Introduction

This document includes tips and trick for how to configure the DKTCOMEGA 797xx series, known as JUMA/Forsete-II.

The examples come from real life experiences, and can be used by operators as part of their integration process.

Please notice that these are examples only, it is a representation of just a sub-set of the switch capabilities; please refer to user syntax guide for the full overview.

For further information, please contact

JESPER BROSZKA
CTO, HEAD OF TECHNOLOGY

Fanoevej 6
DK-4060 Kirke Saaby

www.DKT.dk

Phone: (+45) 46 46 26 26
Direct: (+45) 61 22 57 14

Subjects, overview

How to enable flow control

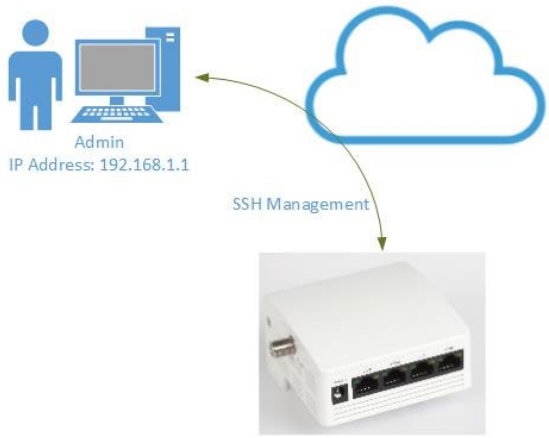
From firmware 05_08 and forward flow control is disabled on all port, prior to this release it was enabled per default. The reason for the change is that QoS is not fully functional when flow control is enabled.

In order to enabled flow control please add the following commands

```
conf <<EOF
# Enable flow control on a single port (e.g. port 0):
conf port set auto-nego port 0 ability 10h 10f 100h 100f 1000f flow-control
asy-flow-control

# Enable flow control on all ports:
conf port set auto-nego port 0-4 ability 10h 10f 100h 100f 1000f flow-
control asy-flow-control

exit
EOF
```

How to allow SSH management from one IP address only

The following example demonstrates how to configure SSH management taking place from one IP address only. All other IP addresses are excluded from managenet via SSH.



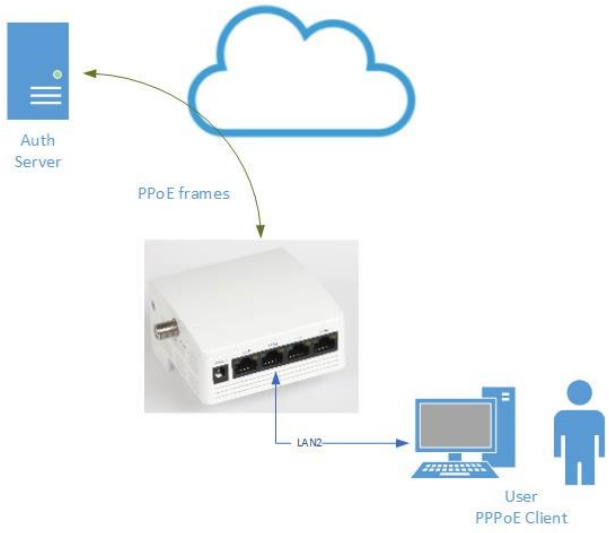
```
conf <<EOF
# Step 1. enable LAN port and WAN port
acl state acl set port 0-4 state enable

# Step 2. packet field selector setting
# This command can configure acl user defined field. Each field can set 16-
bits content of packet which user wants to filter for acl uage. From pure
raw packet to layer-4 content as tcp or udp, the field can be set for
parsing content of packet inside the first 256 bytes.

field-selector set index 1 format ipv4-header offset 8 field-selector set
index 2 format ip-payload offset 2

# Step 3. acl template setting
# This command can use to configure content of editing template. Each
template contains limited packet pattern bits for acl rule matching. The
ordering of editing template pattern bits is depended on editing order.
Notice sip equals source ipv4 address

acl clear template
acl set template user-field 1
acl set template user-field 2
acl set template sip
acl add template entry 1

# Step 4. acl entry 2: SSH with specific source IP address, trap these
frames to cpu port.
# This command can edit wanted rule contents before being added to device.
The rule content is depended on which template is used. So, configuring
used template is more important before setting rule, which was done above.

acl set rule template entry 1
acl set rule state valid
acl set rule port 0-4

# ipv4-header offset 8 value = 0x0006 meas TCP
```

```
acl set rule user-field 1 data 0x0006 mask 0x00ff

# TCP destination port = 0x16 means SSH protocol

acl set rule user-field 2 data 0x16 mask 0xffff
acl set rule sip data 192.168.1.1 mask 255.255.255.255

# This command can clear configured actions of editing rule. It should be
execute before adding new configured rule.

acl clear action

# This command can edit wanted actions of rule will be added to device. Let
us trap frames to CPU port

acl set action trap-to-cpu

# Execute the entry

acl add entry 2

# Step 5. acl entry 3: SSH with source IP not matching entry2, drop these
frames

acl set rule template entry 1
acl set rule state valid
acl set rule port 0-4

# ipv4-header offset 8 value = 0x0006 meas TCP

acl set rule user-field 1 data 0x0006 mask 0x00ff

# TCP destination port = 0x16 means SSH protocol

acl set rule user-field 2 data 0x16 mask 0xffff
acl clear action

# This command can edit wanted actions of rule will be added to device. Let
us trap frames to nowhere
```

```
acl set action redirect port none

acl add entry 3

exit
EOF
```

