Unmanaged Ethernet Redundant Ring

A Simple and Cost Effective Solution using the KSZ8863 / KSZ8873 3-Port Ethernet Switch family

A WHITE PAPER

By Mike Jones
Senior FAE, Micrel Inc.

Micrel, Inc. • 2180 Fortune Drive • San Jose, CA 95131 • USA • tel + 1 (408) 944-0800 • fax 1 (408) 474-1000 •
http://www.micrel.com

Introduction

Unlike office Ethernet 'star' networks, industrial control applications tend to favour 'ring' topology. The 'ring' simplifies cabling and provides inherent redundancy. The basic building block for a 'ring' network is the 3-Port switch as shown in Figure 1 below.

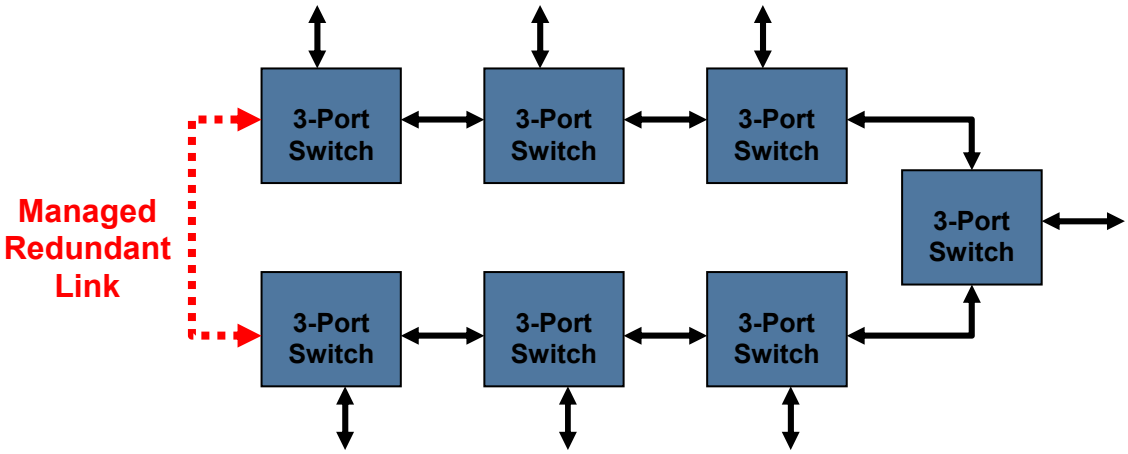


Figure 1. Ring Network using Ethernet 3-Port Switch

Managing the Ethernet Ring

Unlike a token-ring network, which has been the basis of much of the evolution of the IEEE 802.3 specification it is in fact forbidden to configure Ethernet as a true ring. Any loops within an Ethernet network will result in the duplication of packets that are forwarded in endless loops, quickly degrading the bandwidth and efficiency of a network. However, simply by 'breaking' and managing one of the links in the ring, redundancy can be provided for a single point failure.

There is no commonly available standard that deals with the management of Ethernet Ring topology. Various different techniques can be used for the implementation of network ring management. One popular protocol is Spanning Tree or Rapid Spanning Tree. Spanning Tree Protocol (STP) and Rapid STP, IEEE 802.1d and 802.1w create a 'spanning tree' within a mesh network of Ethernet switches by blocking duplicate links, to form a single active path between any two network nodes. The redundancy in the network can provide automatic backup paths if any of the active links fail.

Using STP as the basis for redundancy is relatively expensive in the overheads for the software stack and processor power required to implement the solution. This is undesirable for simpler applications which either do not require or simply cannot afford such performance. For such applications a means of providing reasonable redundancy performance in the ring with minimal management would be highly desirable.

MAC Source Address Filtering

A hardware mechanism of 'Learning' and 'Forwarding' is implemented in all common Ethernet switches today. A switch will 'learn' and then store the ingress packet MAC Source Address and the associated port in a 'Forwarding' Table. A port forwarding decision is then made by looking up the packets MAC Destination Address in the 'Forwarding' Table. If a match is found then the packet is forwarded to the associated port in the table entry. Failure to find a match results in the packet being broadcast to all egress ports except the port it arrived at.

