

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

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

How to allow SSH management from one IP address only

=====
=====SSH ACL policy=====

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 means 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 means 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
=====SSH =====

```

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 8864/64

```

# FOLLOWING IS ENABLING DEVICE TEMPLATES
# 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 0xffff8

# 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

```

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 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 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 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 <<EOC

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 1
vlan set pvid port 6 1
exit
EOC
```