How to pass only certain frames, discard others

The following example shows how to discard all frames on LAN 2, except for PPPoE with Ethertype 8863/64.




```
conf << EOF
# Clear device templates
acl clear template

# Set device template Destination MAC address
acl set template dmac

# Set device template C-Tag VLAN
acl set template ctag

# Set device template Source MAC address
```

```
acl set template smac

# Set device template Ethertype
acl set template ethertype

# Add ACL entry, index 0
acl add template entry 0

# FOLLOWING IS DEFINING ACL RULE CONTENT
# ACL clear is typically needed, before modifying rules, etc.
acl clear

# Create a rule for template entry 0, defined above
acl set rule template entry 0

# enable the rule
acl set rule state valid

# apply the rule for port 1, which is LAN port 2
acl set rule port 1

# The following is set, in order to enable the NOT bit. By setting this
bit, the comparison result of ACL rule will be converted. That is, un-
matched packet would be assigned a “matched” result and a matched packet
will be considered as “un-matched”
acl set rule operation reverse-state enable

# Check for PPPoE frames, The following is set, in order to enable the NOT
bit, correct? By setting this bit, the comparison result of ACL rule will
be converted. That is, un-matched packet would be assigned a “matched”
result and a matched packet will be considered as “un-matched”
acl set rule ethertype data 0x8863 mask 0xfff8

# FOLLOWING IS ACL RULE ADDING
# ACL clear is typically needed, before modifying rules, etc.
acl clear action

# all other frames than PPPoE defined above should be discarded, define a
rule for this
```

```
acl set action redirect port none

# Defined rule as entry 1
acl add entry 1

exit
EOF
```

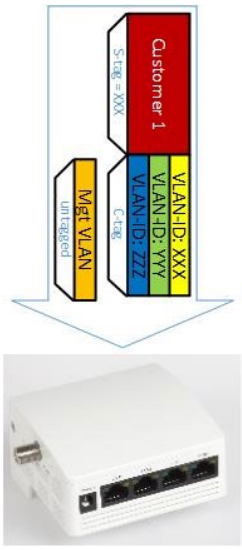
How to mirror from WAN to LAN1

```
conf mirror set mirroring-port 0 mirrored-port 4 rx-mirror tx-mirror
```

How to isolate LAN ports from each other

```
conf port set isolation port 0 mode0 egress-port 4
conf port set isolation port 1 mode0 egress-port 4
conf port set isolation port 2 mode0 egress-port 4
conf port set isolation port 3 mode0 egress-port 4
```

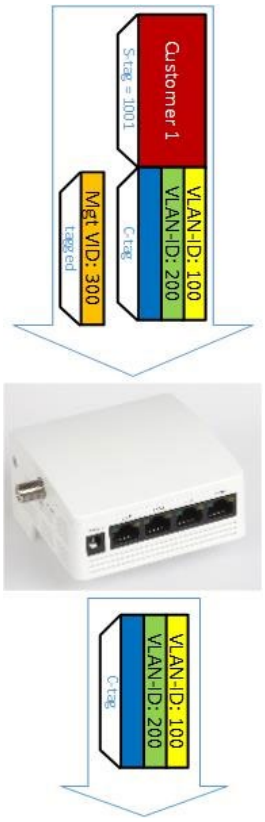
How to untag management vlan, when Service Provider mode is enabled




```
conf <<EOF

# initialize VLAN configuration
vlan init

# Enable s-vlan tag function on WAN port and set s-vlan tpid 0x88a8

# 4 means WAN port
svlan set service-port 4
svlan set tpid 0x88a8

#define management svlan
svlan create svlan-table svid 2
svlan set svlan-table svid 2 member 4,6
svlan set svlan-table svid 2 untag-member 4,6

#for upstream
svlan set port 6 svid 2
```

```
#for downstream
svlan set untag assign-svlan svid 2

exit

EOF
```

How to create S-VLAN 1001 on WAN, C-VLANs 100, 200 on LAN1…4, Mgt VLAN 300, keep tpid 8100 for S-tag




```
switch --enable-lan

conf <<EOF
```

```
# initialize VLAN configuration
vlan init

# create C-VLAN 100
vlan create vlan-table vid 100

# create C-VLAN 200
vlan create vlan-table vid 200

# define member ports for C-VLAN 100, LAN 1...4 and WAN
vlan set vlan-table vid 100 member 0-4

# define member ports for C-VLAN 200, LAN 1...4 and WAN
vlan set vlan-table vid 200 member 0-4

# defined config port based vid for LAN port 1...4
vlan set pvid port 0 100
vlan set pvid port 1 100
vlan set pvid port 2 100
vlan set pvid port 3 100

# enable SVLAN on WAN port
svlan set service-port 4

# define members for management vlan 300, CPU and WAN ports
svlan create svlan-table svid 300
svlan set svlan-table svid 300 member 4,6
svlan set svlan-table svid 300 untag-member 6

# defined config port based vid for CPU port
vlan set pvid port 6 300

# define CPU as untagged member of VLAN 300
vlan set vlan-table vid 300 untag-member 6

# create vid 0 for management, dummy vid
svlan create svlan-table svid 0

# add CPU and WAN ports to dummy vid
```

```
svlan set svlan-table svid 0 member 4,6

# dummy vid is untagged on CPU and WAN port
svlan set svlan-table svid 0 untag-member 4,6

# downstream SVLAN untag action, downstream SVLAN untag action for vid 0
svlan set untag assign-svlan svid 0

# assign vid 0 to CPU port
svlan set port 6 svid 0

# set tag protocol identifier of STAG to 9100, note other identifiers will
be excluded
svlan set tpid 0x8100

# create S-VLAN 1001
svlan create svlan-table svid 1001

# defined member ports for S-VLAN 1001, LAN 1...4 and WAN
svlan set svlan-table svid 1001 member 0-4

# defined config port based vid for LAN port 1...4
svlan set port 0 svid 1001
svlan set port 1 svid 1001
svlan set port 2 svid 1001
svlan set port 3 svid 1001

# S-VLAN 1001 is untagged on LAN port 1,,,4
svlan set svlan-table svid 1001 untag-member 0-3

# define tagged frames on LAN 1...4 and WAN only
vlan set accept-frame-type port 0 tag-only
vlan set accept-frame-type port 1 tag-only
vlan set accept-frame-type port 2 tag-only
vlan set accept-frame-type port 3 tag-only

# Isolate LAN port 1...4 from each other, allow only communication between
LANx and WAN
port set isolation port 0 mode0 egress-port 4
```

```
port set isolation port 1 mode0 egress-port 4
port set isolation port 2 mode0 egress-port 4
port set isolation port 3 mode0 egress-port 4

exit
EOF
```

How to make transparency to any VLANs so VLAN tagged frames are not filtered.


```
switch --enable-lan


conf <<EOF

vlan init
```

