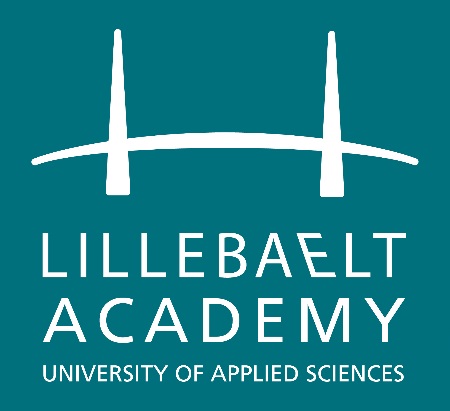
**IT Technology**

**Project Network report**



LILLEBAELT ACADEMY

UNIVERSITY OF APPLIED SCIENCE

Author

Michal Skorczewski

Martin Grønholdt

1st June 2017

**Table of Contents**

[1 Introduction 4](#_Toc484169604)

[1.1 Introduction 4](#_Toc484169605)

[1.2 Project Plan 4](#_Toc484169606)

[1.3 Responibilities 5](#_Toc484169607)

[2 Ethernet L2 & L3 Switching 5](#_Toc484169608)

[2.1 VLAN Implementation 7](#_Toc484169609)

[2.2 VLAN Trunking 7](#_Toc484169610)

[2.3 VLAN L3-Interface 8](#_Toc484169611)

[2.4 Virtual Routers 8](#_Toc484169612)

[2.5 Rthernet OAM 9](#_Toc484169613)

[2.6 Troubleshooting & Monitoring 9](#_Toc484169614)

[3 Intermediate Routing 12](#_Toc484169615)

[3.1 Ipv6 12](#_Toc484169616)

[3.2 OSPF (all of configurations are located in Appendix) 12](#_Toc484169617)

[3.2.1 OSPF 12](#_Toc484169618)

[3.2.2 OSPF areas 12](#_Toc484169619)

[3.2.3 Designated Router 13](#_Toc484169620)

[3.2.4 Quick configuration example 13](#_Toc484169621)

[3.2.5 Sources: 15](#_Toc484169622)

[3.3 IS-IS 16](#_Toc484169623)

[3.4 Route Re-Distribution (OSPF/IS-IS) 17](#_Toc484169624)

[3.5 BGP 18](#_Toc484169625)

[3.5.1 BGP 18](#_Toc484169626)

[3.5.2 AS Path and Attributes 19](#_Toc484169627)

[3.5.3 Internal BGP: (Full configuration in Appendix) 22](#_Toc484169628)

[3.5.4 External BGP: (Full configuration in Appendix) 24](#_Toc484169629)

[3.5.5 Problems with BGP 24](#_Toc484169630)

[3.5.6 Route Reflector 25](#_Toc484169631)

[3.5.7 Proofs, monitoring and troubleshooting 25](#_Toc484169632)

[3.6 Route Re-Distribution (BGP/OSPF) 27](#_Toc484169633)

[4 Security 28](#_Toc484169634)

[4.1 Routing Policies 28](#_Toc484169635)

[4.2 Route Redistribution 29](#_Toc484169636)

[4.3 RE/PFE 30](#_Toc484169637)

[4.4 Firewall Filters 31](#_Toc484169638)

[4.5 CoS 31](#_Toc484169639)

[5 Conclusion 31](#_Toc484169640)

# Introduction

## Introduction

## Project Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Task Name | Duration | Start | Finish |
| **Project Network 2. Semester** | **31.14 days?** | **Mon 1/9/17** | **Mon 5/8/17** |
| **Stage-1: Ethernet L2 & L3 Switching** | **9 days** | **Mon 1/9/17** | **Tue 2/14/17** |
| VLAN Implementation | 1 day | Mon 1/9/17 | Tue 1/10/17 |
| Vlan Trunking (802.1q) | 1 day | Tue 1/10/17 | Mon 1/16/17 |
| VLAN L3-Interface | 1 day | Mon 1/16/17 | Tue 1/17/17 |
| Virtual Routers (SRX & EX) | 2 days | Tue 1/17/17 | Tue 1/24/17 |
| Ethernet OAM | 1 day | Tue 1/24/17 | Mon 1/30/17 |
| Troubleshooting & Monitoring | 1 day | Mon 1/30/17 | Tue 1/31/17 |
| **Stage-2: Intermediate Routing** | **15 days** | **Tue 1/31/17** | **Mon 4/3/17** |
| IPv6 | 2 days | Tue 1/31/17 | Tue 2/14/17 |
| OSPF | 2 days | Tue 2/14/17 | Tue 2/21/17 |
| IS-IS | 2 days | Tue 2/21/17 | Tue 2/28/17 |
| Route Re-Distribution (OSPF/IS-IS) | 3 days | Tue 2/28/17 | Tue 3/7/17 |
| BGP (iBGP & eBGP w. OSPF) | 3 days | Tue 3/7/17 | Mon 3/20/17 |
| Route Redistribution (BGP/OSPF) | 3 days | Tue 3/21/17 | Mon 4/3/17 |
| **Stage-3: Security** | **12 days** | **Mon 4/3/17** | **Tue 5/16/17** |
| Routing Policies | 2 days | Mon 4/3/17 | Mon 4/10/17 |
| Route Redistribution | 2 days | Mon 4/10/17 | Mon 4/17/17 |
| RE/PFE | 2 days | Mon 4/17/17 | Tue 4/25/17 |
| Firewall Filters | 3 days | Tue 4/25/17 | Mon 5/8/17 |
| CoS | 3 days | Mon 5/8/17 | Tue 5/16/17 |
| **Finalize Report** | 4 days | Tue 5/16/17 | Mon 6/5/17 |
| ***Project End/Hand-in*** | 0 days | Mon 6/5/17 | Mon 6/5/17 |

## Hardware

## Responibilities

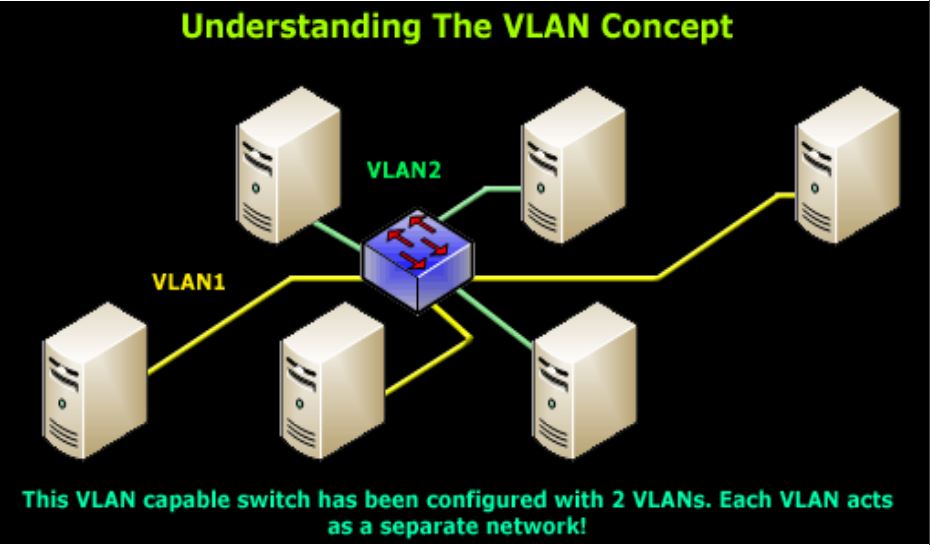
|  |  |
| --- | --- |
| Vlan Implementation | Michal Skorczewski |
| Vlan Trunking (802.1q) | Michal Skorczewski |
| Vlan L3- Interface | Michal Skorczewski |
| Vurtual Routers | Michal Skorczewski |
| Ethernet OAM | Michal Skorczewski |
| Troubleshooting & Monitoring | Michal Skorczewski |
|  |  |
| Ipv6 | Martin Gronholdt |
| OSPF | Martin Gronholdt |
| IS-IS | Martin Gronholdt |
| Route Re-distribution OSPF/IS-IS | Martin Gronholdt |
| BGP | Michal Skorczewski |
| Router Redistribution BGP/OSPF | Michal Skorczewski |
|  |  |
| Routing Policies | Michal Skorczewski |
| Route Redistribution | Martin Gronholdt |
| RE/PFE | Michal Skorczewski |
| Firewall Filters | Martin Gronholdt |
| CoS | Michal Skorczewski |

# Ethernet L2 & L3 Switching

Ethernet LANs were originally designed for small networks that primarly carried text, over years the type of data carried by LANs grew to include voice, graphics, and video. This complex data, when combined with the speed of transmission became too much of a load for the original Ehernet LAN design, packet collisions were slowing down larger LANs. The IEEE 802.1D-2004 standard helped evolve LANs to cope with the higher data and transmission requirements by defining the concept of bridging.

