



Mechanisms to Resolve Issues Faced by Payload Loopback in Bridge Network


Linda Dunbar: ldunbar@futurewei.com

Bob Sultan: bsultan@futurewei.com


Steve Plote: steve.plote@lglass.net

Lucy Yong: lucy.yong@wiltel.com

Sept, 2005

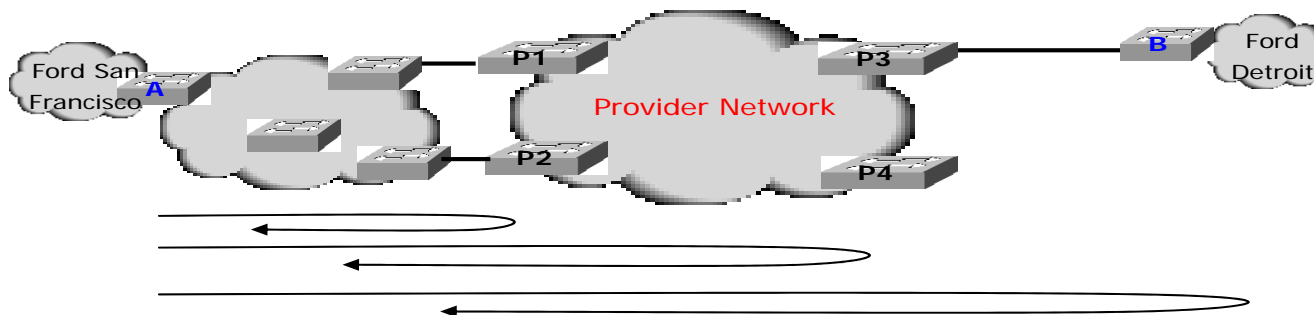
- 
- **Why need payload loopback**
 - **Issues facing payload loopback in packet based networks**
 - **Mechanism to resolve the issues facing payload loopback in packet based networks**
 - **Request to IEEE802.1ag**

Network Maintenance and Operation Need


- 
- **The 802.1ag defined Loopback <==> Ping used in IP network**
 - The current 802.1ag loopback is more like Ping in IP network. It is not payload loopback.
 - **While Ping like messages can test network connectivity, it can't test payload packet drops, bandwidth issues, or network bottleneck triggered by particular traffic flows.**
 - **Following maintenance need can't be addressed by Ping or Echo like messages:**
 - When customer encounters packets drops or bandwidth between two end points (A and B) not meeting the SLA, customer and provider need to trouble shoot where the packet drops occur and where the bottleneck is for the bandwidth drop.

Payload Loopback

- **Two types of payload loopback**
 - **Payload Loopback initiated by network management system (NMS)**
 - Mainly used for trunk circuits, such as OC-n
 - **Payload Loopback initiated by embedded code without network management system intervention.**
 - Mainly used for customer circuits, such as DS1 and DS3
 - Even with NMS triggered loopback, embedded code triggered loopback is still more widely used for user circuits in circuit based networks
- **Why need payload loopback?**
 - To segment the network to small pieces, and to isolate faults
 - To re-produce the problem/faults with the same environment where the problem/faults occurred.
 - Payload loopback has been used by circuit based network for many years. It has been proven that it is very effective in trouble shooting and is widely used by all network operators and end users.

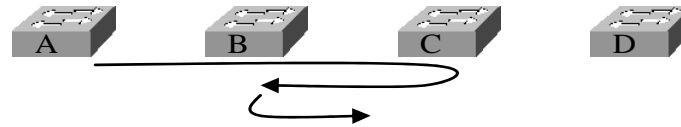


Why Need Embedded Code Triggered Payload Loopback?