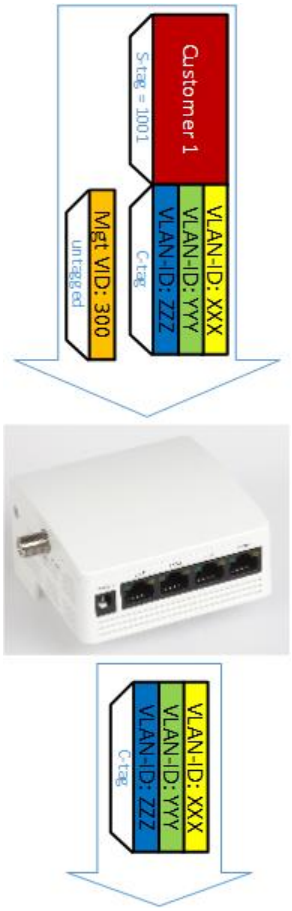
```
# Enable VLAN transparency
vlan set ingress-filter port 0-4 state disable
vlan set transparent state enable
vlan set egress port 0-4 keep-tag ingress-port 0-4 state enable
l2-table set lookup-miss port 0-4 ip-mcast action flood-in-vlan
l2-table set lookup-miss port 0-4 ip6-mcast action flood-in-vlan
exit
EOF
```

How to make transparency to any VLANs and have VLAN 50 as mgt.


```
switch --enable-lan
```

```
conf <<EOF
vlan init
vlan create vlan-table vid 50
vlan set vlan-table vid 50 member 4,6
vlan set vlan-table vid 50 ext-member 0-5
vlan set pvid port 6 50
vlan set vlan-table vid 50 tag-member 4

# Remove untagged management from CPU
vlan set vlan-table vid 1 member 0-4

# Enable VLAN transparency
vlan set ingress-filter port 0-4 state disable
vlan set transparent state enable
vlan set egress port 0-4 keep-tag ingress-port 0-4 state enable

# CPU port egress tagged
vlan set egress port 6 keep-tag ingress-port 4 state disable

# Configure classifier rule for management VLAN 50
classf set cf-sel-port pon enable
classf set upstream-unmatch-act permit
classf clear
classf set rule direction downstream
classf set rule tag-vid data 50 mask 0xfff
classf set downstream-action uni-forward-act forced port 6
classf set downstream-action cvlan-act del
classf add entry 0
l2-table set lookup-miss port 0-4 ip-mcast action flood-in-vlan
l2-table set lookup-miss port 0-4 ip6-mcast action flood-in-vlan
exit
EOF
```

How to make transparency to any VLANs and have untagged vlan as management and SVLAN 1001 on WAN




```
switch --enable-lan
conf <<EOF

vlan init
```

```
# the following allows VLAN C-tag transparency, so we accept any VLAN tag
on LAN 1/WAN
vlan set state enable
vlan set transparent state enable

# disable ingress filter
vlan set ingress-filter port 0 state disable
vlan set ingress-filter port 4 state disable

# disable egress filter
vlan set egress port 4 keep-tag ingress-port 0 state enable
vlan set egress port 0 keep-tag ingress-port 4 state enable
l2-table set lookup-miss port 0-4 ip-mcast action flood-in-vlan
l2-table set lookup-miss port 0-4 ip6-mcast action flood-in-vlan
svlan init

# svlan set tpid 0x88a8
svlan set service-port 4
svlan create svlan-table svid 0
svlan set svlan-table svid 0 member 4,6

# Egress towards CPU is untagged
svlan set svlan-table svid 0 untag-member 4,6

# management packets have no stag, so should default assign a dummy stag
svlan set untag assign-svlan svid 0
svlan set port 6 svid 0
svlan create svlan-table svid 1001
svlan set svlan-table svid 1001 member 0,4
svlan set port 0 svid 1001
svlan set svlan-table svid 1001 untag-member 0
exit
EOF
```

How to use IVL vs. SVL

This example shows how to use IVL instead of SVL, so same MAC can be present on several VLANs.

```
switch --enable-lan
conf vlan init
# Untag vlan 1 on WAN and CPU for DKT management
conf vlan create vlan-table vid 1
conf vlan set vlan-table vid 1 member 4,6
conf vlan set vlan-table vid 1 ext-member 0-5
conf vlan set pvid port 4 1
conf vlan set pvid port 6 1
# Tag vlan 25 on WAN and LAN1 for CPE management
conf vlan create vlan-table vid 25
conf vlan set vlan-table vid 25 member 0,4
# Tag vlan 250 on WAN and Untag on LAN1 for CPE data
conf vlan create vlan-table vid 250
conf vlan set pvid port 0 250
conf vlan set vlan-table vid 250 untag-member 0
conf vlan set vlan-table vid 250 member 0,4
# Active independant vlan learning, for same mac-different vlan
conf vlan set vlan-table vid 1 hash-mode ivl
conf vlan set vlan-table vid 25 hash-mode ivl
conf vlan set vlan-table vid 250 hash-mode ivl
conf << EOF
```

How to make a filter, so only certain MAC addresses can send on VLAN 25

This example shows how only a certain range of SMACs with prefix 00:19:9f:XX:XX:XX can send frame upwards, applicable to VLAN 25 only. Also IVL is activated.

```
switch --enable-lan
conf vlan init
# Untag vlan 1 on WAN and CPU for DKT management
conf vlan create vlan-table vid 1
conf vlan set vlan-table vid 1 member 4,6
conf vlan set vlan-table vid 1 ext-member 0-5
conf vlan set pvid port 4 1
conf vlan set pvid port 6 1
```

```
# Tag vlan 25 on WAN and LAN1 for CPE management
conf vlan create vlan-table vid 25
conf vlan set vlan-table vid 25 member 0,4
# Tag vlan 250 on WAN and Untag on LAN1 for CPE data
conf vlan create vlan-table vid 250
conf vlan set pvid port 0 250
conf vlan set vlan-table vid 250 untag-member 0
conf vlan set vlan-table vid 250 member 0,4
# Active independant vlan learning, for same mac-different vlan
conf vlan set vlan-table vid 1 hash-mode ivl
conf vlan set vlan-table vid 25 hash-mode ivl
conf vlan set vlan-table vid 250 hash-mode ivl
conf << EOF
acl clear template
acl set template dmac
acl set template ctag
acl set template smac
acl set template ethertype
acl add template entry 0
exit
EOF
#end with configure acl template 0
conf << EOF2
acl clear
acl set rule template entry 0
acl set rule state valid
acl set rule port 0-3
acl set rule smac data 00:19:9f:00:00:00 mask ff:ff:ff:00:00:00
acl set rule ctag data vid 25 priority 0 cfi 0 mask vid 0xfff priority 0
cfi 0
acl clear action
acl set action copy port none
acl add entry 1
exit
EOF2
#end with conf acl 1
conf << EOF3
acl clear
acl set rule template entry 0
acl set rule state valid
acl set rule port 0-3
acl set rule ctag data vid 25 priority 0 cfi 0 mask vid 0xfff priority 0
cfi 0
acl clear action
```

```
acl set action drop
acl add entry 2
acl set port 0-3 state enable
exit
EOF3
#end with conf acl 2
```

