RIVERSTONE NETWORKS
ADVANCED TECHNICAL PAPER SERIES

Frame Relay Design Guide

Nick Slabakov, Riverstone Networks

ABSTRACT

This paper explains theory and principles behind frame relay and building networks with frame. Fundamentals related to frame relay are covered before moving on to network topology and the use of routing protocols including OSPF, RIP, ARP and BGP. Example configurations are also given using the Riverstone platform.

Riverstone Networks, Inc.
5200 Great America Pkwy, Santa Clara, CA 95054 USA
(877) 778-9595, (408) 878-6500, www.riverstonenet.com

Version 1.0, 18 Dec 2001

TABLE OF CONTENTS

Introduction

Frame Relay is perhaps the single most popular WAN technology deployed today. The Riverstone RS line is unique among the L3/L4 switches on the market with its WAN support. The purpose of this document is to detail the capabilities of the RS for supporting Frame Relay, and to provide design guidelines for building Frame Relay networks with it.

Many ISPs provide Internet access to their customers via frame relay. They either use carrier frame relay networks and take advantage of their distance-insensitive pricing, or build frame relay backbone on their own and benefit from the statistical oversubscription of trunk bandwidth frame relay allows.

This paper assumes good understanding of network technology and jargon. Only the non-trivial acronyms are explained.

Frame Relay - Basics

Frame Relay is a packet switching technology, which operates at layer 2 of the OSI model (Data Link Layer). It specifies interfaces for connecting user devices to the network, as well as methods for addressing and transport through the network. Frame Relay can be built either by a telco (ILEC/CLEC), or by a service provider or enterprise, using private Frame Relay switches connected with leased lines.

Most often, customers purchase Frame Relay services from carriers. In this document, we will assume that the frame relay network is built by a CLEC/ILEC and the customer (ISP or Enterprise) connects to it. The cases where the customer builds their own frame relay network only simplify the assumptions in this document.

The Basic Terminology

Figure 1 illustrates the components of a Frame Relay network and points where each term applies. CSU/DSU units are omitted from the drawing for simplicity, as they are not components unique to Frame Relay.

Frame Relay Switch – Any frame relay network is a mesh of switches. Some of them are edge switches, optimized to present standardized interfaces (ports) to the customer, while others are core switches, designed primarily to perform fast switching between trunk lines. The protocols switches run amongst themselves are typically proprietary; therefore a provider would typically standardize on one brand of Frame Relay switches throughout its network.

Trunk Lines – High-capacity circuits connecting the switches in frame relay network. Those lines statistically multiplex all customers' traffic and are therefore shared by all customers of this provider. Design, reliability, and capacity planning of the trunk lines are the provider's responsibility. It must be understood

however that the trunk lines are with finite capacity and are shared among customers, so if a trunk line is congested due to traffic from one customer, other customers' traffic passing through this trunk *will be affected*.

Frame Relay Port – An interface of the frame relay switch that connects to the customer's equipment via the local access circuit. The speed of the frame relay port determines the speed at which the frame relay network will accept traffic.

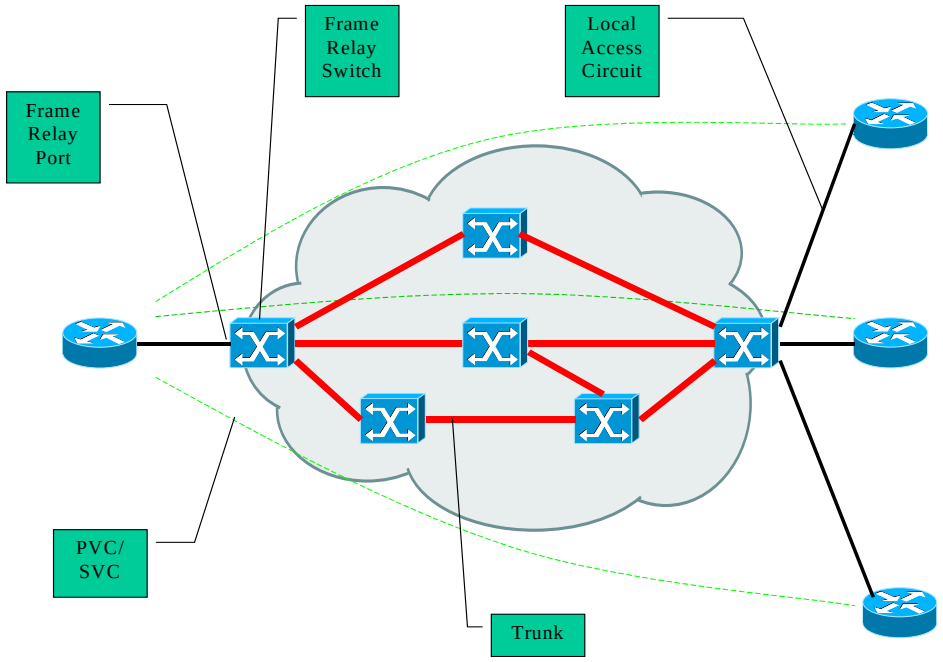


Figure 1: Basic terms of frame relay

Local Access Circuit – A leased line (typically from a ILEC/CLEC), which connects the customer premise to the closest frame relay port in the provider's network. The bandwidth of this circuit is normally equal to that of the Frame Relay port it connects to.

PVC (Permanent Virtual Circuit) – A virtual path through the frame relay network, which is provisioned statically in the frame relay switches. It connects customer premise "A" with customer premise "B" and in that (and only that) respect is analogous to a point-to-point circuit. The PVC has defined bandwidth.

In some cases, **asymmetrical PVCs** are provisioned, and those have different bandwidth levels defined in the two directions of traffic flow. Even though it would seem that asymmetrical PVCs would be very attractive for most types of traffic (WEB Browsing, FTP transfers, Video streams, etc.), the pricing most providers offer for those is not enticing enough to justify the added complexity of their maintenance. Therefore symmetric PVCs are predominant in the industry.

SVC (Switched Virtual Circuit) – Same as PVCs, however those are set up and torn down on demand via signaling protocol (Q.933). Using SVCs is more complex than using PVCs, there are issues with scalability, billing for usage, and general complexity. They do have an advantage over PVCs though, in that they provide instant, transparent any-to-any connectivity. Carriers have only lately begun offering SVC services and they are not widely available yet. The RS does not support SVCs at this time and their mentioning here is only for completeness.

UNI (User-to-Network Interface) – In the context of frame relay, the interface specification between a frame relay switch and a customer device (router).

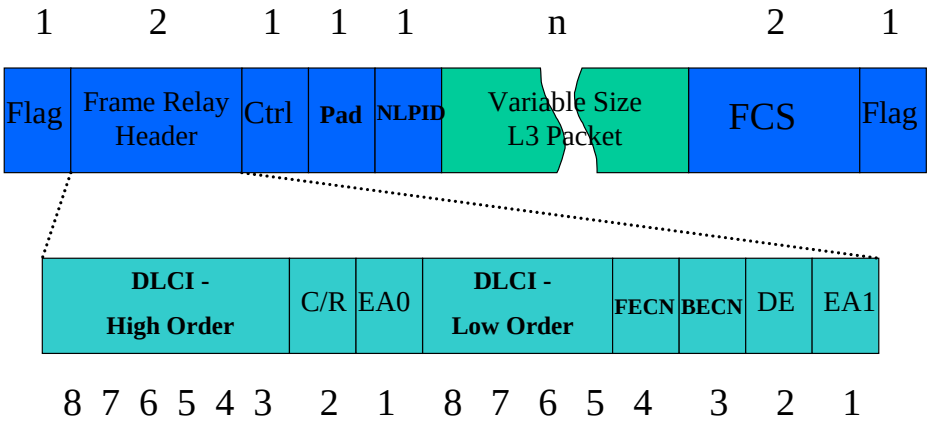
NNI (Network-to-Network Interface) – The specification to connect two provider's frame relay networks (this is a subject outside of the scope of this document).

LMI (Local Management Interface) – Part of UNI – the protocol for signaling between a frame relay switch and a user device (router).

The Framing of Frame Relay

Three standardization bodies deal with frame relay, and here they are along with examples of the types of documents they produce.

ITU - (Q.922, Q.933)
IETF - RFCs
FR Forum - Voluntary Implementation Agreements (FRF x)



Flag - a value of 0x7E which identifies the beginning and end of the frame
Ctrl - the Q.922 Control field (always set at 0x03)
Pad - optional 1-byte padding to align the frame to a 2-byte boundary
NLPD - Network Layer Protocol ID - specifies the L3 protocol that is in the payload (0xCC for IP)

DLCI - Data Link Connection Identifier
C/R – Command/Response bit
FECN – Forward Explicit Congestion Notification
BECN – Backward Explicit Congestion Notification
DE – Discard Eligibility Indicator
EA0 and EA1 – Extension bits – indicate the header is 3 or 4 bytes long

Figure 2: Frame relay frame format

Frame Relay is essentially a Layer 2 technology. As such, it encapsulates L3 PDUs (packets) in a frame and prepends the header shown on Figure 2, defined in ITU Q.922. Since many L3 protocols (besides IP) can be encapsulated in Q.922 framing and sent on the same PVC, a method was needed to multiplex/de-multiplex them. RFC 2427 (Multi-protocol Encapsulation for Frame Relay) specifies this method (it obsoletes RFC 1490). It goes even further and also defines the format for encapsulating Layer 2 frames, thus allowing bridging to occur across frame relay PVCs. There is a disagreement in the Internet community if that is a good or a bad thing however. FRF 3.1 has a similar purpose to RFC 2427, but enhances it to allow SNA encapsulation over frame relay PVCs.

Note on the EA bit: Each header octet has an EA bit, which signifies if this is the last octet of the FR header. If EA is 0, there are more octets in the header, and if EA is 1, this is the last header octet. So in a standard, 2-octet header, EA0=0 and EA1=1. The specification allows for extending the header to 4 octets, which allows for addressing about 8 million DLCIs. Extended headers are uncommon among providers and not supported by Riverstone.

Frame Relay Addressing

DLCI

The **DLCI** uniquely identifies a PVC from the perspective of a router. When the FR header is with the standard size of 2 octets, the DLCI is a 10-bit number, allowing for 1023 addresses. Of those, DLCI 0-15 and 992-1023 are reserved for internal network management, multicast, and other future uses, leaving 976 useable DLCI numbers (16-991). The end user (the router) is responsible for building the frame relay frame, placing the correct destination DLCI in the header. Then it hands the frame to the ingress frame relay switch. All switches in the network have pre-built tables, which allow them to relay traffic with a given destination DLCI through the network down to the egress switch.

If there is any kind of problem with the frame (FCS check invalid, DLCI invalid, etc.), at any point in the network, the frame is simply dropped. There are no provisions in frame relay for error recovery and notification. This responsibility is left to the higher-level protocols. This makes frame relay very efficient when run over error-free lines. **However, error recovery at the higher layers is typically very expensive in terms of bandwidth and CPU, and therefore if frame relay is running over unreliable lines, causing lots of retransmissions, it can be a very high-overhead protocol.**

In order to forward traffic, the frame relay network relies solely on the DLCI value contained in the frame. In this respect frame relay switching bears similarity to other label swapping technologies, such as tag switching, ATM, or MPLS.

Local vs. Global Significance of DLCI Numbers

Care must be taken to understand the meaning a particular frame relay provider places into DLCI numbers. Two types of DLCI addressing are known – local and global. Most providers only support local addressing, but others can do both, in which case the user needs to be aware which addressing type is used. The difference is discussed below.

When **local** addressing is used, DLCI numbers have local significance for a particular frame relay port and can be reused on other ports. As shown in Figure 3, router A will use DLCI 100 to reach router B. However router B will also use DLCI 100 to reach router A. There is no conflict because the DLCIs are of local significance only.

With **global** addressing, each endpoint of a network is uniquely identified by a DLCI number, and therefore DLCI numbers cannot be re-used. In Figure 3, the globally addressed router A is known with DLCI 100 by the entire network, router B is 101, and router C is 102. When router A wants to send a frame to router C, it will address it with DLCI 102. The same will be done by router B.

Local addressing is more scaleable but confusing, while global addressing is simpler. When global addressing is used, the DLCI can be thought of as the MAC address of the destination. In line with the overwhelming majority of local addressing out there, we will assume local addressing throughout this document.

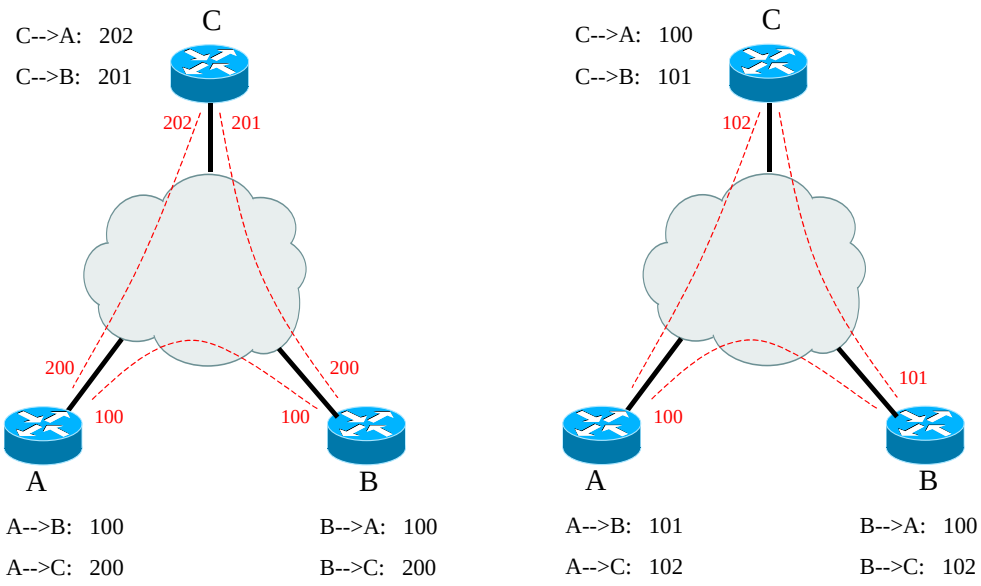


Figure 3: Difference between local and global DLCI addressing

PVC (Permanent Virtual Circuit)

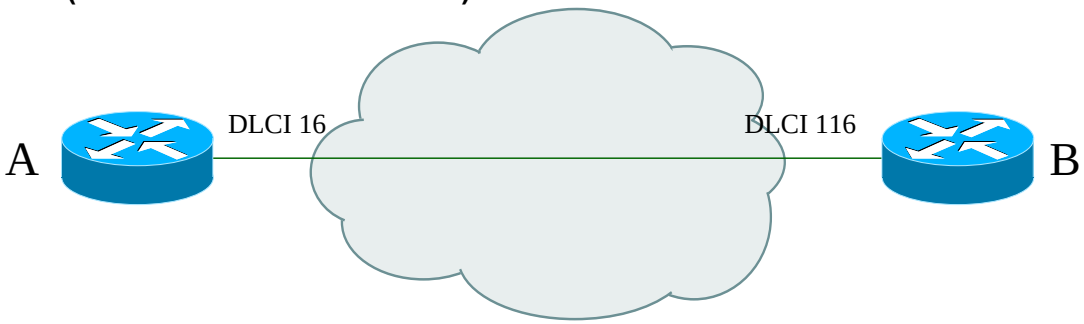


Figure 4: DLCIs on both ends of a PVC

A PVC is a virtual path through the frame relay network, with both of its endpoints associated with a DLCI number. A customer frame relay device (router) only has visibility to the **destination, or remote, DLCI** of the PVC. A router is not aware of the DLCI on its (local) end of the PVC. This rule may be confusing, especially when combined with the typical notation used in drawings. We try to clarify it with Figure 4.

Router A's local DLCI is 116, and router B's local DLCI is 16. However for this PVC the frame relay switch presents DLCI 16 to router A, and DLCI 116 to router B, implying that if router A is to send a frame destined for DLCI 16, it will end up at router B.

Several parameters are defined along with the PVC configuration, which are essential to understanding how traffic management works in a frame relay network, as well as how it can be affected. With the availability of traffic shaping for routers, understanding these parameters became important.

CIR, T_c, B_c, and B_e

The **Committed Burst Rate (B_c)** is the maximum amount of bits the network guarantees to transport during the **bandwidth time interval (T_c).** Values for T_c typically range from 0.5 to 2 seconds. The **Excess Burst Rate (B_e)** is the maximum amount of bits, above and beyond Bc, that users can send during the T_c interval, and the network will attempt to deliver.

CIR (Committed Information Rate) is the average of B_c over time: **$CIR = B_c/T_c$**. Or written otherwise, **$T_c = B_c/CIR$**, which implies that the higher the ratio of B_c to CIR, the longer the switch can accept a sustained burst, the more ingress buffering is needed, and the longer the potential latencies could be through the network. We will discuss this further in the QOS section.

It is important to understand, that data always enters the frame relay network at the speed of ingress frame relay port (data is clocked at the port speed). So even though a CIR can be 32 Kbps, if the port is 56 Kbps, data will be offered to the ingress switch at 56 Kbps. Traffic shaping on the router does not change that; it merely creates a time interval, and ensures that no more than a specified amount of data will be offered during that interval.

T_c, B_c, B_e, and CIR are always configured on the frame relay switch. They can also be optionally configured on the routers, in which case traffic shaping is enabled. Unless specifically noted, we will assume these parameters are NOT configured on the router.

More discussion on this in the QOS section.

Explicit Congestion Notification (FECN/BECN)

Explicit congestion notification is one of the two methods frame relay employs to deal with congestion.

FECN (Forward Explicit Congestion Notification). The FECN bit in the frame relay header is set by the network (frame relay switch) to notify the receiving end-station (router) that the PVC the frame belongs to has experienced congestion. This is solely a notification function. The receiving router may or may not take

action, in most cases it does not, since the cause of the congestion is most likely the sending router. It is therefore more useful to notify the sending router of the congestion condition, so it can slow transmission down until congestion subsides. This mechanism is called **BECN (Backward Explicit Congestion Notification).**

Figure 5 illustrates how FECN and BECN are sent. The congestion occurs at switch B. It may involve single or multiple PVCs through that switch (only one is shown). All PVCs affected are notified through FECNs and BECNs. To send FECNs, switch B simply sets the FECN bit on the frames from PVC 100/101. To send BECNs, switch B waits for data frames that belong to the same PVC to arrive from switch C, destined for switch A. It then sets the BECN bit on those frames.

Handling congestion on a frame relay network is a cooperative effort between the frame relay network provider and the customer equipment. The rules are simple: The provider notifies the customer of the congestion. The customer either reacts and reduces the load offered to the network, or ignores the notification. If the congestion worsens to a condition pre-determined by the provider, the network begins dropping all traffic from the affected PVC, until the throughput improves.

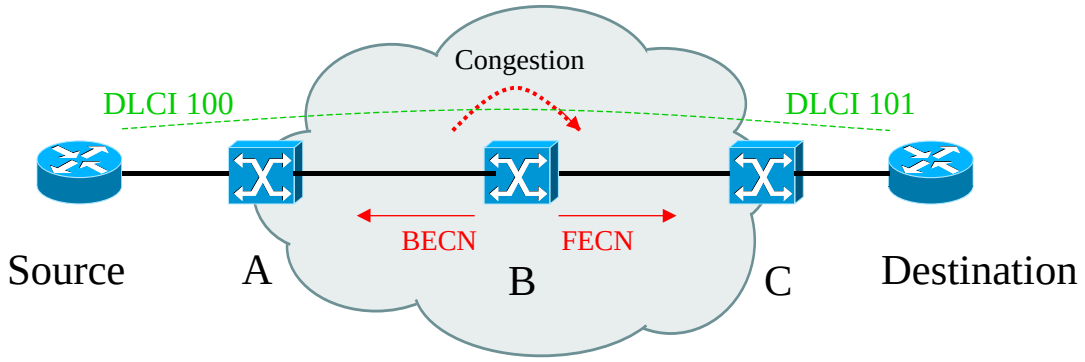


Figure 5: FECN and BECN

The RS ignores FECNs and can respond to BECNs with **BECN adaptive shaping**. More on this in the QoS section.

