSP_a knows it has a path to SSP_d that costs 2+8=10 and another path to SSP_d that costs 4+3=7. As there are no other shorter paths that SSP_a knows about, it puts 7 as its current estimate for the shortest-path from itself (SSP_a) to SSP_d, via SSP_c. The equation as (2)

$$\begin{aligned} D_{SSP_a}(SSP_d) &= \min\{Cost(SSP_a,SSP_b)+D_{SSP_b}(SSP_d), Cost(SSP_a,SSP_c)+D_{SSP_c}(SSP_d)\} \\ &= \min\{2+8, 4+3\} \\ &= 7 \end{aligned} \tag{2}$$

TABLE I. BROADCAST TABLE RECEIVED BY SSPA

(A) BROADCAST TABLE FROM SSPB

dest	D
SSP_a	2
SSP_d	8

(B) BROADCAST TABLE FROM SSPC

dest	D
SSP_a	4
SSP_d	3

After this process, SSP_a update the Route Table as in Table II.

TABLE II. ROUTE TABLE OF SSPA

dest	D	N
SSP_b	2	SSP_b
SSP_c	4	SSP_c
SSP_d	7	SSP_c

B. New SIP Service Provider Joined

Similar to Email system, new SIP service provider may join the peering system at any time. The new joined SSP should construct direct peering relationship with at least one SSP which has already have the Route Table of the SIP Peering ecosystem.

For example, SSP_e is a newcomer SIP service provider, and has already construct direct peering relationship with SSP_d, Cost(SSP_d, SSP_e)=1. New peering model is depicted in Figure 2.

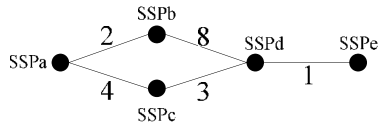

Figure 2. SIP Peering Model-2

Once SSP_e construct peering relationship with SSP_d, SSP_d should update its Route Table, add a record to SSP_e, and broadcast the latest Broadcast Table to its neighbors. The latest Route Table of SSP_d as in Table III.

TABLE III. ROUTE TABLE OF SSPD

dest	D	N
SSP_b	8	SSP_b
SSP_c	3	SSP_c
SSP_e	1	SSP_e

SSP_d broadcast the latest Broadcast Table to SSP_b and SSP_c, then SSP_b and SSP_c renew their Route Tables according to the Broadcast Table. Once SSP_b and SSP_c renew their Route Table, they broadcast the latest Broadcast Tables to SSP_a, as in Table IV.

TABLE IV. BROADCAST TABLE RECEIVED BY SSPA

(A) BROADCAST TABLE FROM SSPB

dest	D
SSP_a	2
SSP_d	8
SSP_e	9

(B) BROADCAST TABLE FROM SSPC

dest	D
SSP_a	4
SSP_d	3
SSP_e	4

SSP_a renews its Route Table according to Broadcast Table from SSP_b and SSP_c. The equation as (3)

$$\begin{aligned} D_{SSP_a}(SSP_e) &= \min\{Cost(SSP_a,SSP_b)+D_{SSP_b}(SSP_e), Cost(SSP_a,SSP_c)+D_{SSP_c}(SSP_e)\} \\ &= \min\{2+9, 4+4\} \\ &= 8 \end{aligned} \tag{3}$$

Now SSP_a gets a path to SSP_e, the cost of this path is 8 within the next hop SSP_c. The latest Route Table of SSP_a as in Table V.

TABLE V. ROUTE TABLE OF SSPA

dest	D	N
SSP_b	2	SSP_b
SSP_c	4	SSP_c
SSP_d	7	SSP_c
SSP_e	8	SSP_c

Till now, SSP_a SSP_b SSP_c SSP_d all have a path to SSP_e .Likewise SSP_e can renew its Route Table, get the latest route information to other SIP Service Providers.

C. Use Case

This part gives an example use case to explain how distance vector algorithm take effect in SIP Peering. We make an assumption that all SSP in our distance vector algorithm model are backbone SIP proxy. They take the role of central SIP proxy in a federation. SSP_a is a central SIP proxy of SSP_{a1} , SSP_{a2} and SSP_{a3} , call from the three SIP service providers are redirect to the backbone SSP_a , SSP_a decide the next hop according to distance vector algorithm. Likewise SSP_d is a central SIP proxy of SSP_{d1} ,SSP_{d2} and SSP_{d3} , as depicted in Figure 3.