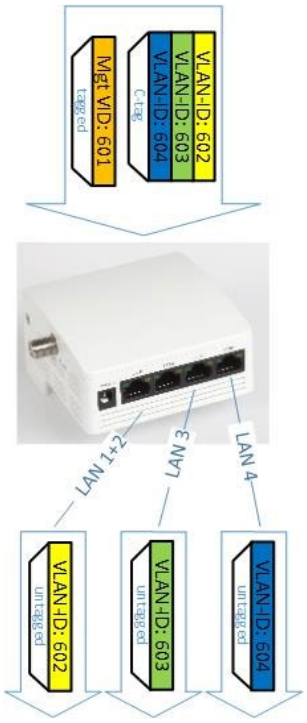
How to make transparency to any VLAN and add S-tag 100 on WAN

This example shows how ANY C-tag on LAN ports will be transparent to the device, and enter on WAN with the ANY C-tag and S-tag of 100.

```
switch --enable-lan

conf <<EOF

vlan init

vlan set ingress-filter port 0-4 state disable
vlan set state disable
svlan set service-port 4
svlan create svlan-table svid 0
svlan set svlan-table svid 0 member 0-4
svlan set port 0-3 svid 0
svlan set unmatch assign-svlan svid 0

classf clear
classf set rule direction upstream
classf set rule uni data 0 mask 0x7
classf set rule cvlan-bit data 1 mask 0x1
classf set upstream-action cvlan-act transparent
classf set upstream-action svlan-act c-tpid
classf set upstream-action svlan-id-act assign 100
classf add entry 2

classf clear
classf set rule direction downstream
```

```
classf set rule cvlan-bit data 1 mask 0x1
classf set rule svlan-bit data 1 mask 0x1
classf set downstream-action cvlan-act transparent
classf set downstream-action svlan-act del
classf set downstream-action uni-forward-act flood port 0-3
classf add entry 3

classf set cf-sel-port pon enable
svlan set untag assign-svlan svid 0
svlan set port 6 svid 0
svlan set svlan-table svid 0 member 0-6
svlan set svlan-table svid 0 untag-member 4,6

exit
EOF
```

How to create 3 VLANs with 1 mgt VLAN




```
# Vlan 601 Mgmt
```

```
# Vlan 602 Bridge Internet LANport1+2
# Vlan 603 Bridge Voip LANport3
# Vlan 604 IPTV LAN4

switch --enable-lan

conf <<EOF
# initialize VLAN configuration
vlan init
# create C-VLAN 601
vlan create vlan-table vid 601
# create C-VLAN 602
vlan create vlan-table vid 602
# create C-VLAN 603
vlan create vlan-table vid 603
# create C-VLAN 604
vlan create vlan-table vid 604
# define member ports for C-VLAN 602, LAN 1, 2 and WAN
vlan set vlan-table vid 602 member 0,1,4
vlan set vlan-table vid 602 untag-member 0,1
# define member ports for C-VLAN 603, LAN 3 and WAN
vlan set vlan-table vid 603 member 2,4
vlan set vlan-table vid 603 untag-member 2
# define member ports for C-VLAN 604, LAN 4 and WAN
vlan set vlan-table vid 604 member 3,4
vlan set vlan-table vid 604 untag-member 3
# define management vlan vid 601
vlan create vlan-table vid 601
vlan set vlan-table vid 601 member 4,6
vlan set vlan-table vid 601 ext-member 0-5
vlan set pvid port 6 601
# define WAN as tagged member of VLAN 601
vlan set vlan-table vid 601 tag-member 4
# define CPU as untagged member of VLAN 601
vlan set vlan-table vid 601 untag-member 6


# defined config port based vid for LAN port 1...4
vlan set pvid port 0 602
```

```
vlan set pvid port 1 602
vlan set pvid port 2 603
vlan set pvid port 3 604
exit
EOF
```

How to reprioritize frames entering on one port to another

If you want to set port-based priority of a port (example priority 1), the following commands can be used:

```
conf qos set remapping port all internal-priority 1
conf qos get remapping port all
```

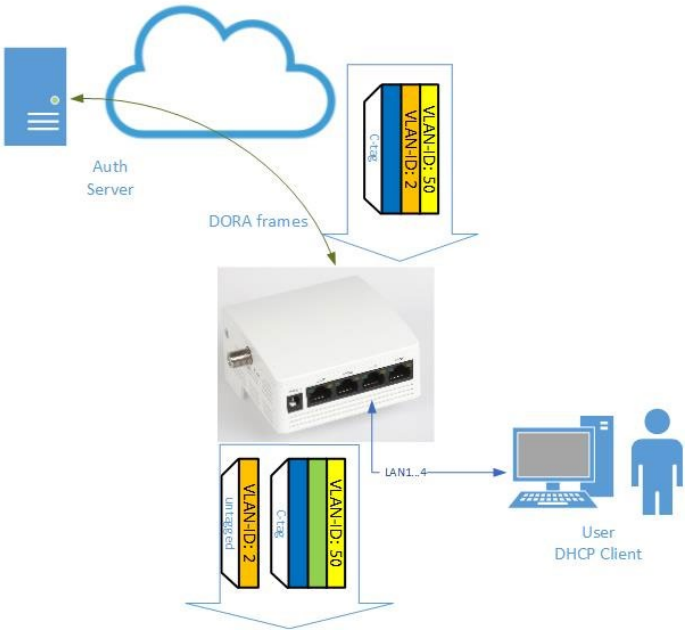
How to prioritize on DSCP value internally