- 
- **No need for interoperable network management systems**
 - The loopback can be triggered and go through 3rd party vendor networks without interoperable network management interface
 - As being proven in reality, it is extremely difficult to have interoperable network management systems to manage end to end operations for network with multiple vendors' equipment
 - **Easy, Convenient, and can be performed without provider's intervention**
 - Whenever a user needs to trouble shoot intermittent network problems, he can initiate the loopback independently, without service provider's assistance, which greatly reduce service provider's maintenance cost.
 - **It has been proven in circuit based network that the embedded code initiated payload loopback is especially important for trouble shoot user circuits, like DS1 or DS3.**

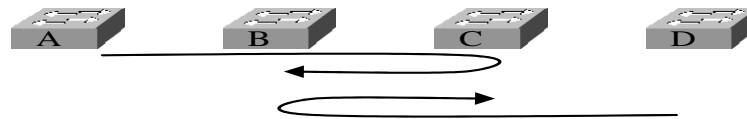
Issues Associated with Payload Loopback in Packet Based Networks

○ Packet Loops



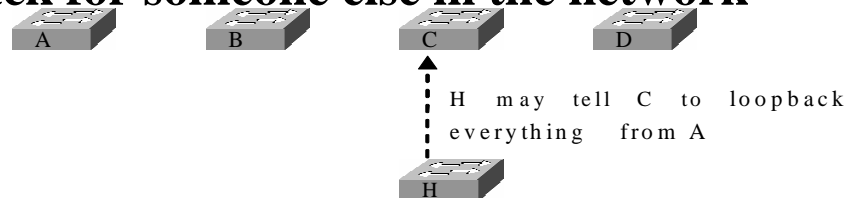
- If C loops back packet from A back to B, B will re-forward the packet back to C.

○ Network overload caused by multiple payload loopback



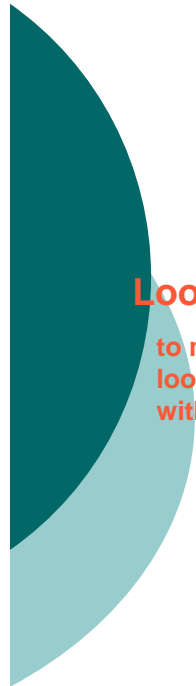
- When A and D both initiated Loopback, the traffic between C and D can exceed the subscribed bandwidth.

○ Hackers initiate loopback for someone else in the network



○ Free run loopback (i.e. never stop) caused by network node or link failure

Simple Steps to Resolve the Issues



Loopback reservation:

to make sure that only one loopback can be activated within a maintenance domain

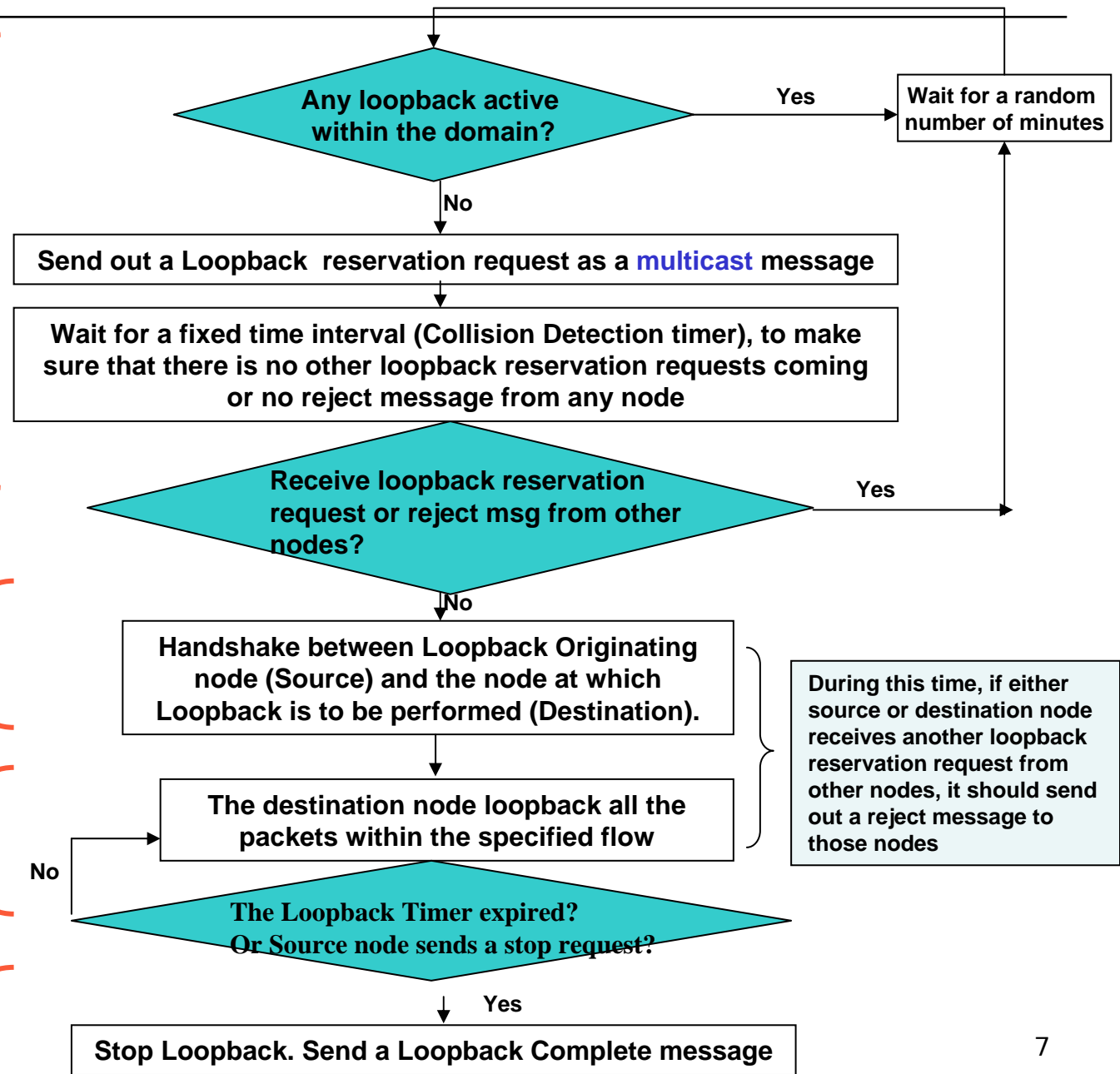
Handshake:

