Technical Disclosure Commons

Defensive Publications Series

May 2020

AUTOMATIC SYMMETRIC NETWORK ADDRESS TRANSLATION DISCOVERY AND TRAVERSE BETWEEN SOFTWARE-DEFINED WIDE AREA NETWORK EDGE DEVICES

Tony Shen

Laxmikantha Reddy Ponnuru

Ajay Kumar Mishra

AUTOMATIC SYMMETRIC NETWORK ADDRESS TRANSLATION DISCOVERY AND TRAVERSE BETWEEN SOFTWARE-DEFINED WIDE AREA NETWORK EDGE DEVICES

AUTHORS:
Tony Shen
Laxmikantha Reddy Ponnuru
Ajay Kumar Mishra

ABSTRACT

The techniques presented herein relate to automatic network address translation (NAT) discovery and traverse for edge devices in a software-defined network in a wide area network (SD-WAN). More specifically, the techniques presented herein relate to techniques for automatic, symmetric NAT discovery and traverse between SD-WAN edge devices that ensure new branches can become part of an SD-WAN overlay network without explicit user (e.g., customer/administrator) involvement. Accordingly, and advantageously, the techniques presented herein can dynamically provide NAT-T hub connectivity for symmetric devices that can be used for service insertion of NAT-T functions. These techniques may vastly simplify SD-WAN deployment in diverse deployments and, thus, may provide significant business value.

DETAILED DESCRIPTION

The recent proliferation of symmetric NATs has reduced NAT traversal success rates in many practical situations, such as for broadband, mobile, and public WiFi connections. The issues mostly stem from the fact that symmetric NAT is a restrictive form of NAT where mapping is done per destination. That is, with symmetric NAT, each destination sees different internet protocol (IP) addresses and/or ports for the same source.

As a specific example of an issue caused by symmetric NAT, when one branch router is behind a symmetric NAT in an SD-WAN, that branch router cannot receive packets from other branch routers. Even if an orchestrator (e.g., a vBond Orchestrator) sees different IP address/port for each branch router and shares this information with the other branch routers, the other branch routers will not have the correct mapping and, thus, the branch router behind the symmetric NAT does not receive packets. One solution to this issue is to use bidirectional forwarding detection (BFD) and/or Internet Protocol

Security (IPsec) protocols to detect the branch router behind the symmetric NAT and change the session to the correct mapping. However, this solution is not necessarily scalable/applicable to all architectures, such as an architecture where one side is symmetric NAT, and the other side is port/IP address restricted.

In fact, the aforementioned solution works because when two branch routers intend to form a data connection, at least one of the two branch routers is able to reach a branch that is not behind the symmetrical NAT/ port/IP address restricted devices, which receives packets from other branches. However, when both branches (e.g., two SD-WAN edge devices) are behind symmetric NAT, the aforementioned solution is unable to detect such a condition. Instead, in these scenarios, a user (e.g., a customer) has to make sure both of these devices connect to a router and form a BFD connection via the router. However, it is hard to manage this issue when an SD-WAN includes a large number of branches (e.g., 10,000 or more). This issue may be further exacerbated in scenarios where a customer has no knowledge of control of the NAT or security settings, such as public broadband internet and/or where NAT techniques are implemented by a service provider (which could then require full-cone NAT that could be problematic and/or challenging for a third party vendor).

In view of the foregoing, the techniques presented herein propose using a dynamically applied centralized control policy based on automatically discovered transport location (TLOC) and symmetric NAT information (which is published across the SD-WAN fabric). Then, the techniques and enforcing routing policies for automatic routing topology via third party network hubs or NAT gateways that are not behind symmetric NAT. That is, at a high-level, the techniques presented herein:

- implement symmetric NAT discovery at orchestrators (e.g., a vBond orchestrator), centralized controllers (e.g., a vSmart controller), and/or edge routers/cloud routers (e.g., cEdge and/or vEdge routers), which may be public or behind full-cone NAT;
- publish symmetric NAT information discovered and owned by edge devices across edge router domains and to the centralized controllers;

- enforce, by the centralized controller, centralized routing policy based on the symmetric NAT information for TLOC , which will build dynamic NAT-Traversal (NAT-T) topology for NAT-T dynamically (e.g., hub-and-spoke topology); and
- select NAT gateways to ensure that two edge routers (e.g., vEdge and/or cEdge devices) can choose the same NAT gateway/hub for data path connectivity (removing the need for a centralized controller to control routing policy dependency).

Accordingly, and advantageously, the techniques presented herein can dynamically provide NAT-T hub connectivity for symmetric devices that can be used for service insertion of NAT-T functions.

