

**MASTER'S PROJECT  
REPORT**

On

**IP LAYER PACKET  
REDIRECTION**

by

Sriram Krishnamurthy

Computer Science

S.U.N.Y. Stony Brook

2004

Approved by \_\_\_\_\_  
Project Advisor

Date \_\_\_\_\_



## TABLE OF CONTENTS

1. Packet Redirection.....	1
2. The OSPF Protocol: .....	4
2.1 Splitting the AS into Areas .....	4
2.2 Classification of routers .....	4
2.2.1 Internal routers.....	4
2.2.2 Area border routers.....	4
2.2.3 Backbone routers.....	5
2.2.4 AS boundary routers.....	5
2.3 Use of external routing information.....	5
2.4 Working of OSPF.....	5
2.5 Originating LSAs.....	6
2.6 The Flooding Procedure.....	7
2.7 The OSPF packet.....	8
3. Modified OSPF daemon:.....	8
3.1 VON as DR: .....	8
3.2 Disable broadcast to ALLSPFROUTERS:.....	9
3.3 Router LSA modifications.....	9
3.4 Originate AS External LSA.....	10
4. Results.....	12
4.1 OSPF routing table at AR: .....	12
4.2 The kernel routing table at AR: .....	12
4.3 The kernel routing table at VON:.....	13
4.4 The kernel routing table at ABR: .....	13
4.5 Time taken for convergence .....	13
4.5.1 Establishing neighborhood:.....	14
4.5.2 Establishing Adjacency:.....	14
4.5.3 Installing LSA:.....	14
4.5.4 Route creation:.....	14
5. Applications:.....	15
6. Case-Study – A virtual overlay network to counter a DDoS attack.....	15
6.1 Filtering at the Virtual Overlay Nodes(VONs):.....	15
6.2 VON Tree .....	16
6.3 Formation of VON tree .....	17
6.4 Construction of MST .....	18
6.5 VON Tree Manager (VTM).....	19
6.6 Percentage computation at the VONs.....	20
6.7 The choice of $C_{\text{thresh}}$ .....	25
References .....	26

## LIST OF FIGURES

Figure 1: Control System.....	1
Figure 2: VON Tree.....	17
Figure 3: Input to VON tree.....	25

## ACKNOWLEDGMENTS

I wish to express sincere appreciation to Professor Tzi-cker Chiueh in the preparation of this report. In addition, thanks to Patrick Tonra for providing a suitable test bed of VMware machines.

## GLOSSARY

### **Adjacency**

Neighboring routers(not necessarily all) establish adjacency(a mutual coupling) in order to exchange routing information.

### **Link state advertisement (LSA)**

Information that indicates the local state of a router/network. This also contains the interface cost for all interfaces of the router.

### **Hello Protocol**

A Protocol that establishes neighborhood between routers. This involves sending hello packets through all the interfaces enabling routers to establish neighborhood dynamically on broadcast networks.

### **Designated Router**

A designated router takes the responsibility of broadcasting/multicasting LSAs to attached routers. It is elected by the hello protocol and serves to optimize the number of LSAs exchanged between the routers.

## 1. Packet Redirection

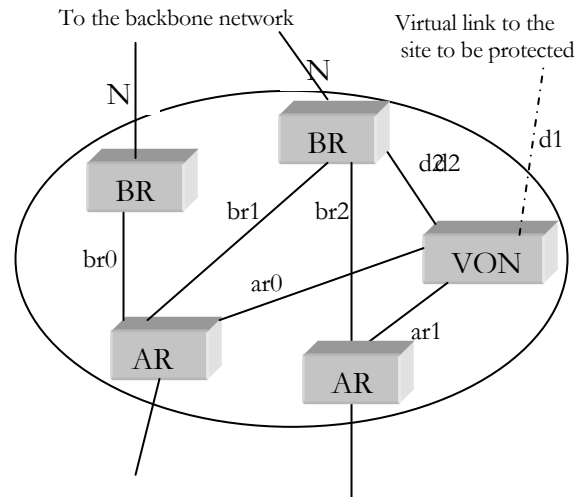


Figure 1: Control System

*Access Routers (AR)* are the routers to which customers directly connect. *Border Routers (BR)* are the exit points for the POP. VON is the virtual overlay node which redirects the packets to itself before forwarding them to the BRs.

For packet redirection to occur interface metrics should be suitably manipulated. Ideally, it should be established that the route to the external site is shortest when forwarded by the VON. The access routers then forward the packets to the VON. A static route could be set up in the VON that forwards all packets destined to the external site to the BR.

The obvious issue is to prevent BRs from forwarding packets back to the VON. To furnish the same, VON should also announce that the interface cost between the BR and the VON is costly enough that VON can no longer act as the next hop router to the external site. Hence, it does not make any changes to its routing table.

The following argument helps us understand this redirection mechanism.

$$d1 + ar_i < br_i + N \Rightarrow d1 < br_i - ar_i + N$$

The VON node announces that it is at distance  $d1$  from the site. This will force all the access routers to send packets destined for that site to the VON node where they are processed.

After the packets arrive at the VON node, they are forwarded to the border router to which the VON node is connected<sup>1</sup>. These nodes must not send the packets back to the VON. The metric to the external site,  $d1$ , and the interface costs  $ar$  and  $d2$  should be appropriately set.