* Bridging divides a single physical LAN (now broadcast domain) into two or more virtual LANs
* By default, system on one VLAN don’t see the traffic associated with systems on other VLANs on the same network

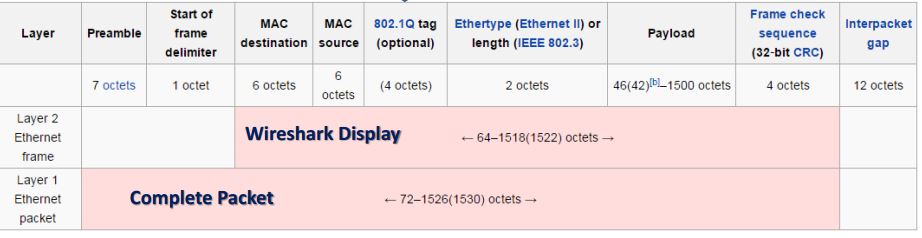
IEEE 802.1Q is the standard defining VLANs. Each VLAN is identified by a unique 802.1Q ID, only IDs 1 through 4094 can be assigned to VLANs during configuration, IDs 0 and 4095 are reserved by Junos OS and cannot be assigned.



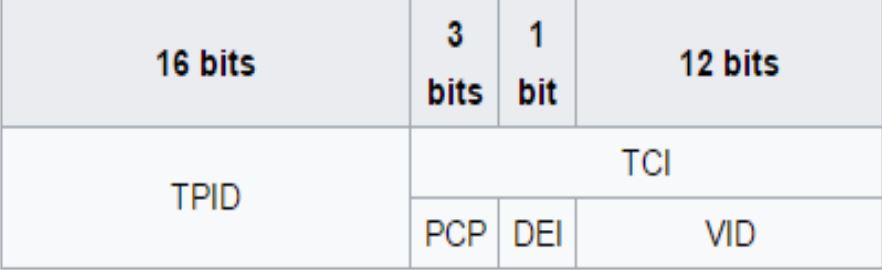
Ethernet packets includes:

* Tag protocol identifier
* EtherType field, which identidies the protocol being transported. When a device within a VLAN generates a packet, this field includes a value of 0x8100, which indicates that the packet is a VLAN-tagged packet.
* The packet also has a VLAN ID field that includes the unique 802.1Q ID, which identifies the VLAN to which the packet belongs

The Ethernet Frame + 802.1q Tag



The “layer 1 Ethernet Packet” is what is transmitted over the wire, bit-by-bit.  
The “layer 2 Ethernet Frame” is the display on monitor interface.



The 802.1q tag:

Tag protocol ID (TPID): a 16 bit field set ro a value of 0x8100

Tag control information (TCI)

* Priority code point (PCP): 3 bit field which refers to the IEEE 802.1p class of service and maps to the frame priority
* Drop eligible indicatior (DEI): 1 bit field, may be used separately or in conjunction with PCP to indicate frames eligible to be dropped in the presence of congestion
* VLAN identifier (VID) a 12 bit field specifying the VLAN to which the frame belongs

## VLAN Implementation

Step 1: Create a layer 2 vlan

*set vlans <vlan-name> vlan-id <vlan-id>*

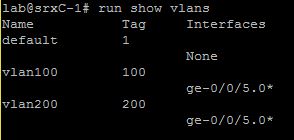
Step 2: Create a logical layer 3 VLAN interface:

*set interfaces vlan unit <unit> family inet address <ip address/mask>*

Step 3: Link the layer 2 VLAN to the layer 3 VLAN interface:

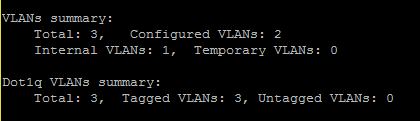
*set vlans <vlan-name> l3-interface vlan.<unit mentioned above>*

The result in project after implementing vlans can be displayed using command: *show vlans.*



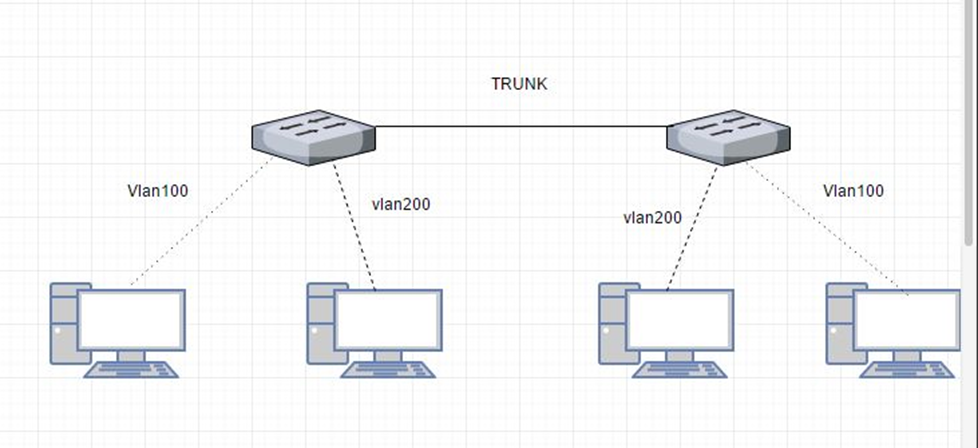
Other command that can be used to display configured vlans is: *show vlans summary.*

In the display, untagged and tagged vlans are shown.



## VLAN Trunking

Trunk mode interfaces are used to connect switches to one another. Traffic sent between switches can then consist of packets from multiple VLANs, with those packets multiplexed so they can be sent over the same physical connection.



The trunk interface is a switched interface, have to have a corresponding interface on a second switch.

QUICK CONFIGURATION:

*set interfaces <interface> unit <unit number> family ethernet-switching port-mode access vlan members <vlan-name>*

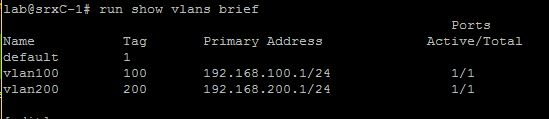
Only VLANs named in members <vlan-name> have access over the Trunk.

## VLAN L3-Interface

In order to configure the switch to perform L3 switching it is necessary to assign VLAN interface to VLAN using command:

*Set vlans <vlan name> l3-interface vlan.<vlan number>*

This is how it looks when the VLAN is implemented with L3 interface.



## Virtual Routers

In Junos Software, a virtual router is a routing instance type. It is a collection of routing tables, interfaces and routing option settings. Routing instance virtual router can act like a normal router, with policies and routing options. It allows to isolate traffic without using multiple routing devices to segment network.

To establish a virtual router it is necessary to follow few steps.

* Create a virtual router
* Assign an interface to a virtual router
* Assign an interface to a zone

It is possible to assign other routing option to virtual router.

To share routes in more than one routing instance it is optional to select physical or logical connection.

Physical connection is a normal interface (for example ge-0/0/0 or so-0/0/0) it can be established using cables or VMNets in VMware.

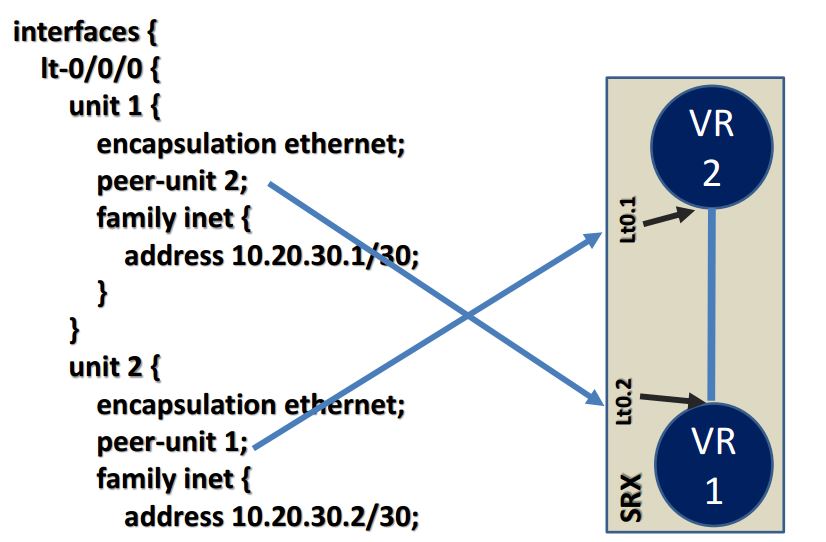
## **Logical Tunnels**

Logical tunnels (lt0 interfaces) can be used only on SRX Juniper devices.

