T1040 L2Switch Software Support

This document introduces QorIQ T series L2Switch Software architecture, describes how to use L2Swith demo application and SMBStax(L2Switch stack) to implement L2Switch supported features, which invoke APIs provided by L2 Switch user space driver.

1. T1040 L2Switch Ports Connectivity

T1040 L2Switch integrates 8 Gigabit switch ports are external and are connected to external Phys and 2 switch ports (2.5G) are connected to FMan ports, please refer to the following figure.

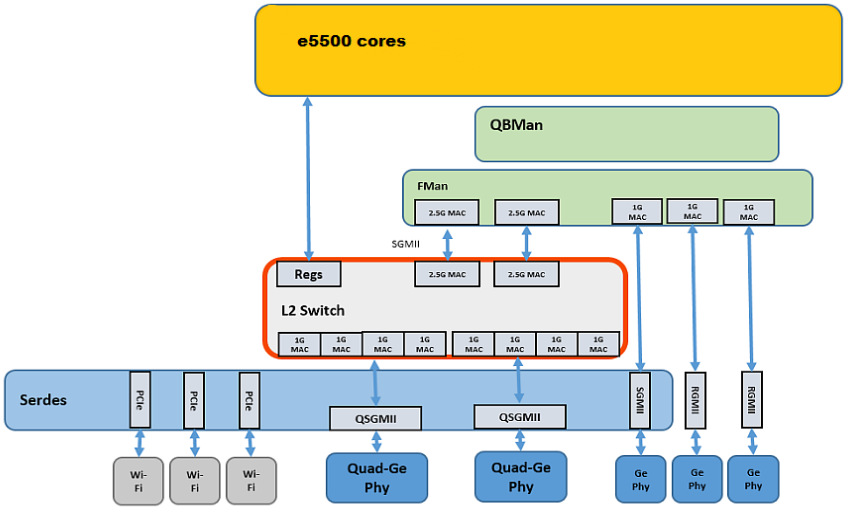

Fig.1 T1040 L2 Switch ports

2. T1040 L2 Switch Software Architecture

The software architecture of T1040 L2 Switch is as the following figure.

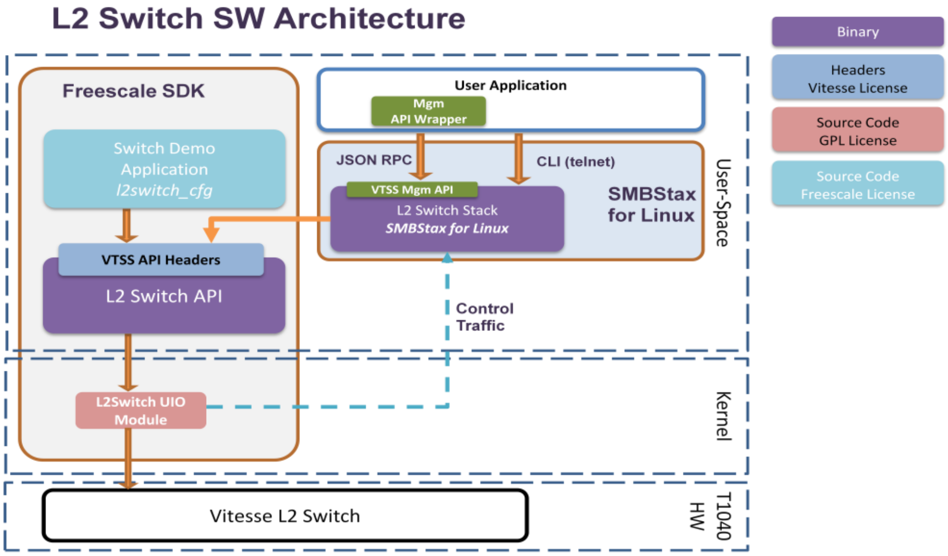

Fig.2 SW Architecture

L2Switch UIO Kernel Module offers access from user space to L2Switch IP and Phy registers, it maps L2 Switch IP registers into userspace, to be accessible to L2 Switch User Space Driver(L2Switch API). Phy registers are accessed through sysfs entries. If offers optimized access to the control traffic, uses HW cache-line support for ingress control traffic, sends/receive frames through a char device.