Labeling Traffic as Discard Eligible (DE)

The DE bit provides a mechanism to inform the network which frames to drop **first** if congestion occurs. It can be set either by the router, or by the ingress frame relay switch.

The customer equipment (router) would set the DE bit based on some pre-configured criteria. The ingress switch may also be configured to set the DE bit for all frames above the B_c, although most frame relay providers do not do it.

If congestion occurs downstream in the network, and traffic needs to be dropped, the switch dropping it will select from its buffer the frames with the DE bit set first. Of course, if this is not sufficient to relieve the congestion, frames with DE bit clear will be dropped next.

The RS can be configured to set the DE bit on all traffic that exceeds B_c. Note that this mechanism does NOT provide for setting the DE bit based on a sophisticated ACL or QoS policy. It merely sets the DE bit on traffic that exceeds certain pre-defined volume.

LMI (Local Management Interface)
LMI is the signaling mechanism between the frame relay switch and the router. Its purpose is to convey configuration information from the switch to the router (such as which PVCs are serviced on this port), and to allow for status information to be exchanged between the two. LMI passes administrative frames between the router and the switch on a designated DLCI, typically 0 or 1023. Currently there are three versions of the LMI specification, which are contrasted in the following table. Of course, the router and the switch must match the LMI specification they use.

Colloquial Designation	Riverstone	Cisco	Nortel Networks	Specification	DLCI used for LMI
LMI, Vendor Forum, Gang of Four	rev1	cisco*	Rev 1 LMI	Frame Relay Forum Implementation Agreement FRF.1, superseded by FRF 1.1	1023
Annex D	ansi617d-1994*	ansi	ANSI T1617.D	ANSI T1.617	0
Annex A	q933a	Q933a	CCITT Annex A	ITU Q.933 (referenced in FRF.1.1)	0

Table 1: The three LMI specifications
*Default setting when LMI is enabled

It is often confusing whether one refers to the generic term LMI, encompassing all three specifications, or to the first specification, called also LMI, or otherwise known as “Gang of Four”. (Named after the four vendors that originated it – Stratacom, DEC, Nortel, and Cisco). LMI and Annex D, the two most commonly implemented LMI specifications, share a set of timers and counters that are used to infer the status of the interface between the switch and the router. Mismatch of those timers between the router and the frame relay switch is also frequently a source of problems. Following is a short description of how the timers work, along with their default values.

- The router is the master in the LMI communication and it polls the switch every 10 seconds (T391 Link Integrity Timer).

- On every 6th poll, the router expects to receive a full status message (N391 Full Status Poll) This is the message which carries the status of all PVCs on the link. It may take up to 60 seconds to communicate a change of a PVC state from the switch to the router.
- If the switch has not received a poll for 30 seconds after the last poll, it logs an error (T392 Polling Verification Timer)
- If three out of four events are errored, then the port is declared failed (Three is the Errored events threshold - N392 and four is the Monitored errored events count - N393)

A typical indication that the timers are mismatched on the switch and on the router is that PVCs are going up and down periodically as a group. Those timers are configurable on the RS and can be adjusted to match the switch settings. Notice the difference in the terminology typically used by frame relay switch providers (above) and the one used by the RS (below).

rs-config# frame-relay set lmi ?	
type	- Set the LMI type
state	- Enable/disable the sending/receiving of LMIs. If LMIs are enabled, the operational state of any VCs is determined by the network equipment. If LMIs are disabled the operation state of any VCs is always up (unless the port is disabled). (Default: Disable).
polling-interval	- T391. Number of seconds between successive status enquiry messages. Valid range: 5-30.
full-enquiry-interval	- N391. Number of status enquiry intervals that pass before issuance of a full status enquiry message. Valid range: 1-255.
error-threshold	- N392. Max number of unanswered status enquiries before declaring an interface down. Valid range: 1-10.
monitored-events	- N393. Number of status polling intervals over which the error threshold is counted. Valid range: 1-10.
ports	- Name of the ports

Finally, LMI is optional. A router can be configured to NOT use any LMI (that setting must be matched by the frame relay switch also). In this case, PVCs would have to be configured statically, and no status information for them will be

available. So even if a PVC goes down, the router will continue to send traffic to it, and will let the upper-layer protocols time-out.

Summary of Frame Relay Operation

To put the various terms in perspective, here is a summary: Data is transmitted over the local access circuit into the frame relay switch. This happens at link access rate. The switch buffers the frame in the ingress buffer and counts the incoming bits on a per-PVC basis, for recurring intervals of T_c. Any bits exceeding the B_c amount are counted as B_e bits, and the switch may set the DE bit for the frames that contain them. These frames are transported if there is no congestion on the network. If the ingress buffer is full or the customer offers more than B_c+B_e bits during any interval T_c, the switch discards new incoming frames until the causing condition clears.

As the ingress buffer is drained, room is freed for more B_c or B_e data. This mechanism is sometimes referred to as “Dual Leaky Bucket” and is well suited for traffic with short bursts.

Frame Relay vs. Private Lines

It is important to keep in mind the following fundamental differences between private lines and frame relay connections. In frame relay

- **Latency is variable.** It depends on the depth and utilization of the ingress buffer, as well as the congestion on the provider’s network, as bandwidth on the trunk links is shared. Private lines tend to exhibit constant latency, as there is no buffering and bandwidth is dedicated.
- **Transmission rates on both ends of a PVC are typically mismatched**. It is not at all uncommon to have a PVC terminating on one end via a T3 local access circuit, and on the other end via a 56 Kbps local access circuit. Traffic is transmitted at the rate of the local circuit. This means that the T3 end can easily overwhelm the other end of the PVC with traffic. This is unlike private lines where there is no mismatch on both ends of the line.

Design Guidelines

Frame Relay Subscription Issues

The most common topology used in frame relay today is the **hub-and-spoke**, shown on Figure 6.

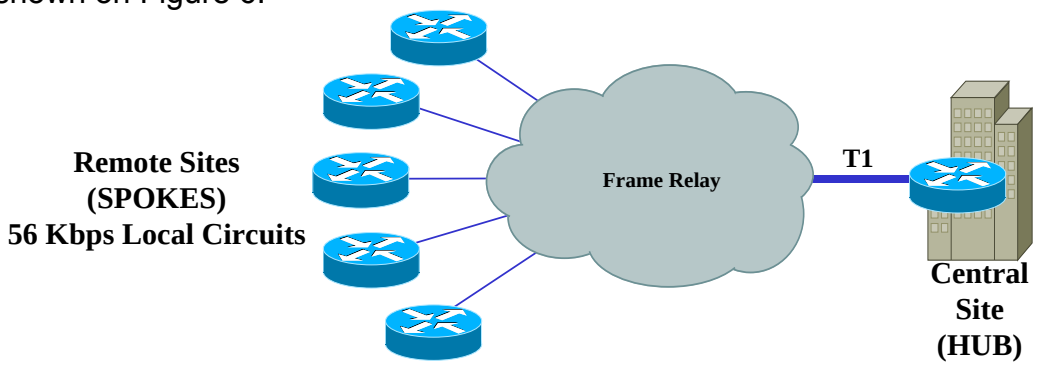


Figure 6: Hub-and-spoke

This topology fits well host-based or central server applications, or a service provider aggregating multiple customers. Oversubscription in this environment is common, especially at the hub site. It is therefore extremely important to know both the customer traffic and the behavior of the routing protocols.

When analyzing subscription rates, one must consider the bursting capabilities of the network. Since bursting in the worst case is limited by the speed of the access circuit, it is the speed of the access circuits that is important, not so much the CIR. Common access speeds at the hub site are T1 (E1), and increasingly T3 (E3). At the spokes, the typical access speed is 56 (64) Kbps or a fractional T1 (E1). The bigger the mismatch between the hub and spokes access speed, the more spokes can be connected to a given hub, the higher the subscription rate, the more pronounced the subscription issues. Two main aspects of subscription need to be considered:

Analyzing the Hub Ability to Handle Inbound Traffic

- **Subscription to the sum of the spoke access speeds.** This is the most conservative subscription formula. If the hub site is connected via a T1 circuit, then it can be subscribed with up to 24 56 Kbps spoke sites. Even if all of them burst to 100% at the same time, the central site should be able to handle the inbound traffic. This level of subscription is safe when the traffic pattern is not well known and excessive latencies cannot be tolerated. Typically however it is considered wasteful and most network managers subscribe to a higher level.
- **Subscription to the sum of the CIRs.** With the above scenario – a T1 hub and 56 Kbps spokes, if the CIRs of the spokes were 16 Kbps, one can concentrate 98 spokes into a hub. To deploy this more aggressive

subscription level, one needs to have very good knowledge of the traffic patterns. Specifically, it must be certain that all PVCs will never burst simultaneously, otherwise they will overwhelm the egress of the frame relay network at the hub site. As illustrated on Figure 7, if all 98 PVCs burst simultaneously for the duration of their T_c interval (let's assume T_c is 1 second), the bandwidth demand on the hub site will be 5.488 Mbps – almost 4 times the available bandwidth. Note that such burst is perfectly legal, and as long as there is no congestion in the frame relay cloud, it will all be delivered to the egress switch. As illustrated on Figure 7, for such a burst to be accepted, the egress switch will have to have 686 KB buffering capacity, which is an order of magnitude larger than what most frame relay switches have. In fact, most frame relay switches do not buffer on egress at all! Either way, such bursting will result in large amounts of traffic being dropped. Allowing such oversubscription rate without knowing the traffic patterns well is clearly a bad idea. It is also easy to see how subscribing the central site to twice the number of PVCs (192) with half the CIR each (8 Kbps) is an even worse idea.

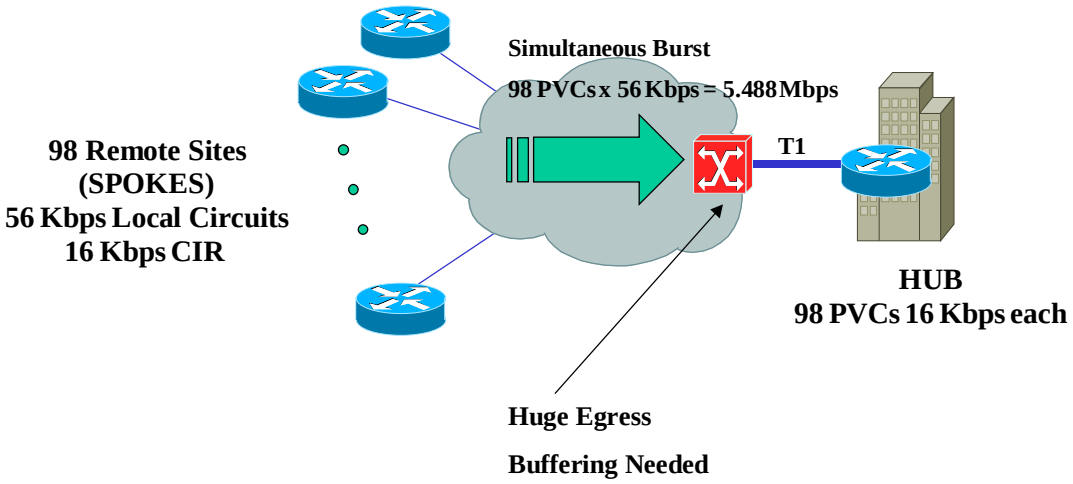


Figure 7: Bursting against the hub site (no traffic shaping)

Analyzing the Hub Ability to Handle Outbound Traffic

Many problems in frame relay are associated with its non-broadcast (and non-multicast) nature. Even though the Frame Relay Forum has developed specifications for multicasting over frame relay, most carriers do not implement those. This leaves the routers with the burden of handling the broadcasts and multicasts over non-broadcast media. They do that by replicating the broadcast on every PVC, as if it were a unicast. This places significant buffering burden on the hub router. The main sources of broadcast are considered below.

Routing Protocols

Routing protocols are responsible for the lion share of the broadcast overhead associated with frame relay connections. A commonly accepted criteria, which can be used as a starting point of an analysis, is that routing overhead should not exceed 15% of the link's bandwidth. With that in mind, we will analyze the overhead associated with different routing protocols.

RIP – version 1 or 2

By default, RIP sends its updates every 30 seconds. A RIP packet can contain up to 25 route entries, 20 bytes each, plus 32 bytes header, for a total packet length of 532 bytes. A router must send its entire routing table (subject to limitations from routing policies) on all of its RIP interfaces every 30 seconds. To use the example on Figure 7, with 98 PVCs defined at the hub, a RIP update for 500 routes on that port will take :

500/25 = 20 full RIP packets per PVC
20*532 = 10,640 bytes = 85,120 bits per PVC
85,120*98 = 8,341,760 bits for all PVCs (every 30 seconds)
8,341,760/30 = **278 Kbps bandwidth needed for RIP only**

It is evident that even with modest number of routes, such subscription will yield significant bandwidth consumption due to traffic replication. The example above is borderline acceptable, and further analysis must be done on the impact of actual user traffic.

With RIP, the problem is typically handled by configuring route filters, which cause only a few necessary routes to be announced over the frame relay interface or, if possible, only a default route to be announced.

IPX RIP and SAP

The RIP version employed in the IPX protocol suite is a bit more efficient than RIP for IP, in a sense that it advertises every 60 seconds, instead of 30 seconds, and contains up to 50 routes per packet, rather than 25. This efficiency is fully negated by the existence of SAP (Service Advertisement Protocol), which broadcasts all services on the network every 60 seconds.

Again, the way to handle these broadcasts on subscribed frame relay interfaces is to put filters in place and advertise only the routes and services that are needed across the frame relay connection.

OSPF

This section assumes good level of understanding of OSPF. For a good reference on OSPF, see [OSPF1].

OSPF is a link state routing protocol. An OSPF interface of a router is always in one and only one OSPF area. When the state of an interface of an OSPF router

changes (goes up or down), the router sends a Link State Update (LSA) out all of its interfaces that belong to the same area. This is called “flooding” of the LSA. All routers receiving the LSA must in turn flood it out their interfaces participating in that area). If there is no interface change for 30 minutes with regard to a particular LSA, the originating router will flood it anyway through the area, for the purposes of database synchronization. Every router maintains a database of all LSAs it receives or originates. It is called Link State Database (LSDB) and its content must be the same throughout the OSPF area.

Every OSPF interface that participates in an area is subject to the above behavior. If that interface happened to be frame relay, the LSA flooding process must is replicated on each PVC.

- For a stable OSPF area (without many interface changes), the overhead of OSPF can be evaluated quite predictably, based on the size of the LSDB in an area. When there is flapping in the area, there is a mechanism in OSPF that limits the impact of the flapping on the bandwidth consumption – it is a rule that no LSA will be flooded more frequently than 5 seconds. This rule can limit the impact of a worst-case scenario. In general, the overhead of OSPF can be placed between best-case and worst-case boundaries based on the following set of rules:
- The size of an LSA is variable, depending on type, but is typically 36 bytes.
- An LSA cannot be flooded more often than 5 seconds.
- An LSA is guaranteed to be flooded at least every 30 minutes. **NOTE:** An LSA is flooded every 30 minutes by its originating router. The originating router keeps separate timers for each of its LSAs. Normally this means that the 30-minute LSA flooding is staggered over time as different LSAs age out at a different time (see [1] page 160). **However , if the frame relay hub router happens to be the originating router for a large number of LSAs, then all of those will probably age out at the same time, resulting in a large, replicated blast over the frame relay port**. One must always be mindful of the topological placement of the hub router and the number of LSAs it might generate at the same time. As a rule, the hub routers should not have many interfaces in the same OSPF area as a heavily subscribed OSPF interface. If possible, those should be placed in different areas, thus minimizing he effect of the 30-minute flooding.
- All routers in an area must have the same image of the LSDB
- Within a single area, OSPF packets travel only a single hop; If router A connects via point-to-point link to router B which in turn connects via point-to-point link to router C, OSPF packets go from A to B, and from B to C, but not from A to C directly.
- There is no easy way to correlate the size of the network (number of routes) with the size of the LSDB, since different network types produce significantly different LSDB sizes. However it is the LSDB size that matters when evaluating the bandwidth demands of OSPF, so the easiest

way to evaluate it is to count its entries. Issue **ospf monitor lsdb** on any router in the area, and count the entries.

BGP4

BGP4 is very efficient routing protocol in a sense that it does not have the regular routing updates like RIP and (to an extent) OSPF does. It only transmits changes as they occur, and groups prefixes with like attributes in one efficient update. It has added efficiency in the fact that BGP communications are always point-to-point, between a pair of BGP speakers. So in certain situations, we would not have the PVC replication issue we would with RIP or OSPF. BGP4 can create a large burst of traffic in the beginning of a BGP session with a peer, and this should be kept in mind especially over slow PVCs (<32 Kbps).

Inverse ARP at Startup

Inverse ARP is enabled by default and is used to associate MAC addresses (DLCIs) with an interface's IP address. For each IP interface that comes up, only three small packets are exchanged between the peers. Normally, InARP traffic will create a short-term bust when a frame relay port is brought up, which may result in a minor congestion if many IP interfaces are defined on this port. This traffic may become an issue only if there are frequent changes of the frame relay port state (up and down) causing multiple frequent executions of InARP.

InARP can be disabled (or the peer on the other end may not support it). In this case frame relay peers can be manually configured, in which case no InARP traffic will flow.

"frame-relay set peer-addr ip-addr 10.10.1.1/30 ports se.4.1.16"

User Traffic

User traffic falls into three categories: Unicast, Multicast, and Broadcast. As mentioned before, most frame relay networks do not have multicast or broadcast capabilities, therefore multicast and broadcast is handled as **replicated unicast** on frame relay interfaces.

The higher the subscription rate of an interface is, the more important it is to know the percentage of multicasts and broadcasts to be transmitted over a frame relay interface. A circuit which is sized correctly for the volume of traffic in general, may be easily congested if large portion of its traffic is multicast/broadcast, because of the replication.

Analyzing Router Scalability

LMI Constraints

The LMI protocols require that all PVC status reports must fit in a single frame. The current hardware on the RS limits the MTU size to 1504 bytes, which allows status information for about 300 PVCs to be placed in a single LMI packet.

If LMI is NOT enabled on a port, 1024 PVCs can be defined. In reality, frame relay providers reserve DLCI 0-15 and DLCI 992-1023 for administrative, multicast, and broadcast, leaves 992 usable DLCI assignments.

Memory Considerations

All data structures for supporting frame relay, such as queues, buffers, VC definitions, etc., are maintained on the WAN card of the RS. The memory on the WAN cards is partitioned between code, heap and packet buffers. The heap is big enough to accommodate 992 PVCs per port, and to set up all the output queues associated with them (shaped queues per port and priority queues per PVC).

CPU Constraints

One significant way in which the WAN cards differ from the rest of the RS architecture, is that they processes all packets in software. Therefore traffic through the WAN card is a lot more CPU-bound than the rest of the traffic, which is generally flow-switched in hardware.

The WAN CPU (WCPU) is designed to process up to 100K packets/second. In a worst-case scenario (minimum packet size of 64 bytes), a WAN card should be able to process about 51 Mbps of traffic (a half-duplex T3).

More Terms Related To Frame Relay

Ports, Interfaces, Subinterfaces, VLANS, and PVCs

This section intends to define the relationship among the above terms in the context of the RS, and to establish some common terminology.

A WAN physical port defined for frame relay operation, is a **frame relay port.**

A frame relay port can have one or more **PVCs.**

PVCs in a port may be grouped in one or more **VLANs.**

IP or IPX addresses apply to VLANs, to create IP/IPX **interfaces**. There can be one or more interfaces assigned to a VLAN.

The above 4 terms are illustrated on Figure 8.