With this mechanism the MAC Source Address is only ever learnt and never used in the decision making when forwarding the packet. However, if the switch could filter packets based on the MAC Source Address (instead of the MAC Destination Address) a true ring network can be realised. Now the switch can detect and filter (drop) any packet that arrives with a MAC Source Address matched to the local processor MAC Address. As a consequence packets are always removed from the ring following one complete loop. Figure 2 shows such an example of how this can work.

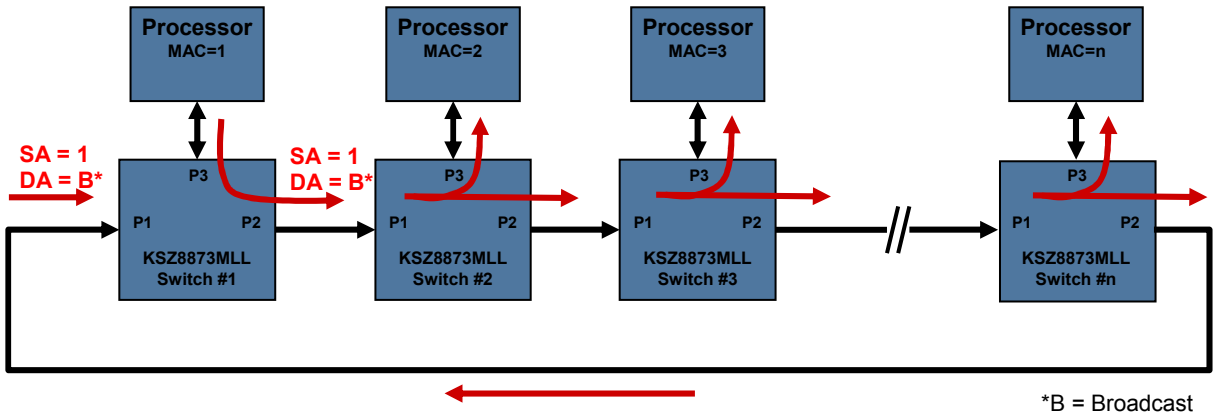


Figure 2. True Ethernet Ring Network using Source Address Filtering

1. Switch #1 receives broadcast packet at port 3 (processor) with Source Address 1
2. Packet is forwarded along the ring until it arrives back at Switch #1
3. Switch #1 has enabled Source Address Filtering on MAC Address 1
4. Packet is then dropped, preventing it from continuing in an endless loop

Unfortunately Fast Ethernet switches today do not offer Source Address Filtering, except for the KSZ8873 and KSZ8863 families of 3-Port Ethernet switches from Micrel. These switches provide two independent MAC Addresses that can be programmed and used for MAC Source Address filtering.

Redundancy using MAC Source Address Filtering

The simplest method of providing redundancy with MAC Source Address Filtering is by sending packets received from the processor in both directions on the ring. Here even if there is a fault on one of the links at least one packet will always get to the destination. Fault detection can also be alerted when a processor only receives a single packet instead of the usual duplicates from either direction. Figure 3 below shows and example of this redundancy method.

