

Subnet address reservations

A subnet + vlan-id can conditionally be created when the subnet is considered full.

The prerequisite is that the subnet can be lifted from an ip-plan. A custom subnet can be used as well, but requires the subnet-name to used throughout the design and may not be changed by an operator.

The Service-type commands for this behaviour are:

ADD	SUBNET	srv	NET_NAME_NOTFULL	value	net	find and conditionally add subnet from an IP-plan by name should the existing be 'full'
LOCATE	SUBNET	srv	NET_NAME_NOTFULL	value	net	find and conditionally add subnet from an IP-plan by name should the existing be 'full'

Both command forms offer the same functionality.

Currently the behaviour to create a new subnet instead of just locating an existing one is effected when any of the following conditions is met:

- No subnet of the required name can be found in the 'srv'
- The Subnet Full flag is set (and optionally locked) by the operator
- The number of ports assigned to the subnet without topology is equal to or exceeds the subnet size (in host addresses)

This mechanism works fine for fine-grained ip-plans where plenty of small subnets for user or server services are available and vlan management is (largely) automated.

However, it does not allow for situations where many devices can potentially share a subnet but where the port-assignment to this subnet are made later on, or where the number of ports may increase over the life-span of a device as is the case with stacks.

To deal with these designs, the current subnet-full mechanism must be extended to include address-reservations for a subnet. Reservations in the sense of the number of host-addresses 'in-use' for a subnet.

In this case the 'add a new subnet' conditions would become:

- No subnet of the required name can be found in the 'srv'
- The Subnet Full flag is set (and optionally locked by the operator)
- The number of ports assigned to the subnet (without topology) is equal to or exceeds the subnet size
- The number of address-reservations of the subnet is equal to or exceeds the subnet size

To realise this extension, the Ip_subnets table would require the numeric field 'Ip_subnet.add_reservations' with the default value of '0'.

The manipulation of these reservations would be done (strictly?) using Service-types or Service-tasks when new nodes or (stack) chassis are added to removed.

Three new service-type commands are required to offer the means to manipulate the `addr_reservation` count:

ASSIGN	SUBNET	net	ADDR_RESERVATION	value	set the subnet's address reservation count to value
ASSIGN	SUBNET	net	ADDR_RESERVATION_INC	value	increase the subnet's address reservation count with value
ASSIGN	SUBNET	net	ADDR_RESERVATION_DEC	value	decrease the subnet's address reservation count with value

A secondary problem arises when increasing the `Net_reservations` after a subnet was added/located. This could cause the reservation to exceed the subnet size. The optional resolution could be to perform the conditional add/locate subnet AGAIN, but this suffers from the problem that the new subnet that might have been create the second time still has its `Net_reservation` set to zero, not the desired reservation set between the add/locate subnets.

To resolve this situation and avoid the double add/locate subnet calls, a modification can be made to the Service-type to expand to TWO value fields, say `Type_option`. This `Type_option` field allows modifiers to be passed to the service-type-call. The best implementation is to have the option field become a text field using attribute / value pairs. In that case the solution is self-expanding for future cases and variables. Especially for the PROMPT-class service type calls would this be useful.

The changes to the environment are significant: `Service_type` table, `Service_type_syntax` table, various `service_type` calls and more significantly, the front-end subform wizard.

For the case at hand, the call for the `ADD | SUBNET | NET_NAME_NOTFULL` would receive the `Type_option` value of `RESERVATION_ADD=48`. The service type routine would use the reservation add value to see if the

```
ADD|SUBNET|srv|NET_NAME_NOTFULL|netname-value|Reservation_add=value|net-alias
```

find and conditionally add subnet from an IP-plan by name should the existing be 'full' AFTER the `resadd-value`

```
LOCATE|SUBNET|srv|NET_NAME_NOTFULL|netname-value|Reservation_add=value|net-alias
```

find and conditionally add subnet from an IP-plan by name should the existing be 'full' AFTER adding the `resadd-value`

From:
<https://wiki.netyce.com/> - **Technical documentation**

Permanent link:
https://wiki.netyce.com/doku.php?id=guides:user:subnet_reservations

Last update: **2024/07/03 12:31**