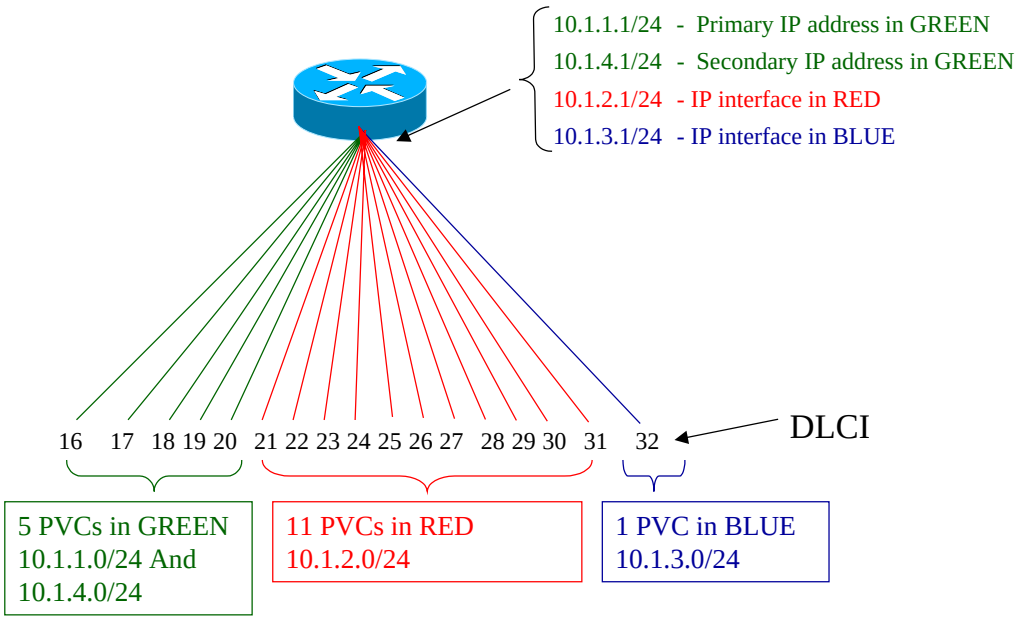


Figure 8: Placement of PVCs in VLANs

Note that VLAN GREEN has a secondary IP interface. Also observe that IP/IPX addressing for VLANs with multiple PVCs will be assigned is if it was a LAN, in this case with a /24 mask. Following is the configuration of the above setup.

```
; The "hub" port is se.4.1 and the circuit speed is 1,536,000 bps
PORT SET SE.4.1 WAN-ENCAPSULATION FRAME-RELAY SPEED 1536000

; We enable LMI on the circuit.  The RS defaults to Annex D LMI, we do
not want to override
; this
FRAME-RELAY SET LMI STATE ENABLE PORTS SE.4.1

; PVCs are defined
frame-relay create vc port se.4.1.(16-32)

; Create VLANs so PVCs can be grouped
vlan create GREEN port-based id 10
vlan create RED port-based id 11
vlan create BLUE port-based id 12

; Assign PVCs to VLANs
vlan add ports se.4.1.(16-20) to GREEN
vlan add ports se.4.1.(21-31) to RED
vlan add ports se.4.1.32 to BLUE

; Assign primary IP Interfaces to VLANs
interface create ip GREEN_INT address-netmask 10.1.1.1/24 vlan GREEN up
```

```
interface create ip RED_INT address-netmask 10.1.2.1/24 vlan RED up
interface create ip BLUE_INT address-netmask 10.1.3.1/24 vlan BLUE up

; Add a secondary IP address to VLAN GREEN
INTERFACE ADD IP GREEN_INT ADDRESS-NETMASK 10.1.4.1/24
```

A frame relay **subinterface** is created when a PVC is defined as a point-to-point interface. This is a new feature introduced in firmware release 3.1. A subinterface can only contain one PVC, and is associated with a separate dedicated VLAN. There is a subtle distinction between subinterfaces, on one hand, and single PVCs placed in separate VLANs, on another. This distinction is of special importance for OSPF, so we will discuss it in more details in the OSPF section. Subinterfaces are illustrated on Figure 9.

Please note that the above definition of subinterface differs from that adopted by Cisco Systems. Cisco's definition of subinterfaces is given in http://www.selsius.com/warp/public/779/smbiz/service/knowledge/wan/subifs.htm. Since a Cisco router lacks the concept of a VLAN, it defines point-to-point and point-to-multipoint subinterfaces as a way to express VLANs with a single or multiple PVCs.

The RS handles VLANs natively and makes no fundamental distinction between VLANs with a single PVC and VLANs with multiple PVCs. Here, the notion of a subinterface is reserved for the special case of defining a FR PVC as a point-to-point interface (during interface definition), WITHOUT the possibility of later adding another PVC to the same VLAN. Subinterface here is strictly an emulation of a point-to-point circuit.

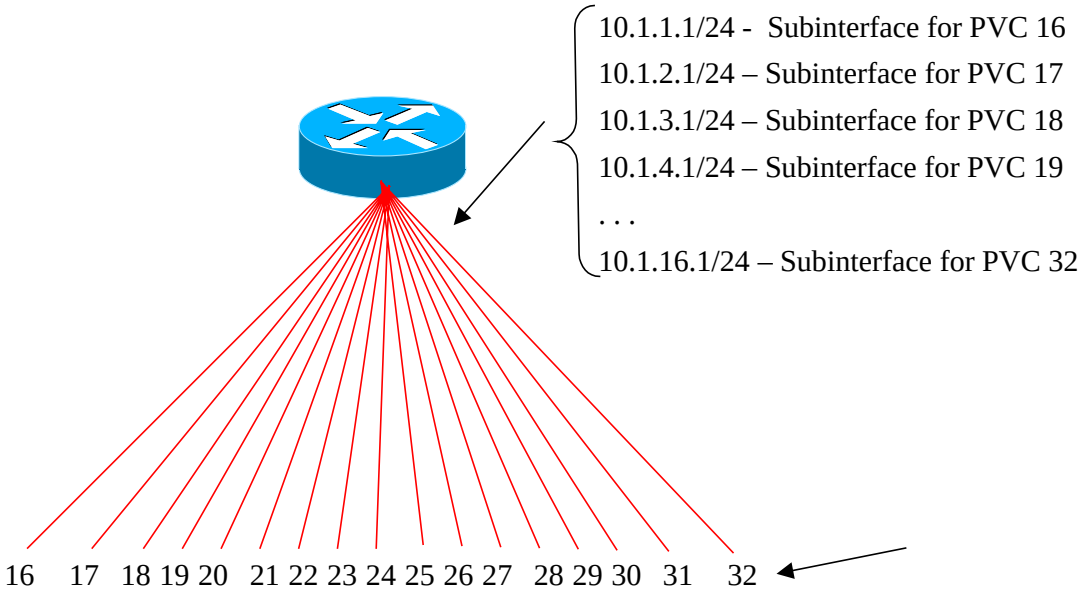


Figure 9: Subinterfaces (version 3.1 and above)

Here is the configuration that complements Figure 9: The difference from the previous configuration is in the VLAN definition and assignment (as commented below).

```
port set hs.4.1 wan-encapsulation frame-relay speed 51840000
frame-relay set lmi state enable ports hs.4.1
frame-relay create vc port hs.4.1.(16-32)

; VLANs will NOT be explicitly created and PVCs will NOT be assigned to VLANs, as in the
previous configuration.  Instead, IP interfaces are configured in Point-to-Point mode,
which creates their VLANs implicitly.

interface create ip SUB_1 address-netmask 10.1.1.1/24 port hs.4.1.16 type point-to-point
interface create ip SUB_2 address-netmask 10.1.2.1/24 port hs.4.1.17 type point-to-point
interface create ip SUB_3 address-netmask 10.1.3.1/24 port hs.4.1.18 type point-to-point
interface create ip SUB_4 address-netmask 10.1.4.1/24 port hs.4.1.19 type point-to-point
interface create ip SUB_5 address-netmask 10.1.5.1/24 port hs.4.1.20 type point-to-point
interface create ip SUB_6 address-netmask 10.1.6.1/24 port hs.4.1.21 type point-to-point
interface create ip SUB_7 address-netmask 10.1.7.1/24 port hs.4.1.22 type point-to-point
interface create ip SUB_8 address-netmask 10.1.8.1/24 port hs.4.1.23 type point-to-point
interface create ip SUB_9 address-netmask 10.1.9.1/24 port hs.4.1.24 type point-to-point
interface create ip SUB_10 address-netmask 10.1.10.1/24 port hs.4.1.25 type point-to-
point
interface create ip SUB_11 address-netmask 10.1.11.1/24 port hs.4.1.26 type point-to-
point
interface create ip SUB_12 address-netmask 10.1.12.1/24 port hs.4.1.27 type point-to-
point
interface create ip SUB_13 address-netmask 10.1.13.1/24 port hs.4.1.28 type point-to-
point
interface create ip SUB_14 address-netmask 10.1.14.1/24 port hs.4.1.29 type point-to-
point
interface create ip SUB_15 address-netmask 10.1.15.1/24 port hs.4.1.30 type point-to-
point
interface create ip SUB_16 address-netmask 10.1.16.1/24 port hs.4.1.31 type point-to-
point
interface create ip SUB_17 address-netmask 10.1.17.1/24 port hs.4.1.32 type point-to-
point
```

Network Type: Point-to-Point, Broadcast, NBMA, Point-to-Multipoint

The above terms are typically used in two contexts – (1) General Networking Topology, and (2) OSPF Interface Modeling. We would like to distinguish those and to avoid the common misunderstanding that accompanies usage of these terms. When it is not specifically stated, this document defaults to the first, colloquial usage of the terms.

Context #1 – General Networking Topology

The terms above are colloquially used in general discussions of network topology to mean the following:

- **Point-to-Point Topology** – a network segment which has two devices attached to it. A serial link running PPP is the typical example, but a frame relay interface with a single PVC can also be considered point-to-point in this general context. So can the connection between two routers, connected over an Ethernet link, addressed with a /30 subnet mask.
- **Broadcast Topology** – a network segment that has broadcast capabilities – most LAN media falls into this category.

- **NBMA Topology (Non-Broadcast Multi-Access)** – a network segment built with Layer 2 technology which supports multiple device attachments (more than 2), but does not support broadcasting (multicasting) capabilities. Examples are X.25, Frame Relay, SMDS, ATM.
- **Point-to-Multipoint Topology** – a special case of NBMA, where the Layer 2 connectivity is such that it allows for communication between a central site and a number of remote sites, but not between the remote sites themselves, otherwise referred to as "Hub-and-Spoke".

Context # 2 – OSPF Network Modeling

The same four terms are also used to describe an OSPF Interface configuration parameter for an interface. In this context, the terms have somewhat different, and much stricter meaning, which is defined below. The terms are defined as used by the RS, and in general, by GATED. The statements below are RS-Specific, and may not be correct or complete for other vendor's equipment.

- **OSPF Point-to-Point Network Type** – Either a PPP interface, or (in release 3.1 and later) a frame relay subinterface.
- **OSPF Broadcast Network Type** – an OSPF network type, which describes broadcast LAN segments – Ethernet, Token Ring, FDDI. When OSPF is configured on these types of interfaces, broadcast is the default network type. OSPF broadcast network type assumes transient connectivity – if router A can talk to router B and router B can talk to router C, then router A can talk to router C. In addition, OSPF uses Layer 3 multicasting for communicating with other OSPF routers on networks of this type. Layer 3 multicast is translated to Layer 2 multicast for media that supports it (such as Ethernet) and into Layer 2 broadcast (for media that is multicast – incapable, such as Token Ring). Please note that **OSPF Broadcast Network Type CAN be configured on interfaces that are not broadcast capable, such as Frame Relay. This however is strongly discouraged, as it relays incorrect information to OSPF and will lead to malfunction.**
- **OSPF NBMA Network Type** – OSPF models NBMA networks in exactly the same way as Broadcast networks, but there are differences in establishing adjacencies. An interface configured as NBMA will not multicast OSPF updates as it knows the media is multicast- and broadcast- incapable. Instead, the interface will unicast the OSPF update to all of its neighbors out that interface. Therefore, the neighbors must be configured with the **"ospf add nbma-neighbor"** command. Like "broadcast", this network type can be configured on any non- Point-to-Point interface, although it is primarily meant for Layer 2 technologies that rely on virtual circuits, such as X.25, Frame relay, and ATM. Please observe that the transient connectivity assumption, described in "OSPF Broadcast Network Type", remains in this network type as well. **This**

means that there must be full mesh connectivity between all devices participating in an NBMA network. Partial mesh, such as frame relay "Hub-and-Spoke" is NOT a good candidate for NBMA network type. Below we will illustrate a possible deviation from this rule, when hub-and-spoke network is built with RSs only. It requires careful configuration, and should be avoided if the alternatives (point-to-point subinterfaces, or point-to-multipoint network types) can be implemented.

- **OSPF Point-to-Multipoint Network Type** – Applicable to any interface where NBMA is applicable. Created to relax the requirement for a full mesh connectivity NBMA imposes. This is the most appropriate OSPF network type for frame relay interfaces. In contrast to the colloquial usage, point-to-multipoint OSPF network type can be applied to frame relay hub-and-spoke interfaces, frame relay partial (or full) meshes, as well as single frame relay PVC links between two routers.

Routing Protocols Specifics with Frame Relay

This section does not aim to be a comprehensive guide to the routing protocols discussed. Its only goal is to point the specifics in the behavior of routing protocols when run through frame relay networks.

These specifics must be well understood, as most Interior Gateway Protocols (IGPs) were originally designed to run over broadcast media with transitive connectivity. Such is not the case in frame relay (as well as most other NBMA technologies). Each of the routing protocols has a specific behavior in an NBMA network, and a set of tools to tune that behavior.

RIP (ver. 1 and 2) Specifics

Routing updates for RIP were originally designed for Local Area Networks. There was an underlying assumption that each port of a router corresponds to exactly one IP interface. Frame relay allows the following variations, which are of importance to RIP:

- A frame relay port with a single PVC defined – as far as RIP is concerned, this configuration will behave as a point-to-point network; no special consideration is needed.
- A frame relay port with several VLANs, each containing a single PVC. This is considered by RIP as an equivalent to several point-to-point interfaces. Again, no special considerations are needed.
- A frame relay port with several subinterfaces. Again, the same point-to-point treatment.
- A frame relay port with a VLAN which contains more than one PVC (hub-and-spoke). This is the situation which needs more attention, as it is modeled by RIP as a LAN, but in reality it is an NBMA.

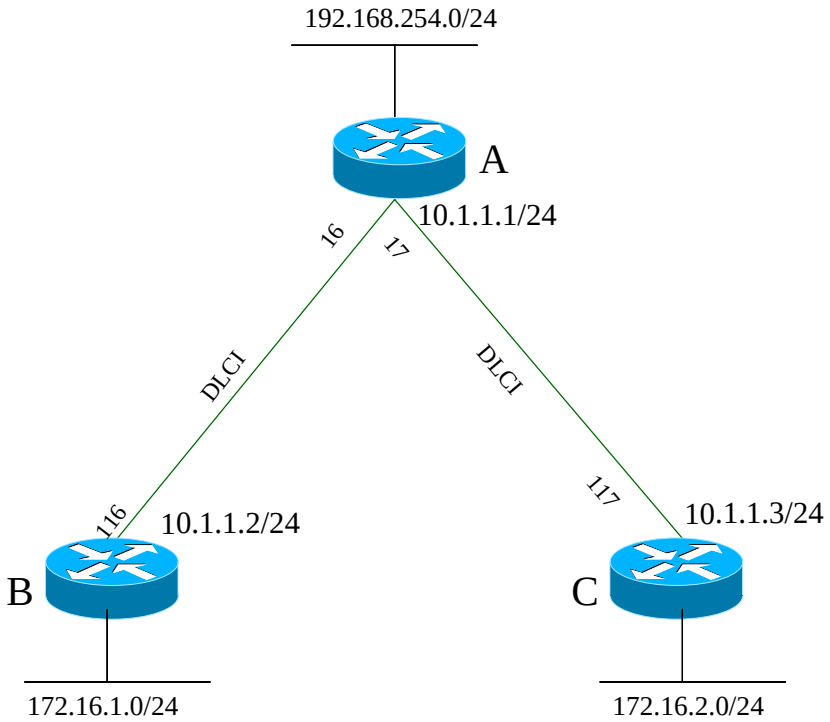


Figure 10: RIP, OSPF, and Broadcast Issues

Consider Figure 10 for a discussion of the problems with RIP in a hub-and-spoke environment. There are two PVCs from router A to routers B and C respectively. Both PVCs are configured in the same VLAN in router A, therefore the IP addressing is assigned as if the frame relay network was a LAN. All interfaces on all routers run RIP. There is no difference between the behavior of RIPv1 and RIPv2 in a frame relay environment. RIPv1, however, has specific rules, which govern when and how specific subnets are announced, and when they are aggregated. These rules are outside of this paper's scope, but can be found in [RIP1]. **For simplicity, throughout this section we will assume RIPv2 is configured.**

RIP with Split Horizon

A well-known mechanism for preventing routing loops that RIP employs is split horizon. It dictates that a router will not announce a route via an interface this route was learnt from. Split Horizon is effective in preventing looping conditions when connectivity to the advertised route is lost. However, in partially meshed frame relay environments, it breaks certain communications.

In this simplest hub-and-spoke configuration, shown on Figure 10, Router A will learn route 172.16.1.0/24 from Router B and route 172.16.2.0/24 from router C. It will install them in its routing table and connectivity from Router A's LAN to either of those subnets will be in place. However, due to the Split Horizon rule, it

will not advertise the route 172.16.1.0/24 to router C and 172.16.2.0/24 to router B, as both of those routes came from interface 10.1.1.1. There is no direct PVC between routers B and C, so there is no way for the spoke routers to learn about each other's LAN.

As a result, even though there is a physical path between spoke sites (B could reach C via A) due to Split Horizon, there will be connectivity between the Hub site and each of the Spoke sites, but not between the two Spoke sites.

RIP with Poisoned Reverse

This is a modification of the standard Split Horizon, where instead of simply not advertising routes out certain interface, routes are advertised with a metric of 16. The result is the same at the receiving router – it will consider the route unreachable. In our example, Router A will advertise both routes, 172.16.1.0/24 and 172.16.2.0/24 out its frame relay interface with a metric of 16. Routers B and C will receive this advertisement but will not consider it, as it represents an unreachable network.

Even though poisoned reverse provides some advantages over standard split horizon, it has the disadvantage of increasing routing traffic by explicitly announcing unreachable networks. It is also equally inappropriate for frame relay hub-and-spoke.

RIP with Actual Metric Announcement

Split Horizon and Split Horizon with Poisoned Reverse are router-wide settings in the global RIP setup of the RS. They apply to all RIP interfaces of the router. The RS provides means to override those global settings on a per-interface basis, using the command:

"rip set interface <NAME> xmt-actual"

This command will override the router-wide setting for split horizon/poison reverse for that interface, and will cause all RIP routes to be announced out of that interface with their actual metric. Announcing the actual metric opens the possibility for creating the very routing loops split horizon aims to prevent. This is generally safe to do in hub-and-spoke topologies, where there is no possibility for a loop. The consequences of allowing RIP announcements with actual metric in any other topology should be carefully evaluated. See [RIP1] for more insight on when disabling split horizon is appropriate.

RIP Design Recommendations

Following are design recommendations for some of the most popular frame relay configurations with RIP.

Hub-and-Spoke with no Inter-Spoke Communications

In the classical hub-and-spoke case, where communication is required only between the hub site and its spokes, running RIP with the default split horizon is recommended for both the hub and the spoke sites. This will prevent the spoke sites from talking to each other, but it fits many companies' policies, where only the hub is meant to talk to the spokes. Using Poison Reverse is discouraged as it creates unnecessary additional traffic on the frame relay subnet. Considering Figure 10, following are the configurations of the three routers with split horizon.