To prevent packets being redirected from the border router **directly** to the VON

$$d2 + d1 > N \Rightarrow d1 > N - d2$$

To prevent packets being redirected from the border router **indirectly** to the VON (ex: through an access router):

$$ar_i + br_i + d1 > N$$

for all possible paths from the border routers to the VON through access routers. This distance will prevent any border routers from sending packets back to the VON node and cycles are avoided.

Also, from all the links between the VON and the access routers, we choose the link having the maximum metric value ( $ar_{max}$ ) and the minimum metric link of all links between access routers and the border routers ( $br_{min}$ )

Thus the following equations were derived keeping the above constraints in mind:

$$d1 < br_{min} - ar_{max} + N$$

$$d1 > N - d2$$

$$d1 > N - br_{min} - ar_{min}$$

Thus,

$$N - d2 < d1 < br_{min} - ar_{max} + N$$

and

$$N - br_{min} - ar_{min} < d1 < br_{min} - ar_{max} + N$$

---

<sup>1</sup> For a given site, a particular border router is generally chosen as the main egress router. Eg. The egress router for Yahoo & Hotmail may be different.

For all practical purposes,  $d_2 < br_{\min} + ar_{\min}$

Thus,  $N - d_2 > N - (br_{\min} + ar_{\min})$

It is always possible to satisfy the above inequalities if  $ar_1$ ,  $d_1$  and  $d_2$  can be controlled. In order to understand how such metrics can be controlled a better understanding of how the OSPF protocol has been implemented is very much necessary. The following sections explain the OSPF protocol implementation that conforms to the RFC 2328. Certain definitions and relevant sections have been extracted from the RFC.

## **2. The OSPF Protocol:**

OSPF, an interior gateway protocol, is based on link-state or SPF technology. OSPF is a dynamic routing protocol as it keeps status of all the links in the Autonomous System. It computes paths without cycle on any detection of topological changes in the AS.

The convergence of routing information all through the AS takes place quickly and in an optimal manner in terms of the amount of information needed to exchange to achieve the convergence.

Each router holds a link state database (LSDB), which describes the Autonomous System's topology. Adjacent routers have identical databases. A tree of shortest paths with itself as root is derived from the LSDB providing the shortest route to each destination in the Autonomous System. Externally, derived routing information appears on the tree as leaves. An area is defined as a group of networks, the topology of which is hidden from the rest of the AS.

### **2.1 Splitting the AS into Areas**

Each area runs a separate copy of the basic link-state routing algorithm and holds has its own link-state database. Similarly, the internal routers are not aware of the topology outside the area they belong to. Every router maintains a separate link-state database for each area it is connected to.

### **2.2 Classification of routers**

The routers are classified based on the function they perform.

#### *2.2.1 Internal routers*

An internal router is internal to an area running a single copy of the routing algorithm. These routers are attached to no more than one area.

#### *2.2.2 Area border routers*

Area border routers (ABRs) are connected to multiple areas. The link-state databases are identical for all routers belonging to the same area. These run multiple copies of the basic algorithm, one copy for each attached area. The provide summary LSAs (concise topological information of their

attached areas) to the backbone network. The backbone in turn distributes the information to the other areas.

### *2.2.3 Backbone routers*

A router that has an interface to the backbone area. These routers also have an interface to more than one area (i.e., area border routers). However, backbone routers are not necessarily area border routers.

### *2.2.4 AS boundary routers*

AS boundary routers (ASBRs) advertise routing information from other Autonomous Systems all through its AS. They also exchange routing information with routers belonging to other Autonomous Systems. External routes advertised by an ASBR are valid only if the router knows the route to the ASBR itself. An ASBR can be a ABR, internal router or a backbone router.

## **2.3 Use of external routing information**

The external routing information can be another routing protocol's routing information such as BGP, or be statically configured (static routes). External routing information is flooded throughout the AS. There are two types of external metrics: type 1 and type 2.

Type – 1: This metric is used to express OSPF interface cost (i.e., in terms of the link state metric).

Type – 2: This metric is an order of magnitude larger than any Type 1 metric within an AS.

It is important to note that Type 2 external metric is considered to be a major cost of routing a packet while routing between AS'es,. Hence, the external costs are not converted into the internal kind. Type 1 and Type 2 external metrics can be both co-exist in the AS, simultaneously. Type 1 internal costs are considered smaller than any Type 2 cost.

## **2.4 Working of OSPF**

OSPF's Hello Protocol establishes neighborhood with the attached routers. On broadcast and point-to-point networks, the router dynamically detects its neighboring routers by sending its Hello packets to the multicast address AllSPFRouters. On non-broadcast networks, some configuration information may be necessary in order to discover neighbors. A Designated router (DR) for the network is elected in the broadcast and NBMA networks.

The DR originates a network-LSA on behalf of the network and plays a pivotal role in the LSDB synchronization process. The DR is elected based on its router Priority, which is configurable on a per-interface basis. In order to optimize the flooding procedure on broadcast networks, the Designated Router multicasts its Link State Update Packets to the address AllSPFRouters, rather than sending separate packets over each adjacency.

