HOW TO READ A 128T CONFIGURATION

Abstract

When troubleshooting a 128T router or conductor it is best to understand the intent of the configuration so you can know where to look to resolve the issues. However, when you issue a `show config running` or `show config candidate` or view the configuration using the Configuration Explorer, the amount of information returned can be overwhelming. With this guide, we will describe a methodology for reading a configuration that will help to understand the intent of the architect who set up your 128T devices.

30 May 2019

I. SERVICES

Services make up the heart of the 128T data model. The 128T data model is built upon the idea that one should build their network around the applications that they will be accessing and not determine which applications their network can access based off of the way their network is set up. In the 128T data model, a `service` is a traffic destination being accessed by constituents in your network.

For this reason, it is best to start your investigation of a 128T configuration by looking at the services the network is configured to deliver.

```
service  video
    name          video
    enabled       true
    security      encryption_only

    transport     tcp
        protocol   tcp

        port-range  8554
            start-port  8554
        exit
    exit

    transport     udp
        protocol  udp
    exit
    address       172.16.128.2/32

    access-policy   corp
        source  corp
    exit
    service-policy  prefer_broadband
exit
```

The `address` and `transport` fields indicate the IP address(es), transport protocols, and ports that make up this service. The `access-policy` elements indicate the user populations (typically defined as the name of a `tenant`) that are allowed access to this service.

Note: The naming convention you choose when modeling your network is very important when you are going back to look at what you configured. Giving your services meaningful names can remove a lot of the guesswork as to what the service you defined actually is – particularly for those that support this configuration in the future.

II. SERVICE POLICIES

Your Service may be configured with a Service Policy. A Service Policy tells each 128T how to handle traffic destined for that Service. It will define things such as whether the Service will be load balanced and with what strategy, whether session resiliency is set up, and selecting path preference with Vectors. The Service Policy will give you a better understanding of how important a Service is to a deployment and what paths that traffic will take to get to that Service.

```
service-policy  prefer_broadband
    name                      prefer_broadband
    lb-strategy               hunt

    vector                    internet
        name      internet
        priority  ordered
    exit

    vector                    mpls
        name      mpls
        priority  ordered
    exit
    session-resiliency        none
    path-quality-filter       false
    best-effort               true
    transport-state-enforcement  reset
```

III. VECTORS

Defined in the Service Policy, vectors let you set path priority with your 128T. You will see the Vectors configured at the Neighborhoods which are sub-elements of the Network Interface. The Service Policy is where you can determine the priority of each Vector when the 128T is deciding which path to send traffic down and thus which Network Interfaces to use.

```
network-interface  mpls1
    name                mpls1
    global-id           3
    vlan                0
    type                external
    conductor           false

    neighborhood          mpls
        name              mpls
        peer-connectivity   bidirectional
        topology          spoke
        vector            mpls
        qp-value          0

        udp-transform
            mode              auto-detect
            detect-interval     300
            nat-keep-alive-mode  disabled
        exit

        path-mtu-discovery
            enabled   false
            interval  600
        exit
    exit
exit
```

IV. NEIGHBORHOODS

Neighborhoods are a means of specifying which Network Interfaces in your 128T Authority are connected to each other, helping your 128T conductor understand your network topology. By using Neighborhoods, your 128T will auto-configure certain elements, such as Peers, Adjacencies, and Service Routes, to ensure that the Network Interfaces in the same Neighborhood connect to each other. By viewing the Network Interfaces and seeing which ones share the same Neighborhood labels, you can get a picture for which Network Interfaces are connected to each other. In the Neighborhood configuration, you can also see a field called `topology`. The values for Topology are:

- mesh
- hub
- spoke

Meshes will connect to everything, hubs connect to spokes and meshes, and spokes connect to hubs and meshes. Knowing this information can help you visualize which Network Interfaces are connected to each other.

V. NETWORK INTERFACES

Network Interfaces tell your 128T which networks it is participating in. You can have multiple Network Interfaces per Device Interface. Network Interfaces have `global-id` that are used by the 128T software to identify which Network Interface to send traffic out of. If two Network Interfaces have the same `global-id` then they are treated as redundant for one another, or as a SHARED INTERFACE.