Notably, although some known techniques can sometimes detect a NAT type of certain SD-WAN edge devices behind the NAT (e.g., by using two controllers as stun-servers to discover public address and port information per transport/interface basis), the present techniques enhance this NAT type detection and discovery. For example, although some SD-WAN edge devices are able to successfully detect and discover NAT device types, the techniques presented herein cause edge devices to publish public TLOC and NAT-T type information over an SD-WAN fabric, across orchestrators, controllers, and edge routers, after discovering such information. In turn, this published information allows for symmetric NAT-T via third party gateways or hubs, including even in scenarios that might have previously prevented NAT-T (e.g., where on device of a connection is behind symmetric NAT and the other is behind symmetric NAT or port/IP address restricted). More specifically, the published information allows SD-WAN hubs to automatically and dynamically apply a centralized policy and provide data-plane connectivity to edge devices behind symmetric NAT. Moreover, the dynamic NAT-T hub connectivity can be used as part of service insertion for NAT-T traverses functions.

To achieve this, and as is noted in the first bullet point above, the techniques presented herein can identify symmetric NAT at controllers and orchestrators. However, when the symmetric NAT is discovered at a single orchestrator, different IP address/Port pairs are utilized for the same NAT discovery and traversal. Additionally, in accordance with the present techniques, orchestrators can proactively establish Datagram Transport

Layer Security (dTLS) connections to edge devices to discover NAT-T type and full-cone types (e.g., full-cone, address restricted full-cone, port restricted full-cone, without NAT, or unknown). Still further, the present techniques can leverage additional controllers in an SD-WAN to discover post-NAT IP address per-transport, to more effectively discover possible symmetric NAT based on the dTLS connections across multiple controllers for the same transport. Then, edge devices behind a public of full-cone NAT can discover symmetric NAT based on the IPSec/BFD.

Once edge devices publish public TLOC and NAT-T type information over the fabric (e.g., across orchestrators, controllers, and edge routers), NAT gateways/hubs can be selected to achieve symmetric NAT-T via third party gateways/hubs. According to a first example embodiment, a centralized routing control policy is utilized to build a hub-and-spoke NAT-T topology with the published public TLOC and NAT-T type information. According to a second example embodiment, one or more SD-WAN devices are introduced for the role of the NAT-T gateway within SD-WAN Fabric and functions to provide NAT-Traversal. Each of these embodiments is discussed in further detail below.

First, with the former (e.g., first) embodiment, a centralized routing control policy can be built based on symmetric NAT-T type to build NAT-T topology for devices behind Symmetric NAT-T devices that cannot establish direct bi-directional data-paths. Then, the NAT-T topology can be built based on availability of a Hub with public or full-cone NAT IP addresses. Devices behind symmetric NAT will build intermediate tunnels to the NAT-T gateway if the remote edge device is also behind Symmetric NAT (e.g., similar to how service-insertion builds an end-to-end path between a first edge router (“Router A”) and a first edge router (“Router B”) by building two tunnels: one between Router A and a third edge router (“Router C”); and one between Router C and Router B). As a specific example, the following centralized routing control policy sets the NAT-T and TLOC information for routes published via a TLOC that is attached to TLOC NAT-T attributes that are symmetric for public-internet:

```
policy
   lists
    site-list branch-sites
      site-id 100-200
   control-policy change-tloc-nat
    sequence 10
```

```
        match route
          site-list branch-sites
          color public-internet
          tloc nat-t symmetric
        action accept
          set
            tloc-action nat-traverse #### specific action for nat-traverse
            tloc 10.1.1.1 color public-internet encap ipsec
apply-policy
  site branch-sites control-policy change-tloc-nat out
```

Second, with the latter (e.g., second) embodiment, the one or more SD-WAN devices introduced for the role of the NAT-T gateway within SD-WAN Fabric are not behind symmetric NAT and, thus, can provide NAT-T gateway functionality for other edge devices. Once the one or more devices publish their gateway capabilities and TLOC information, edge devices in the fabric can build overlay topology via the public Nat-T gateway devices when both local and remote are behind symmetric NAT. Moreover, device-specific information, such as site-ID, TLOC, and transport related information can be hashed when multiple devices are available as NAT-T gateways so that each of two edge devices behind the NAT can select the same NAT-T gateway.

Figure 1 illustrates an example workflow for executing the first example embodiment laid out above in connection with example SD-WAN devices. In this diagram, "vSmart" is a controller, "vEdge1," "vEdge2," and "vEdge3" are edge routers (with vEdge3 being designated a Hub), and "vBond" is an orchestrator.

