# Diagnosing data dependent and data driven connectivity faults

**Mick Seaman**

This note follows up my presentation to the 802.1 March 2006 meeting and the ensuing discussion. It proposes an extension to the basic capabilities of P802.1ag CFM to support detection, diagnosis, and isolation of connectivity faults that affect frames containing particular data or data patterns. The ideas described arose from the discussion of Linda Dunbar's loopback presentations.

## 1. Introduction

There are two broad types of connectivity faults that affect only frames carrying certain data, addresses, or data address combinations. Simple data dependent faults result in the repetitive loss of each of those frames, independent of any other frames, and are usually the result of simple misconfiguration or of a failure to appreciate the consequences of a configuration option—installing protocol specific filters, for example. Data driven faults are more complex: the presence (or absence) of some data frames cause or contribute to the loss of other frames. While the services supported by bridged networks are notionally data independent, data driven techniques enable enhanced service delivery. To give three examples: multicast frame filtering and consequent bandwidth saving is facilitated by IGMP snooping; stateful firewalls are used to protect users connected to managed services; and efficient allocation of frames to the individual links of an aggregation is often based on spotting conversations by looking at frame data.

Data driven faults can be expected to become the majority of data related faults, as the reasons for simple data dependent errors are easier to find and correct. Data driven faults can be provoked both by errors in complex software, and by unexpected customer behavior. In the absence of real traffic they can be expected to disappear, or to revert to an entirely expected state—no IGMP related multicast traffic, for example. Diagnosis has to be carried out while the network is actually running, and the diagnostic tools themselves must not introduce further data dependent faults. If one segment of a data path is to be tested in one direction, the test frames have to be delivered to (and possibly collected from) that segment without triggering any other data driven behavior on the delivery (and collection) path—it is no good looping back plain data traffic through a firewall, for example, as the firewall will see an IP address moving rapidly from one side to another with a changing MAC address, and will likely block the traffic in consequence.

This note proposes standardization of two additional CFM opcodes†1, together with an encapsulating 'partial mirror' functionality, to enable data testing of selected path segments†2. The goal is to support detection, isolation, and diagnosis of both data dependent and data driven faults.

The purpose of the first additional opcode is to deliver the elements of a frame†3 to a point (a decapsulator) where it can be sent as a normal data frame. In my presentation I called this the "send this back to" opcode, but have shortened this to 'SF' or "send frame" in this note, since the useful functionality is not limited to "sending back". The purpose of the 'mirror' is to extract or copy certain frames from a given service instance, and then to encapsulate those frames using the second opcode and a pre-programmed address associated with the mirror. In my presentation I called this second opcode "sending this back", but have shortened that to 'RF' or 'returning frame' in this note.

The reasons for providing this functionality within the CFM context are:

a) to insulate the SF and RF frames against interpretation by data driven functions and against additional CFM handling;

b) to allow the decapsulator and encapsulating mirror to be positioned at an appropriate MA Level and associated with a MIP or MEP as appropriate;

c) to ensure that the encapsulating mirror does not return CFM frames at its own level (or below).

The next section provides some examples, and is followed by a more detailed description of the SF and RF frames, of the functionality provided by the decapsulator and mirror encapsulator, and by a review of security considerations.

†1 A single opcode with two subtypes is a possibility, but using two does have the advantage of encouraging supportive behavior by other network elements. Some firewalls, for example, may be happy to let SFs 'out' of their protected domain, and let RFs both 'in' and 'out', but would be unhappy to let SFs 'in'.

†2 I am using the term "path segment" to mean "part of a service instance, bounded by MEPs or MIPs. We may already have a name for this, if not it seems an important concept and I would welcome suggestions for a better (and ITU friendly) term.

†3 It is worth noting at the outset that the 'frames' discussed are strictly the parameters of a service request/indication, conveyed on a service instance, and that the mirror is also particular to a given service instance. The proposed mechanism discussed does not provide a way to violate service instance segregation.

## 2. Examples

Figure 2 illustrates testing of the 'forward path' from one end of a point-to--point service instance using a simple 'loopback-like' operation. In 2(a) a mirror is installed at a MIP. Frames sent from system A are 'reflected', i.e. copied and encapsulated behind an RF opcode and sent to an address specified when the mirror was installed, in this case back to A. This allows A to check the connectivity to the mirror for various data frames. Moving the mirror allows testing of successive segments of the data path.