L2Switch Kernel driver is provided as Kernel module provided in Linux SDK, the package “uio-seville” needs to be added in Yocto rootfs image recipe, in SDK 1.7, uio-seville has already been appended for T1040 platform in meta-fsl-networking/images/fsl-image-private.inc.

```
IMAGE_INSTALL_append_t1040rdb = " ar ceetm l2switch  uio-seville "
```

If the Kernel module is not inserted automatically when Kernel boots up, please insert the module into Kernel as the following.

```
# insmod /lib/modules/3.12.19-rt30-QorIQ-SDK-V1.7+g6619b8b/extra/uio_seville.ko
```

If the module is inserted successfully, the following console message will be displayed.

```
seville ffe800000.l2switch: Found Seville Switch, UIO device - IRQ 26, id 0x099530e9.
```

L2 Switch API delivers a comprehensive, user-friendly and robust function library that supports all the Ethernet switch features for L2 Switch IP integrates in T1040 devices. It includes these basic functions device initialization, port map setup, port reset and configuration, port status polling and configuration based on auto-negotiation, FDB Management, multicast, mirroring, Link aggregation, port VLAN statistics, advance functions such as Quality of Service (QoS) configurations, CPU interface functions for packet control, port filters and Access Control Lists (ACLs), advanced VLAN support, sFLOW.

L2Switch Demo Application is included in Freescale Linux SDK by default, it is built on the top of the L2Switch API, it supports port control, port statistics, static link aggregation, MAC address table operations and VLAN configuration functionalities, does not handle L2 control traffic, only static configuration.

The demo application “l2switch” is included by default in the rootfs generated by Linux SDK 1.7, and it is configured to start at boot. It could be setup manually as the following.

```
root@t1040rdb:~# l2sw_bin -i /tmp/il2sw -o /tmp/ol2sw&

[1] 2142

fifo /tmp/il2sw is used for stdin

stdout is redirected to /tmp/ol2sw

Using UIO: /dev/uio0

Mapped register memory @ 0xb7ae8000

Chipid: 099530e9

root@t1040rdb:~# /etc/init.d/l2switch start
```

```
l2switch: Already running. Use 'restart' to restart the switch.
```

SMBStax for Linux is an L2 Switch control stack that uses the L2 Switch API(User Space Driver) integrated in the Freescale SDK. SMBStax for Linux is not included in Freescale SDK, it is provided as a 32/64 binary and requires a special SLA with Freescale.

SMBStax for Linux provides advanced Layer 2 managed Ethernet switching functionality, with comprehensive capabilities including VLAN trunking, Multiple VLAN Registration Protocol and CLI/JSON RPC Management to access the rich set of features.

SMBStax provides the Speed, mode, flow control, frame size, status statistics Port Control features; QoS feature such as traffic classes, priority, storm control, port policing and shapping; L2 Switching feature such as IEEE-802.1Q, IGMPv3 snooping/filtering/throttling, MVR, Port mirroring; Secure feature such as Port-based IEEE 802.1x, MAC-based authentication, DHCP snooping.

3. L2Switch Demo Application Feature and Usage

Some L2 protocols require a MAC address for each switch port for proper functioning. An address can be specified in u-boot, that will serve as a seed for generating MAC addresses for each port:

```
setenv l2switchaddr 00:11:22:33:44:55
saveenv
```

This address will be available as a sysfs entry:

```
cat /sys/bus/platform/devices/ffe800000.l2switch/mac_address
00:11:22:33:44:55
```

The following method also can be used to add MAC address.

```
root@t1040rdb:/media/l2_switch# l2sw_bin
Using UIO: /dev/uio0
Mapped register memory @ 0xb7799000
Chipid: 099530e9
fsl_dpa ethernet.17 fm1-gb0: Err FD status = 0x00040000
l2switch>
l2switch>mac add 00:e0:0c:00:55:00 8
mac add 00:e0:0c:00:52:00 8
```

Interaction with the L2 Switch is possible via l2switch-cfg script

3.1 Control Ports and Statistics

```
> l2switch-cfg port <port_no> state enable
> l2switch-cfg port <port_no> state disable
>l2switch-cfg port all config show
Port 2 state: enabled link: up mode: auto speed: 1G duplex: full
Port 3 state: enabled link: up mode: auto speed: 1G duplex: full
Port 4 state: enabled link: down mode: auto speed: ??? duplex: half
Port 5 state: enabled link: down mode: auto speed: ??? duplex: half
Port 6 state: enabled link: down mode: auto speed: ??? duplex: half
Port 7 state: enabled link: down mode: auto speed: ??? duplex: half
Port 8 state: enabled link: down mode: static speed: 2.5G duplex: full
Port 9 state: enabled link: down mode: static speed: 2.5G duplex: full
root@t1040rdb:~# l2switch-cfg port 0 stats all packets
```

```
Port 0 statistics:
-------------------------------------------------------
Rx Packets:         0   Tx Packets:         0
-------------------------------------------------------
Port 1 statistics:
-------------------------------------------------------
Rx Packets:         0   Tx Packets:         0
-------------------------------------------------------
Port 2 statistics:
-------------------------------------------------------
Rx Packets:         0   Tx Packets:         0
-------------------------------------------------------
```