To connect two routing instances with a logical connection logical tunnel interface should be configured for each instance. Then, it is mandatory to configure a peer relationship between the logical tunnel interfaces, thus creating a point-to-point connection. To create a point-to-point connection logical tunnel has to configured using the lt-fpc/pic/port format.

Each logical tunnel interface should be configured with a proper encapsulation type.

It is important to configure only one peer unit for each logical interface. (Unit 0 cannot peer with both unit 1 and unit 2.)



## Ethernet OAM

Ethernet Operations, Administration, and Maintenance

Ethernet OAM is a set of tools that network manager use to know the way how Ethernet links are working. Ethernet OAM should:

* Rely only on the media access control (MAC) address or virtual local area network (VLAN) identifier for troubleshooting
* Work independently of the actual Ethernet transport and function over physical Ethernet ports, or a virtual service such as pseudowire, and so on.
* Isolate faults over a flat (or single operator) network architecture or a nested or hierarchical (or multi-provider) networks.

## Troubleshooting & Monitoring

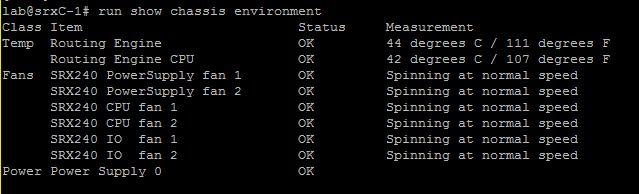
**Troubleshooting**

Juniper SRX device provides set of commands that can be used for troubleshooting. Mostly it is giving an opportunity to view log files, environment of router and alarms.

Few troubleshooting commands:

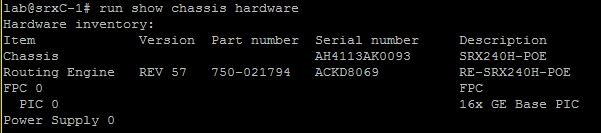
* show chassis environment

Command above allows to display temperatures, fans and power supply on SRX device.



* show chassis hardware

Using command above causes display hardware informations like serial numbers, part numbers and version.



* show chassis alarms

This command is used for displaying information about alarms in real time.

alarms

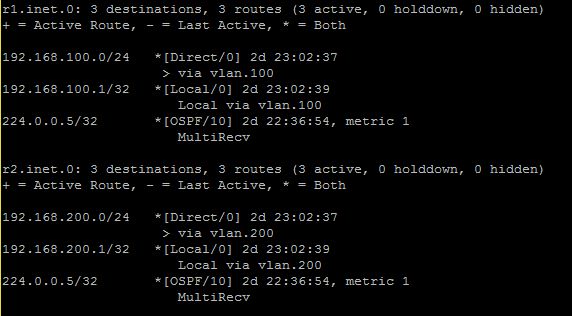
Troubleshooting commands above were used on Juniper SRXC-1 in schools lab.

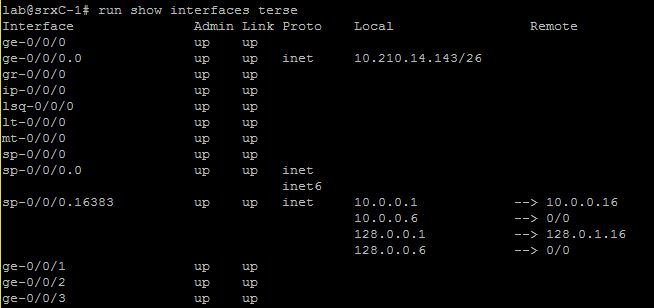
**Monitoring**

Except troubleshooting Juniper gives command to monitor interfaces, traffic and routes on device.

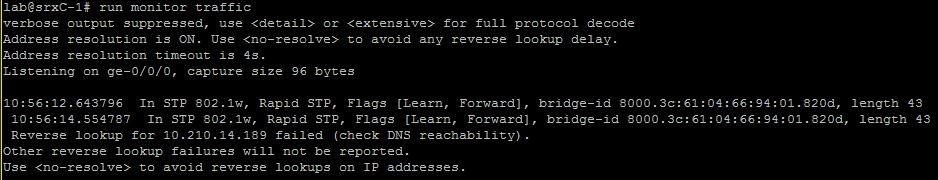
Most common command for monitoring routes in router is: *show route.*

Where next-hops, routes, preferences and protocols.



*show interfaces terse* allows to see admin and link state of interfaces as well as protocol its using and addresses.

In Juniper Devices it is possible to monitor traffic inside the router, using *monitor traffic.*



# Intermediate Routing

## IPv6 (all of configurations are located in Appendix)

IPv6 is the most recent version of the IP protocol an important change from IPv4 is that IPv6 uses 128-bits for address space instead of the 32-bit address space of IPv4. This is important since the IPv4 address pool is all but exhausted. IPv4 provides about 4.29 billion addresses which means that far from every person on earth can have a device with an IPv4 address.

## IPv6 headers

The IPv6 simpler is simpler than that of IPv4 (though larger in size), which allows for faster processing. The IPv6 header is fixed in size at 40 bytes and includes options as extension headers that are only there when actually used. This makes the design easier to extend for future additions.

|  |  |  |  |
| --- | --- | --- | --- |
| **1 byte** | **1 byte** | **1 byte** | **1 byte** |
| Version | Traffic Class | Flow Label | |
| Payload length | | Next Header | Hop Limit |
| Source Address | | | |
| Destination Address | | | |

Illustration 1: The IPv6 fixed header format.

Referring to Illustration 1 above this is a description of the header fields:

* **Version:** Protocol version number = 6.
* **Traffic Class:** The 6 most-significant bits are used fto classify packets. The remaining 2 bits are used in order to signal impending congestion.
* **Flow Label:** Used to label sequences of packages that are to be treated the same way by allowing for more efficient processing by routers.
* **Payload Length:** Length of the payload limited to 64K y the size of this field. IPv6 can use an extension called Jumbo frames to send larger payloads. The length counts everything after the fixed header, included the extension headers.
* **Next Header:**
  + If the next header is UDP or TCP, this field will contain the same protocol numbers as in IPv4, for example, protocol number 6 for TCP or 17 for UDP. The protocol numbers are available through IANA at <https://www.iana.org/assignments/protocol-numbers/>
  + If extension headers are used, this is the type of the next extension header.
* **Hop Limit:** Decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero.
* **Source Address:** 128-bit address of the source of the packet.
* **Destination Address:** 128-bit address of the intended recipient of the packet, this might be the ultimate recipient if a Routing header is present.

Below is an illustration of the extension headers and payloads organisation, coming after the fixed IPv6 header.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fixed header | Extension header 1 | Extension header ... | Extension header *N* | Payload |

Illustration 2: IPv6 fixed, and extension -header followed by the payload.

## IPv6 addressing

IPv6 addresses are either unicast, anycast, or multicast addresses

* Unicast addresses identify a single network interface and the IP packages will be sent to that interface.
* Anycast addresses are assigned to a group of interfaces. Anycast addresses has the same format as unicast addresses and differ only by the fact that they are present on more than one interface in the network. An anycast packet is delivered to one member, typically the nearest host according to the routing protocols definition.
* Multicast addresses are assigned to a group of interfaces, each member processes a message send to a multicast address.

An IPv6 address has 128 bits and address is divided into eight 16-bit hexadecimal blocks

separated by colons. For example:

2001:0DB8:0000:0000:00C0:FEFE:BADA:5500

There are some rules for shortening addresses like this:

* Leading zeroes can be skipped.
  + 2001:DB8:0:0:C0:FEFE:BADA:5500
* A double colon can replace a string of consecutive zeroes in more than one 16-bit address block, but a double colon can only be used once in the address
  + 2001.DB8::C0:FEFE:BADA:5500

Beacuse of the transition from IPv4 to IPv6 going on a special syntax has been introduced where the least 32 bits of an address can be written in the familiar dot notation of IPv4:

::ffff:c000:0280 and ::ffff:192.0.2.128 is the same, but the second form will be more recognisable if you have ever worked with IPv4.