The router forms adjacencies with some of its newly acquired neighbors. Link-state databases are synchronized between pairs of adjacent routers. Each router describes its database by sending a sequence of Database Description packets to its neighbor. Each Database Description Packet describes a set of LSAs belonging to the router's database.

Routing updates are sent and received only on adjacencies. A router periodically advertises its link state. Such advertisements are also sent when a router's state changes. LSAs are flooded throughout the area. The flooding is explained in detail later. OSPF supports five distinct types of LSAs classified based on the functionality they provide with. The collection of LSAs forms the link-state database.

Router-LSAs and network-LSAs describe how an area's routers and networks are interconnected. Summary-LSAs provide a way of condensing an area's routing information. AS-external-LSAs provide a way of transparently advertising externally-derived routing information throughout the Autonomous System.

## **2.5 Originating LSAs**

Each router originates a router-LSA. A DR, additionally, originates network-LSAs for the networks it is designated as a DR. ABR originates a single summary-LSA for each area for which it is an ABR. ASBRs originate a single AS-external-LSA for each external destination. A change in any single route is flooded one at a time without reflooding the entire collection of routes. Many LSAs are carried in a single Link State Update packet during flooding.

In a router-LSA, the B bit indicates if a router is ABR and E indicates if it is an ASBR. Bit B should be set whenever the router is actively attached to two or more areas, even if the router is not currently attached to the OSPF backbone area. Finally, in the LSA, the link output cost is also mentioned. The output cost of a link is configurable. The flooding procedure after a LSA is originated or a

LSA Update packet is received on a particular interface is explained in the following section.

## 2.6 The Flooding Procedure

If there is already a database copy of the LSA, and if the LSA is installed less than `MinLSArrival` seconds ago, the new LSA is discarded (without acknowledging it) and the next LSA (if any) listed in the Link State Update packet is examined.

A LSA is flooded after installation only on eligible interfaces, which is determined based on its LS type.

An AS-external-LSAs is always flooded throughout the entire AS, with the exception of stub areas. The eligible interfaces are all the router's interfaces, excluding virtual links and those interfaces attaching to stub areas.

### Other LS types

The other types of LSAs are exchanged only within a specific area. If the new LSA was received on the interface from either a DR or the Backup then the other routers part of the network could have received the same LSA on a broadcast network.

On broadcast networks, the Link State Update packets are sent as multicast. The destination IP address is `AllSPFRouters` if the interface state is DR or backup. Otherwise, the address, `AllDRouters` is used.

On non-broadcast networks, separate Link State Update packets are unicast, to each adjacent neighbor. The destination IP addresses for these packets are the neighbors' IP addresses.

### AllSPFRouters

The `AllSPFRouters` multicast address is assigned 224.0.0.5. All routers running OSPF receive packets sent to this address. Hello packets and certain other packets sent during the flooding procedure are always sent to this destination

### AllDRouters

This multicast address has been assigned 224.0.0.6. Both the DR and Backup DR receives packets destined to this address.

## 2.7 The OSPF packet

The OSPF packet types are listed below

Type	Packet name	Protocol function
1	Hello	Discover/maintain neighbors
2	Database Description	Summarize database contents
3	Link State Request	Database download
4	Link State Update	Database update
5	Link State Ack	Flooding acknowledgment

OSPF's Hello protocol uses Hello packets to discover and maintain neighbor relationships. The Database Description and Link State Request packets are used in the forming of adjacencies. OSPF's reliable update mechanism is implemented by the Link State Update and Link State Acknowledgment packets.

Each Link State Update packet carries a set of new link state advertisements (LSAs) one hop further away from their point of origination. A single Link State Update packet may contain the LSAs of several routers. Each LSA is tagged with the ID of the originating router and a checksum of its link state contents.

### 3. Modified OSPF daemon:

The implementation of the OSPF daemon had to be modified in order to facilitate the redirection mechanism discussed earlier.

#### 3.1 VON as DR:

For the sake of convenience, the network mode is set to broadcast. As was mentioned earlier, in the broadcast mode, a new LSA packet received on an interface is flooded on all interfaces except the ones whose state is set to DR or backup. If VON has to announce two different interface costs it requires that BR and the AR don't flood to each other after receiving the announcements (Router-LSAs).

VON, if made the DR, then, AR and BR are assured that the other routers part of the network will have reliably received the LS update packets sent by the VON. To enable VON as the designated router it's priority, a configurable

priority, is set to maximum to ensure it always wins the DR election process. For details about DR election process it is recommended that RFC 2328 is consulted.

### 3.2 Disable broadcast to ALLSPFROUTERS:

In order to optimize the exchange of routing information the DR broadcasts LS update packets to a broadcast address, ALLSPFROUTERS. Disabling broadcast for Link State update process multicasts LS update packets to all adjacent routers part of the network.

### 3.3 Router LSA modifications

A router-LSA with both the link state id and the advertising router's id set to VON's router-id announces the interface output cost for the VON. To announce distinct costs to the BRs and ARs, the outgoing packets are subject to modification. If the IP packet's destination is BR, modify the interface cost to  $d_2$  (Refer to figure 1). If the packet is destined to AR set the link state cost to  $ar$ . A router can be determined if it is a ABR from its router LSA. The B bit in the router LSA is used to dynamically determine the area border router.