Router A

```
port set hs.4.1 wan-encapsulation frame-relay speed 51840000
frame-relay set lmi state enable ports hs.4.1
frame-relay create vc port hs.4.1.16
frame-relay create vc port hs.4.1.17

vlan create WAN port-based id 10
interface create ip WAN_INT address-netmask 10.1.1.1/24 vlan WAN
interface create ip LAN_A address-netmask 192.168.254.253/24 port
et.1.1
vlan add ports hs.4.1.16-17 to WAN

rip add interface LAN_A
rip add interface WAN_INT
rip set interface LAN_A version 2
rip set interface WAN_INT version 2
rip start
```

Router B

```
port set hs.4.1 wan-encapsulation frame-relay speed 51840000
frame-relay set lmi state enable ports hs.4.1
frame-relay create vc port hs.4.1.116

vlan create WAN port-based id 10
interface create ip WAN_INT address-netmask 10.1.1.2/24 vlan WAN
interface create ip LAN_B address-netmask 172.16.1.1/24/24 port et.1.1
vlan add ports hs.4.1.116 to WAN

rip add interface LAN_B
rip add interface WAN_INT
rip set interface LAN_B version 2
rip set interface WAN_INT version 2
rip start
```

Router C

```
port set hs.4.1 wan-encapsulation frame-relay speed 51840000
frame-relay set lmi state enable ports hs.4.1
frame-relay create vc port hs.4.1.117
```

```
vlan create WAN port-based id 10
interface create ip WAN_INT address-netmask 10.1.1.3/24 vlan WAN
interface create ip LAN_C address-netmask 172.17.1.1/24/24 port et.1.1
vlan add ports hs.4.1.117 to WAN

rip add interface LAN_C
rip add interface WAN_INT
rip set interface LAN_C version 2
rip set interface WAN_INT version 2
rip start
```

Notice that the default RIP setting is split horizon, so no special command is needed to configure it.

Hub-and-Spoke with Inter-Spoke Communications

The same topology, illustrated on Figure 10, can provide Inter-Spoke communications via the hub site by a simple change in the RIP configuration at the hub router (Router A), and maintaining the same configuration at the spoke routers.

Router A

```
port set hs.4.1 wan-encapsulation frame-relay speed 51840000
frame-relay set lmi state enable ports hs.4.1
frame-relay create vc port hs.4.1.16
frame-relay create vc port hs.4.1.17

vlan create WAN port-based id 10
interface create ip WAN_INT address-netmask 10.1.1.1/24 vlan WAN
interface create ip LAN_A address-netmask 192.168.254.253/24 port
et.1.1
vlan add ports hs.4.1.16-17 to WAN

rip add interface LAN_A
rip add interface WAN_INT
rip set interface LAN_A version 2
rip set interface WAN_INT version 2
rip set interface WAN_INT xmt-actual enable
rip start
```

The result is that Router A will announce **all** of its RIP routes via interface WAN_INT, including those learned via WAN_INT. For example, router A will learn the route 172.16.1.0/24 from Router B with a metric of 1 and will install it this way in its routing table. Then it will announce it **back to Router B with a metric of 2**. In the overwhelming majority of cases this will work just fine, as Router B knows that 172.16.1.0/24 is directly attached to it and will disregard the announcement it received from Router A with regard to that route.

There are failure modes however which will cause a routing loop for several minutes, until RIP "counts to infinity". For example, if Router B loses connectivity to 172.16.1.0/24, it will continue to receive announcements about that network from router A. Router B to think that 172.16.1.0/24 is reachable through Router A, which would create a routing loop. The reader is encouraged to consult [RIP1] RFC 1058, Routing Information Protocol for more generic discussion on the situations in which the lack of split horizon is a problem.

OSPF Specifics

OSPF affords a lot of flexibility and opportunity for mistakes when configured on NBMA interfaces. Please see section 0 for definition of the OSPF Network Modeling terms, we will not repeat this discussion here. We will only summarize it in Table 2.

OSPF NETWORK TYPE	AVAILABLE FOR FRAME RELAY ?	IS DR/BDR ELECTED?
Point-to-Point	YES*	NO
Point-to-Multipoint	YES*	NO
Broadcast	YES	YES
NBMA	YES	YES

Table 2: OSPF Network Type Comparison

* Not available in firmware releases prior to 3.1

We can see that a frame relay interface can be configured with all possible OSPF network types (and with only two of them before ver. 3.1). In the following discussion we do not consider the Broadcast network type, as it is inappropriate for frame relay interfaces[1]. Among the remaining network types, it is significant whether or not a DR/BDR is elected. NBMA network type has to select DR/BDR before OSPF link state updates can occur. Since it is imperative that all routers on a subnet must have connectivity to the DR and the BDR, it is not acceptable to leave the DR/BDR selection to the protocol for NBMA networks. Things are simplified in Point-to-Point and Point-to-Multipoint networks, where there is no DR/BDR election.

Let us consider again Figure 10 for an illustration of the DR/BDR election issues. If we configure router's A, B, and C frame relay interfaces as NBMA, the OSPF processes in those routers will interpret the frame relay network as a multi-access LAN, where any router can talk to any other router. This is not true for this network though. Router A can talk to both routers B and C, but routers B and C cannot communicate directly to each other, as there is no PVC to connect them. Generically, in a hub-and-spoke topology, only the hub router can talk to any other router, therefore only the hub router is capable of performing the DR/BDR function.

[1] There is one obscure application where configuring OSPF Broadcast interface on a frame relay interface is appropriate; it is discussed at the end of section 0.

However OSPF does not know that – it assumes full connectivity because of the network type we have configured. Consequently, it will follow its normal algorithms for DR/BDR election, which can very well lead to the election of, say, router B as a DR, and router A as a BDR. **This is broken! Router C cannot talk to the DR for the segment. Unpredictable routing anomalies will result from such DR/BDR election.** Therefore, configuration changes must be made to ensure that:

1. Only router A can be elected as a DR
2. No other router can elected for DR or BDR

This will be accomplished by explicitly configuring the router priority of router A to 1, and the router priority of all other routers to 0.

On router A:

```
ospf set interface <NAME> priority 1
```

On routers B and C:

```
ospf set interface <NAME> priority 0
```

Note that a Backup Designated Router will not be elected on that subnet. This was intended with the above configuration, since there are the same connectivity requirements for the BDR as there are for the DR, but in a hub-and-spoke topology, there is no way to satisfy them. As a result, the subnet will operate without a BDR.

Let us now consider Figure 11, which illustrates a partial, non-hub-and-spoke frame relay connectivity between four routers.

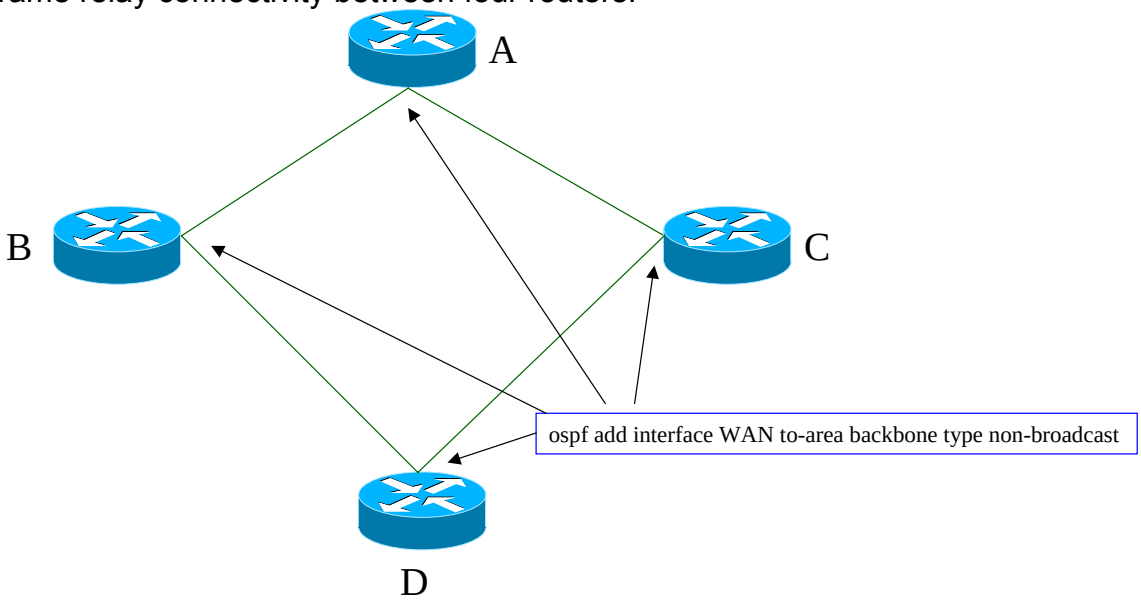


Figure 11: Problems with OSPF Over a Partial Frame Relay Mesh

No router in this network has connectivity to **all** other routers. For that reason no router can be a DR or BDR. **Therefore the configuration shown on Figure 11 will not work.** In order for it to work, the OSPF network type on all routers should be set to **point-to-multipoint**, a new option, available in version 3.1.
ospf add interface WAN to-area backbone type point-to-multipoint

Below we consider several typical frame relay topologies and provide configuration recommendations for them. The recommendations are different in the cases of pre- and post- 3.1 firmware, because this release level introduces significantly more network types and allows more flexibility.

OSPF Design Recommendations

For RS Firmware 3.1 and Later

Beginning with version 3.1, the RS firmware supports two additional OSPF network types for frame relay interfaces - point-to-multipoint and point-to-point. These are the most appropriate interface type for frame relay topologies.

Point-to-Multipoint network type can be configured on any frame relay interface, regardless of the number of PVCs it contains. It is defined at the time of adding the interface to the OSPF routing process:

```
interface create ip WAN address-netmask 10.1.1.1/24 vlan WAN
…
ospf create area backbone
ospf add interface WAN to-area backbone type point-to-multipoint
ospf start
```

Point-to-Point network type, in contrast, can be configured only on frame relay subinterfaces, i.e. interfaces that contain a single PVC and are defined as point-to-point. Contrary to the configuration logic of all other OSPF network types, the point-to-point nature of the interface is declared when the interface is defined. Later, when the interface is added to the OSPF routing process, its OSPF network type cannot be configured – it is implicitly assumed to be point-to-point, and cannot be overridden:

```
interface create ip WAN address-netmask 10.1.1.1/24 port hs.4.1.116
type point-to-point

ospf create area backbone
ospf add interface WAN to-area backbone
ospf start
```

Both point-to-multipoint and point-to-point network types represent each PVC as a point-to-point link. No DR/BDR selection occurs. If a PVC goes down, there is no problem beyond losing connectivity between the two routers the PVC connects. In all other interface types, there are typically more significant consequences. A minor disadvantage of these two network types is that they

install a host route for the IP address at the far end of each PVC in the forwarding table. This inefficiency is more than made up with the resilience against misconfiguration this network type provides.

On the following three figures, we show various topologies, configured with point-to-multipoint and point-to-point network types, as appropriate


```
port set se.4.1 wan-encapsulation frame-relay speed 1536000
frame-relay set lmi state enable ports se.4.1
frame-relay create vc port se.4.1.16

vlan create WAN port-based id 10
interface create ip WAN_INT address-netmask 10.1.1.1/30 vlan WAN
vlan add ports se.4.1.16 to WAN

ospf create area backbone
ospf add interface WAN_INT to-area backbone type point-to-multipoint
ospf start
```

```
port set se.4.1 wan-encapsulation frame-relay speed 1536000
frame-relay set lmi state enable ports se.4.1
frame-relay create vc port se.4.1.116

vlan create WAN port-based id 10
interface create ip WAN_INT address-netmask 10.1.1.2/30 vlan WAN
vlan add ports se.4.1.116 to WAN

ospf create area backbone
ospf add interface WAN_INT to-area backbone type point-to-multipoint
ospf start
```

Figure 12: Two routers connected with a single PVC (Point-to-Multipoint Configuration)


```
port set se.4.1 wan-encapsulation frame-relay speed 1536000
frame-relay set lmi state enable ports se.4.1
frame-relay create vc port se.4.1.16

interface create ip WAN_INT address-netmask 10.1.1.1/30
                port se.4.1.16  type point-to-point

ospf create area backbone
ospf add interface WAN_INT to-area backbone
ospf start
```

```
port set se.4.1 wan-encapsulation frame-relay speed 1536000
frame-relay set lmi state enable ports se.4.1
frame-relay create vc port se.4.1.116

interface create ip WAN_INT address-netmask 10.1.1.2/30
                port se.4.1.116 type point-to-point

ospf create area backbone
ospf add interface WAN_INT to-area backbone
ospf start
```

Figure 13: Two routers connected with a single PVC, defined as a subinterface (Point-to-Point configuration)

The difference between Figure 12 and Figure 13 is only in the network type of the frame relay interface. Either network type is appropriate for this link, and both

are represented internally in approximately the same way. In some cases, the decision to use point-to-multipoint vs. point-to-point is governed by interoperability concerns.

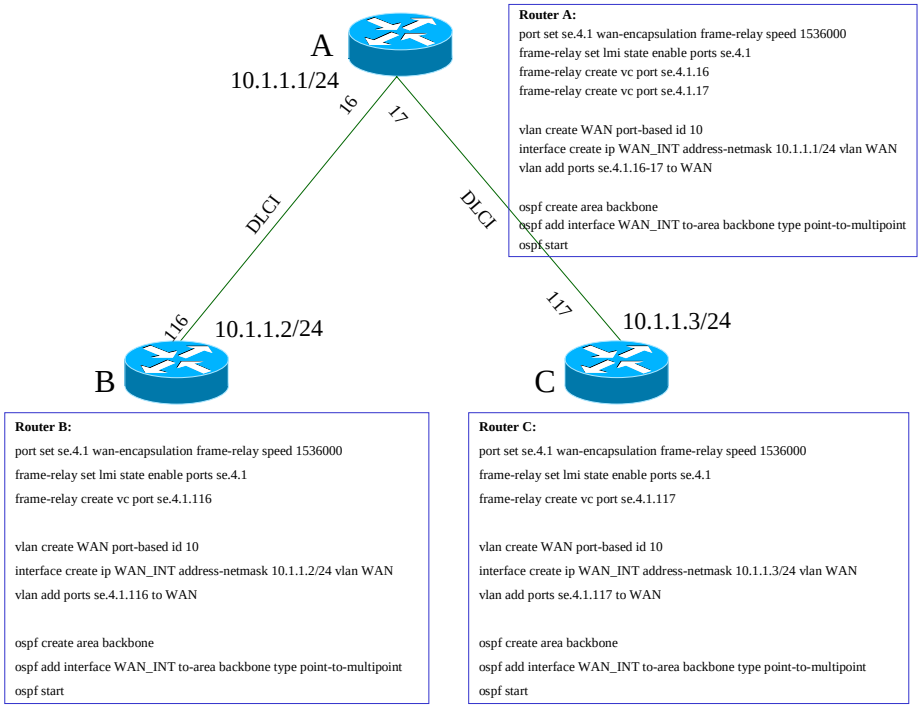


Figure 14: Hub-and-Spoke, one VLAN, Point-to-Multipoint

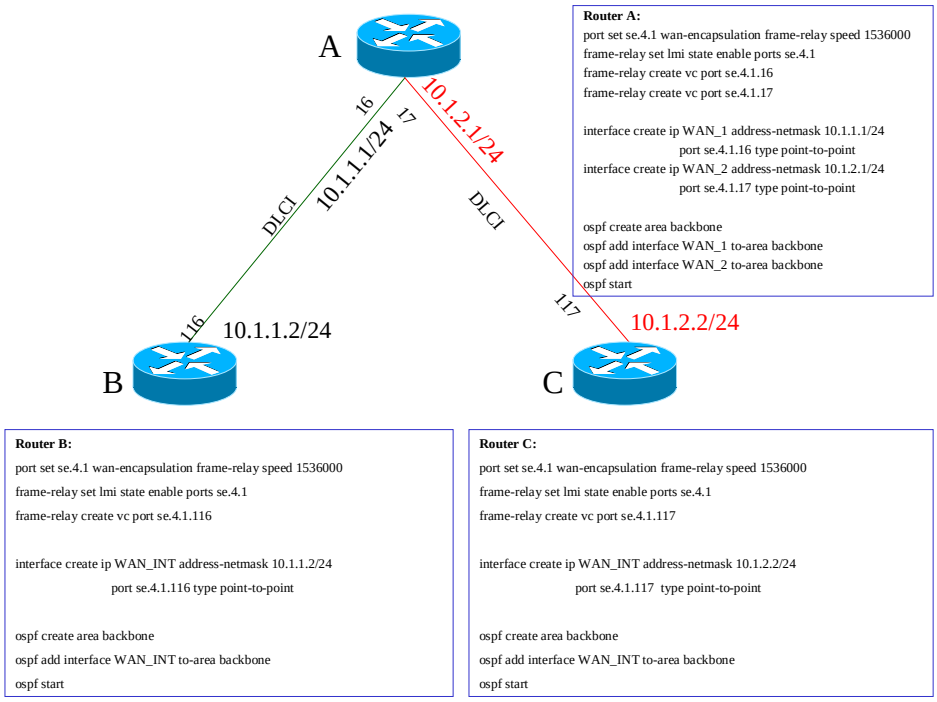


Figure 15: Hub-and-Spoke, subinterfaces, Point-to-Point

Figure 14 and Figure 15 illustrate two ways to configure the same physical hub-and-spoke topology. The first one is to treat it as a LAN, and assign IP addressing as if it were a LAN. The second assigns each PVC in a separate VLAN, utilizing subinterfaces.

Note that the IP configuration shown on Figure 15 (each PVC is its own IP subnet) can also be implemented using point-to-multipoint interface type, however the configuration will be longer.

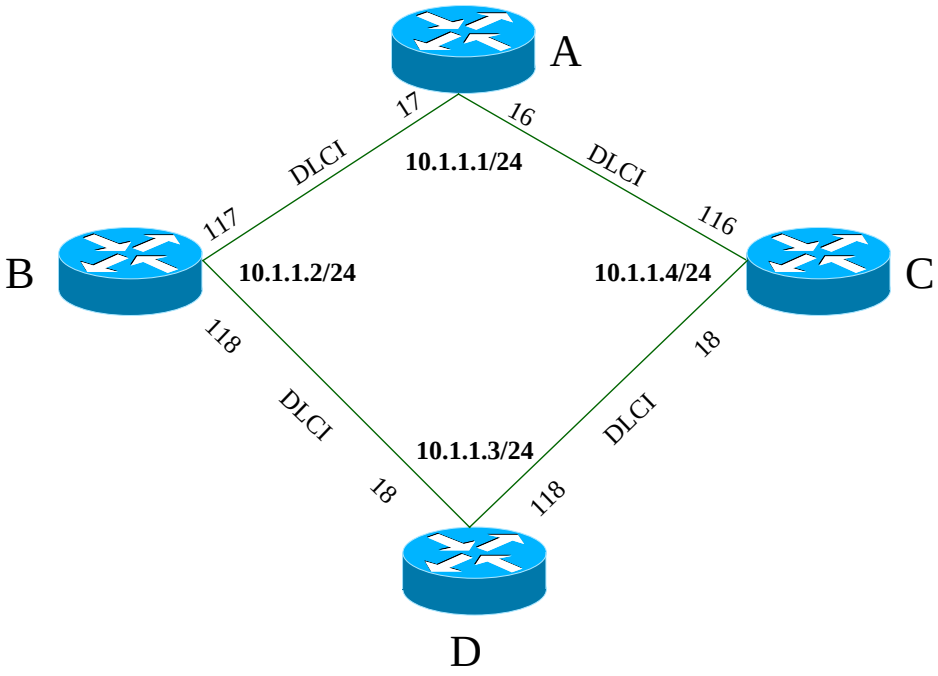


Figure 16: Partial mesh, point-to-multipoint

Point-to-multipoint OSPF network type is particularly appropriate for partial mesh topologies, such as the one shown on Figure 16. We only show the configuration of router D below, the remaining routers' configuration is similar.