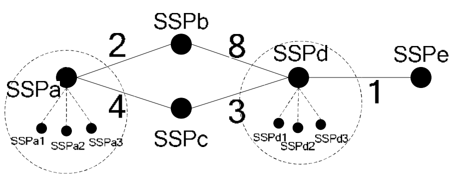

Figure 3. SIP Peering Model-3

On the assumption that Alice is a subscriber of SSP_{a2} ,Bob is a subscriber of SSP_{d3} .If Alice want to communicate with Bob, she should send Invite message with Bob's SIP URI BOB@SSP_{d3} .

Between Alice and Bob, five SIP Service Providers have been involved, each SSP route Alice's Invite request based on federation principle or local Route Table generated by distance vector algorithm. Each SSP on this path is trustable. SSP_a and SSP_c may use UDP protocol and their private certificate, while SSP_c and SSP_d may use TCP protocol and other private certificate they shared. The path is the least cost, can guarantee QoS and security of communication.

D. Analysis

With distance vector algorithm, SIP service providers can dynamically and quickly achieve equilibrium state. The path between SIP service providers is the best one, can guarantee the QoS of communication. SIP messages are always transmitted between trusted SIP service providers, the parameters of communication have been negotiated, so this can prohibit VoIP spam, and SIP service providers can negotiate about agile billing mechanism.

SIP Peering with distance vector algorithm has excellent extension, new joined SIP service provider only need to construct direct relationship with several SIP service providers, then can get the path to other SIP service provider dynamically and automatically.

This mechanism can be used with other SIP peering method, for instance collaborating with federation case. Distance vector algorithm mechanism only used among backbone SIP service providers, other small SIP service providers direct peering with several fixed backbone SIP service providers, once received outbound SIP calls, just redirect to those backbone SIP service providers, the backbone SIP service providers use distance vector algorithm calculate the next hop. With this collaboration, can reduce the size of SSP's Route Table and reduce transmit cost of Broadcast Table.

V. CONCLUSION

We introduce distance vector algorithm into SIP peering subject, with the mechanism, numerous SIP service providers can automatically find the best path, reduce the communication cost. This algorithm can also prohibit VoIP spam, and facilitate SIP service providers to negotiate about agile billing mechanism. The mechanism has great extension, makes new SIP service provider joined effortless. As we know, distance vector algorithm has a shortage of "slowly converge"[9], we can use other mechanism to decrease the influence, such as hold down mechanism. And distance vector algorithm can be used with other peering mechanism, for instance, only use distance vector algorithm among backbone SIP service providers, these can reduce each SIP service provider's pressure and improve the pace of converge.

REFERENCES

[1] H. Schulzrinne and J. Rosenberg: "The Session Initiation Protocol:Internet-Centric Signaling", IEEE Communication Magazine, vol.38, 10,pp-134-141, Oct.(2000)

[2] J.Rosenberg, et.el.: "SIP: Session Initiation Protocol", RFC3261,http://www.ietf.org/rfc/rfc3261.txt, June(2002)

[3] O.Lendl,2008.VoIP Peering:Background and Assumptions, IETF, Work in progress,http://tools.ietf.org/html/draft-lendl-speermint-background-01

[4] H. Schulzrinne and J. Rosenberg: "Session Initiation Protocol (SIP):Locating SIP Servers", RFC3263,http://www.ietf.org/rfc/rfc3263.txt, June(2002)

[5] Hedrick, C., "Routing Information Protocol", STD 34, RFC 1058, http://www.faqs.org/rfcs/rfc1058.html, June 1988

[6] G. Huston, "Interconnection, peering, and settlements," in Proc. INET, June 1999.

[7] R. Albrightson, J. J. Garcia-Luna-Aceves, and J.Boyle, ÒEIGRP-A Fast Routing Protocol Based on Distance Vectors,Ó Proceedings Networld/Interop 94, Las Vegas, Nevada, May 1994.

[8] C. Cheng, R. Reley, S. P. R Kumar and J. J. Garcia-Luna-Aceves, ÒA Loop-Free Extended Bellman-FordRouting Protocol without Bouncing Effect'', ACM Computer Commun. Review, Vol.19, No.4, 1989, pp. 224-236.

[9] D. Bertsekas and R. Gallager, Data Networks, Prentice Hall, Inc., Second Edition 1992