Note: do not assign a `global-id` to a network-interface on your own. Your 128T will generate a unique `global-id` for a standalone network-interface, and will automatically assign a common `global-id` for shared interfaces when it detects that they each reference a common `shared-phys-address`. For more information on shared interfaces, refer to the High Availability Best Practices documentation.

```
network-interface    wan1
    name                 wan1
    global-id            5

    neighborhood         internet
        name      internet
        topology  hub
        vector    internet
    exit
    inter-router-security  aes1

    address              3.3.3.128
        ip-address     3.3.3.128
        prefix-length  24
        gateway        3.3.3.1
    exit

    adjacency            1.1.1.128
        ip-address           1.1.1.128
        peer-connectivity    bidirectional
        peer                 seattlesite1
        generated            true
        inter-router-security  aes1
```

```
      cost              0
      qp-value          0
      vector            internet
    exit

    adjacency           2.2.2.128
      ip-address          2.2.2.128
      peer-connectivity     bidirectional
      peer                dallassite1
      generated           true
      inter-router-security  aes1
      cost              0
      qp-value          0
      vector            internet
    exit
  exit
```

VI. DEVICE INTERFACES

The Device Interface tells you which port on the physical server that you are using. The Device Interface will consist of a name, the type of interface it is, and an identifier for how to tell the 128T software which port it is taking control of. For example, if the `type` is `ethernet`, then you will need to put in a `pci-address` as your identifier.

```
device-interface  wan1
    name               wan1
    pci-address        0000:00:09.0
    shared-phys-address  00:01:00:A1:CA:A5

    network-interface    wan1
        name                 wan1
        global-id            5

        neighborhood          internet
            name      internet
            topology  hub
            vector    internet
        exit
        inter-router-security  aes1

        address               3.3.3.128
            ip-address     3.3.3.128
            prefix-length  24
            gateway        3.3.3.1
        exit

        adjacency             1.1.1.128
            ip-address            1.1.1.128
            peer-connectivity     bidirectional
            peer                  seattlesite1
            generated             true
```

```
            inter-router-security  aes1
            cost              0
            qp-value          0
            vector            internet
        exit

        adjacency         2.2.2.128
            ip-address        2.2.2.128
            peer-connectivity    bidirectional
            peer              dallassite1
            generated         true
            inter-router-security  aes1
            cost              0
            qp-value          0
            vector            internet
        exit
    exit
exit
```

If two Device Interfaces have the same `shared-phys-address` that means that they are a redundant pair. You can issue `show device-interface summary` to find out which one is currently active and which is standby.

VII. REDUNDANCY GROUP

A Redundancy Group is a group of Device Interfaces that are tied to each other. This means that when one goes down, they will all switch over to their redundant partner.

```
redundancy-group    datacenter1
  name     datacenter1

  member   datacenter1 mpls1
    node      datacenter1
    device-id  mpls1
  exit

  member   datacenter1 srv1
    node      datacenter1
    device-id  srv1
  exit

  member   datacenter1 srv2
    node      datacenter1
    device-id  srv2
  exit

  member   datacenter1 srv3
    node      datacenter1
    device-id  srv3
  exit

  member   datacenter1 wan1
    node      datacenter1
    device-id  wan1
  exit
  priority  1
```

```
exit
```

VIII. ADJACENCY

Another sub-element of the Network Interface is the Adjacency. The Adjacency tells your 128T how it can reach its Peer. Peers can be reachable from multiple Network Interfaces, so this is just one of the ways that a particular Peer is reachable.