Router D:

```
port set hs.4.1 wan-encapsulation frame-relay speed 51840000
frame-relay set lmi state enable ports hs.4.1
frame-relay create vc port hs.4.1.18
frame-relay create vc port hs.4.1.118

vlan create WAN port-based id 10
interface create ip WAN_INT address-netmask 10.1.1.3/24 vlan WAN
vlan add ports hs.4.1.18 to WAN
vlan add ports hs.4.1.118 to WAN

ospf create area backbone
ospf add interface WAN_INT to-area backbone type point-to-multipoint
ospf start
```

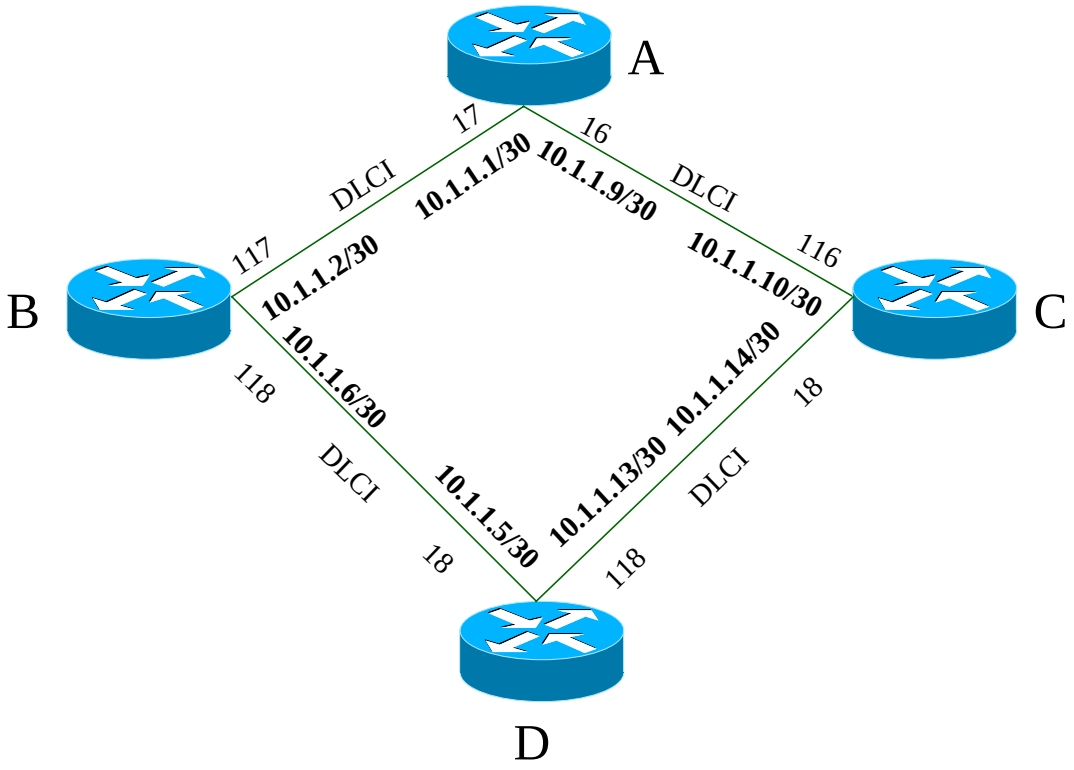


Figure 17: Partial mesh, subinterfaces, point-to-point

Point-to-point network type can be used to configure the same partial mesh topology, but in this case each PVC must be defined as its own IP subnet, resulting in potential IP address waste. This configuration is illustrated on Figure 17.

Router D:

```
port set hs.4.1 wan-encapsulation frame-relay speed 51840000
frame-relay set lmi state enable ports hs.4.1
frame-relay create vc port hs.4.1.18
frame-relay create vc port hs.4.1.118

interface create ip WAN_1 address-netmask 10.1.1.5/30 port hs.4.1.18 type point-to-point
interface create ip WAN_2 address-netmask 10.1.1.13/30 port hs.4.1.118 type point-to-point

ospf create area backbone
ospf add interface WAN_1 to-area backbone
ospf add interface WAN_2 to-area backbone
ospf start
```

As a general rule of thumb, when the frame relay subnet has to be addressed as if it is a LAN, point-to-multipoint network type is the only choice. If each PVC can be treated as separate VLAN, then either point-to-multipoint or point-to-point network type can be chosen.

Also, we are taking the opportunity here to demonstrate the **local significance of the DLCIs**, a property discussed in more details in section 0. This is

demonstrated on Figure 17 router D, where DLCIs 18 and 118 point to routers B and C respectively. This is not a problem, despite the re-use of those DLCIs elsewhere in the network. (This property is unrelated to the current discussion; we just mention it as a reminder since Figure 16 and Figure 17 provide the occasion).

For RS Firmware Prior to ver. 3.1

In firmware releases prior to ver. 3.1, there is no support for OSPF network types of point-to-multipoint and point-to-point. The choices for configuring network types are much more limited,

port set se.4.1 wan-encapsulation frame-relay speed 1536000
frame-relay set lmi state enable ports se.4.1
frame-relay create vc port se.4.1.16

vlan create WAN port-based id 10
interface create ip WAN_INT address-netmask 10.1.1.1/30 vlan WAN
vlan add ports se.4.1.16 to WAN

ospf create area backbone
ospf add interface WAN_INT to-area backbone type non-broadcast
ospf add nbma-neighbor 10.1.1.2 to-interface WAN_INT
ospf start

port set se.4.1 wan-encapsulation frame-relay speed 1536000
frame-relay set lmi state enable ports se.4.1
frame-relay create vc port se.4.1.116

vlan create WAN port-based id 10
interface create ip WAN_INT address-netmask 10.1.1.2/30 vlan WAN
vlan add ports se.4.1.116 to WAN

ospf create area backbone
ospf add interface WAN_INT to-area backbone type non-broadcast
ospf start

Figure 18: Pre- 3.1: On a single PVC between two routers, configure the OSPF network type as NBMA

so are the topologies supported. For subnets having pre-3.1 or other vendor's routers, incapable of either point-to-point or point-to-multipoint network type, the following is recommended for the
respective topologies:

When only two routers are connected over a single PVC, there is really no disadvantage in configuring the link as NBMA. One router will be chosen as DR, the other will be a BDR. If the link fails, the situation will be correctly reflected in the LSDB, and there will be no confusion as to who is a DR/BDR. This however is not the case in the following topologies.

Notice the use of the
`ospf add nbma-neighbor <IP Address> to-interface <Interface>`
command. It needs to be issued at least on one end of the PVC for OSPF adjacency to be established.

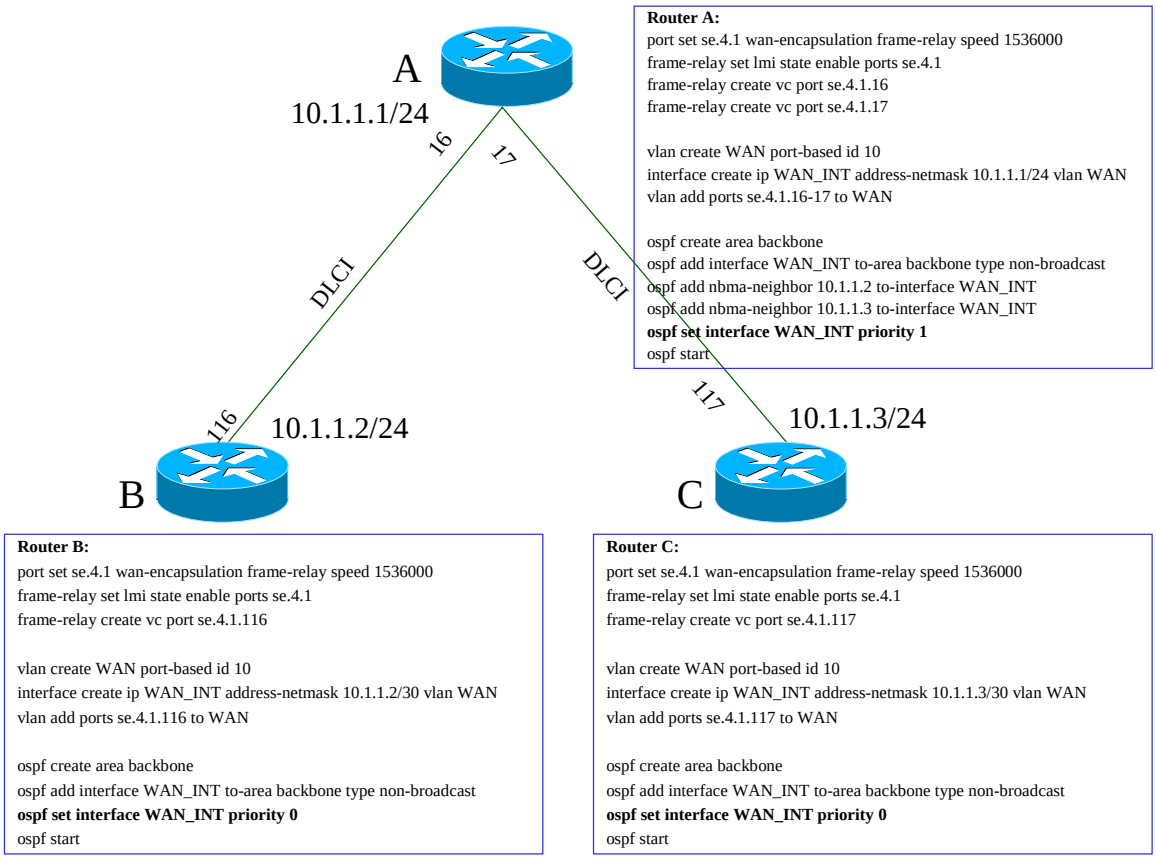


Figure 19: Hub-and-Spoke with NBMA interface type: Interface priorities must be manually set

It is normally a requirement for NBMA type networks to have full mesh connectivity between all routers. The reason for that is that for proper DR/BDR election every router must be able to send HELLOs to every other router on the subnet. If that does not happen, various problems will occur – more than one DR can be selected, some routers will not be able to communicate with the DR, etc., details are discussed in the beginning of section 0.

The requirement for full mesh connectivity can be bent slightly in cases where the selection of the DR/BDR is deterministic, and connectivity from all other routers to the DR/BDR can be ensured. That can be accomplished in a hub-and-spoke topology with the following stipulations:

1. The DR will be the hub router
2. There will be no BDR
3. The spoke routers can never become DR or BDR.
4. The hub router is an RS (for details on this requirement, see section 0 - ARP discussion)

The first three rules are enforced by manually configuring the router priority of the appropriate interfaces. The hub router frame relay interface is set to priority 1,

and the spoke routers' interfaces are set to priority 0. This makes the hub router the only one eligible for election as a DR.

A common mistake is made when the hub router is configured simply with higher priority than the spoke routers, and the spoke router priority is different than zero. This is an incorrect setting, since if the hub router goes down, another one will elect itself as a DR. When the hub router comes back up, it will NOT be re-selected as a DR, unless the running DR disappears. This is typically very hard to troubleshoot, and special care should be taken to ensure that **the interface priority of the spoke routers is 0**.

Notice the use of the
`ospf add nbma-neighbor <IP Address> to-interface <Interface>`
command at the hub router. It needs to be issued at least on one end of each PVC for OSPF adjacency to be established. This is so because the router cannot use broadcast or multicast to send the HELLOs, and it does not know at this point who its neighbors are.

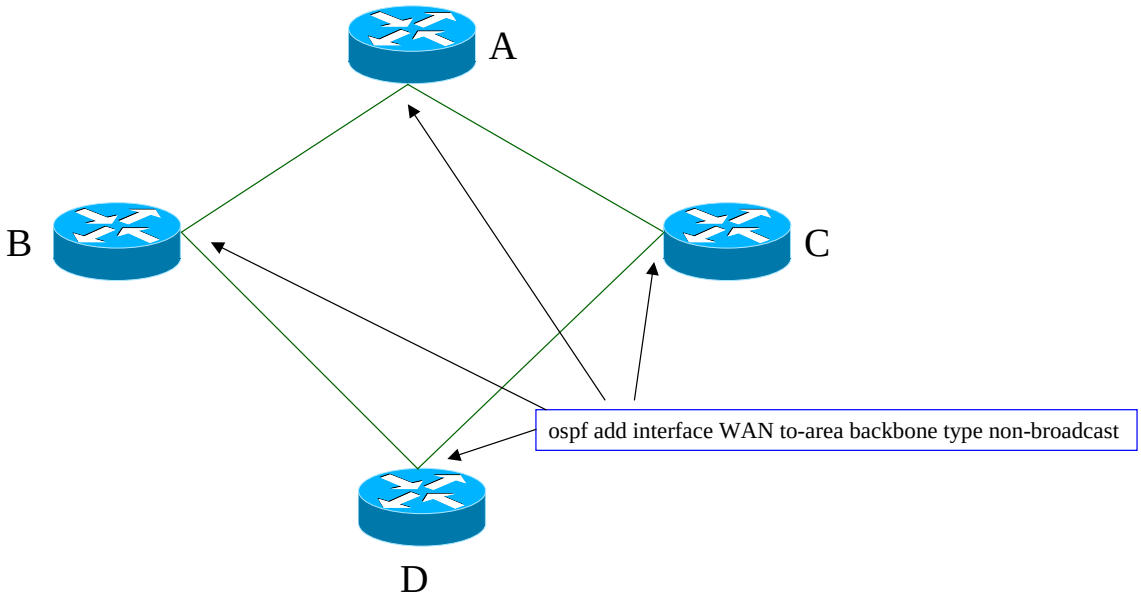


Figure 20: Partial mesh with NBMA - does not work

Lastly, we consider the partial mesh connectivity shown on Figure 20. This type of connectivity cannot be supported with NBMA Network type, and therefore cannot work in pre- 3.1 firmware. In order to support this subnet within the bounds of NBMA network type, two additional PVCs must be added – between routers A and D, and between routers B and C.

> ***Note on using Broadcast OSPF network type on NBMA interfaces:***
> *When an interface that is defined a broadcast OSPF network type comes up, OSPF waits for 40 seconds before beginning to transmit HELLOs and*

establishing adjacency. The reason for this wait is so the router can learn of an existing DR/BDR on the subnet (or otherwise assume that role). This behavior can be helpful in NBMA networks when it is desired for a router to delay its adjacency forming. In that case, the frame relay interface of the router will be defined as "broadcast" even though it is connecting to a NBMA network. This technique should only be used after thorough understanding of its effects and testing it with the specific release of code!

ARP and Other Broadcast Protocols with Frame Relay

ARP in the General Case (non-RS)

Partial mesh topologies, combined with the lack of broadcast capabilities pose special challenges to protocols designed to operate over broadcast-capable media. We specifically discuss ARP in this section, but the logic would apply to any other broadcast protocol.

Consider Figure 21, where we have two workstations attached to the LANs at the two spoke sites. If 172.16.1.2 was to PING 172.16.2.2, it will work, as traffic will be routed via the hub router. If however, 172.16.2.2 was to ping the frame relay interface of router C, that will NOT work. The reason for this anomaly is that router B will ARP for router C's frame relay address, 10.1.1.3. Router C will NOT receive this ARP broadcast , as there is no PVC between routers B and C. Therefore router B will never receive an ARP reply for 10.1.1.3, and will assume the address is not existent.

In the former case (LAN-to-LAN), router B will not ARP for 10.1.1.3, but instead will forward the ping packet to router A (which is his next hop for network 172.16.2.0/24). Router A will in turn forward the packet to router B, then router B will ARP on its LAN for the destination address of 172.16.2.2.

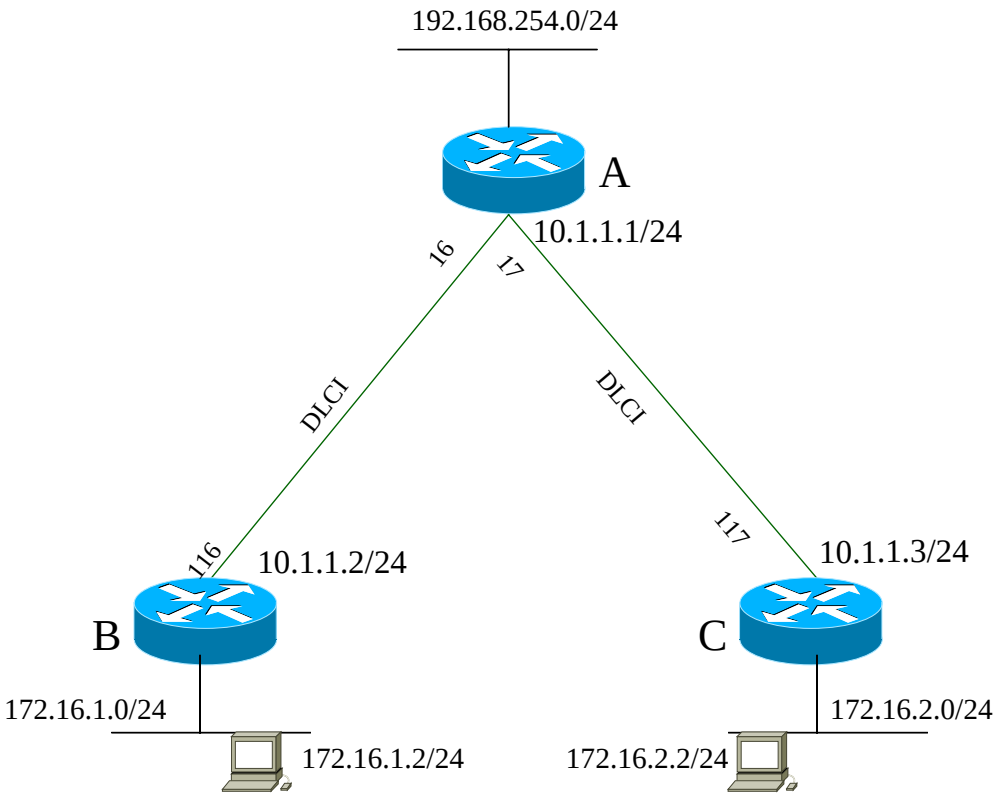


Figure 21: ARP in an NBMA Network

An interesting observation is that this anomaly is only present when a spoke router is forced to ARP for the frame relay interface of another spoke router. There is no problem when the hub router ARPs for any of the spokes, or a spoke ARPs for the hub router's interface, since the direct PVC connectivity in this case is in place.

The anomaly described above does not present a problem for normal LAN-to-LAN traffic between spoke sites. It is only a problem if troubleshooting or network management needs to originate from a spoke site. This behavior is characteristic for broadcast protocols in an NBMA subnet; most routers will behave as described here. However, the RS behaves slightly differently, and improves the handling of broadcast protocols in an NBMA subnet.

How the RS handles ARP in NBMA subnets?

In order to improve the connectivity in NBMA subnets and to avoid the anomaly described above, the RS has a different behavior for ARP requests/replies that it receives on a PVC, which is part of a VLAN – it replicates the ARP request/reply to all other PVCs, member of that VLAN. So if router A on Figure 21 is an RS, and it receives an ARP request originated from router B, trying to resolve router C's MAC address, router A will forward that ARP packet onto PVC 17. The request will be received and replied to by router C, and the reply will be replicated back to PVC 16. As a result, even though routers B and C do not have

direct connectivity between each other, they can resolve each other's MAC addresses.

To generalize, with regard to PVCs in the same VLAN, the RS behaves as a transparent bridge. Only traffic that needs to be replicated across PVCs will be replicated. That includes unknown (unresolved) unicast (flooding) and broadcast/multicast. Unicast traffic for known (learned) addresses will only be sent to the appropriate PVC for those addresses.

This broadcast emulation behavior of the WAN card may be helpful in some cases and confusing in others. It is generally recommended that the standard restrictions of NBMA media are observed in all designs, and this "broadcast emulation" capability is only used when absolutely needed.

The interested reader may want to read Appendix 0 for details on the mechanics of handling ARP on frame relay interfaces.

Recommendations
If possible, in a hub-and-spoke topology, all network management and troubleshooting traffic should originate from the hub site. In other partial mesh topologies, it should originate from a site that has direct PVC connectivity to **ALL** other sites.