3.2 Configure port based VLAN

By default all ports belongs to VLAN1.

```
root@t1040rdb:~# l2switch-cfg vlan show
Printing all VLANs...
VLAN 1: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
```

Adding a port to a specific VLAN:

```
root@t1040rdb:~# l2switch-cfg vlan 10 add 3
Adding port 3 as member for VLAN 10
VLAN 10: 3,
root@t1040rdb:~# l2switch-cfg vlan 11 add 4
Adding port 4 as member for VLAN 11
VLAN 11: 4,
root@t1040rdb:~# l2switch-cfg vlan show
Printing all VLANs...
VLAN 1: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
VLAN 10: 3,
VLAN 11: 4,
```

3.3 Control MAC table

Adding a static entry:

```
root@t1040rdb:~# l2switch-cfg mac add 00:11:22:33:44:55 4 12
root@t1040rdb:~# l2switch-cfg mac dump
Type VID MAC Address Ports
------ --- ----------------- -----
Static 12 00:11:22:33:44:55 4
Static entries: 1
Dynamic entries: 0
```

Deleting a static entry:

```
root@t1040rdb:~# l2switch-cfg mac del 00:11:22:33:44:55
```

Lookup for specific MAC address:

```
Type VID MAC Address Ports
------ --- ----------------- -----
Static 1 00:11:22:33:44:66 5
Static 1 00:11:22:33:44:67 5
Static 1 00:11:22:33:44:68 5
Static 10 00:11:22:33:44:60 3
Static entries: 4
Dynamic entries: 0
root@t1040rdb:~# l2switch-cfg mac lookup 00:11:22:33:44:67
Type VID MAC Address Ports
------ --- ----------------- -----
Static 1 00:11:22:33:44:67 5
root@t1040rdb:~# l2switch-cfg mac lookup 00:11:22:33:44:50
No entries found
```

4. SMBStax Implement L2 Switch functions

SMBStax is provided as a binary package outside the Linux SDK, users need to install SMBStax for Linux package separately.

SMBStax for Linux is started via an init.d script.

```
root@t1040rdb:~# /etc/init.d/vtss_smbstax.sh
```

This section introduces how to use SMBStax ICLI(Configuration of the Industrial Command Line Interface) to implement L2 Switch related functions.

4.1 Mirroring SPAN

For debugging network problems or monitoring network traffic, the switch system can be configured to mirror frames for multiples ports to a mirror port.

Mirror the traffic of Port x to Port y

```
On board:
# conf t
(config)# monitor session 1
(config)# monitor session 1 source interface GigabitEthernet 1/1 both
(config)# monitor session 1  destination interface GigabitEthernet 1/3
(config)# mac address-table static 00:10:94:00:00:01 vlan 1 interface GigabitEthernet 1/1
(config)# mac address-table static 00:10:94:00:00:02 vlan 1 interface GigabitEthernet 1/2
(config)# monitor session 1 source cpu both
```

```
# show monitor

Session 1
---------
Mode                     : Disabled
Type                     : Mirror
Source VLAN(s)          :
Source Ports            :
Both                   : Gi 1/1
Destination Ports       : Gi 1/3
CPU Port                 : both
```

4.2 802.1 X NetWork Access Server Configuration

NAS provides access control on a port basis, 802.1x provides 3 kinds, port based 802.1x, single 802.1x and multi 802.1x. There are 3 terms that are used in the 802.1x context. Supplicant(client PC with some 802.1x software), Authenticator(the switch), Authentication Server(a RADIUS Server). The supplicant is connected to the authenticator/switch on some port, and the authenticator can reach an authentication server. The idea is that when the supplicant wants to assess to the port it sends some message to the authenticator which turn asks the authentication server if this supplicant can be accepted. If so, the authenticator opens the port for the supplicant and communication can begin.

Client->L2switch->Server

Setup XSupplicant for 802.1x client

Setup freeradius for 802.1x server

On the target board

```
(config)# radius-server key 111111
(config)# radius-server host 192.85.1.1
```

Initiate EAP Authentication from 802.1x client.