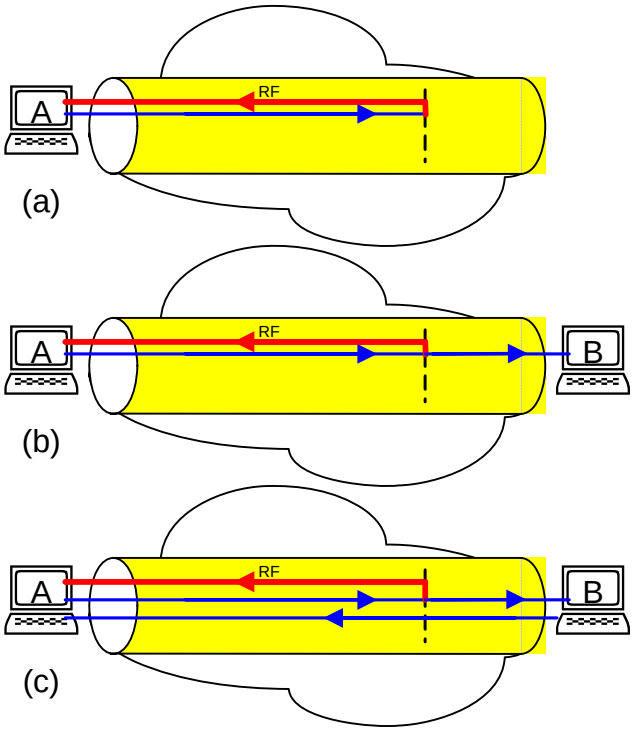


**Figure 1—Forward path testing with a loopback**

The reflected frames, and other frames from A, can also be delivered to the other end of the service instance, represented by system B in Figure 2(b), while frames in the other direction can be delivered as normal, as shown in 2(c).

A 'return path' can be tested without installing a mirror, as illustrated in Figure 2. The frame to be returned is encapsulated after an RF opcode, and sent to a MIP that is capable of decapsulation. As in Figure 2, changing the choice of decapsulator allows successive testing of the path segments.

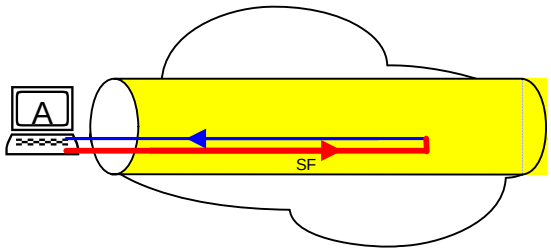


**Figure 2—Return path testing**

Use of both the SF and RF capabilities supports single-ended testing of complete application dialogues. Figure 3(a) shows two systems, A and B, at each end of a point-to-point service instance. Figure 3(b) shows the same application dialogue supported over just part of the path with A and B conveniently located (for diagnosis) at a single customer service interface. A mirror has been installed at a MIP to return frames sent 'in clear' on the forward path from A, while frames transmitted by B are encapsulated and transmitted to the same MIP to be returned in clear.

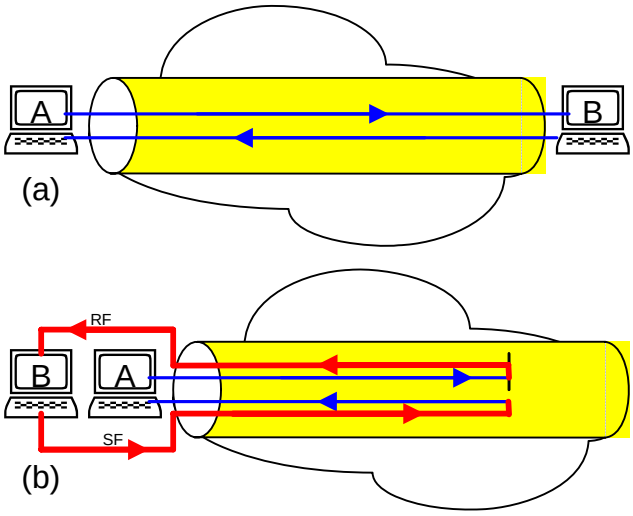


**Figure 3—Single-ended application testing**

Use of both SF encapsulation and a mirror with RF encapsulation allows data driven effects to be restricted to part of the path, as illustrated in Figure 4.

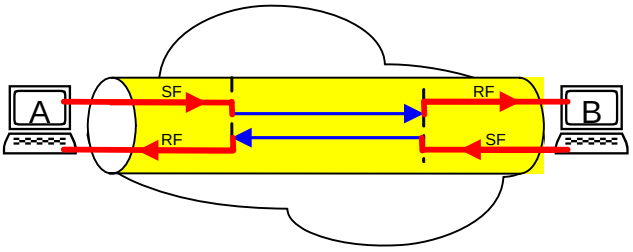


**Figure 4—Partial path testing**

In this case the generality of the 'mirror' function is exploited to encapsulate and send the frame onward.

The use of SF and RF in combination is also interesting when testing a multi-point service instance. A technician located at head office, H in Figure 4, can test connectivity between satellite offices.

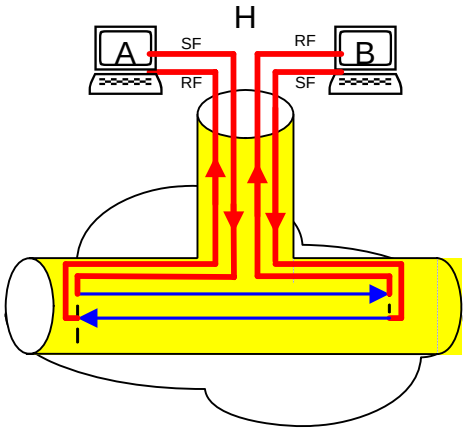


**Figure 5—Remote path testing**

## 3. SF frame encapsulation

The SF (send frame) is destination addressed to the MIP or MEP that is to send the encapsulated frame, and naturally has the source address of the requestor. Apart from the normal CFM format, including MA Level and the SF opcode, it conveys the destination and source address of the frame to be sent by the maintenance point, and the data for that frame.