Modification to LS update packets installs two different router LSAs in the BRs and ARs. The following are the samples of router LSAs in the different routers:

In the following example, 192.168.1.182 is the VON and 192.168.1.183 is the BR. OSPF daemon (broadcast mode) is running on routers 192.168.1.[181-184]. The link states in the BR:

OSPF Router with ID (192.168.1.183)

Router Link States (Area 0.0.0.2)

```
LS age: 35
Options: 2
Flags: 0x2 : ASBR
LS Type: router-LSA
Link State ID: 192.168.1.182
Advertising Router: 192.168.1.182
LS Seq Number: 80000003
Checksum: 0x3b85
Length: 36
Number of Links: 1
```

Link connected to: a Transit Network  
(Link ID) Designated Router address: 192.168.1.182  
(Link Data) Router Interface address: 192.168.1.182  
Number of TOS metrics: 0  
TOS 0 Metric: 11

LS age: 40  
Options: 2  
Flags: 0x3 : ABR ASBR  
LS Type: router-LSA  
Link State ID: 192.168.1.183  
Advertising Router: 192.168.1.183  
LS Seq Number: 80000003  
Checksum: 0x2a93  
Length: 36  
Number of Links: 1

Link connected to: a Transit Network  
(Link ID) Designated Router address: 192.168.1.182  
(Link Data) Router Interface address: 192.168.1.183  
Number of TOS metrics: 0  
TOS 0 Metric: 10

### 3.4 Originate AS External LSA

To announce a virtual cost,  $d_i$ , to the external site an external LSA for each external site is originated in the VON with  $d_i$  as the external metric. For this external LSA to be installed in the adjacent routers, the VON has to be made the ASBR. Fortunately, a router can be both ASBR and DR. While explaining ASBR, it was mentioned that all the routers in the AS know routes to ASBRs. An AS external LSA also specifies the forwarding address which specifies the route needed to take to reach the external site. If the forwarding address is 0.0.0.0, the advertising router (VON in our case) is assigned as the forwarding address.

When the external LSA is installed in the attached routers, an external route to registered site through VON is inserted into the routing table if the LSA

announces a shorter path. It may be noted that, an external LSA is deleted from the LSDB if the external site is actually unreachable through the forwarding address. This requires that we regularly originate the external LSA. The following is the sample of the external LSA installed in the LSDBs of the routers.

OSPF Router with ID (192.168.1.183)

#### AS External Link States

LS age: 150  
Options: 2  
LS Type: AS-external-LSA  
Link State ID: 140.40.40.40 (External Network Number)  
Advertising Router: 192.168.1.183  
LS Seq Number: 80000002  
Checksum: 0x43e7  
Length: 36  
Network Mask: /32  
    Metric Type: 2 (Larger than any link state path)  
    TOS: 0  
    Metric: 20  
    Forward Address: 192.168.1.1  
    External Route Tag: 0

LS age: 150  
Options: 2  
LS Type: AS-external-LSA  
Link State ID: 140.40.40.42 (External Network Number)  
Advertising Router: 192.168.1.183  
LS Seq Number: 80000002  
Checksum: 0x2ff9  
Length: 36  
Network Mask: /32  
    Metric Type: 2 (Larger than any link state path)  
    TOS: 0  
    Metric: 20  
    Forward Address: 192.168.1.1  
    External Route Tag: 0

It is important to note that modifying the LSAs embedded in the outgoing packets introduces a checksum error. The checksums in the LSA header and the OSPF packet header need to be recalculated after modification.

#### 4. Results

In a broadcast network, 192.168.1.182 (vm152) is chosen as VON, which is additionally configured as both ASBR and DR. 192.168.1.183 (vm153) is the BR and has an external route to the external site through the default gateway (Refer to external LSAs). 192.168.1.181 (vm151) and 192.168.1.184 (vm154) are the access routers.

The routing tables of the access routers reflect in the redirection caused by VON for packets destined to an arbitrarily chosen ip addresses, 140.40.40.40/32 and 140.40.40.42/32.

##### 4.1 OSPF routing table at AR:

```

===== OSPF network routing table =====
N 192.168.1.0/24 [10] area: 0.0.0.2
    directly attached to eth0

===== OSPF router routing table =====
R 192.168.1.182 [10] area: 0.0.0.2, ASBR
    via 192.168.1.182, eth0
R 192.168.1.183 [10] area: 0.0.0.2, ABR, ASBR
    via 192.168.1.183, eth0

===== OSPF external routing table =====
N E2 140.40.40.40/32 [10/20] tag: 0
    via 192.168.1.1, eth0
N E2 140.40.40.42/32 [10/20] tag: 0
    via 192.168.1.1, eth0

```

##### 4.2 The kernel routing table at AR:

```

Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
140.40.40.40 vm152 255.255.255.255 UGH 105 0 0 eth0
140.40.40.42 vm152 255.255.255.255 UGH 105 0 0 eth0
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0

```

169.254.0.0	*	255.255.0.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	192.168.1.1	0.0.0.0	UG	0	0	0	eth0