If you want to set-up internal prioritization for DSCP, e.g. value 60 should have highest priority.

```
# internal forwarding priority.
conf qos set remapping dscp dscp 60 internal-priority 7
```

How to enable dhcp relay and let the frames keep their origin tags

If both tagged/untagged vids are to be DHCP relayed to CPU (DHCP option 82), then the following example can be used. VLAN vid 1 is default CPU management vlan; VLAN vid 2 is untagged VLAN on LAN ports, tagged VLAN on WAN; VLAN vid 50 is tagged VLAN on LAN/WAN ports




```
switch --enable-lan
/etc/init.d/dhcprelayd start -i

conf <<EOF
vlan create vlan-table vid 2
vlan create vlan-table vid 50

vlan set vlan-table vid 2 member 0-6
vlan set vlan-table vid 2 untag-member 0-3
vlan set vlan-table vid 2 tag-member 4-6

vlan set vlan-table vid 50 member 0-6
vlan set vlan-table vid 50 tag-member 0-6

vlan set pvid port 0 2
vlan set pvid port 1 2
vlan set pvid port 4 2
vlan set pvid port 6 2
exit
EOF
```

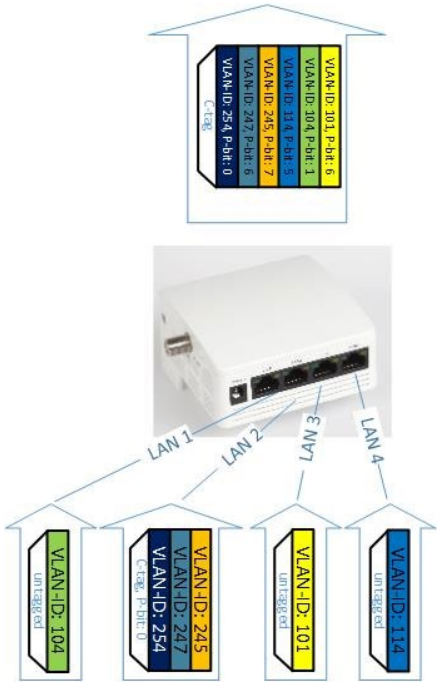
How to rate limit traffic based on VLAN VID

Here is an example for rate limiting VLAN 100 on LAN port 1 with a limitation of 10Mbps. Other VLANs on LAN port 1 will not be affected by this rate limitation. Please note that VLANs are NOT defined, remember to define the VLANs (tagged or untagged)


```
conf <<EOF

# step 0, set a meter
meter set entry 1 rate 10000

# step 1, enable port 0 acl state
acl set port 0 state enable

# step 2, and a template to compare vlan id
acl clear template
acl set template ctag
```

```
acl add template entry 2

# step 3, add ACL rule entry
acl clear

# the template entry index
acl set rule template entry 2
acl set rule state valid

# LAN port 0
acl set rule port 0

# VLAN id = 100 will hit this ACL rule
acl set rule ctag data vid 100 priority 0 cfi 0 mask vid 0xfff priority 0x0
cfi 0x0
acl clear action

# use meter index 1
acl set action meter 1

# the ACL rule means:  the traffic (with vlan tag = 100, and rx from LAN
port 0), will rate limit by meter index 1
acl add entry 10
exit
EOF
```

How to enable remarking of p-bit on WAN port for different vlans.




```
# 101, tagged WAN, untagged LAN 3, priority 6 on WAN
# 104, tagged WAN, untagged LAN 1, priority 1 on WAN
# 114, tagged WAN, untagged LAN 4, priority 5 on WAN
# 245, tagged WAN, tagged LAN 2, with priority 7 on WAN
# 247, tagged WAN, tagged LAN 2, with priority 6 on WAN
# 254, tagged WAN, tagged LAN 2, no priority

# Default-VLANs per port:
# LAN port 1 = 93, LAN port 3 = 91, LAN port 4 = 94

switch --enable-lan

conf <<EOF
# VLAN definitions
```

```
vlan create vlan-table vid 101
vlan create vlan-table vid 104
vlan create vlan-table vid 114
vlan create vlan-table vid 245
vlan create vlan-table vid 247
vlan create vlan-table vid 254

# Member port assignments
vlan set vlan-table vid 101 member 2,4
vlan set vlan-table vid 104 member 0,4
vlan set vlan-table vid 114 member 3,4
vlan set vlan-table vid 245 member 1,4
vlan set vlan-table vid 247 member 1,4
vlan set vlan-table vid 254 member 1,4

# Untagged member ports, other members are default tagged
vlan set vlan-table vid 101 untag-member 2
vlan set vlan-table vid 104 untag-member 0
vlan set vlan-table vid 114 untag-member 3

# default vid for ports, untagged frames are mapped into default VLAN
vlan set pvid port 0 104
vlan set pvid port 2 101
vlan set pvid port 3 114

# Enable vlan based prioritization internally in switch
vlan set vlan-table vid 101 vlan-based-priority state enable
vlan set vlan-table vid 104 vlan-based-priority state enable
vlan set vlan-table vid 114 vlan-based-priority state enable
vlan set vlan-table vid 245 vlan-based-priority state enable
vlan set vlan-table vid 247 vlan-based-priority state enable
vlan set vlan-table vid 254 vlan-based-priority state enable

# Assign internal prioritization in switch based on vlan vid
vlan set vlan-table vid 101 vlan-based-priority priority 6
vlan set vlan-table vid 104 vlan-based-priority priority 1
vlan set vlan-table vid 114 vlan-based-priority priority 5
vlan set vlan-table vid 245 vlan-based-priority priority 7
vlan set vlan-table vid 247 vlan-based-priority priority 6
```

```
# the following ports are untagged, so let us define their internal
priority value
qos set remapping port 0 internal-priority 0
qos set remapping port 2 internal-priority 0
qos set remapping port 3 internal-priority 0

# how to distingish between the priorities, vlan must be of a higher value,
as their internal prioritization value must be carried to the egress of the
switch
qos set priority-selector group-id 0 port 3
qos set priority-selector group-id 0 vlan 8

# definition of the remarking values at the egress of the switch (comes
from the internally prioritization values above)
qos set remarking dot1p user-priority 1 dot1p-priority 1
qos set remarking dot1p user-priority 5 dot1p-priority 5
qos set remarking dot1p user-priority 6 dot1p-priority 6
qos set remarking dot1p user-priority 7 dot1p-priority 7

# WAN port should have remarking enabled, so egress on WAN p-bit value is
kept from the internal value through the egress assigned value
qos set remarking dot1p port 4 state enable

exit
EOF
```