4.3 QOS Configuration

All incoming frames are classified to a QoS class, which is used in the queue system when assigning resources, in the arbitration from ingress to egress queues and in the egress scheduler when selecting the next frame or transmission. Bandwidth control in the queues can be done by using Policers or Shapers.

L2Switch QoS Classification based on DSCP

Enable DSCP Trust for DSCP at Port 2.

```
# configure terminal
(config)# interface GigabitEthernet 1/2
(config-if)# qos trust dscp
(config-if)# exit
```

Map DSCP Values 4 and 5 to QoS Class 6.

```
(config)# qos map dscp-cos 4 cos 6 dpl 0
(config)# qos map dscp-cos 5 cos 6 dpl 0
```

L2Switch bandwidth control with QoS

Enable policer on Port 2 and set the policer to 2000Kbps

```
(config)# interface GigabitEthernet 1/2

(config-if)# qos policer 2000 flowcontrol
```

4.4 L2Switch MAC Address Learing and Forwarding

Generate stream block 1 on port 1

SMAC: 00:10:94:00:00:01, DMAC: 00:10:94:00:00:02

stream block 2 on port 2

SMAC: 00:10:94:00:00:02, DMAC: 00:10:94:00:00:01

```
#icli

# show mac address-stable

Type    VID  MAC Address        Ports

Dynamic 1    00:10:94:00:00:01  GigabitEthernet 1/1

Dynamic 1    00:10:94:00:00:02  GigabitEthernet ½
```

L2Switch forwarding with unicast

Add static MAC address entry in FDB

```
# icli

# configure terminal

(config)# mac address-table static 00:10:94:00:00:01 vlan 1 interface GigabitEthernet
1/1

(config)# mac address-table static 00:10:94:00:00:02 vlan 1 interface GigabitEthernet
1/2

(config)# end

# show mac address-table
```

Generate streams, and the unicast forwarding is OK.
stream block 1: SMAC: 00:10:94:00:00:01, DMAC: 00:10:94:00:00:02

stream block 2: SMAC: 00:10:94:00:00:02, DMAC: 00:10:94:00:00:01

4.5 L2Switch LLDP-MED feature

LLDP is used to exchange information between network equipment by advertising information about themselves to the link partners at each interface. A LLDP-MED enabled device is capable to convey its capability and configuration to neighbors, and thus the consistent attributes for the same media service or application across the network could be guaranteed.

LLDP-MED provides device’s additional ability for serving specific application. Depending on LLDP-MED operation mode per interface, device is acting as either a connectivity device or an endpoint device on the designated port. The difference between working as "Network Connectivity Device" and "Endpoint Device" is a matter of initializing the LLDP-MED TLVs transmission. A "Network Connectivity Device" is not starting LLDP-MED TLVs transmission until it has detected an "Endpoint Device" as link partner. An "Endpoint Device" will start LLDP-MED TLVs transmission at once.

The following is the procedure setting up LLDP-MED via ICLI

1. Configure terminal

```
# configure terminal

(config)#
```

2. Specifies the fast start repeat count for LLDP-MED

```
(config)# lldp med fast 4

(config)#
```

3. Specifies the geodetic system type.

```
(config)# lldp med datum wgs84

(config)#
```

4. Assign the geographic coordinate value for the device.

Also, the ELIN identification could be specified.

```
(config)# lldp med location-tlv altitude meters 0

(config)#
```

5. Specifies civic location information.

```
(config)# lldp med location-tlv civic-addr country TW
```

```
(config)# lldp med location-tlv civic-addr zip-code 30055
(config)# lldp med location-tlv civic-addr house-no-suffix 23F, 2B
(config)#
```

6. Creates a network policy, which can be assigned to an interface, for a specific kind of application.

```
(config)# lldp med media-vlan-policy 0 streaming-video untagged
(config)# lldp med media-vlan-policy 1 voice tag 123 l2-priority 3 dscp 21
```

7. Specifies the interface on which you are setting LLDP-MED, and enters interface configuration mode.

```
(config)# interface 10GigabitEthernet 1/2
```

8. Selects either "Network Connectivity Device" or an "Endpoint Device" the interface is working as.

```
(config-if)# lldp med type connectivity
```

9. Specifies the discovery ability with which optional TLV to be transmitted.

```
(config-if)# lldp med transmit-tlv capabilities location network-policy poe
```

10. Specifies the media policy that an interface will apply.

```
(config-if)# lldp med media-vlan policy-list 0-1
```

11. Returns to privileged EXEC mode.

```
(config-if)# end
```

12 Saves your settings in the configuration file.

```
# copy running-config startup-config
```