### 4.3 The kernel routing table at VON:

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
140.40.40.40	vm153	255.255.255.255	UGH	0	0	0	eth0
140.40.40.42	vm153	255.255.255.255	UGH	0	0	0	eth0
192.168.1.0	*	255.255.255.0	U	0	0	0	eth0
169.254.0.0	*	255.255.0.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	192.168.1.1	0.0.0.0	UG	0	0	0	eth0

### 4.4 The kernel routing table at ABR:

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
140.40.40.40	192.168.1.1	255.255.255.255	UGH	0	0	0	eth0
140.40.40.42	192.168.1.1	255.255.255.255	UGH	0	0	0	eth0
192.168.1.0	*	255.255.255.0	U	0	0	0	eth0
169.254.0.0	*	255.255.0.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	192.168.1.1	0.0.0.0	UG	0	0	0	eth0

A *traceroute* run on vm154:

```

1  vm152 (192.168.1.182) 0.538 ms 0.362 ms 0.303 ms
2  vm153 (192.168.1.183) 2.849 ms 0.791 ms 0.886 ms
3  192.168.1.1 (192.168.1.1) 1.513 ms 1.087 ms 1.246 ms
4  130.245.14.1 (130.245.14.1) 1.955 ms 1.908 ms 1.768 ms
5  130.245.1.3 (130.245.1.3) 2.193 ms
6  ...

```

### 4.5 Time taken for convergence

Establishing neighborhood, election of DR, establishing adjacency and eventually synchronizing databases was noted to take 27 secs after the VON was enabled in the broadcast network. Once the databases were synchronized the routing table changes were immediately effected. The different phases involved before a route is created are as follows:

#### *4.5.1 Establishing neighborhood:*

Hello packets are broadcasted regularly to establish neighborhood. Hello packet was sent to 224.0.0.5 after 2 secs 18527 usecs. The DR election follows among all the routers and since the priority of the VON is the highest. The VON is elected the DR.

#### *4.5.2 Establishing Adjacency:*

The VON goes through the different states of the Neighbor State Machine (NSM) and finally establishes adjacency. Any LS update packets or DB description packets received before establishing adjacency is ignored by either neighbors. A DB description packet is scheduled to be sent every 5 secs in the daemon. It takes 5 such DB exchanges before the adjacency with all the neighbors was established. The first DB description packet was sent after the neighborhood which is approx. 2secs. Hence, the total time taken for the LSDBs to get synchronized is  $2 + 5 * 5 = 27$  secs.

#### *4.5.3 Installing LSA:*

Once the DB description is received instantly, the eligible LSAs are installed.

#### *4.5.4 Route creation:*

After the LSA is installed the SPF algorithm is scheduled. Hence, a route is created.

In all, it takes the routers approx. 27-28 secs to effect modifications in the routing tables.

## **5. Applications:**

The IP Redirection mechanism can be employed in useful applications like firewalls, load-balancing among servers etc.. It also finds its application in Network Security. The VON could be part of a virtual overlay network that counters a Distributed Denial of Service attack. A VON is attached to each ABR in an Autonomous System. A suitable filtering mechanism can be employed in the VONs that could perform signature based anomaly detection. The different VONs communicate between themselves in a distributed manner and cooperate to counter a denial of service attack. The following is a case-study.

## **6. Case-Study – A virtual overlay network to counter a Distributed Denial of Service attack.**

### **6.1 Filtering at the Virtual Overlay Nodes(VONs):**

The VONs can now analyze all the traffic that is destined to the registered client. A suitable filtering mechanism needs to be employed at the VONs in order to prevent a distributed form of DoS attacks. A naïve way of preventing such an attack is to signal an attack to all the VONs. All the VONs filter packets to an equal extent, good enough to prevent the DDoS from aggravating. But, this in itself is a DoS, because the registered client is unable to serve the genuine packets. This also implies a loss in business.

To effectively filter packets the genuine packets have to be distinguished from the malicious ones. Unfortunately, when normal workstations are compromised the packets need not be anomalous. The known techniques of anomaly detection can not be applied in our case. But, if there should be an attack then, the malicious traffic has to be growing at acceleration greater than the genuine traffic. The growth in the malicious traffic is due to either the growth in the compromised victims( ignorantly, sending the malicious packets ) or the growth in the rates

from the already compromised victims. In either case, the acceleration from the ASs, to which they belong to, should be high. Intuitively, the arrival rate at a VON is increasing when it is, with high probability, forwarding malicious traffic. Hence, acceleration, at the time of attack, is a very good metric for dropping packets. This is discussed in further detail later where we actually compute the percentage by which the VONs need to drop the packets. We'll notice that the VONs drop packets to the extent to which it has seen an increase in the arrival rate during the previous instant of time. But, this requires VONs to change in arrival rates relatively with respect to other nodes. Hence, VONs need to communicate between themselves and arrive at a solution. In the following section we describe a suitable structure for the overlay network in order to reduce the communication overhead.

## 6.2 VON Tree:

When a registered client reaches a threshold  $C_{\text{thresh}}$  (threshold arrival rate or the capacity) for the border router it signals the VONs of such an occurrence. Broadcasting, this to all VONs around the world requires multicasting this message to all VONs. A bigger problem is to know all the VONs, which is not scalable in terms of the number of VONs. For every new installation of VON each registered client should be intimated. Another disadvantage is when the VONs need to communicate between each other. Clearly, for each node to talk to each other  $n * (n-1) / 2$  communications are required. This is a very costly and a cumbersome approach for a plenty of well known reasons.