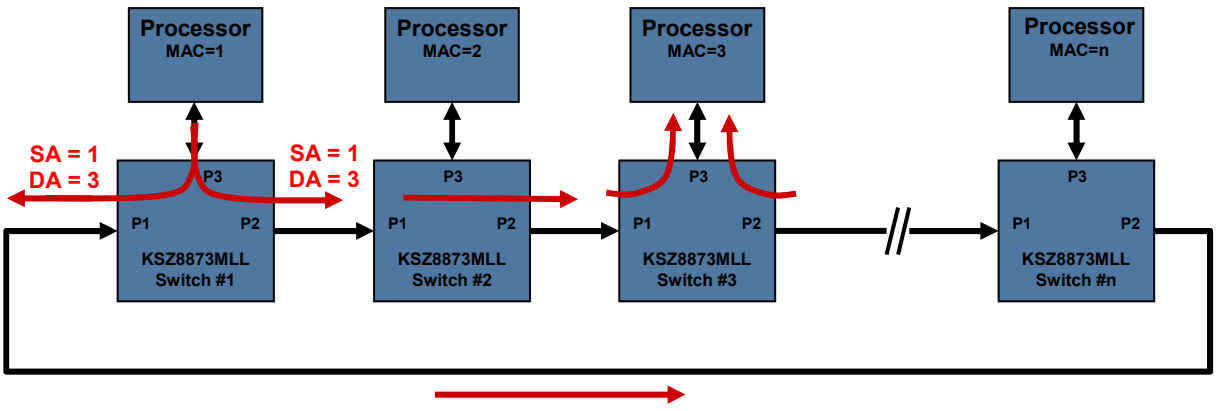


Figure 3. Redundancy using Source Address Filtering

The disadvantage of this method is that receiving duplicate packets increases the processor work load significantly and the bandwidth of the network is effectively halved. Sending duplicate packets on the ring in both directions can also cause 'confusion' for the switches learning mechanism. Here the switch will receive packets with the same MAC Source Address from two different ingress ports, potentially leading to one of the packets being dropped as a 'local packet'. A packet is defined as a 'local packet' and dropped by the switch if the destination port from the 'Lookup' table matches the port from which the packet originated. Figures 4a and 4b demonstrate an example of this potential issue.

Now there is no way of distinguishing whether a dropped packet is the result of a link failure or just this issue.

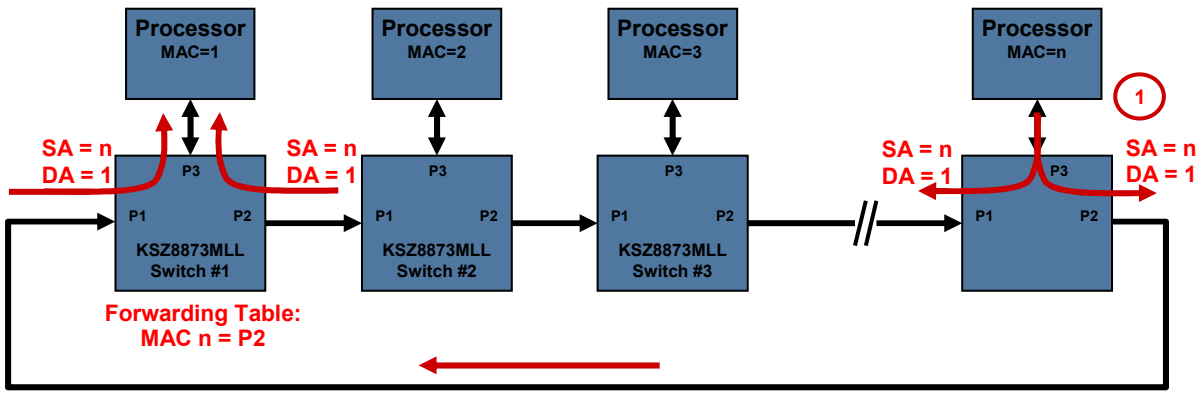


Figure 4a. Potential Redundancy Issue using Source Address

1. Packet '1' received by Switch #n at port 3 (processor) destined for Switch #1
    - Switch #n forwards in both directions (ports 1 & 2).
2. First packet arrives at Switch #1 port 1 (shortest distance)
    - Forwarding Table learns, MAC n = Port 1 and updates
3. Other packet arrives at Switch #1 port 2 later
    - Forwarding Table learns, MAC n = Port 2 and updates