### Special addresses

* ::/128 – The unspecified address that corresponds to the IPv4 0.0.0..0/32 address. This address must never be assigned to an interface, but is for instance used to get software to listen for incoming connections on all interfaces
* ::/0 - The default route address that corresponds to the IPv4 0.0.0.0/0 in IPv4.
* ::1/128 – The loopback address.
* Fe80::/10 – This is a link-local address and compares to the auto-configuration type address of IPv4. The last 64 bits are usually chosen as the interface hardware address in modified EUI-64 format, basically adding ff:ee in the middle of the 48-bit MAC address. Addresses in the link-local prefix are only valid and unique on a single link.
* Fc00::/7 - Unique local addresses (ULAs), these addresses are comparable to IPv4 private addresses (10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16). Like their IPv4 counterparts they are routable only within a set of cooperating sites. There are provisions in this range for doing addresses that will most likely not clash when merging more networks.
* ::ffff:0:0/96 - This prefix is designated as an *IPv4-mapped IPv6 address*.
* 64:ff9b::/96 - Addresses with this prefix are used for automatic IPv4/IPv6 translation.
* 2002::/16 - This prefix is used for 6to4 addressing.
* 2001::/32 — Used for Teredo tunneling, a transition technology that gives full IPv6 connectivity for IPv6-capable hosts that are on the IPv4 Internet, it functions even from behind NAT devices
* 2001:db8::/32 — This prefix is used in documentation.

Juniper configuration.

This is a basic configuration setting IPv6 addresses on a logical tunnel.

interfaces {

lt-0/0/0 {

#vSRX-1

unit 11 {

encapsulation ethernet;

peer-unit 21;

family inet6 {

address fdaa:dead:beef:1::1/127;

}

}

#vSRX-2

unit 21 {

encapsulation ethernet;

peer-unit 11;

family inet6 {

address fdaa:dead:beef:1::1/127;

}

}

}

## Sources:

Wikipedia - Open Shortest Path First - <https://en.wikipedia.org/wiki/Open_Shortest_Path_First>

The Internet Engineering Task Force – RFC2460 - <https://tools.ietf.org/html/rfc2460>

IANA – Protocol number - <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>

EAL - IPv6 Essentials Addressing

## OSPF (all of configurations are located in Appendix)

### OSPF

OSPF is short for Open Shortest Path First and is the name of a routing protocol. OSPF is an interior gateway protocol which means that it is used to exchange routing information within an autonomous system. IOSPF Version 2 was defined in RFC2328 in 1998 and Version 3 in RFC5340 in 2008. Version 3 is an update to support IPv6.

OSPF forms IP datagrams directly and packages them using protocol number 89 and implements its own transport layer error detection and correction functions. OSPF uses multicast addressing for distributing routing information within a broadcast domain.

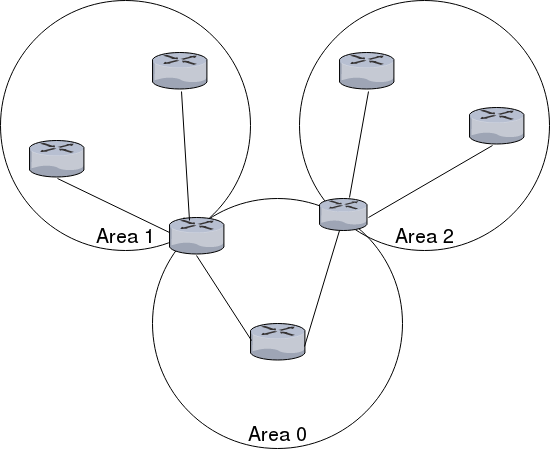
Routers running OSPF communicate with neighbouring routers on connected interfaces to establish the state of connections:

* **Down:** Initial state of the connection indicating that no resent communication has been received.
* **Init:** A HELLO packet has been received from a neighbour but the routers have not established two-way communication.
* **Exchange:** The router is sending its link state database to the neighbour in database description packets. Each packet has a sequence number that is explicitly acknowledged.
* **Loading:** The router requests the most recent link-state advertisements from its neighbor discovered in the Exchange state.
* **Full:** The end state when all adjacent routers has reached the Full state and the link state database of o the neighbours are fully synchronized.

### OSPF areas

Areas in OSPF are used to administratively group networks and host in an AS together, areas are identified by 32-bit numbers. The topology of an area is unknown outside that area

An example of a network split into areas are shown in Illustration 1. The routers fully inside the areas (circles) are called internal routers, these are all connected to devices inside the same area. The routers on the borders between to areas are called area border routers or ABRs. Area 0 has a special role as the backbone area that distributes routes between areas. All ABRs are connected to the backbone, and the backbone area most be contiguous, if not physically, by using virtual links. The backbone has no ABRs and the routers in area 1 has to go through area 0 to talk to routers in area 2. It is the backbones job to redistribute routing information between the other areas.

  
Illustration 1: OSPF areas

### Designated Router

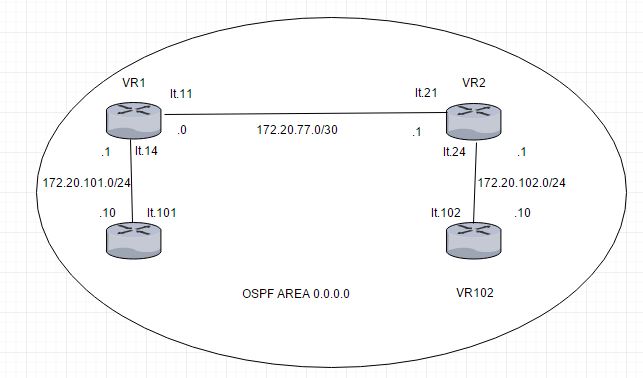
To not load down the network with routing traffic in large networks OSPF uses designated routers. Routers in the same network sends their link state information to the designated router. The designated router send the link advertisements on behalf of the network and participates in synchronising the link state database by establishing adjacencies. The designated router is found through election.

* The router priorities are evaluated and the router with the highest priority is selected as the designated router.
* If there is more than one router with the same priority, the one with the highest router identifier is chosen.
* If the no router ids are configured the election will go by the IP address of the first interface that comes online. This is usually the loopback interface.
* If nothing of the above, the first hardware interface with an IP address will be used for the election.

By default routing devices has a priority of 128. The priorities work like this:

* 0: the router will not be considered in the election.
* 1: the router has the least chance of being elected.

### Quick configuration example

  
Illustration 2: Network diagram og the OSPF example configuration.

The above diagram illustrates an example configuration of virtual router to use OSPF. Connection between the virtual routers are connected using logical tunnels.

This is the routing instance designated “VR1” in the illustration above:

routing-instances {

gangstin {

The logical tunnel interfaces and loopback interface are set up according to the illustration above.

instance-type virtual-router;

interface lt-0/0/0.11;

interface lt-0/0/0.14;

interface lo0.1;

Next comes the actual OSPF configuration. This configuration includes all interfaces in area 0, the backbone area.

protocols {

ospf {

area 0.0.0.0 {

interface lt-0/0/0.11;

interface lt-0/0/0.14;

interface lo0.1;

}

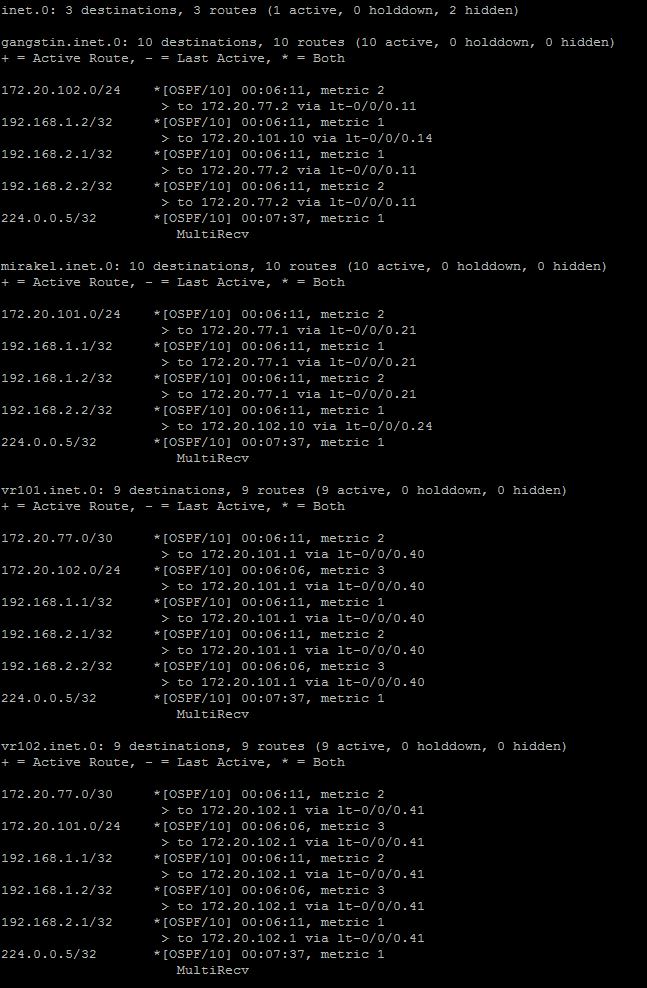
}

}

}

}

This configuration is mirrored on each virtual router, except of course the interfaces change according to the digram above.