A viable solution is to arrange the VONs into a logical tree. This entails the border router to know only the VON at the first level. Besides, the VONs need to communicate only with the siblings, parent and their children. This, indeed, is scalable as one sub-tree need not worry about the changes in another sub-tree if they don't overlap. We group VONs into logical groups depending on their

terrestrial distribution. Each such group contains a designated VON, DVON. A group of DVONs contains another DVON as their parent and this process could recursively carry all the way upto the root to create a logical VON tree structure. All the nodes in this tree except the leaf nodes are DVONs.

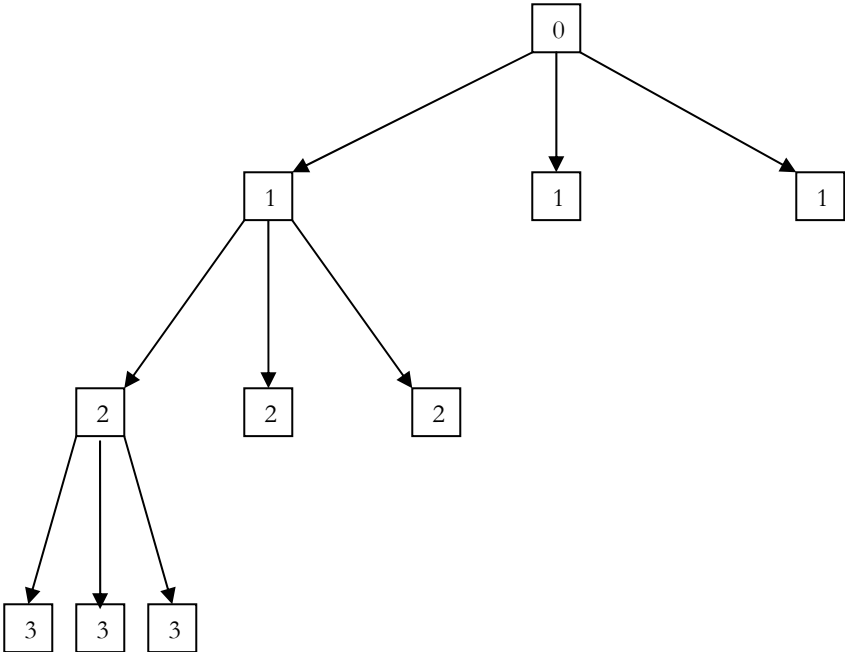


Figure 2: VON Tree

Node marked 0 is the border router of the registered client. Nodes marked 1 and 2 are DVONs. Nodes(leaves) marked 3 are VONs. It should be understood that DVONs are VONs themselves with some additional responsibility of helping the children coordinate. An analogy is the presence of the current directory (/.) within a directory in UNIX file systems.

**6.3 Formation of VON tree:**

Every DVON talks to its parent and its children. Hence, these nodes should be laid out such that communication overhead is minimal in terms of cost between any two adjacent links (say, number of hops). This can be achieved if the total cost incurred by the tree is minimal. Hence, a VON tree, at best, should be a

**minimum-spanning tree (MST)**,  $T_{\min}(V,E)$ , where,  $V$  is the number of nodes and  $E$  is the number of edges in the **VON graph**. Now, every node in the tree should be aware of its parent and its children in  $T_{\min}(V,E)$ . The next few sections focus on the steps involved in the formation of the VON tree.

#### **6.4 Construction of MST:**

All VONs are interconnected through the Internet. Hence, the overlay network is a fully connected graph,  $G(V,E)$ . Every such edge is associated with the cost incurred by a packet to traverse the edge. The cost could be physical distance, time, hop count, etc. In order to minimize the communication overhead between the VONs a MST of  $G(V,E)$ ,  $T_{\min}(V,E)$  needs to be constructed. The MST could be constructed using the Prim's or the Kruskal's algorithm. But, this entails the nodes to have global information namely, a priority queue.

The construction of a MST could be done centrally or in a distributed manner. The latter technique is fault-tolerant and does not cause congestion at any particular node. But, to concurrently perform en(de)queue operations the nodes need to acquire exclusive locks on the queue. Besides, for all the nodes to individually maintain a global entity( the queue itself) they need to be constantly in sync. For every en(de)queue operation all the nodes need to be informed. This is certainly very expensive.

A simpler solution involves performing this computation centrally. With the knowledge of the costs of all the edges in the VON graph, a central **VON Tree Manager (VTM)** announces to each VON in the graph, its relative position in the MST. VTM helps a VON to find its relative position in the tree by informing the VON's parent and the set of children. A major advantage of a centralized arrangement is that VTM is free to implement an algorithm of its choice and the VONs do not have to worry about how the tree is formed. Any modifications to

the graph or the MST can now be centrally updated and this change could be easily reflected in the topology. We will study the working of the VTM in more detail in the following topic.

### **6.5 VON Tree Manager (VTM):**

As a centralized server, VTM owns the responsibility of finding a MST from the VON graph and broadcasting this information to all the nodes in the graph. In order to compute the MST the VTM has to be aware of the edge costs of the graph,  $G(V,E)$ .