IX. PEER

A Peer is another 128T router that your current 128T router is connected to. By understanding which routers are Peers, you can start to draw out a network diagram with your 128T nodes.

```
router        seattlesite1
  name              seattlesite1
  location-coordinates  +47.6062-122.3321/
  description         Test-Changed
  inter-node-security   internal

  peer              bostonsite1
    name           bostonsite1
    authority-name  CompanyX
    router-name     bostonsite1
  exit

  node              branchoffice1
    name             branchoffice1
    asset-id         branch-1-router
    role            combo

    device-interface  mpls1
      name             mpls1
      type            ethernet
      pci-address       0000:00:04.0
      capture-filter     len>0

      network-interface  mpls1
        name                mpls1
        global-id           3
        vlan              0
        type              external
```

```
conductor           false

neighborhood          mpls
   name              mpls
   peer-connectivity   bidirectional
   topology          spoke
   vector            mpls
   qp-value          0

   udp-transform
      mode              auto-detect
      detect-interval      300
      nat-keep-alive-mode  disabled
   exit

   path-mtu-discovery
      enabled   false
      interval  600
   exit
exit
inter-router-security   internal
prioritization-mode     local
source-nat              false
qp-value                0
mtu                     1500
enforced-mss            disabled

address                 10.0.128.0
   ip-address      10.0.128.0
   prefix-length  31
   gateway        10.0.128.1
exit
```

```
        adjacency           10.0.128.1
          ip-address          10.0.128.1
          peer-connectivity    bidirectional
          peer                bostonsite1
          generated           true
          inter-router-security  aes1
          cost                0
          qp-value            0
          vector              mpls

          udp-transform
            mode              auto-detect
            detect-interval     300
            nat-keep-alive-mode  disabled
          exit

          path-mtu-discovery
            enabled   false
            interval  600
          exit
        exit
        icmp                allow
        multicast-listeners     automatic
        multicast-report-proxy  false
        dhcp                disabled
      exit
    exit

    device-interface  lan1
      name            lan1
      pci-address       0000:00:03.0

      network-interface  lan1
```

```
      name      lan1
      global-id  4
      tenant     seattle.corp

      address    192.168.64.1
        ip-address     192.168.64.1
        prefix-length  24
      exit
    exit
  exit

  device-interface  wan1
    name              wan1
    pci-address       0000:00:05.0

    network-interface  wan1
      name                 wan1
      global-id            6

      neighborhood          internet
        name    internet
        vector  internet
      exit
      inter-router-security  aes1

      address              1.1.1.128
        ip-address     1.1.1.128
        prefix-length  24
        gateway        1.1.1.1
      exit

      adjacency            3.3.3.128
        ip-address           3.3.3.128
```

```
                peer-connectivity      bidirectional
                peer                   bostonsite1
                generated              true
                inter-router-security  aes1
                cost                   0
                qp-value               0
                vector                 internet
            exit
        exit
    exit
exit
```

X. ROUTER

The routers are the 128T software systems responsible for receiving and sending packets to their correct destinations, aka routing. You can think of Routers as a logical entity because they consist of one or two Nodes. Routers have `description` and `location-coordinates` fields that should be used to help give you some context as to the function of this router and where it is located. The name given to a Router should be a helpful name that tells the viewer of the configuration as much useful information as possible.

Any elements configured under the Router are considered Local Data. That means that they only exist in that Router and not in the other Routers. Global Data is anything that is configured outside the Router element and applies to your whole Authority, not just one Router.

XI. NODE

The Node is the physical hardware or virtual space that makes up the Router. A Router can consist of one or two Nodes. If a Router consists of two Nodes, then it is referred to as an HA pair.

HA in the 128T is not a pure Active:Standby relationship. You can have different Device Interfaces on different Nodes active while other Device Interfaces on the same Node be in standby.

XII. SERVICE ROUTES

Service Routes are used to tell your 128T how to reach a particular Service. This is Local Data, so it is only specific to a particular Router. It tells that Router that if you see traffic that matches one of your configured Services, then send it to the following destination. The destination can be one or more 128T Peers, a gateway, an IP address, a subnet, a blackhole, etc. If the destination is another 128T, then the 128Ts will use Secure Vector Routing to send the traffic.

```
service-route      webserver-route
  name         webserver-route
  service-name  webserver
  peer         bostonsite1
exit

service-route            local-webserver3-route
    name               local-webserver3-route
    service-name        webserver3
    nat-target          172.36.128.2
    service-route-policy  lb-policy
  exit
exit
```

XIII. SERVICE ROUTE POLICIES

A Service Route Policy is way to set limits on the Service Routes to use when the 128T is determining which path to take. This is necessary for load balancing. You can set limits based on the max number of concurrent sessions to send down one path or the session rate/second for that path. Service Route Policies get applied to the Service Routes.

```
service-route-policy  lb-policy
    name          lb-policy
    max-sessions  1000
exit
```

XIV. ROUTING DEFAULT-INSTANCE

While using Service Routes with the 128T is the preferred method of routing, you may also encounter deployments that use traditional routing methods such as BGP, OSPF, or Static Routes. You will find the settings for these under `routing` under the Router. The Routing element must have a `type` of `default-instance` but the sub-elements under that will have all the settings you need to set for creating Static Routes, BGP peering, and OSPF peering.