to make sure that both source and destination agree upon the loopback term and confirm the loopback

Actual Loopback:

Destination node swap DA and SA and loopback the specified packets until the loopback stop condition is met

Loopback Completed:



Loopback Reservation




- **Loopback Reservation is:**

- To make sure that only one node can initiate payload loopback within one administrative domain. To prevent network overload caused by multiple loopback being initiated within the same time period
- To prevent unwanted loopback initiated by 3rd party which is not agreed by the source.

- **Information included in the Loopback Reservation request:**

- Source node, destination node which is to perform loopback, VLAN ID, and Loopback Timer (Time to live)

Loopback Reservation Reject

- 
- **Loopback Reservation request can be rejected by the following conditions:**
 - There is already an active payload loopback in the administrative domain
 - The very first node, which still has Loopback Timer not being expired, can reject the Loopback Reservation request. This node will not forward the multicast message.
 - The source or destination node specified by the Loopback Reservation request don't want to proceed with the loopback
 - In case the loopback is initiated by the 3rd party which is not agreed by the source.
 - **Loopback Reservation Reject should be sent to both originating node and the node which is to perform the loopback**
 - The reason that the destination node has to be informed of the rejection is to make sure that the destination node will not perform any loopback which is rejected.
 - This step is to prevent any node from requesting a loopback without the proper reservation being granted.
 - **If source or destination node doesn't grant the loopback reservation, they won't perform any loopback.**
 - **If a node receive a reject message, it will erase the record of the loopback reservation request.**

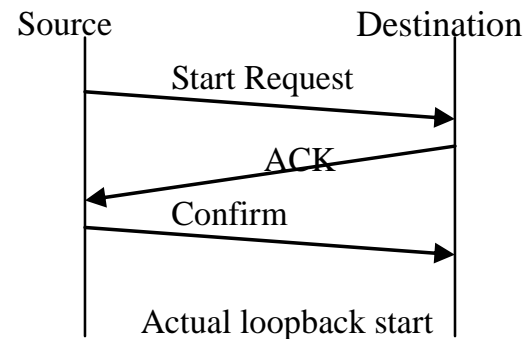
Handshake to Start and Stop the Loopback

- **The conditions to get into handshake stage:**

- The originating node and destination node don't receive any reject message
- There is no other loopback reservation received until the collision timer is expired