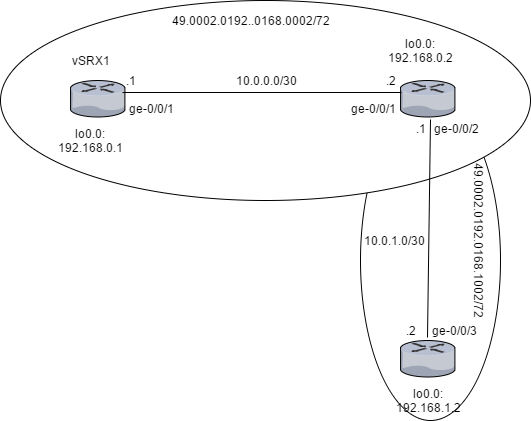
### Sources:

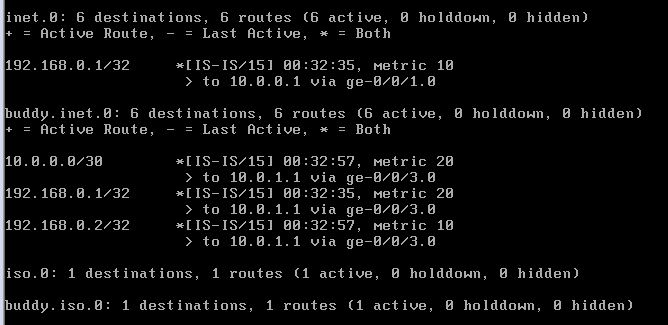
Wikipedia - Open Shortest Path First - <https://en.wikipedia.org/wiki/Open_Shortest_Path_First>

Juniper website - Understanding OSPF Areas - <https://www.juniper.net/documentation/en_US/junos/topics/concept/ospf-routing-understanding-ospf-areas-overview.html>

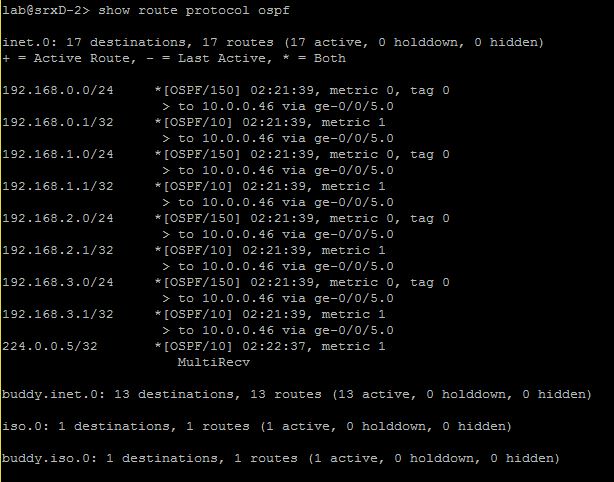
EAL - OSPF/Open Shortest Path First presentation

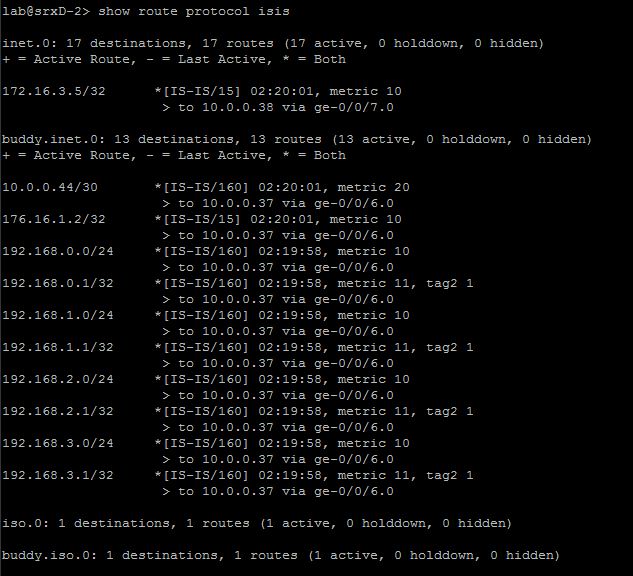
## IS-IS





## Route Re-Distribution (OSPF/IS-IS)

****

****

## BGP

### BGP

Is an exterior gateway protocol, uses TCP on port 179 to establish connection between the routers. It is used to exchange information between routers in different autonomous systems. Routing information of BGP includes the complete route to each destination.

Border Gateway protocol exchange information about network reachability with other BGP systems. It uses network reachability information to create a graph of AS connectivity, which enforce policy decisions at the Autonomous System level. There are two options in connections between networks, it can be private point-to-point link or an exchange.

### BGP Message Types

BGP has four types of messages, each with own role in setting up, maintaining, or tearing down a BGP session. List of messages types:

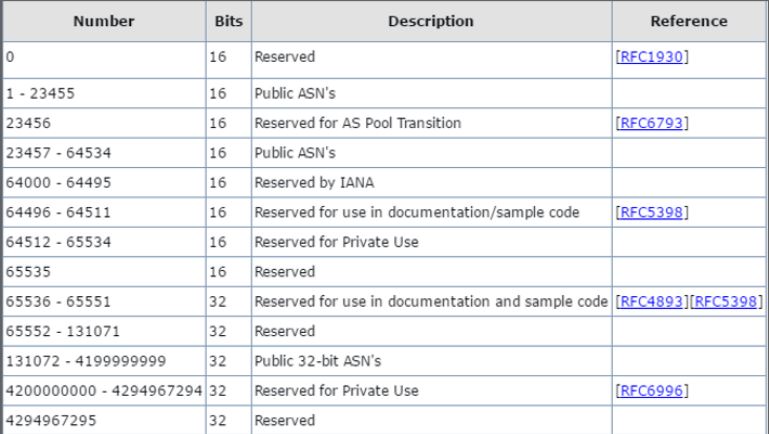
* OPEN message
* UPDATE message
* KEEPALIVE message
* NOTIFICATION message

Important information is that these messages cannot be exhanged until two BGP routers have set up TCP session on port 179. Errors will display BGP NOTIFICATION messages that will close connecion.

### Autonomous Systems (AS)

Autonomous system is a group of router that are under single technical administration. Normally use a common set of metrics to share routing information with the set of routers.

Until 2007 AS were defined as a 16-bit integers, because of the size of Internet and number of devices we are running out of AS numbers. That is why in 2007 32 bit AS numbers were introduced. It allows us to use numbers from 0 to 4 294 967 295.



### AS Path and Attributes

* BGP systems exchange routing information which include complete route to each destination and additional information about the route.
* AS path is the name of the route to each destination, additional route information is included in path attributes.
* BGP uses AS paths and path attributes to determine topology of the network.
* If BGP knows the topology, it can detect and eliminate routing loops, as well as selecting groups of routes to enforce preferences and policy decisions.

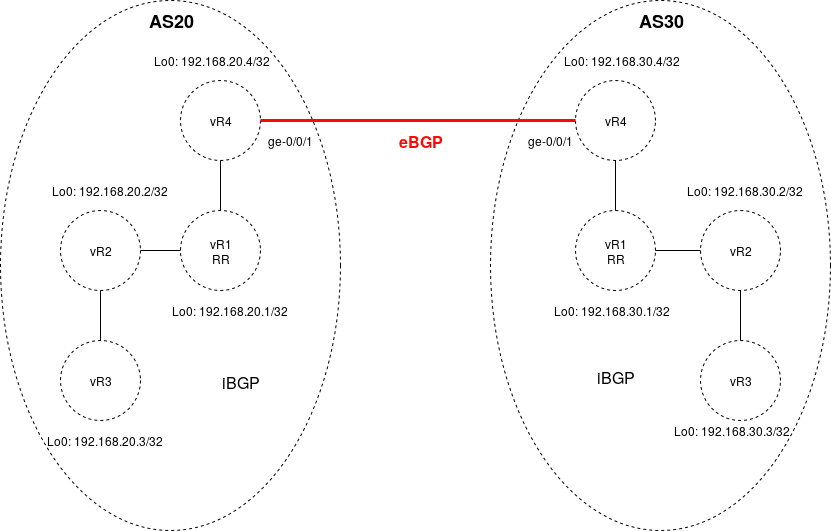
BGP uses modes to communicate with internal and external peers, two primary modes are:

* Internal BGP (iBGP)
* External BGP (eBGP)

Internal BGP runs inside one Autonomous System, when External works just between multiple ASs.