How to enable remarking of VLAN VID from WAN port to LAN port for different vlans

The example is based on VLAN 200 received on WAN and VLAN 100 on LAN, we will swap the VID visa versa

```
# WAN: Ingress with VLAN VID 200 <-> egress on LAN 0-3 as VLAN VID 100
# LAN 0-3: Ingress with VLAN VID 100 <-> egress on WAN as VLAN VID 200

------------------------------------------------------------

switch --enable-lan
conf <<EOF

# The following setting is applicable for WAN port
# WAN: Ingress with VLAN VID 200 <-> egress on LAN 0-3 as VLAN VID 100
# LAN 0-3: Ingress with VLAN VID 100 <-> egress on WAN as VLAN VID 200
```

```
vlan init

# create C-VLAN 100
vlan create vlan-table vid 100

# create C-VLAN 200
vlan create vlan-table vid 200

# define member ports for C-VLAN  100, LAN 1..4 and WAN
vlan set vlan-table vid 100 member 0-4
# define member ports for C-VLAN  200, LAN 1..4 and WAN
vlan set vlan-table vid 200 member 0-4

vlan set pvid port 0 100
vlan set pvid port 1 100
vlan set pvid port 2 100
vlan set pvid port 3 100
vlan set pvid port 4 200

acl set template ctag
acl add template entry 0
acl set rule template entry 0
acl set rule ctag data vid 200 priority 0 cfi 0
acl set rule port 4
acl set port 4 state enable
acl set rule state valid
acl set action cvlan ingress vid 100
acl add entry 14

# The following setting is applicable for LAN ports
acl set template ctag
acl add template entry 0
acl set rule template entry 0
acl set rule ctag data vid 100 priority 0 cfi 0
acl set rule port 0-3
acl set port 0-3 state enable
acl set rule state valid
acl set action cvlan ingress vid 200
acl add entry 15
```

```
exit
EOF
```

How to enable remarking of SVLAN VID from WAN port to VLAN VID of LAN port for different VLANs

```
The example is based on SVLAN VID TO VLAN VID translation example:

LAN port 0 VID = 50 <--> WAN port 4 SVID =100
WAN port 4 SVID = 100 <--> LAN port 0 VID =50


Please Note: svlan-index should be correctly set, it is described how to
look up the index in the middle of the configuration example.

-----------------------------------------------------------

switch --enable-lan
conf <<EOF

vlan init
svlan init

Step 1: global configuration
# svlan set service-port 4
svlan set tpid 0x8100
svlan set lookup-type svlan-table

# create default unmatch svid 0
svlan create svlan-table svid 0
svlan set svlan-table svid 0 member 0-4
svlan set unmatch assign-svlan svid 0
```

```
# Step 2: Downstream vlan translation
# translation list(100 --> 50)
svlan create svlan-table svid 100
svlan set svlan-table svid 100 member 0,4
svlan set svlan-table svid 100 untag-member 0

{
NOTE THIS IS INFORMATIVE ONLY, NOT PART OF THE CONFIGURATION)
# lookup the svlan 100's index in 64-svlan table.
# From the dump, we can see the svlan 100's index is 1.
RTK.0> svlan get entry all
Index SVID Member  TagSet  Spri FidEn   FID EfidEn  Efid
0     0    0-4     0-6     0    Disable 0   Disable 0
1     100  0,3     1-6     0    Disable 0   Disable 0
}

svlan set vlan-conversion sp2c entry 0 state valid
# svlan-index is the svid=100's index in the 64-svlan-entry
svlan set vlan-conversion sp2c entry 0 svlan-index 1
svlan set vlan-conversion sp2c entry 0 vid 50
svlan set vlan-conversion sp2c entry 0 port 0


# Step 3: Upstream vlan translation
# translation list (50 --> 100)
vlan create vlan-table vid 50
vlan set vlan-table vid 50 member 0,4
vlan set vlan-table vid 50 untag-member 4

svlan set vlan-conversion c2s entry 0 member 0
svlan set vlan-conversion c2s entry 0 enhanced-vid 50
#svlan-index is the svid=100's index in the 64-svlan-entry
svlan set vlan-conversion c2s entry 0 svlan-index 1
exit
EOF
```

How to configure IGMP Relay feature

IGMP Relay Agents for LAN

IGMP join/leave frames can be routed from LAN ports to CPU for multicast filter setting processing.

If a join is received on any LAN port (requires that IGMP snooping is enabled), it will be processed by the CPU, and the associated multicast group is defined in the switch core. It means that any multicast received on WAN port associated with this group will be forwarded to the member ports only.

On the contrary any IGMP leave frames will tear down the associated group from the switch core (unless more than 1 port is a member).