- **Handshake between Source and Destination**


- To reach agreement on the loopback term and confirm the loopback
- To make sure that the destination node will not acknowledge any loopback request without prior reservation
- To mark the START and END of the loopback



- **Time to Live:**

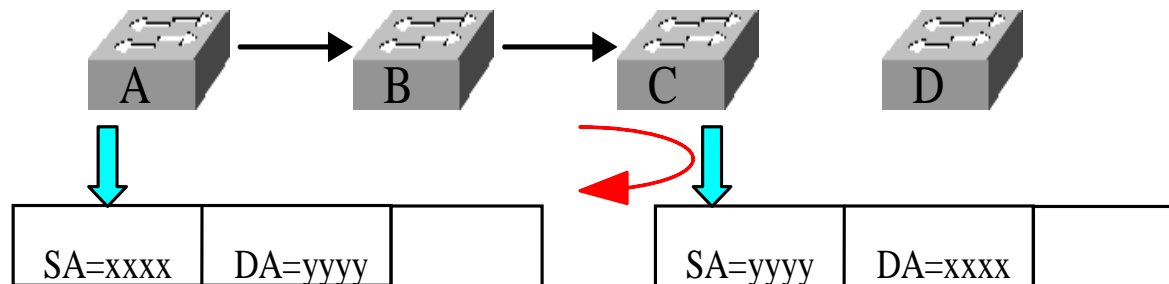
- To prevent free run loopback (i.e. never stop) caused by network node or link failure
- To let all the other nodes keep track of on-going loopback which might be active in the administrative domain.

How the Issues Are resolved?


- 
- **With Loopback Reservation, only one node can initiate payload loopback within an administrative domain**
 - Scenarios 1 & 2: Two nodes send Loopback Reservation at the same time or one node sends out a Loopback Reservation request before receive the other one:
 - Both will receive the reservation request from the other node before the collision timer is expired
 - Both will withdraw, wait for a random number of minutes which is to guarantee that each will send the second request at different time.
 - Scenario 3: if a newly added node, who doesn't have any information on existing loopback, sends out a Loopback Reservation Multicast request, the request will be rejected by the source or destination node, or any node who has record of on-going loopback
 - **Loopback Reservation and Handshake can prevent a 3rd party node to initiate a loopback on behalf of another node who don't agree**
 - Since Loopback Reservation request is a multicast message, the source node can reject the Loopback Reservation request. The reject is sent to both the originating node and the destination node. So Destination node will not perform the requested loopback.

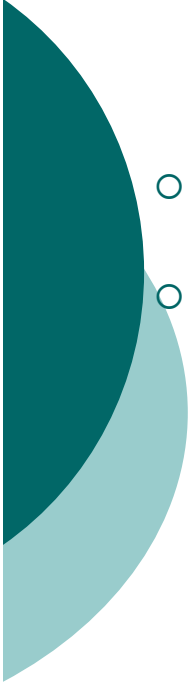
Mechanism to avoid unwanted loops in the network

- **Swap Source and Destination Addresses by the node which perform the loopback**



Impact to Network With Loopback Steps

- 
- **Payload Loopback is an optional feature. There is no impact at all if network operator don't want to enable the feature**
 - **If enabled, the only extra traffic added are 2 multicast messages and 4 unicast messages.**
 - Loopback Reservation Message
 - Multicast message
 - Loopback Reservation reject message
 - To both the originating node and the node which is to perform loopback
 - Loopback Start Request, Acknowledge, and Confirm Messages for proper handshake
 - Loopback Stop message
 - Could be multicast, if needs to stop the loopback timer on other nodes in the administrative domain.
 - But doesn't hurt if it is an unicast message. The only drawback of being an unicast message is that other nodes will wait longer for another loopback request. It doesn't hurt anyone.

- 
- **802.1ag should include the Inband Payload Loopback Protocol to enable embedded code initiated payload loopback**
 - **“Inband Payload Loopback Protocol” consists of**
 - Payload Loopback Reservation
 - Payload Loopback Handshake to start the payload loopback
 - Source and Destination swapping during actual loopback
 - Payload loopback Stop triggered by message or timeout



Thank You