Since,  $G$  is a fully-connected graph, it needs to maintain  $O((\#V)^2)$  entries in its database. But, it is sufficient for every node to know only the cost it takes to reach its physical neighbors. This is because if a VON is not in the physical neighborhood of the VON in question, then, the cost incurred for a packet to traverse the link,  $L$ , between the two VONs is higher than the any of the costs to traverse the links between VONs in the physical neighborhood. Hence,  $L$  can not figure in the MST. Of course, this proof is based on the assumption that link costs are proportional to the physical distance between the VONs. But, this is not always the case. There could be differences in bandwidth capacities in two different routes. Hence, we proceed with an approximate MST, hoping that the overall communication cost is very small if not the least.

Hence, we could consider a graph,  $G'(V,E')$  with  $E'$ , a subset of  $E$ , such that  $e$  belongs to  $E'$  only if the nodes connected by  $e$  are in physical neighborhood. Two VONs are neighbors to each other if, the ASs to which they are associated with, are neighbors (adjacent) themselves. Every VON sends the link costs to all its neighbors to the VTM. Hence, the VTM needs to maintain the costs for the graph  $G'$  which has far lesser entries than  $G$ . Only one of the two neighbors needs to announce the cost of the edge connecting them. The resulting adjacency

graph is, indeed, a sparse matrix, which can be optimally stored as a database. Besides, this matrix is symmetric and hence, only half the entries need to be actually stored.

The VTM, using the cost matrix, computes the MST. It announces this information to all the nodes in the graph,  $G' (V,E')$ . In order, to avoid congestion at the VTM, the announcements can be done batch-wise. This does not harm as the whole process of tree formation is, rather, infrequent. Occasionally, but periodically, the VTM can request the nodes for the costs, update its database, compute MST and broadcast this information.

As mentioned earlier, the VTM identifies the parent and the children for every VON. In this manner, the VON tree is generated. If now, a VON goes down or is removed; its neighbor will inform the edge cost to this VON to be  $\infty$ . This edge will no longer figure in the MST. When a new VON is added, the costs to its neighbors are sufficient to update the VTM database. Since, this cost is symmetrical ( $G'$  being undirected) the neighbors need not explicitly inform (or be informed) the VTM about this change. Hence, the neighbors (and hence, the VON graph) can be transparent of changes in the topology.

With the VON tree in place, we are now in a position to perform percentage computation for packet filtering at the VONs.

### **6.6 Percentage computation at the VONs:**

After an attack has been signaled to all the VONs, every DVON computes the percentage of packets to be dropped by its children during the next time instant. As discussed earlier, the policy we choose to drop the packets should be fair in preserving genuine packets and dropping packets that could potentially cause a

DDoS attack. Thus, any increase in the arrival rates at the VONs during a DDoS attack, with good probability, can be assumed to be caused by the attack traffic. If genuine packets also increase at this instant we play hard on them too, as in a DDoS attack, a spurious packet can be no different from a genuine packet. Hence, our only metric for percentage computation will be the increase in the arrival rates at instant of time of a potential attack which we term as *acceleration*. We now derive a formula to compute the threshold arrival rate for a VON. Within a time instant, if the packets arrive beyond the threshold capacity for that time instant, they are dropped (head filtering).

To compute the percentages, the DVON needs the following:

1.  $C_{\text{thresh}}$ , the threshold arrival rate at the DVON, which it obtains from its parent.
2.  $C_{\text{curr}}$ , the current rates at which packets arrive at the children, which it obtains from its children.

Conventions and symbols:

$C$  – Arrival rate at a VON

$a$  – acceleration at a VON

$C_{\text{thresh}}$  – threshold arrival rate at a VON

$a_{\text{thresh}}$  – threshold arrival rate at a VON

The above physical quantities are a function of *depth* and *instant*. Depth refers to the depth of the VON in the VON tree. Instant refers to the cycle in the sequence of computations. For instance,  $C$  (parent, current) refers to the arrival rate at the parent during the current instant of time.

At a VON, at any instant of time, the percentage computation is done as follows:

1.  $a_{\text{thresh}}(\text{parent,current}) = (C_{\text{thresh}}(\text{parent,current}) - C(\text{parent,current})) / T$
2.  $a(\text{parent,current}) = (C(\text{parent,current}) - C(\text{parent,previous})) / T$
3.  $a(\text{child,current}) = (C(\text{child,current}) - C(\text{child,previous})) / T$
4.  $a_{\text{thresh}}(\text{child,next}) = a_{\text{thresh}}(\text{parent,current}) * (a(\text{child,current}) / a(\text{parent,current}))$
5.  $C_{\text{thresh}}(\text{child, next}) = C(\text{child,current}) + a_{\text{thresh}}(\text{child,next}) * T$

Symbol	Meaning
T	Time period for the cycle of computations.
$C_{\text{thresh}}(\text{parent,current})$	The sum of the threshold arrival rates of all the children at the current instant of time.
$C(\text{parent,current})$	The sum of the arrival rates of all the children at the current instant of time.
$a_{\text{thresh}}(\text{parent,current})$	The sum of the threshold arrival accelerations at all the children at the current instant of time.
$C(\text{parent,previous})$	The sum of arrival rates of all its children at the previous instant of time.
$a(\text{parent,current})$	The sum of the accelerations at the parent at this instant of time which is also the sum of accelerations of all its children at this instant of time.