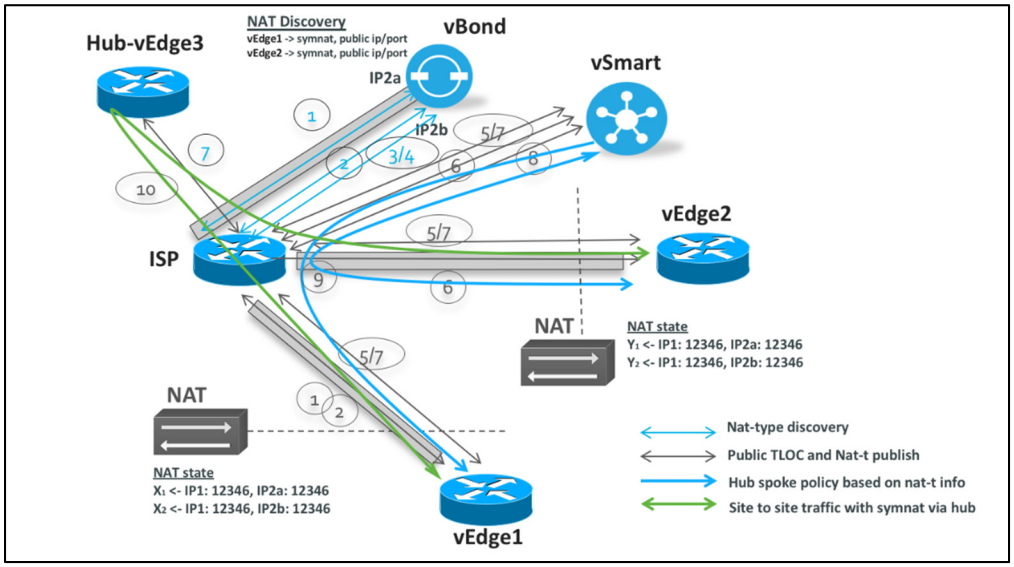

Figure 1

In this workflow, the following steps occur:

1. vBond receives a request from the vEdge1 for control connections. vBond learns the public IP/port of vEdge1, this mapping is IP1: P1.

2. vBond receives a request from vEdge1 device for the control connection. vBond learns the public IP/port of vEdge1, but this would be different from IP1: P1, for example IP1: P2, since the vBond appliance has different TLOC (Public IP and Port) connected via the same transport/cloud (alternatively, a first vBond could receive the request from vEdge1 and a second vBond could receive the request from vEdge2);

3. if vEdge1 is behind symmetric NAT (e.g., if IP1: P1 and IP1: P2 are the same) vBond proactively connects to the discovered public TLOC via a separate or third public IP on vBond to determine a NAT-T type;

4. vBond shares the public IP/Port and NAT-T type of vEdge1, and vEdge1 can learn if it is behind symmetric NAT, and also the NAT-T type;

5. vEdge1 publishes the NAT-T type and public TLOC information to the vSmart (e.g., via OpenMP ("OMP")) and vSmart shares this TLOC information (e.g., over OMP) to routers vEdge1, vEdge2, and vEdge3;

6. In this example, vEdge2 is also behind symmetric NAT and, thus, steps 3-5 are repeated for vEdge2 and public TLOC and NAT-T type information are also published across the whole SD-WAN fabric (e.g., via OMP);

7. vEdge branch3 is not behind the symmetric NAT and used a role of a NAT-T gateway/hub;

8. vSmart controller enforces centralized routing control policy for all branch sites based on discovered NAT-T type info for each pair of branch sites based on TLOC info, which is used to assist the NAT-T based on discovered NAT-T information and the policy (an example policy is laid out above).

9. Based on vSmart centralized routing policy, vEdge1 builds overlay routing topology to vEdge2 via the Hub (vEdge3) based on the policy by setting TLOCs for OMP routes between vEdge1 and vEdge2 with service insertion TLOC directionally;

10. vEdge1 automatically builds a routing path to vEdge2 via vEdge3 (the Hub) based on the vSmart control policy;

Alternatively, after step 7, the techniques presented herein can utilize a different workflow, pursuant to the second example embodiment discussed above. Figure 2 illustrates am example workflow for executing the second embodiment in connection with SD-WAN devices. Since steps 1-7 remain the same between the first embodiment and the second embodiment, only steps after step 7 are described below.

