Adaptive Routing Overlay addressing with OSPF

$Rev: 2$

The separation of adaptive routing from other requirements has helped isolate previously intertwined implementation assumptions. With adaptive routing taken into account, we re-define the problem requirements for providing a flat overlay address space, and propose an alternative implementation.

Note the 100.64/10 IP range is used for “flattened” virtual addressing here, rather than the 172.16/12 range previously discussed. Also, numbering for the private LAN ranges is representative; in practice any private range may be used.

Contents

Context

We aim to provide a “flat” address IP space over multiple networks connected by an OSPF core. Routers' external interfaces have pre-allocated “public” addressing, which cannot be renumbered. One router may possibly have multiple IPs, usually due to providing multiple physical media.

Edge routers typically NAT from private LANs. LANs may be in duplicated (physically isolated) ranged, numbered by the conventional private range allocations. Various devices on a LAN (notably cameras) are to be reached by port forwarding TCP or UDP from those public IPs.

The external interfaces on edge routers do not provide IP aliasing, and therefore LAN IPs cannot be made reachable by direct IP routes—nor would this be desirable, as those LANs are intended to remain unreachable except by port forwarding for sake of security by isolation.

Redefined Problem Statement

So far we have maintained the assumption that VPN style IP tunnelling is to be implemented. However nothing in the above scenario actually requires this.

Consider a TCP session over a route set up by OSPF, and therefore reachable by IP. The source and destination know their respective “public” IPs as OSPF neighbours, and port forwarding is used at the destination to forward to a particular address in the private LAN range.

This is nothing we wouldn't be able to do, if we didn't know that "public" destination IP in the first place. All we want to add is the convenience for these “flat” virtual IPs.

Therefore this isn't a question of tunnelling, but rather a question of how the source node finds that public IP of the destination router, given its virtual address in the flattened space.

Proposal

We propose that OSPF could be used to disseminate the mapping between virtual and public IP addresses, by the same mechanism as neighbour selection, and presented as OSPF Router IDs.

The crux of this idea is that routing is done by remapping from virtual IPs to the “public” IPs by packet re-writing at the source router.

The destination router would essentially be unaware of this, and hence just see a TCP session, routed as per usual by its IP route to a public IP.