A full mesh topology may also be considered if economically feasible.

Quality of Service (QoS)

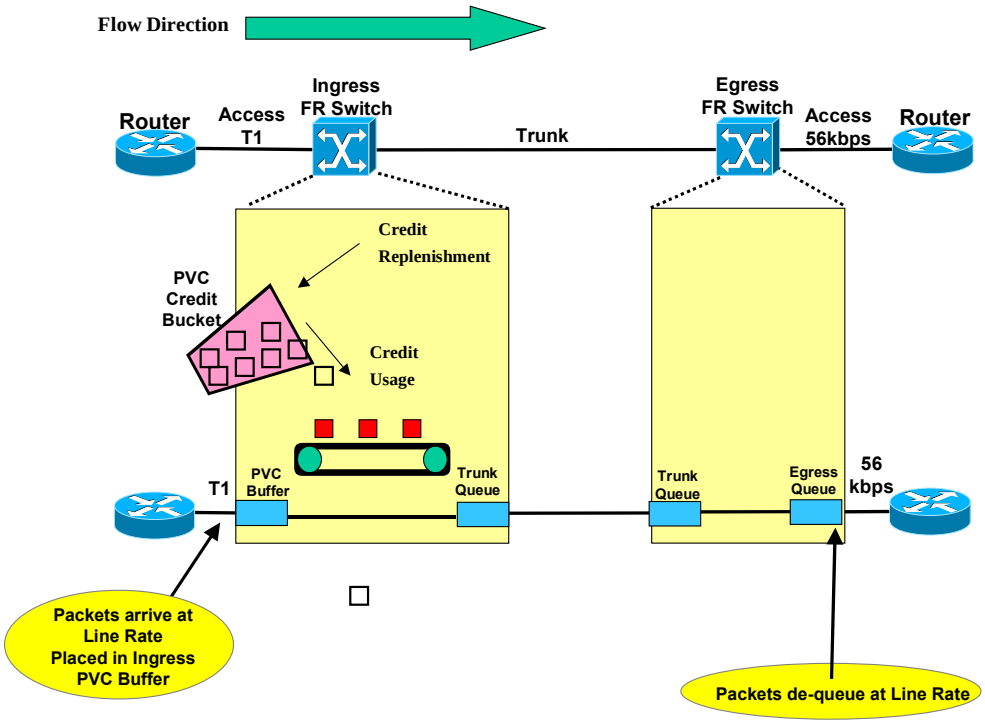
Traffic Flow in a Frame Relay Network

Frequently the frame relay network is depicted as a "cloud" and considered to behave the same as a private line. While this approximation works fairly well for lightly utilized frame relay links, it fails dramatically and leads to very poor performance when moderate-to-high utilization is experienced.

To take advantage of the QoS tools available with the RS, thorough understanding is necessary of the way traffic flows through the frame relay "cloud". Section 0 defines the basic terms in this discussion – we will not repeat these definitions here.

Consider Figure 22, which illustrates a PVC defined between two routers, and the elements involved in passing traffic through it. For most providers, frame processing occurs at ingress, therefore we only show the details of the ingress switch. Traffic always enters and exits the frame relay network at the speed of the relevant ingress/egress access circuits. In our example, traffic enters at a T1 rate, and exits at a 56 Kbps rate. The ingress frame relay switch maintains an input buffer for every PVC; its size can be anywhere from 0 to 64 KB.

Figure 22: Handling of packets in a Frame Relay "cloud"



Of course, traffic on the PVC is bi-directional, here we only illustrate flows that go from left to right. The same processing occurs for flows in the opposite direction. Figure 22 is slightly simplified; in reality there are two credit buckets – one representative of Bc credits, and another for Be credits.

Once traffic is in the PVC input buffer, it gets de-queued using the token bucket mechanism. This mechanism calls for the switch to maintain a token, or credit, bucket for each PVC. The switch places credits in the bucket at a **Credit Replenishment** rate, and takes them out of the bucket at a **Credit Usage** rate. When a packet is processed from the PVC buffer, a token is deleted, or used, from the bucket. No user data can get into the network without a credit.

The **Credit Replenishment Rate** is equal to the CIR, plus some extra credits, which are granted when the PVC has been idle for some time. The **Credit Usage Rate** either:

- The minimum of the access circuit speeds on both sides of the PVC, if the network is not congested (in our case, that is 56 Kbps), or
- The CIR (if the network is congested).

If the DE bit is set by the customer router, handling of those packets depends on the state of the PVC ingress buffer. Normally, DE frames marked by the customer are discarded at ingress only if the PVC ingress buffer reaches very high utilization, usually 90%. Otherwise, DE frames are accepted, and

attempted to be delivered, but could be dropped downstream if congestion occurs. Some providers may also evaluate each frame at ingress whether or not it is within Bc, and if it is not, mark it as DE (marking by the network).

CIR, Bc, Be, and access circuit speeds are all measured in bits-per-second. Credits are measured in bits. The values of CIR, Bc, Be, and circuit speed will determine the availability of credits in the bucket, as well as the rate at which they can be used.

When **Traffic Shaping** is not defined, the router has no knowledge of these three values, and offers traffic to the network at access line speed. When traffic shaping is defined, the router is configured with CIR, Bc and Be, and traffic is offered to the network according to those values. On the RS, traffic shaping is enabled by configuring any of the above three parameters. To configure efficient traffic shaping, one must not only know the CIR, but also the Bc and Be values the provider is setting.

Another important parameter to know from the frame relay provider, is the depth of the PVC Ingress Buffer. It has direct bearing on latency, ability to handle bursts, and the effectiveness of any traffic prioritization done at the router:

- Latency and Jitter – The larger the buffer, the more data can be stored in it when no tokens are available. If traffic arrives at a rate equal or lower than the CIR, then none of it will be stored in the PVC ingress buffer. This traffic will experience a nominal and constant latency, which will only depend on the geographic length of the PVC. On the other hand, if more traffic than the CIR is offered to the network, some of it will be stored in the PVC ingress buffer (up to its capacity). Certain amount of time will pass for the buffer to be drained at **Credit Usage Rate**. If during this interval of time new traffic arrives from the router, it will be placed at the end of the PVC ingress buffer, where it will have to wait its credits. As a general rule, the larger the ingress PVC buffer, the less traffic will be dropped at ingress and the less predictable the jitter for traffic that is accepted.
- Bursting Capabilities – Larger PVC ingress buffer will handle larger bursts.
- Effectiveness of traffic prioritization at the router – larger PVC ingress buffers tend to invalidate traffic prioritization done at the router, smaller buffers tend to promote it. Even if the router expedites high priority traffic, if it arrives in a PVC buffer that already has low-priority traffic in it, the high-priority traffic will have to wait, as the PVC ingress buffer is a FIFO queue. The larger the buffer, the more pronounced that effect will be.

From the above discussion one may conclude that buffering at ingress is a bad thing. Observe however that if there is no PVC ingress buffer, then the capacity of a frame relay PVC will be equal to the capacity of a private circuit at the speed equal to the CIR. The economies of frame relay are attractive when bursting **above** PVC level is available, therefore buffering at ingress is a necessary evil.

A final thought regarding buffering at ingress is that the absolute size of the buffer is not as important as is the ratio buffer size/Credit Usage Rate. Buffer size is typically static (64 KB) with most providers, therefore the higher the credit usage rate, the less pronounced the negative effects of a large buffer are.

Disappointingly, after all this discussion, we have concluded that more bandwidth is better – hardly a breakthrough discovery. Hopefully the discussion in this section helps in understanding the provider side of a frame relay network. Below is a list of some basic questions to be asked of a provider, when provisioning frame relay service.

1. For a PVC, what are the CIR, Bc, and Be values **defined at the switch**?
2. For a PVC, what is the depth of the PVC ingress buffer?
3. Does the provider set the DE bit **in the network**? Under what condition?
4. If the DE bit is set by the customer, what is the criteria for dropping DE packets at ingress?

Knowing the answers to those questions will help fine-tuning QoS settings on the RS side. On the other hand, if QoS is not needed, and only best effort service will be configured, these questions are not as important.

Key QoS Features of the RS

The RS provides a comprehensive set of tools for QoS definition, compliance, and enforcement. Good knowledge of those tools, coupled with understanding how the provider's frame relay network handles traffic, is essential for delivering QoS over frame relay. Here we discuss the RS capabilities and how they can be matched to those of a frame relay network provider.

Hardware Capacity Considerations

The RS performs at wire speed by using a non-blocking, dynamic multipoint switching fabric and custom ASICs that deliver wire-speed switching and routing of Layer 2, Layer 3, and Layer 4 flows. Although true in the general case, the above statement needs a couple of refinements, when WAN traffic is involved:

1. **The WAN card switches traffic in software.** The WAN card, unlike the Ethernet or POS cards, has a dedicated CPU (WCPU), which handles most of the packet forwarding functions on WAN interfaces. As a result, while the forwarding capacity between other line cards is measured in millions of packets-per-second (30 Million pps for RS8600), the WAN card only has capacity of 100,000 pps. In real-life traffic situations, and given the form factor of the card, this capacity translates to line rate forwarding. The worst-case lab scenario is with a dual HSSI card and 64 byte-sized packets, then the throughput of the whole WAN card, based on 100,000 pps, will be 51 Mbps (half-duplex DS3).

2. **Speed mismatch requires buffering**. While the RS performs at wire speed, it cannot transmit traffic at a rate exceeding the speed of the outbound interface (the opposite would be magic). Therefore, if traffic is streamed at a line rate from a Gigabit Ethernet interface into a 10 Mbps interface, buffering must occur at the RS. The larger the speed mismatch between the two interfaces, the larger the buffering needs on the outbound interface are. Since the WAN interfaces (especially ones represented by frame relay PVCs) can be very slow, they are the ones that are the most susceptible to buffer overflow and therefore may require buffer size adjustment and active management.

Traffic Shaping and Queuing on the RS

When traffic enters the RS faster than it can exit it over a certain WAN port, congestion occurs. When it does, traffic is buffered on the WAN card. For frame relay, each port maintains a queuing structure, illustrated on Figure 23, which handles both traffic shaping and queuing for the port. It is important to keep in mind that queuing only occurs when there is congestion. In the absence of congestion, traffic is directly transmitted at line rate.

There are four priority queues for each best effort PVC – control, high, medium, and low. These queues have user-configurable depth, which determines how many packets of the corresponding priority will be buffered in case of congestion. In addition, there are two per-port queues, Interrupt, and Shaped. The Interrupt queue contains WAN control traffic, such as LMI frames (or LCP and NCP packets in the case of PPP interfaces). The shaped queues collect all conforming traffic from the shaped PVCs for that port. There are differences in how traffic is handled for shaped and best-effort PVCs, and we describe both cases in section 0.

The Interrupt queue is always serviced first, and the shaped queue is serviced second. Then, if bandwidth is available, the best-effort PVCs are serviced in a round-robin schedule. Best effort PVCs are the ones for which the CIR, Bc and Be are not defined. On Figure 23, PVCs 1, 2, and 3 are best-effort PVCs, while 4 and 5 are shaped. The priority queues for each PVC can be serviced either in strict priority order or based on WFQ percentages, if WFQ is enabled on the port.

Handling of Traffic in Shaped PVCs

For each shaped PVC the RS maintains a traffic counter, which is reset every Tc interval (Tc=Bc/CIR). At any point in time, the counter represents the number of bits that have been queued for transmission on that PVC during the current Tc interval.

As traffic arrives to be transmitted on a shaped PVC, the counter is incremented with the number of bits the newly arrived frame contains, and this value is compared with Bc+Be. If traffic level is below Bc+Be, the frame is considered

conforming and is placed in the shaped queue for transmission, otherwise it is dropped. Note that even with shaped PVCs, there are still buffering structures which allow conforming traffic to be placed in different groups for the purposes of traffic prioritization within the PVC.

Handling Best Effort (Non-Shaped) Traffic

Traffic that has no CIR/Bc/Be parameters defined is best effort traffic. Configurable number of buffers is maintained for control/high/medium/low priority traffic on every PVC. The PVCs are serviced in a round-robin fashion, and within each PVC the four buffers can be serviced in one of two ways:

- Strict priority – the default method – for the time interval dedicated to PVC n, service the control queue, when there is no more traffic waiting in the control queue, then service the high queue, when that traffic is transmitted, then service the medium queue, and finally, if there is time left, service the low queue.
- WFQ (Weighted Fair Queuing) – configurable per port, and applicable to all PVCs on that port – percentages of time are assigned for each of the four queues and the time slot for servicing the PVC is divided according to those percentages.

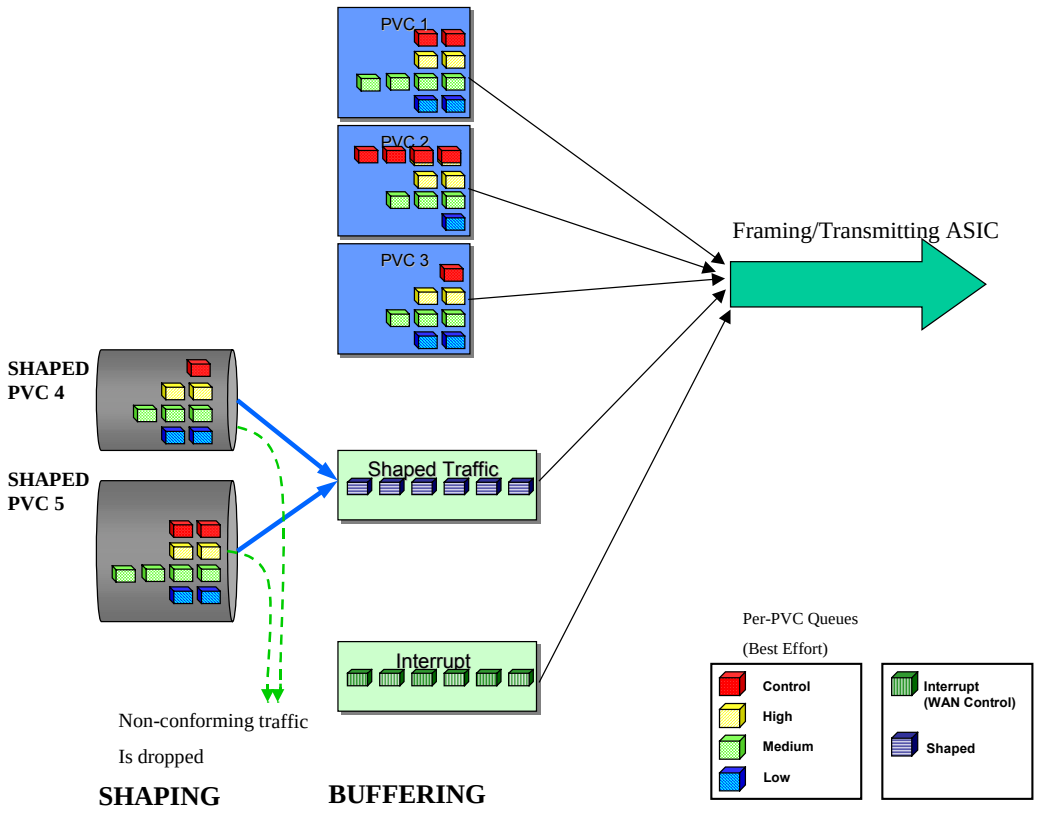


Figure 23: Queuing/shaping structure of a frame relay port

Traffic Shaping Summary

Traffic shaping is a unidirectional, outbound activity. It allows for outbound traffic to be throttled in a way that minimizes the chances for it being dropped in the frame relay network.

Shaped traffic, once classified as conforming, will be serviced with strict priority relative to the remaining, best effort traffic. It is important to realize that a user can oversubscribe a frame relay port with shaped traffic (when the sum of the Bc values for all shaped PVCs exceeds the port speed). The consequences of that will be serious:

1. There will not be enough circuit capacity to service the shaped FIFO queue. As a result, shaped traffic will be tail-dropped.
2. Best effort traffic will NOT BE SERVICED AT ALL! The PVC priority queues are not serviced until the shaped queue is empty.

The obvious recommendation is to NOT oversubscribe a frame relay port with shaped traffic. Ten gallons can't fit into a five gallon bucket!

If symmetrical shaping is required, then the routers at both ends of a PVC must be configured appropriately.

The above described mechanism for shaping/queuing provides great flexibility in tuning the RS behavior with regard to outbound traffic, depending on the business need that the PVCs serve.

- **Constant (low) latency, constant (low) throughput, no traffic loss in the network** – If the goal is emulate a CBR behavior for a PVC, then CIR and Bc should match exactly those set at the provider's switch, and Be should be set to 0. If Bc is unknown, it should be set equal to CIR. This will cause traffic to be offered to the network at the contracted rate; none of it should be dropped or buffered at the switch. While this will emulate CBR behavior, appropriate for voice and video applications, it negates frame relay's benefits of bursting and actually carrying more traffic than the contracted CIR. The PVC will behave very similar to a leased line with the speed of CIR.
- **Variable (low) latency, variable (high) throughput, no traffic loss in network** – when the goal is to maximize throughput, at the expense of some latency, but still maintain very low or no traffic loss in the frame relay network, all three parameters CIR, Bc, and Be should be matched to the provider's settings. For many providers, Bc+Be=(Access Circuit Speed), in other words, they allow bursting up to line speed, so if Be is not available, it can be assumed to be Be=(Access Circuit Speed)-Bc. Care should be taken with other providers, who offer "EIR- Excess Information Rate" contract, that is usually an offering in which Be < =(Access Circuit Speed)-Bc. With these settings, traffic will still be offered to the network at contracted rate, but bursts will be buffered at the provider's switch. Those will cause variable latencies for the traffic that follows. Bursts will never

be large enough to overflow the frame relay switch ingress buffer, as they are bound by Be at the router.

- **Variable (high) latency, variable (high) throughput, possible traffic loss in network –** this is the behavior with no traffic shaping defined. The router is unaware of any PVC-specific parameters, and will offer traffic to the network at access line rate. Given enough traffic, the frame relay switch ingress buffer may be fully utilized (resulting in large latencies) or over flown (resulting in dropped frames).

One final note on queuing and buffering. Because of the illustrative nature of this section, one may conclude that there are several levels of physical queues and buffers on the WAN card. This is NOT the case. All queuing on the card is single-level, and the buffers and queues illustrated in this section are actually counters within a single linear buffer, which yields the maximum performance.

BECN Adaptive Shaping

BECN Adaptive Shaping is a mechanism to throttle down the transmissions from a shaped PVC in case of network congestion. This feature is not available for best effort PVCs.

When configuring BECN adaptive shaping, a threshold number is configured, which specifies how many frames with the BECN bit set will be accepted before throttling action begins.

```
Frame-relay create service <service name> becn-adaptive-shaping
<threshold number>
```

Once the line above enables BECN adaptive shaping for a PVC this service profile applies to, a counter will be maintained every second for the BECNs received. If the number of BECNs exceeds the threshold, the CIR for the next second will be reduced by 1/8, and be will be set to 0. If the next second the number of BECNs also exceeds the threshold, the CIR will be reduced by additional 1/8 and so on, until the number of BECNs per second no longer exceeds the threshold. When that happens, the CIR is increased by 1/16 for the next second. That increase continues until the original CIR is reached or BECNs begin exceeding the threshold again.

BECN adaptive shaping is rather crude flow control mechanism because the feedback information travels in a direction opposite to the congestion, and the amount of feedback does not reflect the congestion severity. If router A is causing congestion when sending packets to router B, router A will receive the BECNs in data packets from router B. Therefore the frequency of the BECNs does not reflect the severity of the congestion, but merely the quantity of packets that router B had to send to A while the PVC was congested. As a result, an operator cannot set the BECN adaptive shaping to any kind of universal value, the appropriate value will depend on the particular higher level protocol running on the PVC. As an example, TELNET or HTTP traffic from A to B will provide

plenty of acknowledgement packets (and therefore frequent opportunity for setting the BECN bit), while an FTP transfer will typically require much less acknowledgement packets).