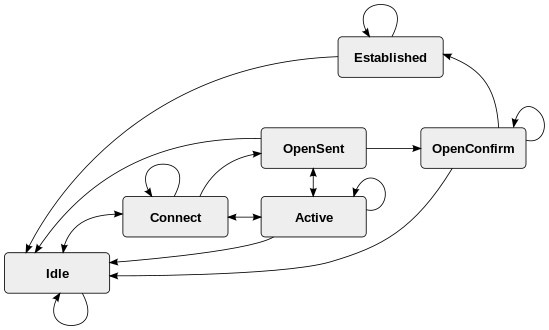
Peer Ass establish links through and external peer BGP session. Then all route advertisement between external peers takes place by means of information exchange.

BGP network used in project:



BGP uses a Finite State Machine to make decisions. It consists of six states:

* Idle;
* Connect;
* Active;
* OpenSent;
* OpenConfirm;
* Established.



**Idle State:**

Router is searching for a routing table to check if a route to neighbor already exist.

**Connect:**

Router found a route to BGP neighbor. Three-way TCP handshake is completed.

**OpenSent:**

Open message with BGP session parameters is sent.

**OpenConfirm:**

Router received agreement for establishing session.

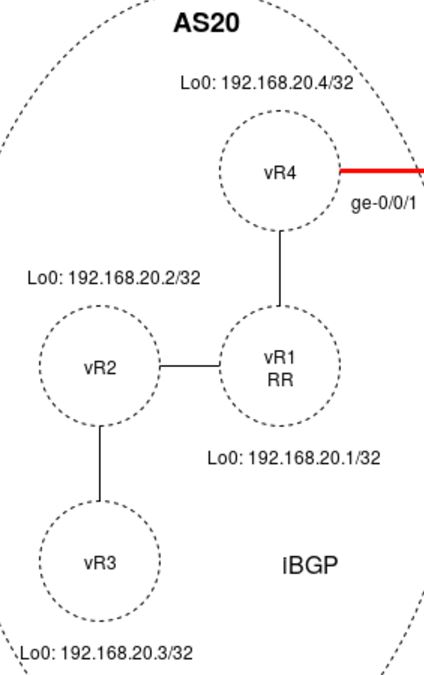
**Active:**

Router didn’t recive agreement for establishing session .

**Established:**

Peering connection is up, routing process begins.

### Internal BGP: (Full configuration in Appendix)

****

In the network used in project all devices in Autonomous System 20 are meshed in the group internal-peers. Each device has configured loopback address, logical tunnel according to connection between each device and router-id which is requirement in order to use BGP and OSPF in a routing instance.

Example from device vR4:

vR4 {

instance-type virtual-router;

interface lt-0/0/0.6;

interface ge-0/0/1.0;

interface lo0.4;

routing-options {

router-id 192.168.20.4;

autonomous-system 20;

}

protocols {

bgp {

group internal-peers {

type internal;

local-address 192.168.20.4;

export [ send-direct send-ospf ];

neighbor 192.168.20.1;

neighbor 192.168.20.2;

neighbor 192.168.20.3;

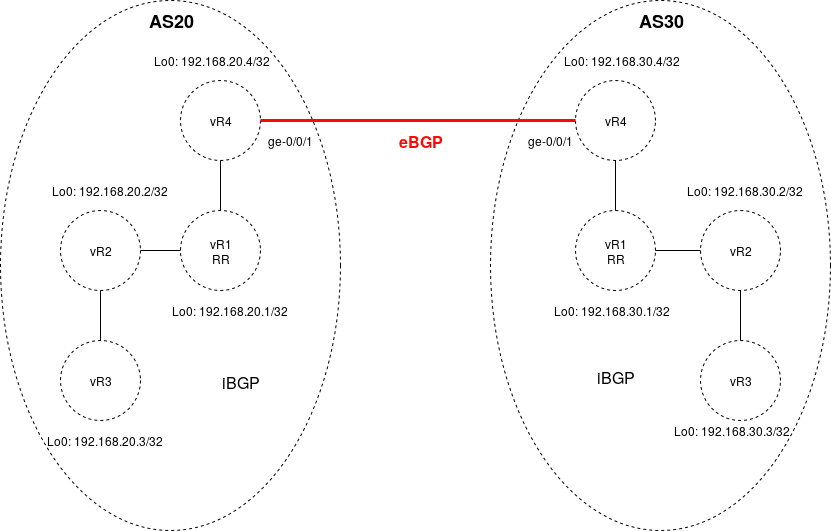
}

It is possible to display the router-id values of a routing instances by using *show route instance detail.*

*root@vSRX1> show route instance detail*

|  |
| --- |
| *vR1:* |
|  | *Router ID: 192.168.20.1* |
|  | *Type: virtual-router State: Active* |
|  | *Interfaces:* |
|  | *lt-0/0/0.2* |
|  | *lt-0/0/0.1* |
|  | *lo0.1* |
|  | *Tables:* |
|  | *vR1.inet.0 : 34 routes (13 active, 0 holddown, 8 hidden)* |
|  |  |
|  | *vR2:* |
|  | *Router ID: 192.168.20.2* |
|  | *Type: virtual-router State: Active* |
|  | *Interfaces:* |
|  | *lt-0/0/0.4* |
|  | *lt-0/0/0.3* |
|  | *lo0.2* |
|  | *Tables:* |
|  | *vR2.inet.0 : 20 routes (13 active, 0 holddown, 3 hidden)* |
|  |  |
|  | *vR3:* |
|  | *Router ID: 192.168.20.3* |
|  | *Type: virtual-router State: Active* |
|  | *Interfaces:* |
|  | *lt-0/0/0.5* |
|  | *lo0.3* |
|  | *Tables:* |
|  | *vR3.inet.0 : 19 routes (12 active, 0 holddown, 0 hidden)* |
|  |  |
|  | *vR4:* |
|  | *Router ID: 192.168.20.4* |
|  | *Type: virtual-router State: Active* |
|  | *Interfaces:* |
|  | *lt-0/0/0.6* |
|  | *ge-0/0/1.0* |
|  | *lo0.* |
|  | *Tables:* |
|  | *vR4.inet.0 : 21 routes (13 active, 0 holddown, 1 hidden)* |

### External BGP: (Full configuration in Appendix)



In network used in project device vR4 in AS20 and vR4 in AS30 has BGP peer sessions to a group of peers called external-peers:

Example from device vR4 in AS20:

*group external-peers {*

*type external;*

*advertise-peer-as;*

*neighbor 172.20.5.2 {*

*peer-as 30;*

*}*

*}*

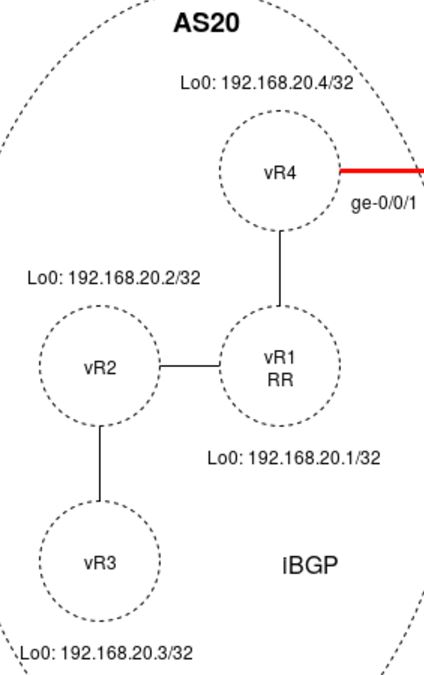
It has defined neighbor with router-id in AS30.