$C(\text{child,current})$	Arrival rate at the child at this instant of time
$a(\text{child,current})$	Acceleration at the child at this instant of time
$a_{\text{thresh}}(\text{child,next})$	Threshold acceleration at the child at the next instant of time
$C_{\text{thresh}}(\text{child, next})$	Threshold arrival rate at the child at the next instant of time

Table 1

Explains symbols used in the derivation

The input to the algorithm are  $C_{\text{thresh}}(\text{parent,current})$  and  $C(\text{parent,current})$ . The former is obtained from the parent. The latter needs the children to report their current arrival rates. The sum of the arrival rates at the children is  $C(\text{parent,current})$  which the parent computes and broadcasts to all its children. It should be noted that this computation is performed only when the arrival rate exceeds the threshold i.e.) if  $a_{\text{curr}}$ , computed at the first step is negative. If not, the rest of the steps are not computed because the DVON is forwarding below its threshold.

The steps 1-5 are performed by every VON taking the input parameters  $C_{\text{thresh}}(\text{parent,current})$  and  $C(\text{parent,current})$  from its parent. The latter is in fact the sum of the current arrival rates of all the children of that parent. Thus, every VON successfully derives a threshold rate for itself. Packets that arrive after  $C_{\text{thresh}}(\text{child, next}) * T$  are now dropped at the child during the time period 'T'.

The other problem with this computation is the requirement of physical quantities during the previous instant. This will delay the computation by another time period which could prove costly and dangerous. This delay needs to be accommodated while choosing the  $C_{\text{thresh}}$  for the registered client.

The synchronization is not achieved across levels. All nodes within a level are synchronized, but, across levels the communication is done asynchronously.

An example:

Referring to Fig 1, the node marked 0 is the border router of the registered client.

Instant	Level	Sender	Receiver	Message
0	0	0	1	$C_{\text{thresh}}(0,0), C(0,0)$
1	1	1	2	$C_{\text{thresh}}(1,1), C(1,1)$
1	0	0	1	$C_{\text{thresh}}(0,1), C(0,1)$
2	2	2	3	$C_{\text{thresh}}(2,2), C(2,2)$
2	1	1	2	$C_{\text{thresh}}(1,2), C(1,2)$
2	0	0	1	$C_{\text{thresh}}(0,2), C(0,2)$

Table 2

The pipeline table indicating the communication at 0<sup>th</sup>, 1<sup>st</sup> and 2<sup>nd</sup> instants of time.

The whole process takes a pipelined structure and all the levels operate concurrently. It may be noticed that the registered client participates in this process at each instant of time sending its current threshold rate to the node at the first level. It is advantageous to announce its threshold capacity at each instant of time as it could act as a control mechanism for distributing the percentages all through the tree. The whole tree could be imagined as a black box performing the DDoS prevention. Varying  $C_{\text{thresh}}$  for every instant could give the client the flexibility of choosing the stringency in packet filtering by the box.

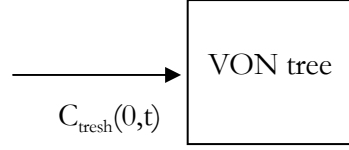


Figure 3: Input to VON tree

### 6.7 The choice of $C_{\text{thresh}}$ :

The registered client maintains a threshold arrival rate,  $C_{\text{thresh}}$  at its border routers, which indicates that the arrival rate has reached its capacity. This also indicates that client is already under attack<sup>1</sup>. To avoid such a situation, the choice of the threshold rate should account for the delay between the announcements to all VONs and the instant when the actual packet filtering happens to suppress the attack. Hence, the border router maintains another parameter  $C_{\text{pseudo-thresh}}$  which is lesser than  $C_{\text{thresh}}$ . The border router announces the VONs that it has reached  $C_{\text{thresh}}$  when it actually  $C_{\text{pseudo-thresh}}$ . Let  $t_{\text{attack}}$  be the time it takes to reach  $C_{\text{thresh}}$  from  $C_{\text{pseudo\_thresh}}$  and  $a_{\text{max}}$  is the maximum acceleration that could possibly be expected. Then,  $C_{\text{pseudo\_thresh}} = C_{\text{thresh}} - a_{\text{max}} * t_{\text{attack}} * \sum$   
 Even if the estimate is wrong, the  $C_{\text{thresh}}$  for the next instant of time can be adjusted to compensate for the error caused during the previous instant.

---

1 – The word ‘attack’ is a misnomer. It is used because even if the border router has exceeded its capacity by merely genuine packets(with no attacks) this could still be considered an attack.

### References:

- 1] Moy, J., "OSPF Version 2", [RFC 2328](#), April 1998.
- 2] Moy, J., "OSPF Version 2", [RFC 1583](#), March 1994.
- 3] Moy, J., "OSPF Version 2", [RFC 2178](#), July 1997.
- 4] A Client Oriented, 1P Level Redirection Mechanism by Sumit Gupta and A. L. Narasimha Reddy [[http://www.iecee-infocom.org/1999/papers/10d\\_03.pdf](http://www.iecee-infocom.org/1999/papers/10d_03.pdf)]