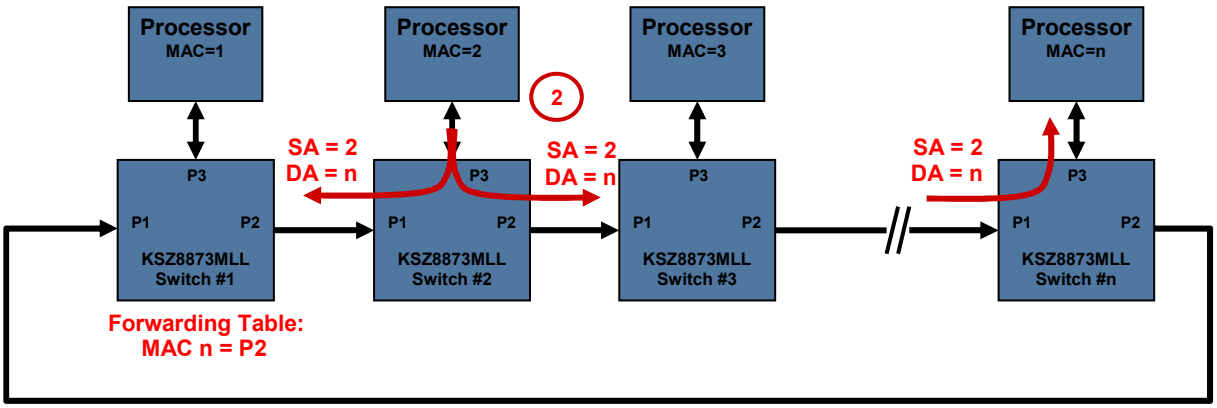


Figure 4b. Potential Redundancy Issue using Source Address Filtering

1. Packet '2' received by Switch #2 at port 3 (processor) destined for Switch #n
   - Forwards in both directions (ports 1 & 2).
2. Packet arrives at Switch #1 port 2 and *DROPPED*
   - 'Local Packet' - Destination Address in Forwarding Table = Ingress Port (P2)
3. Other Packet arrives at Switch #n port 1 OK and forwards to port 3

To ensure that both packets always reach the destination switch, if no link failure exists, you need to disable learning. By disabling learning the 'Forwarding' table will always be empty and hence, incoming packets can never be 'local' and dropped. The major drawback to this approach is an increased overhead to the processor. Now all incoming packets will be forwarded to the processor port (port 3) irrespective of whether or not they are destined for the processor. The KSZ8863/73 switch family can overcome this short fall by the processor manually programming the Forwarding table to ensure any 'unknown' packets are passed on the ring and not to the processor port.

Enhanced Redundancy using MAC Source Address Filtering

The one major disadvantage of the method previously described is that packets destined for the local processor are duplicated, increasing its work load. In the following method we enhance the configuration by:

1. 'Disabling the receiver' on either port 1 or 2 for each switch in the ring
    - Packets will only travel in single direction on the ring
    - Destination processor only receives single packet, reducing work load
    - Disabled port is used for redundancy in fault condition
2. Taking advantage of KSZ8863/73 Link Interrupt pin to signal a network fault.

This solution uses very minimal management by the processor and offers extremely quick fault recognition and correction, even in a large ring network, as demonstrated in Figures 5a and 5b.

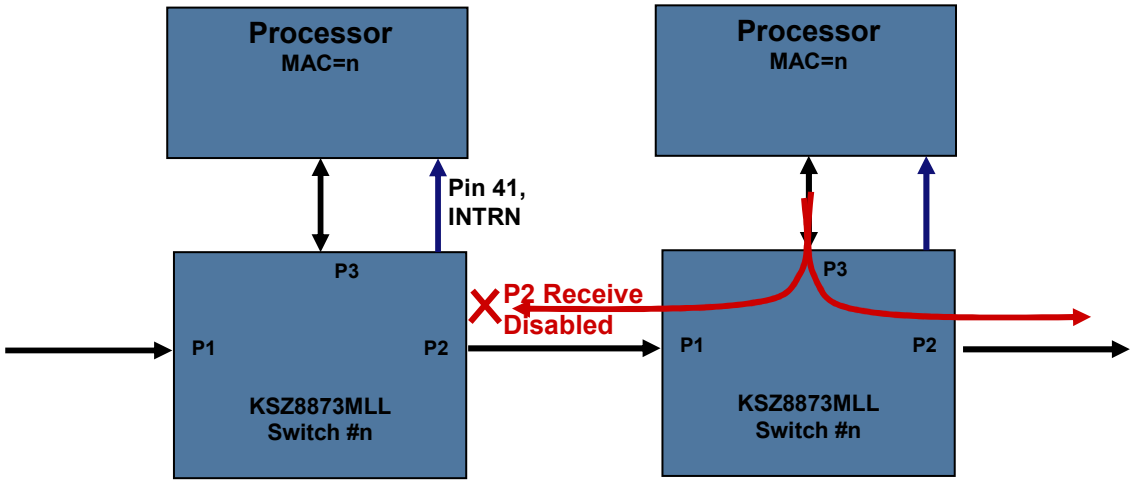


Figure 5a. Initialisation of Switch – Port 2 Receive 'disabled'