BECN adaptive shaping tends to be effective in conditions of mild congestion. It complements traffic shaping well. It only tends to be a performance detriment to the router if many PVCs become congested at the same time, merely having it configured on many PVCs does not have a significant performance impact.

Queue Depth Management and Queue Servicing Discipline

There are four priority queues for each best effort PVC – control, high, medium, and low. The high, medium, and low queues have user-configurable depth, which determines how many packets of the corresponding priority will be buffered. **We will emphasize again, that queuing and buffering ONLY occurs when there is congestion on the frame relay access circuit (i.e. there is more traffic that needs to exit the RS than the access circuit/ingress port can accept). All other congestion situations, such as congestion within the frame relay network, or at the egress of the PVC, will NOT result in any buffering on the transmitting RS.**

From a troubleshooting perspective then, when queuing occurs on the frame relay port of the RS, it is a clear indication of **congestion at the access circuit/ingress frame relay port**. The most likely reason for this congestion is oversubscription. Since oversubscription is a fact of life in frame relay networks (especially at hub sites), queuing will be a fact of life as well. When queuing does occur, it is important to manage it in a way that does not result in a significant traffic loss.

Queuing for each PVC can be observed with the command

```
FRAME-RELAY SHOW STATS PORTS <PVC DESIGNATION>
```

WHICH RESULTS IN THE FOLLOWING OUTPUT:

```
HUB-A# frame-relay show stats ports hs.4.1.16
hs.4.1.16:
    Admin state:               Up
    Operational state:         Up
    Compression:               Disabled/Down
    Local Ip address 10.1.1.1, Peer Ip address 10.1.1.2, netmask ffffff00
    Service Features Status
    -----------------------
    RMON:                      Disabled
    RED:                       Disabled

    Frame Relay DTE Statistics:
    --------------------------
        Rx FECNs             0
        Rx BECNs             0
        Rx Frames            4
        Rx Octets            208
        Tx Frames            1
        Tx Octets            24
        Rx Dropped           0
```

```
        Tx DE marked frames  0
        Rx DE marked frames  0
        Creation Time    0 days 0 hours 0 min 3 secs
        Last Time Change 0 days 0 hours 2 min 46 secs
        cir              16000
        Bc               16000
        Be               0

    Current Output Queue States
    ---------------------------
        High priority queue depth                       20
        Med  priority queue depth                       20
        Low  priority queue depth                       20
        Ctrl priority queue depth                       20
        Max number of pkts enqued in high priority queue  0
        Max number of pkts in enqued med priority queue   0
        Max number of pkts in enqued low priority queue   1
        Max number of pkts in enqued ctrl priority queue  1
        Current number of pkts in high priority queue     0
        Current number of pkts in med  priority queue     0
        Current number of pkts in low  priority queue     0
        Current number of pkts in ctrl priority queue     0
        Frames dropped due to ctl queue depth exceeded    0
        Frames dropped due to high queue depth exceeded   0
        Frames dropped due to med queue depth exceeded    0
        Frames dropped due to low queue depth exceeded    0
HUB-A#
```

This output is self-explanatory with regard to the state of the priority queues and any eventual packet drops. It shows no queuing and no packet drops occurring.

The next output is illustrative of the opposite situation – a severe congestion of the access circuit (we only show here the relevant portion of the output):

```
HUB-A# frame-relay show stats ports hs.4.1.16
(some irrelevant output removed)

    Current Output Queue States
    ---------------------------
        High priority queue depth                       20
        Med  priority queue depth                       20
        Low  priority queue depth                       20
        Ctrl priority queue depth                       20
        Max number of pkts enqued in high priority queue  0
        Max number of pkts in enqued med priority queue   0
        Max number of pkts in enqued low priority queue   20
        Max number of pkts in enqued ctrl priority queue  1
        Current number of pkts in high priority queue     0
        Current number of pkts in med  priority queue     0
        Current number of pkts in low  priority queue     20
        Current number of pkts in ctrl priority queue     0
        Frames dropped due to ctl queue depth exceeded    0
        Frames dropped due to high queue depth exceeded   0
        Frames dropped due to med queue depth exceeded    0
        Frames dropped due to low queue depth exceeded    8361
```

A couple of observations are important in this output:

- PVC 16 is experiencing severe congestion while sending low-priority traffic. Many frames are dropped because of this.

- The reason for this congestion may be traffic from PVC 16, but it may also be traffic from other PVCs, which share port hs.4.1. A prudent next step would be to issue ***frame-relay show stats ports hs.4.1*** and see which OTHER PVCs on the same port may be experiencing the same problem (and possibly contributing to it).

Observing and adjusting queuing sizes and service policies is an important part of QoS management. Several approaches can be taken, depending on the problem.

Increasing Queue Depth

Increasing the queue depth is often the first reaction of network managers to counteract high queue utilization. Unfortunately this only helps in very small fraction of the cases. It is usually effective when the average queue utilization is fairly low, but there are occasional spikes of traffic which cause buffer overflow.

If, over a long period of time, the queue status command yields something along the lines of the following output, then perhaps increasing the queue depth will alleviate the occasional traffic drops during bursts.

```
HUB-A# frame-relay show stats ports hs.4.1.16
(some irrelevant output removed)

    Current Output Queue States
    ---------------------------
        High priority queue depth                        20
        Med  priority queue depth                        20
        Low  priority queue depth                        20
        Ctrl priority queue depth                        20
        Max number of pkts enqued in high priority queue  0
        Max number of pkts in enqued med priority queue   0
        Max number of pkts in enqued low priority queue   20
        Max number of pkts in enqued ctrl priority queue  1
        Current number of pkts in high priority queue     0
        Current number of pkts in med  priority queue     0
        Current number of pkts in low  priority queue     3
        Current number of pkts in ctrl priority queue     0
        Frames dropped due to ctl queue depth exceeded    0
        Frames dropped due to high queue depth exceeded   0
        Frames dropped due to med queue depth exceeded    0
        Frames dropped due to low queue depth exceeded    78
```

The significant lines in this output are highlighted. It is important to note that:

- There were times in the past where the Low priority queue reached its capacity.
- The current utilization of the low priority queue us fairly low, and that is the case over a long period of time (suppose the command was continuously executed for over an hour)

- Few frames are dropped, and the number does not increase rapidly during the observation period.

This is an illustration of a case where queuing occurs very sporadically, and can generally be alleviated by increasing the queue size. Queue sizes should not be increased dramatically from their default values (20). If the queue size is tripled, and frames are still dropped, this will probably be an inadequate method for alleviating the problem, and other QoS methods should be considered, or the CIR of the PVC should be increased.

Queue depth manipulation is done through the service profile. The following two lines define a service profile with a low priority queue depth of 30 frames, and apply it to a PVC.

```
frame-relay define service DEEP_QUEUE low-priority-queue-depth 30
frame-relay apply service DEEP_QUEUE ports hs.4.1.16
```

Traffic Prioritization

Traffic classification is performed when traffic is entering the RS. Based on that classification, traffic is marked as control/high/medium/low priority. If no classification is done, all traffic will by default be placed in the low queue (an exception to that rule is traffic from routing protocols and bridged PDUs, which are always placed in the control queue).

If a queue is highly utilized AND the traffic in it is a mixture of high-priority (real-time or interactive traffic), with low priority (bulk FTP transfers) traffic, then traffic prioritization may help ensued that time-sensitive traffic goes first.

An example of traffic prioritization is provided in section 0. Note that prioritization is done on the traffic as it enters the RS. Therefore, if we want to handle traffic exiting a frame relay port with priorities, we must apply the *qos set IP …* command NOT on the frame relay interface, but on the other interfaces, where traffic enters the RS.

RED (Random Early Detection)

Random Early Detection (6) provides for a mechanism to maintain the utilization of a circuit slightly below its capacity. It monitors traffic volume for reaching key threshold values, and when that happens, the algorithm starts to drop packets from random flows. If the flows are part of TCP connections, the participating TCP hosts will respond to such dropped packets with decreasing the size of the TCP windows and slowing down.

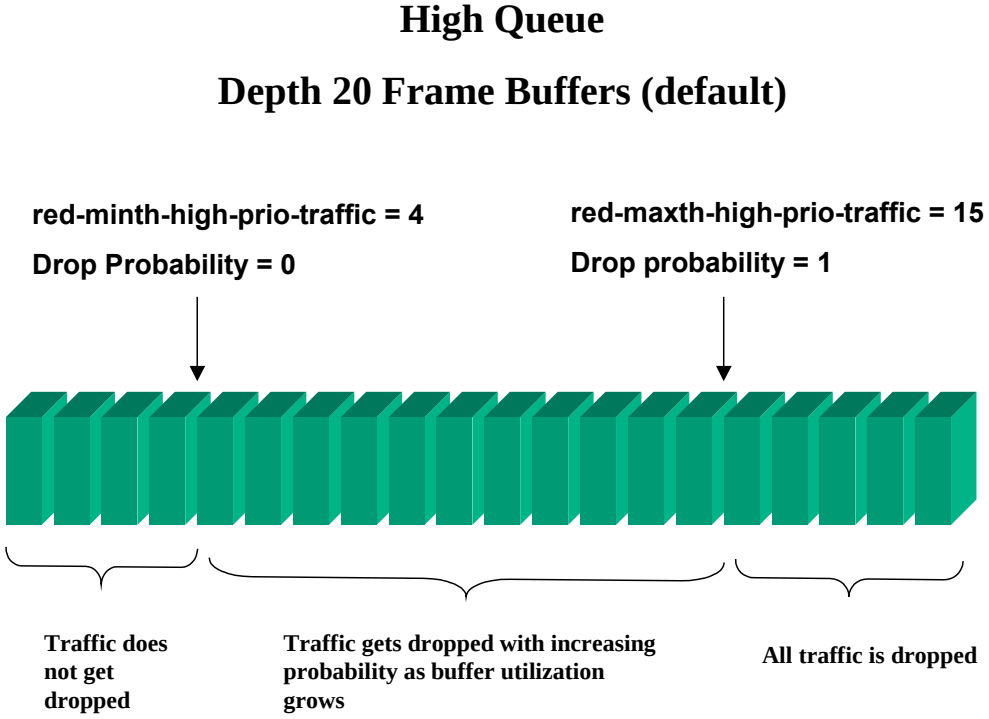


Figure 24: RED thresholds

Since RED slows down all TCP flows contributing to a given circuit congestion, and it does it in a staggered way (it does not slow down all flows at the same time) it is particularly effective when applied to high-bandwidth circuits with many TCP flows. On the other hand, it is completely ineffective when the circuit carries many UDP flows (or any other IP traffic that is not TCP-based.

There are two important thresholds for RED on each queue. Those are defined in a service profile, and are applied a per-queue basis for each PVC the service profile is applied to.

The main logic of RED is as follows: If the queue is empty or very lightly utilized, do not drop any packets. If the queue is moderately utilized, begin dropping random packets and increase the probability of dropping the packets as the queue utilization grows. If the queue utilization reaches predefined maximum threshold, begin dropping all packets.

The ***red-minth*** and ***red-maxth*** values define the level of queue utilization where traffic will be dropped. Normally***, red-minth*** should be set to 1/3 of ***red-maxth***, and ***red-maxth*** should be set to a few buffers less than the absolute depth of the queue. ***Red-maxth*** represents the **average** utilization of the buffers, and it should NOT be set equal to the queue depth – there should be some space for short bursts.

Putting it all Together – Service Profile

All frame relay QoS parameters discussed so far are configured as part of the **frame relay service profile**.

```
HUB-A(config)# frame-relay define service SERVICE_NAME ?
 high-priority-queue-dept - Set the high priority queue depth on frame relay
                            ports(Default: 20, Recommended Range: 5 - 100)
 med-priority-queue-depth - Set the medium priority queue depth on frame relay
                            ports(Default: 20, Recommended Range: 5 - 100)
 low-priority-queue-depth - Set the low priority queue depth on frame relay
                            ports(Default: 20, Recommended Range: 5 - 100)
 de-mark                    - Enable/Disable DE marking for Best traffic (Default:
                            Disabled)
 red                        - Enable/disable RED (Default: disabled)
 red-maxTh-high-prio-traf - Set the red maximum threshold for high priority
                            traffic(Default: 12. In general, this parameter
                            should be set to be less than or equal to the queue
                            depth)
 red-minTh-high-prio-traf - Set the red minimum threshold for high priority
                            traffic(Default: 4. In general, this parameter
                            should be set to be 1/3 of maximum threshold)
 red-maxTh-med-prio-traff - Set the red maximum threshold for medium priority
                            traffic(Default: 12. In general, this parameter
                            should be set to be less than or equal to the queue
                            depth)
 red-minTh-med-prio-traff - Set the red minimum threshold for medium priority
                            traffic(Default: 4. In general, this parameter
                            should be set to be 1/3 of maximum threshold)
 red-maxTh-low-prio-traff - Set the red maximum threshold for low priority
                            traffic(Default: 12. In general, this parameter
                            should be set to be less than or equal to the queue
                            depth)
 red-minTh-low-prio-traff - Set the red minimum threshold for low priority
                            traffic(Default: 4. In general, this parameter
                            should be set to be 1/3 of maximum threshold)
 cir                      - Set committed information rate for frame relay
                            VCs(bps)
```

Not all parameters of the service profile need to be defined on the same command line. Parameters are additive as long as they refer to the same service profile. As an example, the following three lines

```
frame-relay define service SHAPE cir 30 bc 30 be 60
frame-relay define service SHAPE red on
frame-relay define service SHAPE becn-adaptive-shaping 20
```

are equivalent to

```
frame-relay define service SHAPE cir 30 bc 30 be 60 red on becn-
adaptive-shaping 20
```

Once a service profile is defined, it can be applied to one or more PVCs:

```
frame-relay apply service SHAPE ports hs.4.1.16
```

A PVC can only have one service profile applied to it.

A Comprehensive Example

We will now illustrate some of the QoS mechanisms previously discussed, as well as other issues, through this example of a hypothetical ISP. We will first list the requirements and constraints for the network, then we will discuss the design decisions we made to meet the requirements, and finally show a couple of config files to illustrate the implementation of the design decision we made.

Of course, we do all this for illustrative purposes, so some of the design decisions will be biased toward frame relay, and some of the configuration (especially routing protocols) is cursory and needs more work for a real implementation. Refer to Figure 25 for the following discussion.

Requirements

1. ISP Framelover will have two upstream providers, through which full redundancy needs to be achieved.
2. ISP Framelover is a **stub AS**, i.e. there will be no downstream BGP peers and no transit traffic.
3. There will be 4 POP (Points of Presence), where customer connectivity will be aggregated via frame relay.
4. Connectivity between the 4 POPs will also be frame relay.
5. Frame relay services will be purchased from an IXC, and the same IXC will be used for the inter-POP links as well as customer connectivity.
6. Two of the POPs (POP1 and POP2) will also house some content services, among which two Caching servers. Traffic from the Caching servers must be handled with priority throughout Framelover's network.
7. POP4 is extremely constrained in terms of space and facilities. Only one router can be installed there, and only one DS3 circuit can be provisioned.
8. Traffic volume for one customer should not be allowed to interfere with traffic from other customers. Backbone links should not be allowed to reach unmanageable congestion, which would cause tail dropping.
9. ISP Framelover owns the IP address range of 10.0.0.0/8 for its internal use and for distribution of addresses to its customers.

Design Decisions

General:

- The 4 POPs will be interconnected with frame relay PVCs. OSPF will be the internal routing protocol. The backbone PVCs will be in OSPF area 0 and each POP will be its own area.
- Core1 through Core4 will be the core routers, also serving as OSPF Area Border Routers (ABRs). Each POP will have as many Edge aggregation routers as necessary to accommodate all customers terminating there.

Specifically, on the requirements:

1. ISP Framelover will have two upstream providers, through which full redundancy needs to be achieved.

Upstreams will be frame relay-connected. There will be PVCs from both Core1 and Core2 to both upstreams. Core1 and Core2 will accept full internet routing tables so statistical load balancing of outbound traffic will occur.

2. ISP Framelover is a **stub AS**, i.e. there will be no downstream BGP peers and no transit traffic

Only Core1 and Core2 need to share the full Internet routing table. There is no need to propagate it into OSPF. Instead a default route will be injected from both Core1 and Core2 into OSPF. A 10.0.0.0/8 summary route will be advertised from both Core1 and Core2 into both Upstreams.

3. There will be 4 POP (Points of Presence), where customer connectivity will be aggregated via frame relay.

POPs will generally be implemented with a common fast Ethernet network for connectivity between the aggregation and core routers. With the exception of POP4, no customer connectivity will terminate in a core router.

4. Connectivity between the 4 POPs will also be frame relay.

Generally, different frame relay circuits will be used for different purposes. For example, Core1 and Core2 will have one frame relay port (hs.4.1) for PVCs within Framelover, and another port (hs.4.2) for connectivity to the upstreams. Again, an exception will be made at POP4 to accommodate the facilities constraints.

5. Frame relay services will be purchased from an IXC, and the same IXC will be used for the inter-POP links as well as customer connectivity.

This is not really a constraint, but it is nice to know when we come to requirement 7.

6. Two of the POPs (POP1 and POP2) will also house some content services, among which two Caching servers. Traffic from the Caching servers must be handled with priority throughout Framelover's network.

We will use the qos classification capabilities of the RS to place traffic originating from those two servers in the medium queue. We will also enable Weighted Fair Queuing so that the remaining traffic (in the low queue) is not starved.

7. POP4 is extremely constrained in terms of space and facilities. Only one router can be installed there, and only one DS3 circuit can be provisioned.

Since we cannot build out the usual POP infrastructure here, we will have to combine all functionality in one router (Core4). This should normally be avoided, but here it is necessary. The single frame relay circuit, connected to hs.4.1 will contain both the PVCs for customers, as well as for the core. Special attention will be paid to the subscription of this circuit, to ensure that customer traffic (hs.4.1.(20-200)) does not overrun backbone capacity. To that end, we will stipulate that the customer VLAN will be subscribed to the sum of the customer access ports. Since the sum of the backbone CIRs requires 30 Mbps, we want to ensure that customer traffic will never peak above 15 Mbps (45 Mbps – 30 Mbps). Therefore we will conservatively subscribe the "customer" 15 Mbps to the sum of the customer access circuits speed. If the customers connect with 256 Kbps circuits, this will yield a subscription number of 58 customer sites.

8. Traffic volume for one customer should not be allowed to interfere with traffic from other customers. Backbone links should not be allowed to reach unmanageable congestion which would cause tail dropping.

Based on the expected traffic volumes, it is decided that the backbone should be able to handle 20 Mbps of traffic. Therefore the PVCs between Core1/Core4 and Core2/Core3 are defined as 20 Mbps. Additional PVCs are provisioned between Core1/Core3 and Core2/Core4, for redundancy purposes only, and therefore they are provisioned at 10 Mbps. Performance degradation is considered acceptable in cases that force traffic to go through the backup PVC. In normal situation however, traffic should always choose the 20 Mbps PVC. This will be done with proper OSPF costing of the appropriate interfaces. Also, in order to be able to cost the two PVCs coming out of a core router differently, the PVCs in the core will be defined as frame relay subinterfaces.

To prevent backbone links from becoming congested, RED will be enabled on them.