### Problems with BGP

* iBGP scalability – each device has to be configured with peers to each other in a full mash (everyone speak directly to everyone). In a huge networks it can degrade performance of router due to memory or CPU requirements
* Instability – when the device is misconfigured it may get into cycle between up states and down states. It is known as a route flapping, is causes that other devices are informed about broken route and then broken device is removed from routing table
* Routing table growth – Internet infrastructure is growing all the time, and if the older routing devices are not able to handle huge number of routes, it can create a problem that the device will not be a good gateway between the parts of the internet.
* Load-balancing problem – balance in multi-homed networks due to limitations of the BGP peers. This may cause that one link is congested when other links are optimized well.
* Security issues – by default BGP routers accept BGP routes from other devices, it means that Internet is potentially vulnerable for disruptions. BGP is vulnerable for IPhijacking.

### Route Reflector

**Route Reflector** is a device in internal BGP which has peer relationships with all iBGP devices, but also has configured the cluster statement and a cluster identifier. This cause that device vR4 is peered with just vR1 not with all devices in iBGP, the same situation happens in vR2 and vR3, where peering is configured just with route reflector.

****

Route Reflector is configured both in AS20, as well as in AS30.

### Proofs, monitoring and troubleshooting

Command *show bgp group* shows information about groups including, ASs, policy options, peers, addresses, routing tables.

|  |
| --- |
| *root@vSRX1> show bgp group* |
|  | *Group Type: Internal AS: 20 Local AS: 20* |
|  | *Name: internal-peers Index: 0 Flags: <Export Eval>* |
|  | *Export: [ send-direct send-ospf ]* |
|  | *Options: <Cluster>* |
|  | *Holdtime: 0* |
|  | *Total peers: 3 Established: 3* |
|  | *192.168.20.4+54693* |
|  | *192.168.20.2+179* |
|  | *192.168.20.3+179* |
|  | *vR1.inet.0: 3/24/16/0* |
|  |  |
|  | *Group Type: Internal AS: 20 Local AS: 20* |
|  | *Name: internal-peers Index: 1 Flags: <Export Eval>* |
|  | *Export: [ send-direct send-ospf ]* |
|  | *Holdtime: 0* |
|  | *Total peers: 1 Established: 1* |
|  | *192.168.20.1+54090* |
|  | *vR2.inet.0: 3/10/7/0* |
|  |  |
|  | *Group Type: Internal AS: 20 Local AS: 20* |
|  | *Name: internal-peers Index: 2 Flags: <Export Eval>* |
|  | *Export: [ send-direct send-ospf ]* |
|  | *Holdtime: 0* |
|  | *Total peers: 1 Established: 1* |
|  | *192.168.20.1+57784* |
|  | *vR3.inet.0: 3/10/10/0* |
|  |  |
|  | *Group Type: Internal AS: 20 Local AS: 20* |
|  | *Name: internal-peers Index: 3 Flags: <Export Eval>* |
|  | *Export: [ send-direct send-ospf ]* |
|  | *Holdtime: 0* |
|  | *Total peers: 3 Established: 1* |
|  | *192.168.20.1+179* |
|  | *192.168.20.2* |
|  | *192.168.20.3* |
|  | *vR4.inet.0: 0/7/6/0* |
|  |  |
|  | *Group Type: External Local AS: 20* |
|  | *Name: external-peers Index: 4 Flags: <>* |
|  | *Options: <AdvertisePeerAs>* |
|  | *Holdtime: 0* |
|  | *Total peers: 1 Established: 1* |
|  | *172.20.5.2+179* |
|  | *vR4.inet.0: 2/3/3/0* |
|  |  |
|  | *Groups: 5 Peers: 9 External: 1 Internal: 8 Down peers: 2 Flaps: 0* |
|  | *Table Tot Paths Act Paths Suppressed History Damp State Pending* |
|  | *vR1.inet.0 24 3 0 0 0 0* |
|  | *vR2.inet.0 10 3 0 0 0 0* |
|  | *vR3.inet.0 10 3 0 0 0 0* |
|  | *vR4.inet.0 10 2 0 0 0 0* |
|  | *vR1.mdt.0 0 0 0 0 0 0* |
|  | *vR2.mdt.0 0 0 0 0 0 0* |
|  | *vR3.mdt.0 0 0 0 0 0 0* |
|  | *vR4.mdt.0 0 0 0 0 0 0* |

Command *show bgp summary* displays number of groups, peers up and down, ASs, and state

|  |
| --- |
| *root@vSRX1> show bgp summary* |
|  | *Groups: 5 Peers: 9 Down peers: 2* |
|  | *Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...* |
|  | *172.20.5.2 30 10 11 0 0 3:11 Establ* |
|  | *vR4.inet.0: 2/3/3/0* |
|  | *192.168.20.1 20 14 12 0 0 2:25 Establ* |
|  | *vR2.inet.0: 3/10/7/0* |
|  | *192.168.20.1 20 12 11 0 0 2:17 Establ* |
|  | *vR3.inet.0: 3/10/10/0* |
|  | *192.168.20.1 20 9 12 0 0 2:05 Establ* |
|  | *vR4.inet.0: 0/7/6/0* |
|  | *192.168.20.2 20 11 14 0 0 2:25 Establ* |
|  | *vR1.inet.0: 0/7/4/0* |
|  | *192.168.20.2 20 0 4 0 0 3:43 Active* |
|  | *192.168.20.3 20 10 12 0 0 2:17 Establ* |
|  | *vR1.inet.0: 0/7/7/0* |
|  | *192.168.20.3 20 0 4 0 0 3:43 Active* |
|  | *192.168.20.4 20 12 10 0 0 2:05 Establ* |

https://en.wikipedia.org/wiki/Border\_Gateway\_Protocol

## Route Re-Distribution (BGP/OSPF)

# Security

## Routing Policies

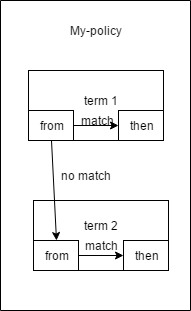
Routing Policy (all of configurations are located in Appendix)

Routing Policy allows user to control the flow of routing information between the routing protocols and the routing tables and between the routing tables and the forwarding table. Routing policy allows you to control which routes the routing protocols store in and retrieve from the routing table.

Routing policy was used because:

* It was not an intention to import all routes into routing table. Routes were specified by proper match criteria in terms.
* According to route redistribution It was intended to transfer active routes learned from another routing protocol (BGP/OSPF, IS-IS/OSPF)

Routing Policy Flow:



Number of terms in routing policy is equal or bigger than 0, the software evaluates terms until it reaches a terminating action or end policy. Names of policies and terms are defined by user.

Each term contains “from” and “then” statement. The first describe match condition(s) and the second one describe action that is taken if a “from” statement is matched.

Match criteria that was used in project:

* from protocol direct
* from protocol ospf
* route-filters (192.168.0.0/22 longer; 10.0.0.44/30 exact; 10.0.0.36/30 exact)

“from” statement describes match conditions

Match types used in project: longer, exact.

* Exact match the specified prefix and mask exactly (10.0.0.36/30 exact)
* Longer match routes that have longer masks (192.168.0.0/22 longer)

Actions

“then” statement describes the actions to take if a “from” statement is matched.

Just one terminating action was used in project, it was “accept”

Implementing Routing Policy.

Defined routing policy is always located under the [edit policy-options] hierarchy on Juniper Device. (configuration in Appendix)

Quick set-up:

[edit policy-options]

set policy-options policy statement ospf-isis term 1 from protocol ospf

set policy-options policy statement ospf-isis term 1 from route-filter 192.168.0.0/22 longer

set policy-options policy statement ospf-isis term 1 then accept

|  |  |
| --- | --- |
| policy-options { |  |
|  | policy-statement ospf-isis { |
|  | term 1 { |
|  | from { |
|  | protocol ospf; |
|  | route-filter 192.168.0.0/22 longer; |
|  | } |
|  | then accept; |
|  | } |
|  | } |

Applied routing policies as import or export policies can be found at different hierarchy levels (for example under routing-instances or protocols)

[edit protocols]

set isis export ospf-isis

set isis interface ge-0/0/7.0

set isis interface lo0.0

|  |  |
| --- | --- |
| protocols { |  |
|  | isis { |
|  | export [ ospf-isis send-direct-to-isis-neighbors ]; |
|  | interface ge-0/0/7.0; |
|  | interface lo0.0; |
|  | } |

Sources:

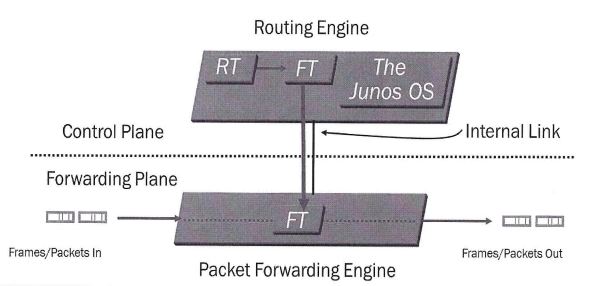
Routing Policy presentation

https://www.juniper.net/documentation/en\_US/junos/information-products/pathway-pages/config-guide-policy/config-guide-policy.html

## Route Redistribution

## RE/PFE

All platform running the JunosOS share a common design goal: clean separation of control and forwarding functions.