Initialisation:

1. Disable learning on ports 1 & 2 of each switch *(Port Control Reg2.0 =1)*
2. Configure the Static MAC Forwarding Table to forward packets with DA = local uP to port 3
3. Enable 'Unknown MAC Address Forwarding' *(Reg14.7 = 1 and Reg14.2-0 = 011)*
4. Disable Port 2 Receive *(Port 1 Control Reg2.1 = 0)*
    - Other methods do exist including tail-tagging, VID etc.
5. Enable Link Status Interrupt *(Reg187.1:0 = 1,1)*

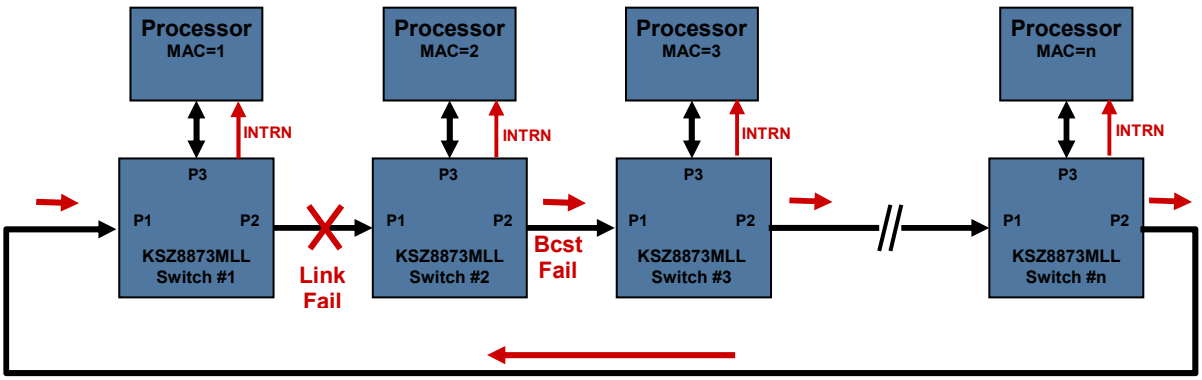


Figure 5b. Switchover

Fault Detection Routine:

1. All traffic is received on port 1
    - Traffic travels in a clockwise direction
2. Link between Switch #1 and Switch #2 fails
3. Switch #1 & #2 Interrupt active
    - Perform switchover routine

Switchover Routine:

The Switchover routine is simple and acts with minimal latency and processor workload.

1. Read Interrupt Register *(Reg 188)*
2. If Port 1 link down then
    - *Send a broadcast message to notify all switches to enable Port 2 Receive*
    - *Enable Port 2 Receive (Port 1 Control Reg2.1 =1)*
    - *Write 0xff to register 188 to clear interrupts*
3. If Port 2 link down then
    - *Ignores 'enable port 2 Receive' request*
    - *Perform Fault Diagnostics*

Fault Correction Routine:

The KSZ8863/73 offers a cable diagnostic feature LinkMD™ that can be used (in this example Switch #1) to identify and locate location of fault.

When link has been fixed the Interrupt signal will indicate a 'link change status' ie. Link is good.

1. Read Interrupt Register *(Reg 188)*
2. If Port 1 link good then
    - *Send a broadcast message for all switches to disable Port 2 Receive*

All traffic travels back in the clockwise direction again……. until the next fault!