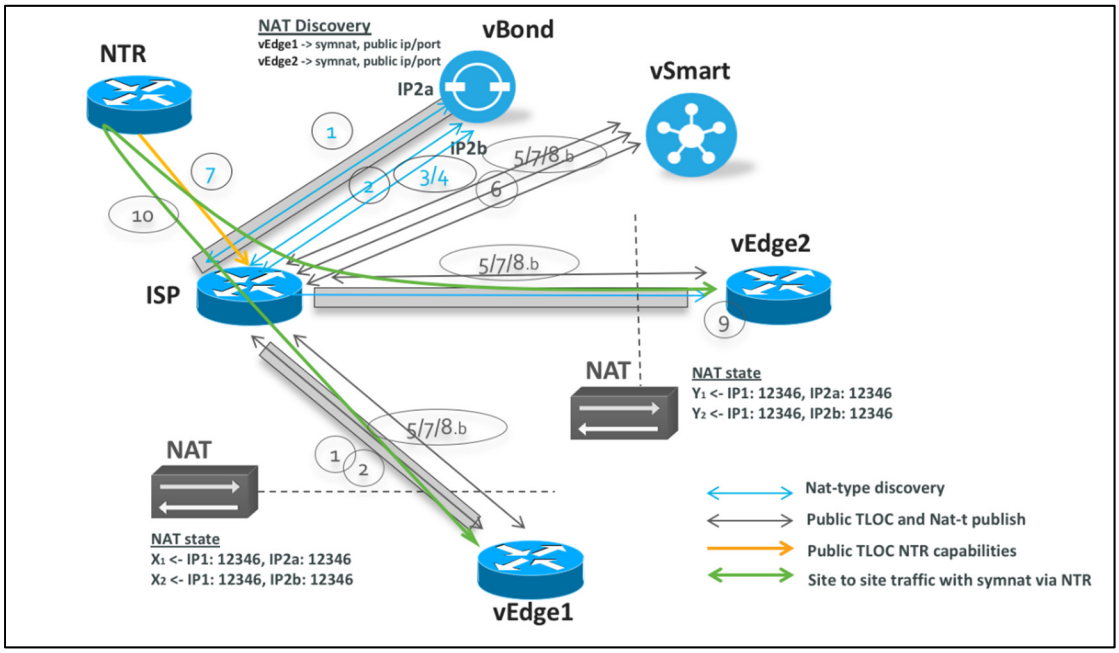

Figure 2

8b. When the TLOC and NAT-T information is shared with vEdge1 and vEdge2, these routers realize that they are behind symmetric NAT;

9b. Based on the techniques laid out above, vEdge1 and vEdge2 both choose vEdge3 for a gateway data connection;

10b. BFD sessions come up between vEdge1 and vEdge3 and between vEdge2 and vEdge3 (e.g., using the existing BFD based symmetric NAT detection);

11b. vEdge3 is selected as a gateway, which can automatically do reverse routing injection for connected routers, or advertise default route or summary route to ensure bidirectional routing traffic across vEdge1 and vEdge2 via vEdge3.

Notably, these example embodiments allow NAT discovery and traverse in a wide variety of use cases, including scenarios where previous techniques have not allowed for NAT discovery and traverse. For example, the techniques presented herein may allow for

NAT-T when a first device is behind symmetric NAT and a second device is behind symmetric NAT or IP address/port restricted. In fact, the techniques presented herein may facilitate NAT-T in at least the following scenarios (with "Site A" and "Site B" representing two devices forming a connection):

Site A	Site B
Public	Public
Full Cone	Full Cone
Full Cone	Port/Address Restricted
Port/Address Restricted	Port/Address Restricted
Public	Symmetric
Full Cone	Symmetric
Symmetric	Port/Address Restricted
Symmetric	Symmetric

For the first use case (public-public), tunnels (e.g., direct IPSec tunnels) may be used for NAT-T (e.g., the first example embodiment may be utilized). However, for the last two use cases (symmetric-symmetric and symmetric-port/address restricted), traffic may traverse a hub (e.g., the second example embodiment may be utilized) and tunnels ((e.g., direct IPSec tunnels) may be unnecessary.

In summary, techniques are presented herein for automatic network address translation (NAT) discovery and traverse for edge devices in a software-defined network in a wide area network (SD-WAN). The techniques provide automatic, symmetric NAT discovery and traverse between SD-WAN edge devices while ensuring that new branches become part of an SD-WAN overlay network without explicit user (e.g., customer/administrator) involvement. Accordingly, and advantageously, the techniques presented herein can dynamically provide NAT-T hub connectivity for symmetric devices that can be used for service insertion of NAT-T functions. These techniques may vastly simplify SD-WAN deployment in diverse deployments and, thus, may provide significant business value.