Routing Engine is located in control plane, it is the brain of the Juniper Device, responsible for performing protocol updates and system management, it runs various deamons that reside inside a protected memory environment. The Routing Engine maintains the routing tables, bridging table and forwarding table and conntects to the Packet Forwarding Engine through an internal link. The RE provides the CLI in addition to the J-Web GUI.

The Packet Forwarding Engine is responsible for forwarding transit traffic through the device. In many Juniper platforms the PFE uses application-specific integrated circuits (ASICs) for increased performance.

The PFE receives the layer 2 and layer 3 forwarding table (FT) from Routing Engine. FT updates are a high priority for the Junos OS kernel and are performed incrementally. It implements various services such as policing, stateless firewall filtering, and class of service

Transit Traffic consists of all traffic that enters an ingress network port, is compared against the forwarding table entries, and is forwarded out an egress network port toward its destination.

Exception Traffic does not pass through the local device but rather requires some form of special handling. Examples of exception traffic:

* Packets addressed to the chassis (telnet, pings traceroutes)
* Traffic that requires the generation of ICMP messages

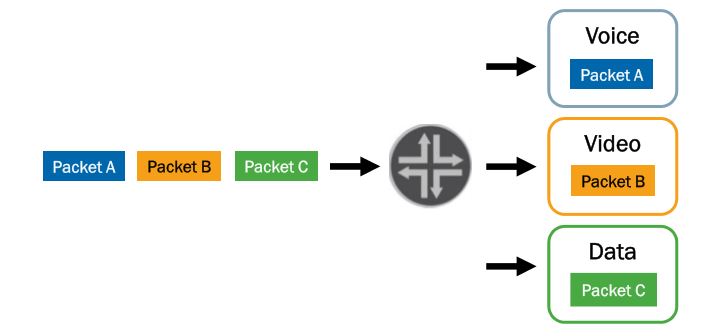
Exception traffic is rare-limited on the internal link to protect the RE from potential DoS attacks

Source: Introduction to the Junos Operating System – Student Guide Revision V-15.a

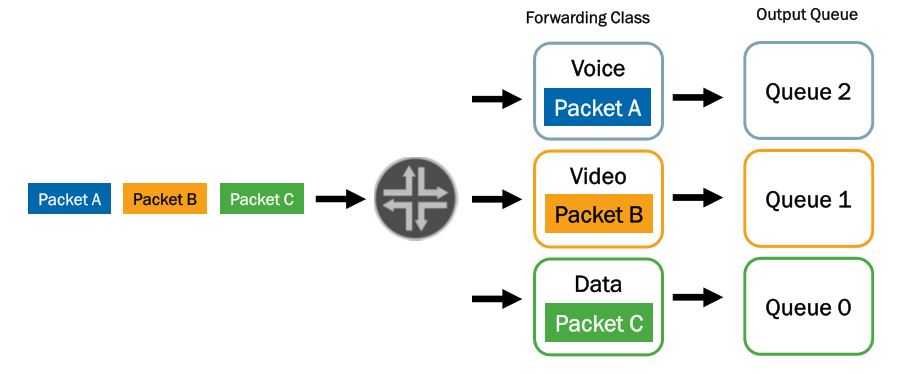
## Firewall Filters

## CoS

### Class of Service

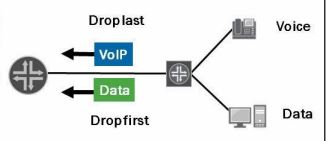
Class of service provides to the device mechanisms for categorizing traffic with different types of payload. The classification is based mostly on the type of application, which can be divided in three main categories voice, video, data.

CoS prioritize latency-sensitive traffic (for example Voice over Internet Protocol), it can allocate bandwidth of connection for different classes. Class of Service to assign an output to a packet, it has to associate packet with the forwarding class. Forwarding classes also can identify traffic that should receive proper treatment.



### Loss Priority

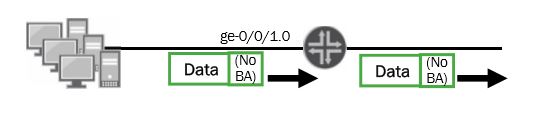
Loss priority can be configure if user want to tell the system which priority it should give to packet to choose what should be dropped at first.



### CoS Deployment Models

There are two prime deployment models:

1. Multifield classifier is used in in-the-box model

It is defined under [edit firewall family inet] hierarchy with “from” and “then” statement and then applied under [edit interfaces <interface> unit <unit number> family <family name> filter] with *input/output <name of classifier>*

1. In across-the-network model Behavior Aggregate rewrite, and BA classifier is in the core and multifield classifier at edge.

Behavior Aggregate rewrite is defined under [edit class-of-service]

interfaces {

ge-0/0/2 {

scheduler-map my-sched-map;

unit 0 {

rewrite-rules {

inet-precedence default;

}

}

}

While BA Classifier is defined also under [edit class-of-service]

interfaces {

ge-0/0/3 {

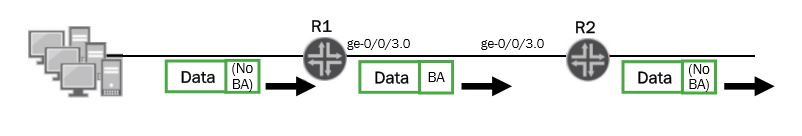
scheduler-map my-sched-map;

unit 0 {

classifiers {

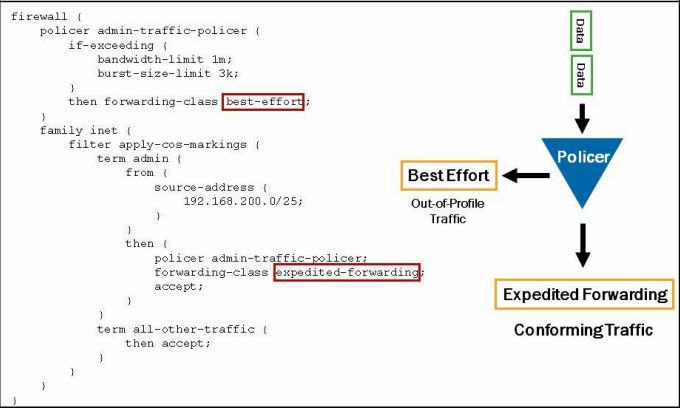
inet-precedence default;

}



### Policers

Policers allows to limit traffic to a specified bandwidth and burst size. It is possible to configure Junos device to assign forwarding class or loss priority to traffic exceeding the configured imits. These policers can be configured using *forwarding-class* and/or *loss-priority* statement after “then” statement of policer.



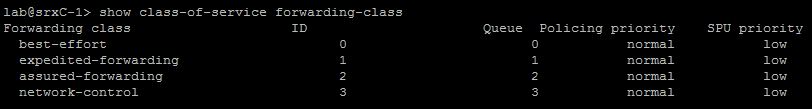
### Queuing

Juniper devices are associating each forwarding class with queue number. On most devices that runs Junos OS default queues mapping looks like:

* 0: best-effort
* 1: expendited-forwarding
* 2: assured-forwarding
* 3: network-control

It is possible to display current queue and forwading class mapping using *show class-of-service forwarding-class*

Display:



### Defining Forwarding Classes

Forwarding classes can be configured under [edit class-of-service forwarding-class], using command

*Set forwarding-classes queue <number> <name>*

# Conclusion

## Conslusion about Project

## Project Management

During the project we have been working together on configurations.

# Sources?

Sources:

Vlan presentation from Peter

<http://searchnetworking.techtarget.com/definition/virtual-LAN>

<https://kb.juniper.net/InfoCenter/index?page=content&id=KB11000>

<https://www.juniper.net/documentation/en_US/junos/topics/reference/configuration-statement/l3-interface-bridging.html>

<https://www.juniper.net/documentation/en_US/junos/topics/task/configuration/bridging-vrf-qfx-series-cli.html>

<https://kb.juniper.net/InfoCenter/index?page=content&id=KB21260>

https://www.juniper.net/documentation/en\_US/junos12.3/topics/concept/layer-2-802-1ag-ethernet-oam-overview-mx-solutions.html