Switchover Latency:

Latency = $T_{interrupt}$ + $T_{read\ interrupt}$ + $T_{broadcast\ message}$ + $T_{enable\ P2\ Receive}$

$T_{interrupt}$ = 100us approx (assumed for this example)

$T_{read\ /\ write}$ = 4.8us for 5MHz SPI clock
(Single register write = 3bytes @ 5MHz)

$T_{message}$ = (n-1) x 7.7us
Delay from switch #2 transmits message to switch#n-1 receives it
Assumes 64-byte packet = approx 7.7us switch latency

Therefore:

Latency = 100us + 4.8us + (n-1)x7.7us + 4.8us

For example, 16 node ring:

Latency = 225us approx

Maximum number of nodes in a ring:

For example a 1ms switchover delay:

Latency = $T_{interrupt}$ + $T_{read\ interrupt}$ + $T_{broadcast\ message}$ + $T_{enable\ P2\ Receive}$

Therefore:

1ms = 100us + 4.8us + (n-1)x7.7us + 4.8us
n < (890.4 / 7.7) + 1

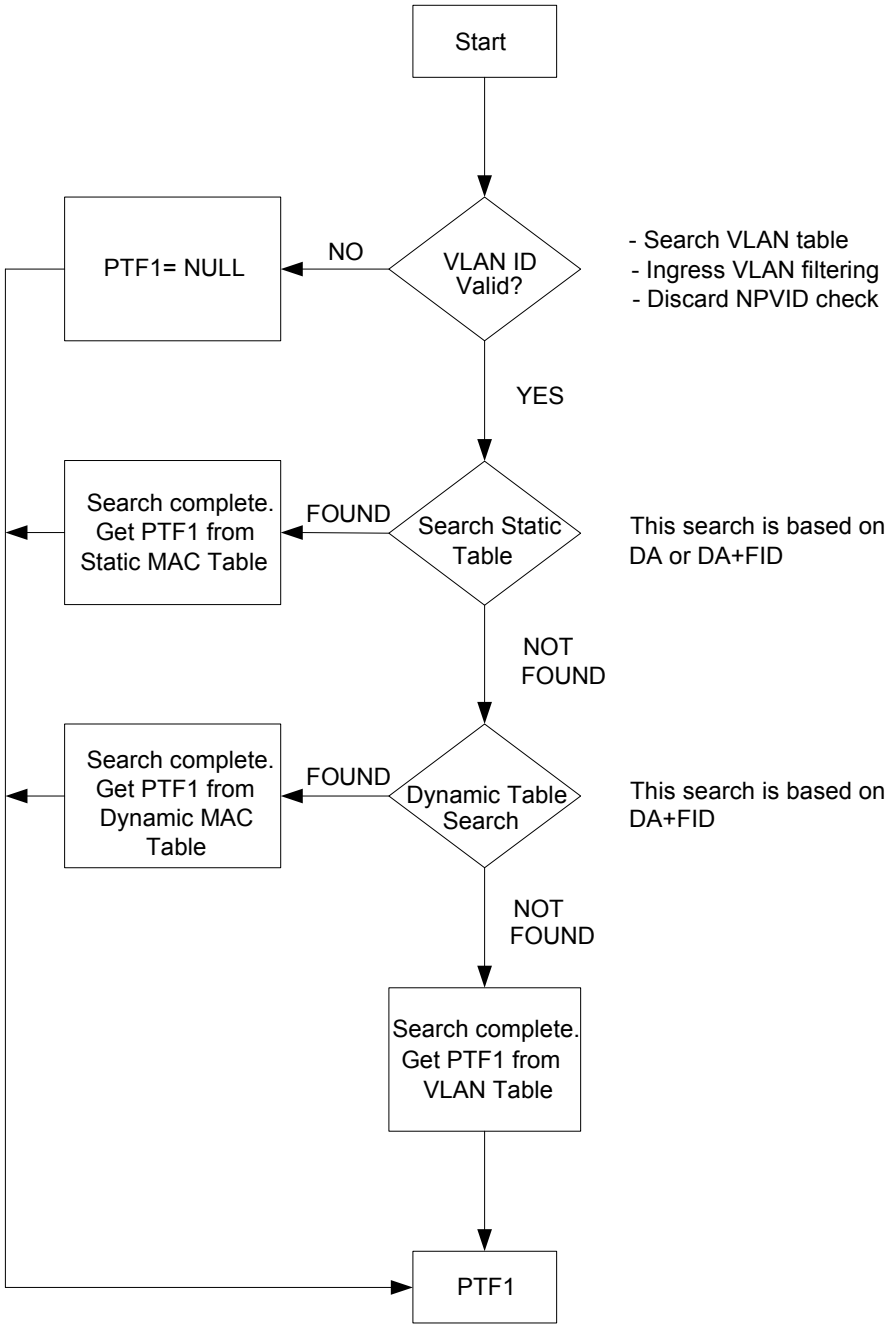
Maximum number of nodes = 116

Conclusion

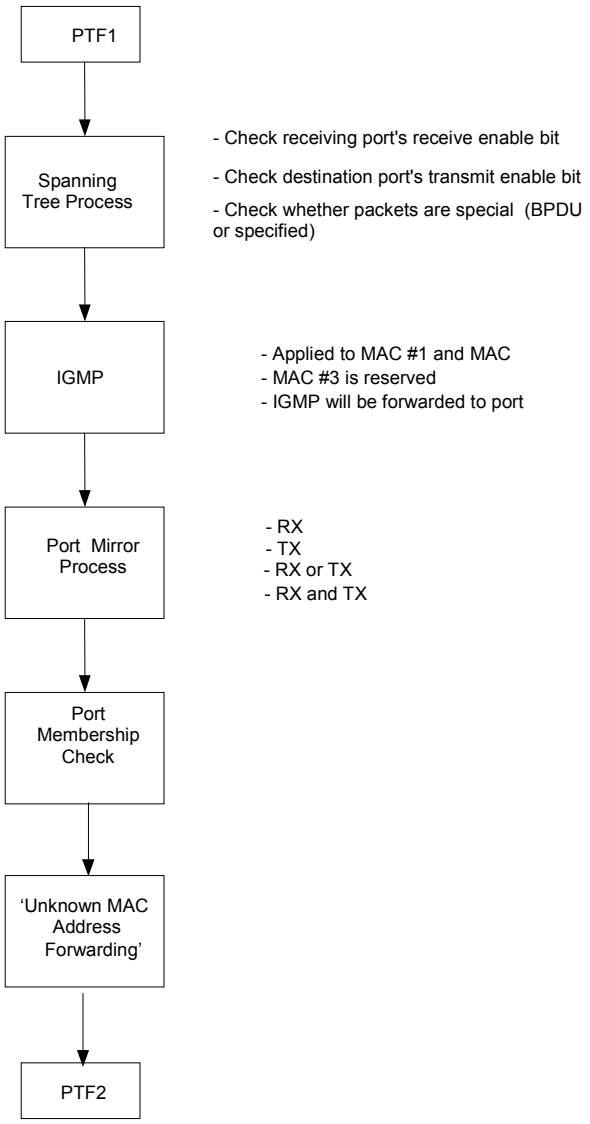
'Ring' topology is commonly implemented in Industrial Control networks. However, such a configuration is generally forbidden in Ethernet networks. Packets will endlessly circulate the network in the presence of a loop, quickly resulting in bandwidth degradation. Complex high layer protocols, such as Spanning Tree, can be utilised to manage the ring and provide redundancy via the 'broken' link. Complexity and cost are significantly increased as a result. In this paper we have shown that by providing the unique feature 'Source Address Filtering' supported by the Micrel KSZ8863 and KSZ8873 3-Port Ethernet switch family, a simple and cost effective alternative is available.

For further details on Micrel Ethernet Solutions go to http://www.micrel.com/ethernet.

Appendix – KSZ8863/73 Ethernet Switch Forwarding Algorithm



Search engine looks up the VLAN ID, Static MAC Table and Dynamic Mac Table for the destination address, and comes up with “port to forward 1” (PTF1).



PTF1 is then further modified by Spanning Tree, IGMP Snooping, Port Mirroring, Port VLAN and 'Unknown MAC Address Forwarding' processes to come up with "Port to Forward 2" (PTF2).

The packet is sent to PTF2.