When the 128T is making routing decisions, it will use traditional administrative distances to figure out which route to use with one exception: Service Routes get priority over every other route. You can see what route will get chosen with `show fib`. You can see the traditional routing decisions with `show rib`.

```
routing             default-instance
  type              default-instance

    routing-protocol  bgp
      type            bgp
      local-as        100
      router-id       1.1.1.128

      address-family  ipv4-unicast
        afi-safi              ipv4-unicast

        default-route-distance
          external  69
          internal  13
          local     55
        exit

        network             192.168.64.0/24
          network-address  192.168.64.0/24
          policy           mark-vrfl
        exit
      exit
    exit
exit
```

XV. TENANTS

Tenants are a way to define endpoints that you want to identify on your network. When you define Tenants, you can create access policies on your Services based on the Tenant name.

```
tenant        corp
    name  corp
exit

tenant        seattle.corp
    name  seattle.corp
exit

tenant        dallas.corp
    name  dallas.corp
exit

tenant        _internal_
    name        _internal_
    description  "Auto generated tenant for internal services"
    generated    true
exit
```

As you can probably notice from the configuration above, to configure a Tenant, all you need to do is give it a name. Your name should be descriptive. Optionally, there is a description field you can use to give more context to anyone reviewing your configuration.
Tenants can be children of other Tenants. This means that if you apply an access policy to a parent Tenant, all the children will inherit it. However, if you set the access policy at the child, then the parent will not get that access from the child. You create the parent-child relationship by how you name your Tenants. The name format is child.parent. So for example, if the tenant is `dallas.corp`, then the parent is `corp` and `dallas` is the child. You can have an arbitrary number of levels of tenants within a tenant, such as greatgrandchild.grandchild.child.parent.

Traffic gets associated with a Tenant in one of three ways:

1. Tenancy was assigned by an upstream 128T and communicated in metadata
2. By arriving on a Network Interface that has a tenant configured
3. By arriving on a network-interface that has a neighborhood configured, and the source IP address of the IP packet is defined within a tenant's `member > address` definition

In the metadata that 128T routers send to each other when using SVR, they include the Tenant traffic has already been assigned. So, if my Seattle Router assigns the `seattle.corp` Tenant to a session and then sends that session to my Boston Router, my Boston Router will already know that this session belongs to the `seattle.corp` Tenant and use that information in determining if this session has access to the Service it is trying to access.

At the Network Interface, you can also assign a Tenant and if any traffic that doesn't already have a Tenant assigned ingresses this Network Interface, it will belong to that Tenant.

```
device-interface  lan1
    name             lan1
    pci-address      0000:00:03.0

    network-interface  lan1
        name       lan1
        global-id  4
        tenant     seattle.corp

        address    192.168.64.1
            ip-address     192.168.64.1
            prefix-length  24
        exit
    exit
exit
```

Lastly, if you are using neighborhoods, then you can associate tenancy based on the originating subnet. Within the tenant configuration, you will find a sub-element called `member`. The `member` element defines the association of IP address ranges within a neighborhood to a tenant. If a network interface belongs to that neighborhood, and the source address of the traffic that ingresses that network interface falls into one of the subnets assigned to that tenant, then the traffic will be associated to that tenant. This technique allows you to correlate traffic into an array of tenants when it arrives on a single network interface.

```
tenant         corp
    name    corp

    member  mpls
        neighborhood  mpls
        address       192.168.64.1/24
        address       10.0.0.0/24
```

```
    exit
  exit
```

In this example, any traffic arriving on an interface that is part of the `mpls` neighborhood, will be associated with the `corp` tenant if it is sourced from 192.168.64.0/24 or 10.0.0.0/24.

Note: some configurations elect to use this technique for tenant association in lieu of the previous technique (assigning a tenant to a network-interface). This is done by associating the network-interface with a neighborhood, and setting the `member > address` prefix to `0.0.0.0/0`.