Topology

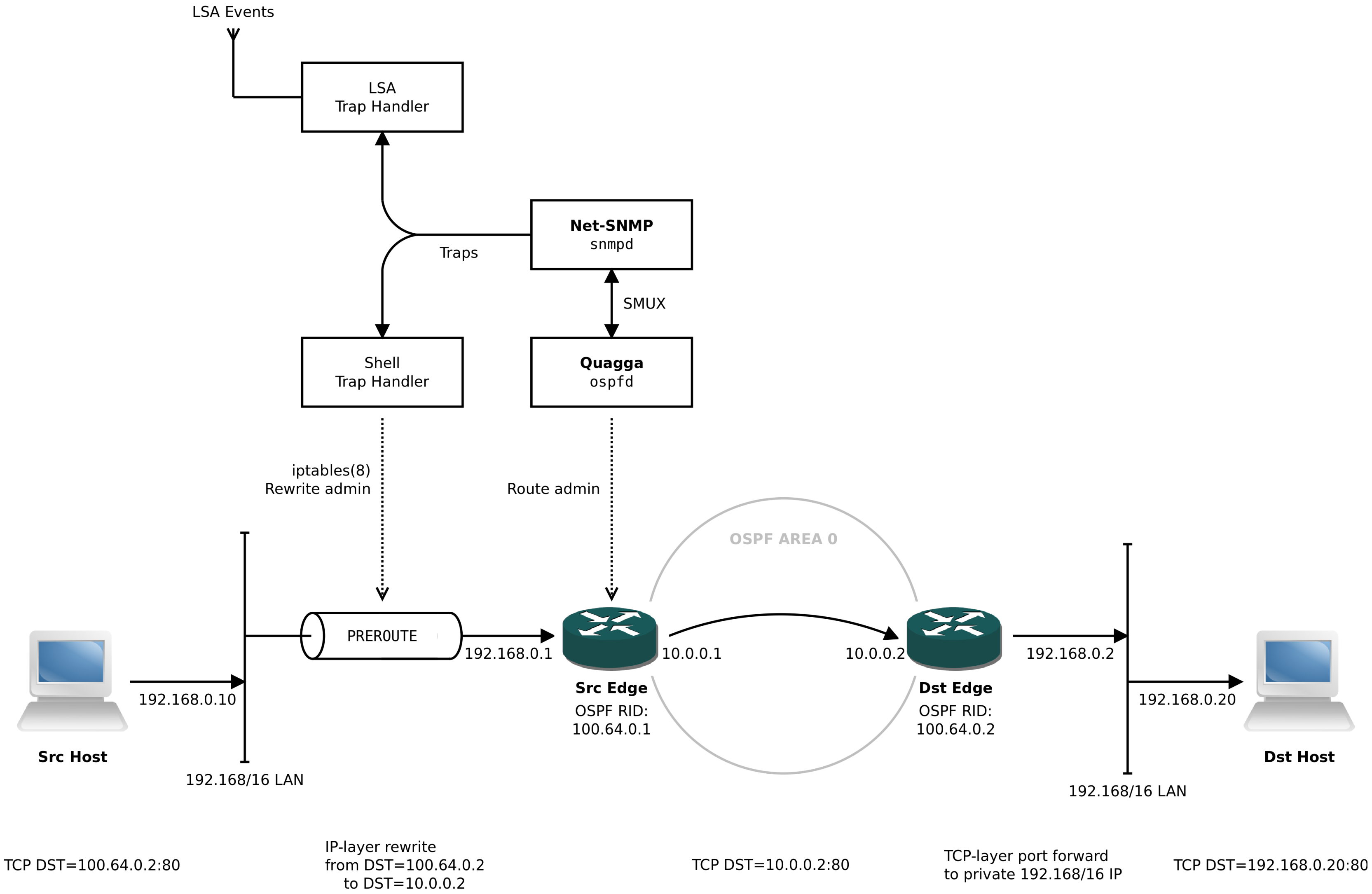

Figure 1. Overlay topology

Setup Sequence

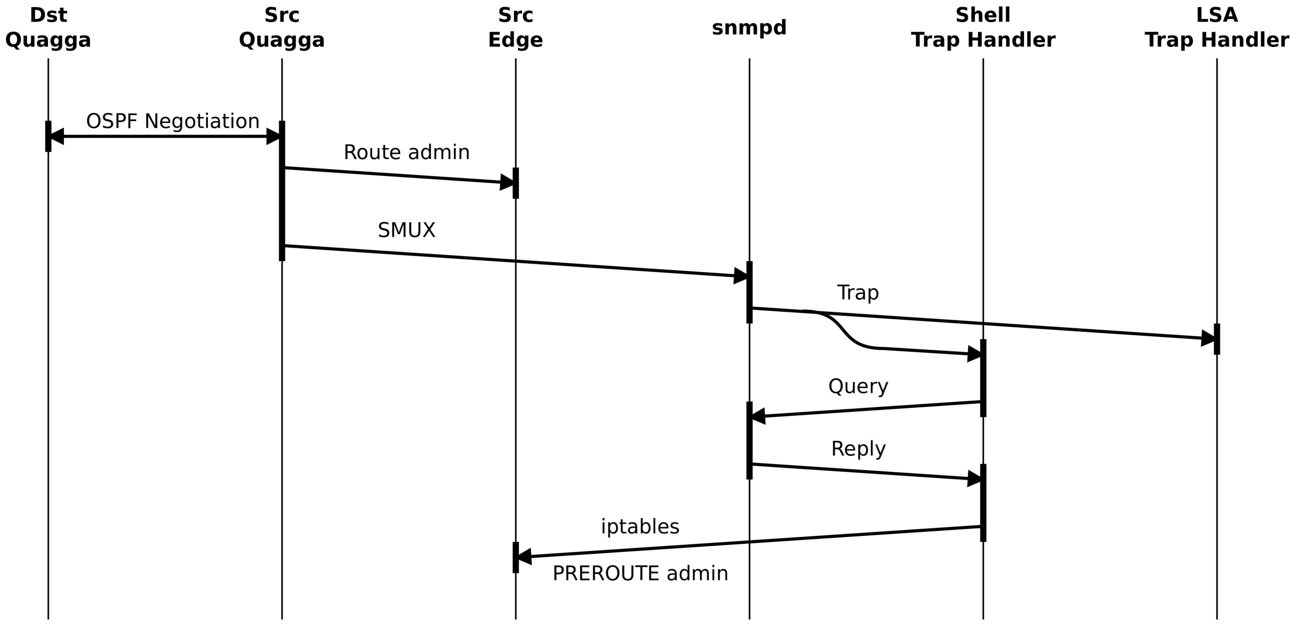

Figure 2. OSPF Negotiation and SNMP Traps

1. OSPF establishes neighbours for Quagga instances
2. OSPF adds local routes
3. OSPF raises SNMP trap to indicate neighbour change
4. Trap handler queries SNMP for neighbour list
5. Trap handler updates iptables to add (or replace) IP rewrite rules