IGMP snooping is enabled with the following commend

```
# Start IGMP snooper
/etc/init.d/igmp start
```

The following limitation is observed to the IGMP snooping feature: “if untagged VLAN is defined for LAN ports, and management of the device is on any tagged VLAN, the relayed IGMP frames will get the VID of the management VLAN.
If untagged IGMP frames are desired on WAN, the following script should be added to the configuration file (this will bypass the tagging egress on WAN port):

```
# Enable LAN to WAN switching
switch --enable-lan

conf <<EOF

# step 1, add an acl rule to match IGMP packet and source port = CPU
port 6 # IGMP protocol is in IPv4 header offset 8 field-selector set
index 1 format ipv4-header offset 8

# enable cpu port 6 acl state
acl set port 6 state enable

acl clear template
acl set template user-field 1
```

```
acl add template entry 3

acl clear
# use the template index 3
acl set rule template entry 3
acl set rule state valid
# port 6 will hit this acl rule
acl set rule port 6
# IGMP protocol = 0x02
acl set rule user-field 1 data 0x0002 mask 0x00ff

acl clear action
# latch index to classify rule
acl set action latch-index

acl add entry 11

# step 2, add a classify rule to remove IGMP packet's vlan tag #
enable classfiy classf set cf-sel-port pon enable

classf clear
classf set rule direction upstream
# the packet which match acl index 11 will hit this classfiy rule
classf set rule hit-acl data 11 mask 0x3f classf set upstream-action
cvlan-act del classf set upstream-action svlan-act del classf add
entry 2

exit
EOF
```

It is possible to enable support for blocking of unknown multicast received on the WAN port (port 4), via the following commands:

```
# the packets which are IPv4 will drop
conf l2-table set lookup-miss port 4 ip-mcast action drop

# the packets which are IPv6 will drop
conf l2-table set lookup-miss port 4 ip6-mcast action drop

# the packets which are not IPv4 or IPv6 will drop
conf l2-table set lookup-miss port 4 multicast action drop
```

How to configure DHCP Relay feature

DHCP Relay Agents for LAN

By default only the DHCPv4 agent will be started if nothing else is configured.

What agents to start is configured by setting ***dkt_ip_mode***

Unset or zero, IPv4 only

```
fw_printenv dkt_ip_mode 0
```

IPv6 only, not supported.

```
fw_printenv dkt_ip_mode 1
```

Dual stack mode

```
fw_printenv dkt_ip_mode 2
```

DHCPv4 Relay Agent

Forwarding rules

In a pure IPv4 setup *dhcprelayd_ext_rules* causes the DHCPv4 Relay Agent to use the files `/mnt/flash/saved_configuration/dhcprelayd_start` and `/mnt/flash/saved_configuration/dhcprelayd_stop` whenever `/etc/init.d/dhcprelayd start / stop / restart` is executed.

This allows a customer to setup special forwarding rules for DHCPv4 bootp

Circuit ID

If the files `/tmp/dhcprelayd-t[0-3].txt` exists the content of these files will be used for option 82 else they must be given using the command line using options --t1 to --t4

On the fly ACL creation

It is possible to setup on the fly ACL rules by setting the parameter *dhcprelayd_filtering*

```
fw_setenv dhcprelayd_filtering 1
```

This causes the file */mnt/flash/saved_configuration/dhcprelayd_filter* to be called at every DHCP event

dhcprelayd_filter <action> <port> <IP address>

<action >
1: Discover
2: Offer
3: Request
4: Decline
5: ACK
6: NAK
7: Release
8: Inform

<port>
0 - 3 is LAN
4 is WAN

<IP address>
ciaddr in dotted notation

DHCPv6 Relay Agent

Intro

Requires dkt_ip_mode 2 for the time being but to provide full documentation assume dkt_ip_mode 1 to be supported.

If only dkt_ip_mode is set to enable IPv6, default forwarding rules will be used allowing IPv4 and/or IPv6 DHCP to be passed from LAN to WAN and reverse.

External custom rules

The *dhcprelayd_ext_rules* applies to IPv6 too.
The files used are the same as for IPv4 described above, only the content must be changed to include IPv6.
It is the resposibility of the customer to provide forwarding rules matching the setting of *dkt_ip_mode*

Remote Id - DHCPv6 option 37

The content of the Remote Id option can be configured in two ways:
By using the files
`/tmp/dhcprelayd6-c[0 - 3].txt`

or
Command line options --c1 to --c4

The files `/tmp/dhcprelayd6-c[0 - 3].txt` might be created when the external forwarding files are executed or from a custom config.sh file

The files `/tmp/dhcprelayd6-c[0 - 3].txt` are only read if `dhcprelayd_ext_id` is set as `fw_setenv dhcprelayd_ext_id 1`

Special feature
Customers in need of putting Remote Id into option 18 Interface Id may do this by executing
`fw_setenv dhcprelayd_rem_id_opt 18`

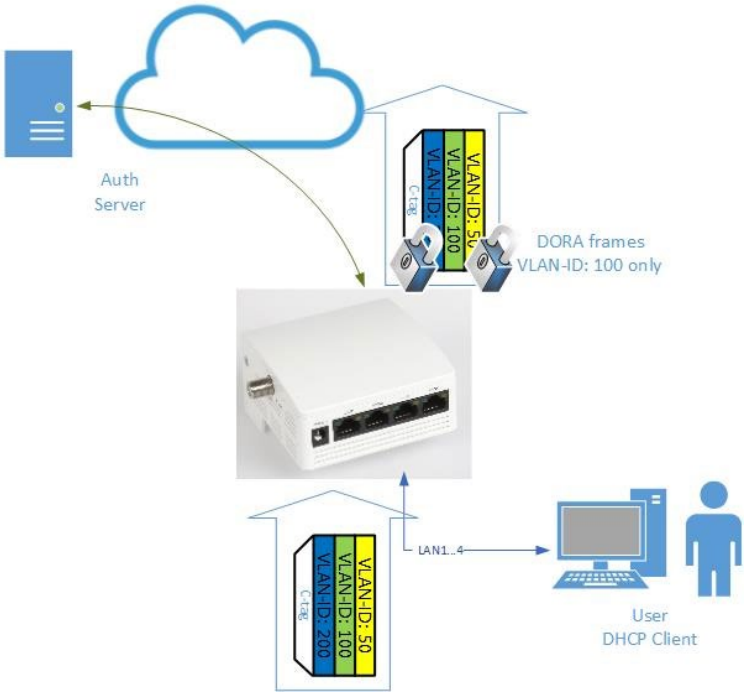
This causes the DHCPv6 Relay Agent's usage of option 18 and 37 to be swapped.
By default option 18 is used by DKTComega to store port number and MAC address of client being served.