The addressed maintenance point can be either an 'inbound' MP, i.e. the data flow through the MP proceeds into the MAC Relay Entity of a bridge, or an 'outbound' MP, i.e. the data flow is proceeding from the bridge's MAC Relay Entity in the direction of physical transmission. The SF specifies whether the encapsulated frame is to be sent inbound or outbound (independently of the MP type). On a multipoint service inbound frames will benefit from frame filtering by the bridge.

## 4. RF frame encapsulation

The encapsulated RF (returning frame) has as its destination an address specified when the mirror (or encapsulating point) is created, and the source address of that mirror. As for the SF frame, the mirror can be associated with an inbound or outbound MP and the RF can be specified as being sent inbound or outbound. This is necessary on a point-to-point service as the bridge itself has not necessarily learnt the RF's destination, and can equally apply to a multipoint service that uses only two ports of the bridge.

## 5. Mirror functionality

The 'mirror' functionality provided by an MP that encapsulates a non-CFM frame requires that MP to hold state, and thus may be unattractive for MIPs that support many service instances†1. However in many cases the number is not large, and reflects use of the underlying service by a single customer. The mirror needs to be configured with the following parameters:

a) the mirror's orientation, i.e. does it 'reflect' frames that are 'inbound' or 'outbound'—a given MP could be configured with mirrors for both orientations

b) the destination address for the RF encapsulated (reflected) frame;

c) whether frames are passed through the mirror as well as being reflected;

d) whether frames impinging on the reverse side of the mirror are passed through transparently†2;

e) whether encapsulated frames are to be reflected, i.e. sent in the direction from which they were received, or simply diverted i.e. sent onward once encapsulated;

f) which frames are to be reflected.

The last of these parameters is the most complex, and could lead to very extensive discussion. Reflecting all frames, as might be done for a very simple loopback, could be very performance intensive for the equipment supplying the mirror, as well as adding significantly to the network load in the direction of the reflection. Reflecting frames for specific destination addresses is probably the easiest to manage, but multicast and broadcast frames may also be required for full application testing. A full ACL capability is unlikely to be popular or effective as it could easily impact the performance provided to other users of the equipment. One option would be to consider mirrors as being installed per destination address per direction through the MIP, as use of just two or three of such mirrors would be likely to meet even the most complex testing needs.

It should be clear that a mirror never reflects OAM frames at the same level, and never moves frames from one level to another. Since SF and RF are OAM frames, facing mirrors will not endlessly bounce frames between themselves, and there is no need for an overall configuration protocol to guard against that eventuality.†3

## 6. SF decapsulation functionality

In principle an MP does not have to hold state or participate in any prior agreement to be used as the target for an SF frame that is to be decapsulated, but see Security considerations below.

## 7. Security considerations

All DDCFM 'frames' are confined to a single service instance, they are generated and processed by MPs within the scope of that service instance, and their 'wire format' includes those tags and/or addresses that segregate one service instance from another.

If a service is secured by MACsec running across a provider's network over the service instance between Provider Edge Bridge C-VLAN aware components (see P802.1AE Figures 11-12 and 11-13) then, in the absence of special arrangements, customer systems behind those interfaces will not be able to send or receive SF and RF frames. This is believed to be desirable†4, as it limits the use of SF and RF to maintenance operations initiated at the provider edge equipment.

The use of SF, as described above, does not require holding state at an MP, but it may be desirable to only allow decapsulation of SF frames from certain addresses for transmission to certain other addresses, with those parameters being preset by a secure dialogue with the target MP. The SF capability is powerful and can cause serious disruption, within the service instance, if abused.

## 8. MTU size considerations

A perennial bugbear: but I do not believe that it is worth trying to assemble one frame from two to allow testing to the maximum supportable MTU size. A provider network should be configured with a little size to spare over that offered to its customers so that testing for size related failures is not unduly difficult.

†1 Whether there is one MIP per service instance above a MEP, or one for all service instances, is a moot point if the MIP has no associated state. As has been pointed out there is a considerable difference between having even one state variable with an associated MIB and having none.

†2 Yes, by default. If two potential mirrors are always associated with an MP this parameter is redundant, as its predecessor can be used (for the other paired mirror) to serve the same purpose.

†3 If it were possible to decapsulate RF frames within the network and automatically forward them it would be possible to setup such a data loop, but it is not. Guarding against this eventuality is a benefit of clearly distinguishing SF and RF frames.

†4 To be honest I have not fully made up my mind about this. However constructing a frame to be sent by someone else for successful receipt by MACsec is far from trivial and I cannot recommend it. There is a non-negligible chance of a serious security breach.