Traffic Sequence

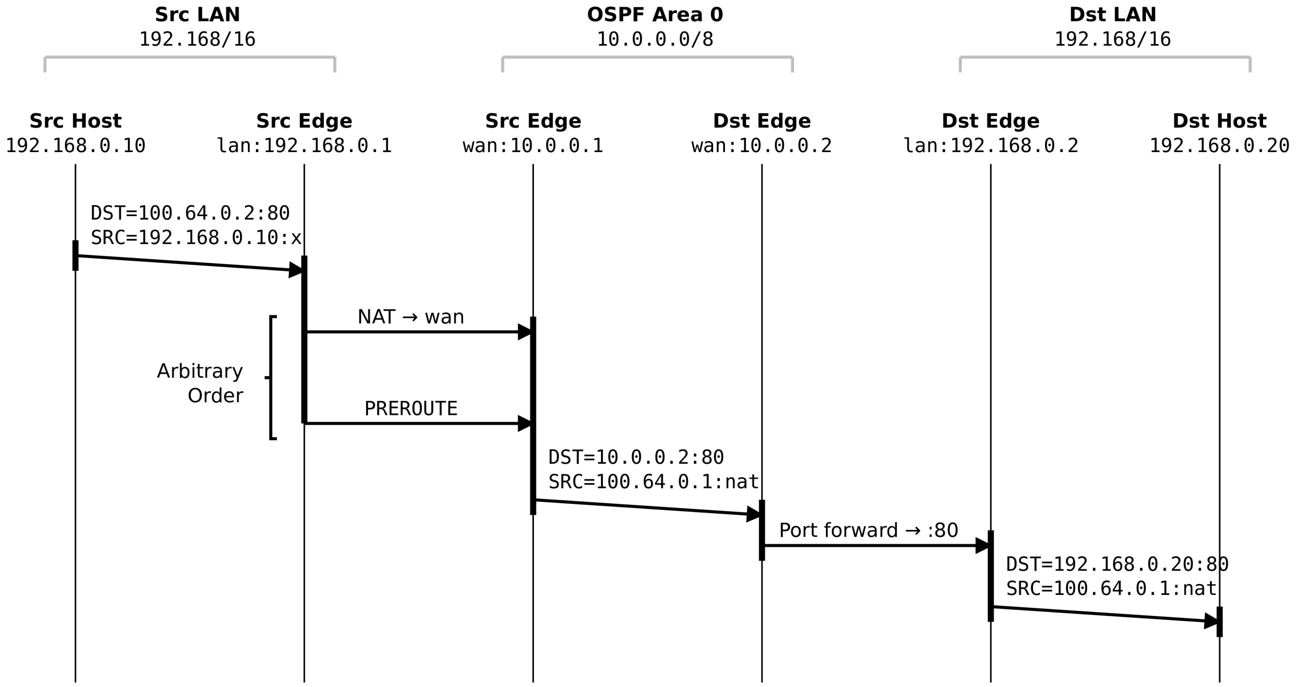

Figure 3. Routing, NAT and Port Forwarding

1. Src Host on 192.168/16 sends outgoing packet to DST=100.64.0.2
2. Src Edge NATs to its wan interface on 10.0.0.1

   Src Edge iptables rewrites SRC and DST in PREROUTE
3. Route over OSPF Area 0
4. Dst Edge port forwards to Dst Host on 192.168/16

Return path is same as above, in reverse

Security Concerns

There is a race between SNMP trap delivery to the LSA handler, and SNMP trap delivery to the shell handler for setting up `PREROUTE` rewriting. This could be avoided by combining both into a single handler, or alternatively by having the shell trap handler in turn raise a trap when it has completed the PREROUTE admin by itables, and having the LSA handle that trap instead.

There is no VPN-style tunnelling, and therefore no throughput overhead due to layering or data encapsulation. As a result, this side-steps the issue of the overlay network itself causing fragmentation for previously un-fragmented MTU-sized packets.

This is beneficial for security in preference to a VPN-style solution as fragmentation is often mis-implemented, where headers for trailing fragments are erroneously taken to overwrite the headers from a head fragment, during reassembly. That can lead to bypassing firewall and IDS conditions which (correctly) only take the head fragment into account.

Traffic on the network is exactly as if it were routed via OSPF. Therefore there is no impact to existing network fabric. This is just a mechanism to “know” the ever-changing destination IPs.

OSPF could be substituted for any other adaptive routing protocol which has similar functionality. In particular, this design requires being able to convey semantically relevant information (the overlay IPs) in an otherwise arbitrary identifier (an OSPF Router ID).