WARNING and BE AWARE:
This only works for DHCPv6 servers returning option 37 "as is". This requirement is only a "MAY" in the RFC. You most likely need to patch your DHCPv6 server to make it return option 37

How to select DHCP DISCOVERs from “valid” VLAN versus “invalid” VLAN when using DHCP relay feature (DHCP option 82)

Applicable to firmware release 05_22 and later

This relay feature is integrated in the DHCP option 82. Typically with DHCP Option 82 all DHCP DISCOVERs/REQUESTs are relayed to CPU. With this add-on feature, the operator can specify from which VLAN(s) they would like to have this support. DHCPs received from other than the VLAN(s) defined will be dropped. Notice following example is made with VLAN vid 100.



The following flag is to be set in order for specific DHCP relay function (other than default) to work

```
fw_setenv dhcprelayd_ext_rules 1
```

Please copy text between -------- below into a file on device flash disc, alternatively tftp the file from TFTP server: /etc/config/flash/saved_configuration/dhcprelayd_start

```
# After download or copy, please start DHCP relay function.
/etc/init.d/dhcprelayd start
#--------
conf <<EOF
# Define template for forwarding rule
field-selector set index 1 format ipv4-header offset 8
field-selector set index 2 format ip-payload offset 2
acl clear template
acl set template user-field 1
acl set template user-field 2
acl set template ctag
acl add template entry 1
acl set rule template entry 1
acl set rule state valid
acl set rule port 0-3

# ipv4-header offset 8 value = 0x1100 meas UDP
acl set rule user-field 1 data 0x1100 mask 0xff00

# UDP source port = 0x44 means DHCP Client
acl set rule user-field 2 data 0x0044 mask 0xffff
acl set rule ctag data vid 100 priority 0 cfi 0 mask vid 0xfff priority 0x0
cfi 0x0
1

# DHCP Server trap to CPU
field-selector set index 1 format ipv4-header offset 8
field-selector set index 2 format ip-payload offset 2
acl clear template
acl set template user-field 1
acl set template user-field 2
acl add template entry 1
acl set rule template entry 1
acl set rule state valid
acl set rule port 0-3
# ipv4-header offset 8 value = 0x1100 meas UDP
```

```
acl set rule user-field 1 data 0x1100 mask 0xff00
# UDP source port = 0x43 means DHCP Server
acl set rule user-field 2 data 0x0043 mask 0xffff acl clear action
acl set action redirect port 6
acl add entry 2

acl set port 0-3 state enable
acl set port 0-3 permit enable

# Define rule for how to handle untagged frames, notice that ACL index
number must be higher than previous ACL rules.

acl set rule template entry 1
acl set rule state valid
acl set rule port 0-3

# ipv4-header offset 8 value = 0x1100 meas UDP
acl set rule user-field 1 data 0x1100 mask 0xff00x

# UDP source port = 0x44 means DHCP Client
acl set rule user-field 2 data 0x0044 mask 0xffff
acl clear action

#acl set action trap-to-cpu
#acl set action redirect port 6
acl set action drop
acl add entry 4
exit
EOF

#--------
```

How to change management vlan from native to specific vlan, once and for all?

Device typically will boot in native vlan, however this can be changed, either by applying DHCP option 125 in DHCP offer or using the following syntax in configuration script

```
# Please specify the VLAN VID within the <>

VID=`fw_printenv vlan`
   if [ "x$VID" = "x" ]; then
       fw_setenv vlan <VLAN VID>
       reboot
   else

…# some other commands
   fi
```

Comment to Dying gasp support

Dying Gasp OAM packet

If OAM is enabled (default disabled), then a Dying Gasp packet will only be send after the OAM controller enters an operational state.

To avoid this or simply to receive an OAM packet whenever Dying Gasp is activated then execute

```
# echo 1 > /proc/dkt_config/force_dying_gasp_oam
```

This forces an OAM packet to be sent at power out regardless of OAM state or usage.

Fine tuning Ingress rate limitation

Many end customers use various speed test tools to check their Internet speed connectivity. The switch core is configured for raw layer 2 forwarding. It means the ingress rate may not always seem accurate to the customer. The following example for LAN port 1.

```
switch --enable-lan
conf bandwidth set egress port 0 rate 11000
conf bandwidth set ingress port 0 rate 11000
```

It is possible to fine tune the switch ingress buffer, please add the following command to the configuration file

```
conf register set 0x2308c 0xba
```

Alternatively if flow control (asymmetric) is enabled on the port, it will give a more accurate result:

```
conf port set auto-nego port 0 ability 10h 10f 100h 100f 1000f flow-
control asy-flow-control
conf bandwidth set ingress flow-control port 0 state enable
```

Change of DHCP client, broadcast mode → Unicast mode

DHCP multicast or unicast

By default the dhcp client in the device uses broadcasting. The dhcp client in the firmware can be forced using unicasting by setting the parameter **dhcp_unicast**

```
fw_setenv dhcp_unicast 1
```

Change of SNMP user name / password

By default the SNMP feature is enabled with public / private default values. It is possible to change the default values to customer specific values, please e.g. TELNET into target

```
cp /usr/share/defaults/snmp/snmpd.conf /mnt/flash/saved_configuration/
fw_setenv cust_snmpd_conf 1
reboot
```

Then please add the following two lines to /mnt/flash/saved_configuration/snmpd.conf
(vi editor is available on target)

```
rwuser AESUser
createUser AESUser SHA VeryLongAESDemoPasswordForTestingThisToWork AES
```

Finally please reboot the device.

It is now possible to access the SNMP using the new user name and password, please see example below
(examples made from a Linux PC, CPE is at 192.168.10.119)

```
snmpget -v 3 192.168.10.119 .1.3.6.1.4.1.27304.15.1.3.0.0 -l authPriv -u
AESUser -a SHA -A VeryLongAESDemoPasswordForTestingThisToWork -x AES -X
VeryLongAESDemoPasswordForTestingThisToWork -t 60 -r 0
```

Change of MTU size, jumbo frames

The max jumbo size is 16K bytes in the switch. Jumbo frame size is a global setting, you should setting these two commands. The following example is 8000 bytes jumbo.

```
# profile index 0
conf switch set max-pkt-len index 0 length 8000
# profile index 1
conf switch set max-pkt-len index 1 length 8000

# use index 0, or use index 1 for the individual ports
conf switch set max-pkt-len ge port 0 index 0
conf switch set max-pkt-len ge port 0 index 1
```

Suggestion is to use index 0 and index 1 using the same value.