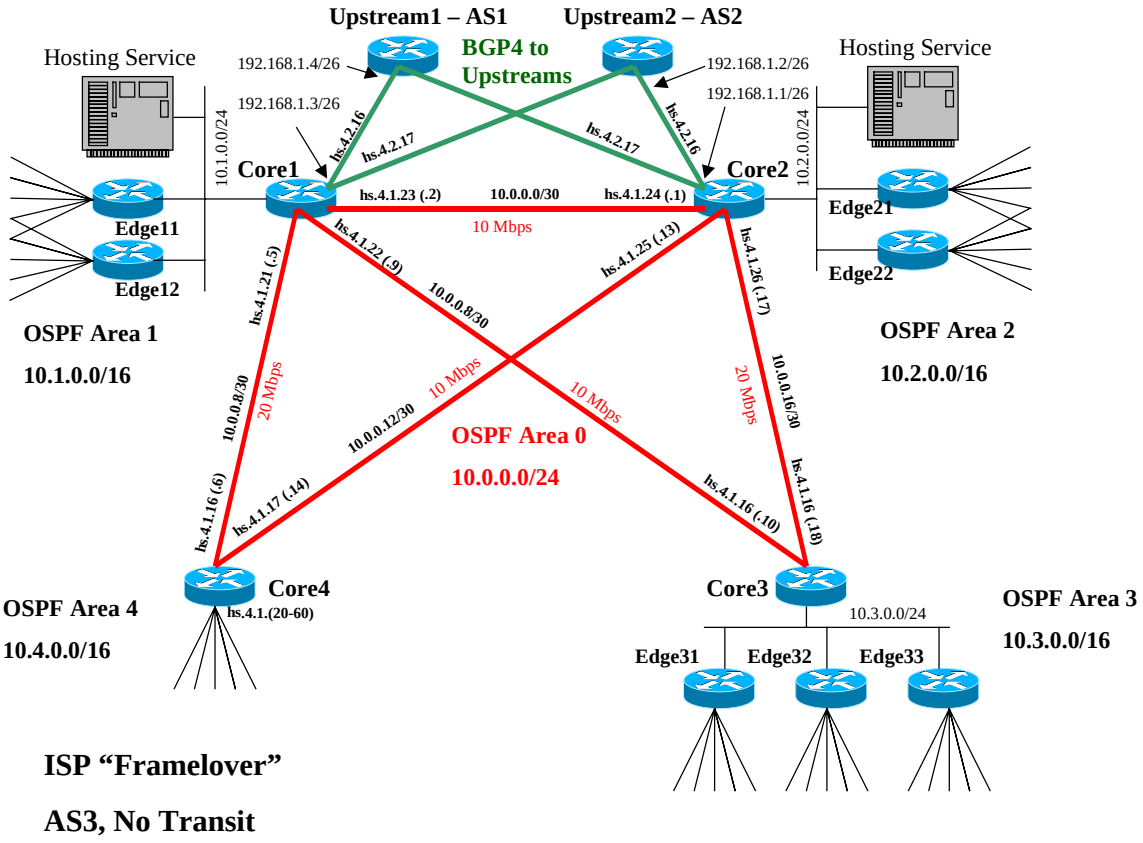


Figure 25: Frame Relay implementation of ISP Framelover

9. ISP Framelover owns the IP address range of 10.0.0.0/8 for its internal use and for distribution of addresses to its customers. Each area will be assigned a Class “B”-sized CIDR block from the 10.0.0.0/8 range. Summarization will be done at area borders.

Sample Configuration Files

Following are the configuration files for Core2 and Core4, which illustrate most of the QoS concepts discussed so far, as well as other frame relay issues covered in this document.

```
Core2# system show active-config
Running system configuration:
     !
     ! Last modified from Console on 2000-02-29 21:50:21
     !
 1 : port set hs.4.1 wan-encapsulation frame-relay speed 51840000
 2 : port set hs.4.2 wan-encapsulation frame-relay speed 51840000
     !
 3 : frame-relay set lmi type ansi617d-1994 ports hs.4.1 state enable
 4 : frame-relay set lmi type ansi617d-1994 ports hs.4.2 state enable
 5 : frame-relay create vc port hs.4.2.(16,17)
 6 : frame-relay create vc port hs.4.1.(24-26)
```

```
 7 : frame-relay define service Upstreams red on
 8 : frame-relay define service Core red on
 9 : frame-relay apply service Core ports hs.4.1.(24-26)
 10 : frame-relay apply service Upstreams ports hs.4.2.(16-17)
     !
 11 : vlan create To_Upstreams ip id 10
12 : vlan add ports hs.4.2.(16,17) to To_Upstreams
     !
13 : interface create ip LAN address-netmask 10.2.0.1/24 port et.1.1
14 : interface create ip To_Core1 address-netmask 10.0.0.1/30 port hs.4.1.24 type point-
to-point
15 : interface create ip To_Core4 address-netmask 10.0.0.13/30 port hs.4.1.25 type point-
to-point
16 : interface create ip To_Core3 address-netmask 10.0.0.17/30 port hs.4.1.26 type point-
to-point
17 : interface create ip BGP_Peers address-netmask 192.168.1.1/26 vlan To_Upstreams
18 : interface add ip lo0 address-netmask 10.2.254.2/32
     !
19 : qos set ip Cache_Server1 medium 10.1.0.20/32 any
20 : qos set ip Cache_Server2 medium 10.2.0.20/32 any
21 : qos set queueing-policy weighted-fair port hs.4.1
22 : qos set weighted-fair control 20 high 10 medium 50 low 20 port hs.4.1
23 : qos set queueing-policy weighted-fair port hs.4.2
24 : qos set weighted-fair control 20 high 10 medium 50 low 20 port hs.4.2

     !
25 : ip-router global set router-id 10.2.254.2
26 : ip-router global set autonomous-system 3
     !
27 : ip add route default gateway 192.168.1.2  preference 60 no-install
28 : ip add route default gateway 192.168.1.4 preference 70 no-install
29 : ip add route 10.0.0.0/8 gateway 10.2.254.2 no-install
     !
30 : ip-router policy redistribute from-proto static to-proto ospf network default
31 : ip-router policy redistribute from-proto static to-proto bgp target-as 1 network
10.0.0.0/8
32 : ip-router policy redistribute from-proto static to-proto bgp target-as 2 network
10.0.0.0/8
33 : ip-router policy redistribute from-proto bgp source-as 1 to-proto bgp target-as 3
34 : ip-router policy redistribute from-proto bgp source-as 2 to-proto bgp target-as 3

     !
35 : ospf create area backbone
36 : ospf create area 2.2.2.2
37 : ospf add interface To_Core1 to-area backbone
38 : ospf add interface To_Core3 to-area backbone
39 : ospf add interface To_Core4 to-area backbone
40 : ospf add interface LAN to-area 2.2.2.2
41 : ospf add summary-range 10.2.0.0/16 to-area 2.2.2.2
42 : ospf add stub-host 10.2.254.2 to-area 2.2.2.2 cost 1
43 : ospf set interface To_Core1 cost 1000
44 : ospf set interface To_Core3 cost 500
45 : ospf set interface To_Core4 cost 1000
46 : ospf set interface LAN cost 100
47 : ospf start
     !
48 : bgp create peer-group AS2 type external autonomous-system 2
49 : bgp create peer-group AS1 type external autonomous-system 1
50 : bgp create peer-group AS3 type routing autonomous-system 3 proto any
51 : bgp add peer-host 10.1.254.1 group AS3
52 : bgp add peer-host 192.168.1.2 group AS2
53 : bgp add peer-host 192.168.1.4 group AS1
54 : bgp set peer-group AS3 local-address 10.2.254.2
55 : bgp start
     !
56 : system set name Core2
Core2#
```

Comments on Core2 config:

Lines 1-6 are the basic frame relay configuration for the two ports.

Lines 7-10 define and apply QoS parameters for the core and upstream PVCs. At this point only RED is enabled to prevent sever congestion on those interfaces. Although currently with the same content, the service profiles are kept separate in case we need to do different modifications to each of them in the future.

Lines 11-12 create a VLAN for the two PVCs that will go to the two upstreams. Notice we are not creating a VLAN for the core PVCs, as those are defined as subinterfaces (because we want to apply different OSPF costing to each of them). In reality, we will probably want to define the upstream PVCs as subinterfaces as well, but we wanted to demonstrate we can do otherwise here.

Lines 13-18 define IP interfaces – one for the POP LAN, 3 subinterfaces for the core PVCs, one interface for the upstreams to talk to, and a secondary loopback.

Lines 19 and 20 place traffic originating from the two caching servers in the medium priority queue. We will have to do the same on all routers in the network.

Lines 21-24 enable WFQ on the two frame relay ports. Since we placed our priority traffic in the medium queue, we don't want to starve the low queue where all the remaining traffic will go by default.

Lines 27-29 place static routes in the RIB, but not in the FIB. We will need those for redistribution into routing protocols, but this router does not need them, in fact, it is not a good idea to have them. Therefore we prevent them from being placed in the FIB.

Lines 30-34 export the 10.0.0.0/8 summary, representing the whole AS, to the two upstreams, inject a default route into OSPF for all internal routers to use, and send all routing information from the upstreams to the IBGP neighbor (Core1).

Lines 35-40 create OSPF areas and place appropriate interfaces into them.

Line 41 summarizes the address range for Area 2 and injects the summary into Area 0

Line 42 allows the host route, associated with the router ID, to be included in the area summary (instead of being flooded as an OSPF external route, which cannot be summarized).

Lines 43-46 assign appropriate costing to the different interfaces so traffic can actually choose the fasted PVC. The following commonly used formula is applied when figuring the OSPF costs:

OSPF cost = 10 Gbps / CIR

This results in 10 Mbps PVCs being assigned a cost of 1000, 20 Mbps PVC – 500, 100 Mbps Ethernet – 100.

Lines 48-55 are the BGP configuration, which is only for illustration purposes here. A lot more needs to be done to ensure this As behaves as a good NET-izen.

```
Core4# system show active-config
Running system configuration:
     !
     ! Last modified from Console on 2000-02-29 19:20:37
     !
 1 : port set hs.4.1 wan-encapsulation frame-relay speed 51840000
     !
 2 : frame-relay set lmi type ansi617d-1994 state enable ports hs.4.1
 3 : frame-relay create vc port hs.4.1.(16,17)
 4 : frame-relay create vc port hs.4.1.(20-25)
 5 : frame-relay define service Police_Fast_Customers cir 128000 bc 128000 be 50000
                                    becn-adaptive-shaping 40 de-mark on
 6 : frame-relay define service Police_Slow_Customers cir 56000 bc 56000 be 20000
                                    becn-adaptive-shaping 15 de-mark on
 7 : frame-relay define service Core_Connections red on
 8 : frame-relay apply service Police_Fast_Customers ports hs.4.1.20-29
 9 : frame-relay apply service Police_Slow_Customers ports hs.4.1.30-60
10 : frame-relay apply service Core_Connections ports hs.4.1.(16,17)
     !
11 : vlan create Customers ip id 10
12 : vlan add ports hs.4.1.(20-60) to Customers
     !
13 : interface create ip To_Core1 address-netmask 10.0.0.6/30 port hs.4.1.16 type point-
to-point
14 : interface create ip To_Core2 address-netmask 10.0.0.14/30 port hs.4.1.17 type point-
to-point
15 : interface create ip To_Customers address-netmask 10.4.0.1/24 vlan Customers
16 : interface add ip lo0 address-netmask 10.0.254.4/32
     !
17 : qos set ip Cache_Server1 medium 10.1.0.20/32 any
18 : qos set ip Cache_Server2 medium 10.2.0.20/32 any
19 : qos set queueing-policy weighted-fair port hs.4.1
20 : qos set weighted-fair control 20 high 10 medium 50 low 20 port hs.4.1
     !
21 : ip-router global set router-id 10.0.254.4
     !
22 : ospf create area backbone
23 : ospf create area 4.4.4.4
24 : ospf add interface To-Core1 to-area backbone
25 : ospf add interface To_Core2 to-area backbone
26 : ospf add interface To_Customers to-area 4.4.4.4 type point-to-multipoint
27 : ospf add stub-host 10.0.254.4 to-area backbone cost 1
28 : ospf add summary-range 10.4.0.0/16 to-area 4.4.4.4
29 : ospf add summary-range 10.0.0.0/16 to-area backbone
30 : ospf set interface To_Core1 cost 500
31 : ospf set interface To_Core2 cost 1000
32 : ospf start
     !
33 : system set name Core4
Core4#
```

Following are some comments on Core4 config features, absent from Core2.

The main feature of this config is that it creates two subinterfaces and one VLAN with 40 PVCs on the same frame relay port. Traffic shaping is then performed on the VLAN with the 40 PVCs to which customers are connected, to ensure that bursts from one customer don't affect another.

Customers are segregated between fast ones (whose local access circuit is 256 Kbps and their PVCs are 128 Kbps, and slow ones, whose circuit is 128 Kbps with PVCs at 56 Kbps.

Lines 5-10 ensure proper traffic shaping (policing) for the two customer categories, and enable RED on the backbone links.

Appendix

Broadcast Handling in a Hub-and-Spoke Environment

Here we describe the mechanism that allows the WAN card to emulate broadcast on NBMA media. Normally, in a frame relay environment, the last two bytes of the MAC address of a frame represent the destination DLCI for that frame. As an example, on Figure 26, if router C wants to send a frame to router A, it will forward it on DLCI 117. As a result, the destination

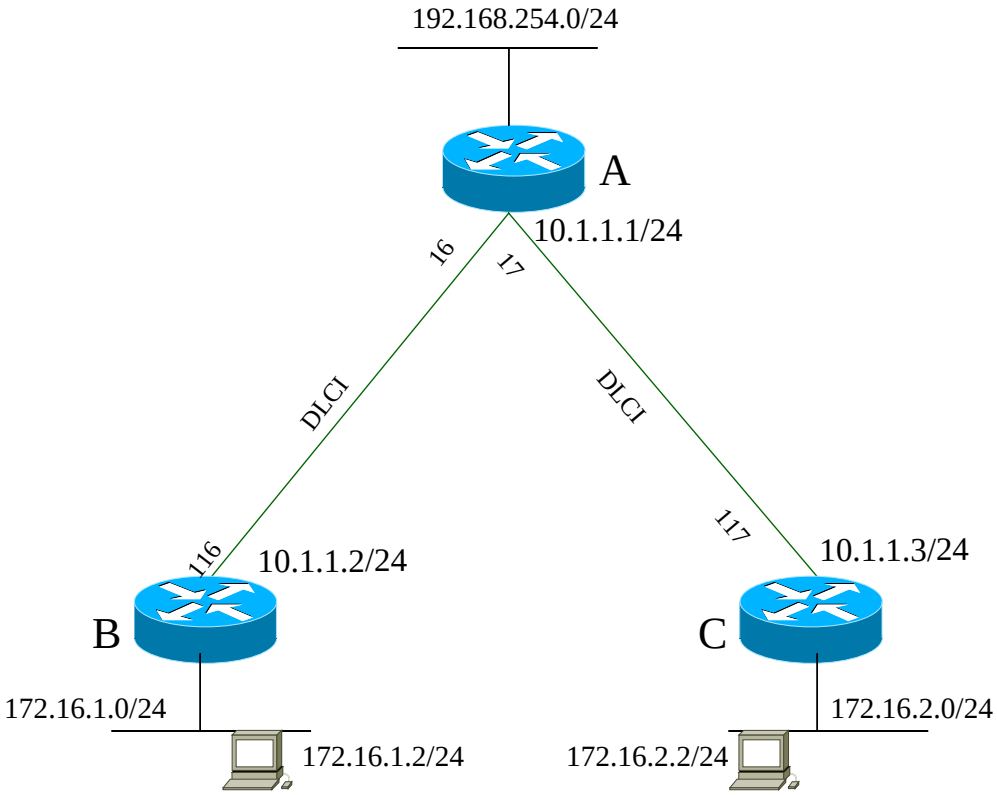


Figure 26: Broadcast in Frame Relay (for RS)

MAC address of the frame will contain xx:xx:xx:xx:00:75. Similarly, a frame from router B to router A will be addressed to xx:xx:xx:xx:00:74. Following this logic, there would be no way for router B to address a frame to router C, as there is no direct PVC between the two. This is a normal restriction in NBMA media, but the RS has a mechanism to overcome it.

For each IP interface defined on the WAN card, there is a corresponding MAC address. In addition, there are the MAC addresses associated with PVCs, as described above. When the WAN card is sending a frame to another router connected directly with a PVC, it composes the outgoing frame with the MAC

address derived from the PVC. Otherwise (in the absence of a direct PVC), the MAC address of the destination WAN card is used.

Let's go through a step-by-step example of a PING from Router C to Router B in order to illustrate this mechanism. First, when the network on Figure 26 comes up, the three routers will have the following information in their ARP caches.

Router	WAN Interface	ARP Cache Content
A	se.4.1.(16-17) → 10.1.1.1→ 00:E0:63:34:BF:8E	10.1.1.2 → 02:E0:63:40:00:10 → se.4.1.16 10.1.1.3 → 02:E0:63:40:00:11 → se.4.1.17
B	se.4.116 → 10.1.1.2 → 00:E0:63:34:1F:8E	10.1.1.1 → 02:E0:63:40:00:74 → se.4.1.116
C	se.4.117 → 10.1.1.3 → 00:E0:63:0B:FC:DA	10.1.1.1 → 02:E0:63:40:00:75 → se.4.1.117

Table 3: State of routers before traffic between B and C

Before any traffic is sent between routers B and C, the ARP caches of all routers are comprised solely of the entries derived from the PVCs (those came from InARP – see section 0). Next we issue the following command from the console of Router C:

Router_c# ping 10.1.1.2

The following sequence of events occurs:

1. Since Router C does not have an ARP cache entry for 10.1.1.2, it sends a broadcast ARP request for that address. The source MAC address for the broadcast is Router C's WAN card MAC address, 00:E0:63:0B:FC:DA.
2. Router A, acting as a transparent bridge for the VLAN composed of PVCs 16 and 17, receives this broadcast on DLCI 117, and forwards it to DLCI 16. The WAN card of Router A also records an association between DLCI 17 and Router C's MAC address, 00:E0:63:0B:FC:DA.
3. Router B receives this broadcast and replies to it with its own WAN Interface MAC address, 00:E0:63:34:1F:8E. It installs in its ARP cache this entry: 10.1.1.3 → 00:E0:63:0B:FC:DA → se.4.1. Its WAN card also records an association between DLCI 116 and Router C's MAC address, 00:E0:63:0B:FC:DA.
4. Router A bridges the ARP reply packet back to DLCI 17, while recording in its WAN card the association between router B's WAN MAC address, and DLCI 16.
5. Lastly, Router C receives the ARP reply message and installs an ARP cache entry of 10.1.1.2 → 00:E0:63:34:1F:8E → se.4.1. Its WAN card also remembers the fact that 00:E0:63:34:1F:8E is reachable through DLCI 117.
6. A PING packet is formed at router C with a source MAC address of 00:E0:63:0B:FC:DA and a destination MAC address of 00:E0:63:34:1F:8E.

Luckily everyone along the way now knows how to forward that frame based on the following ARP entries:

Router	WAN Interface	ARP Cache Content
A	se.4.1.(16-17) → 10.1.1.1→ 00:E0:63:34:BF:8E	10.1.1.2 → 02:E0:63:40:00:10 → se.4.1.16 10.1.1.3 → 02:E0:63:40:00:11 → se.4.1.17
B	se.4.116 → 10.1.1.2 → 00:E0:63:34:1F:8E	10.1.1.1 → 02:E0:63:40:00:74 → se.4.1.116 10.1.1.3 → 00:E0:63:0B:FC:DA → se.4.1
C	se.4.117 → 10.1.1.3 → 00:E0:63:0B:FC:DA	10.1.1.1 → 02:E0:63:40:00:75 → se.4.1.117 10.1.1.2 → 00:E0:63:34:1F:8E → se.4.1

Table 4: State of routers after the ARP exchange between B and C

Note that the additional ARP cache entries built in routers B and C point to the WAN card (port) itself, rather than a particular PVC. This is because the WAN card maintains the association between these remote MAC addresses and the PVCs they are reachable through.

References

1. [OSPF1] John T. Moy, OSPF, Anatomy of an Internet Routing Protocol Addison Wesley 1998, ISBN 0-201-63472-4
2. [RIP1] RFC 1058, Routing Information Protocol
3. [RIP2] RFC 1388, RIP Version 2
4. [OSPF2] RFC 2328, OSPF Version 2
5. [INARP] RFC 2390, Inverse Address Resolution Protocol
6. [RED] Floyd, S., and Jacobson, V., Random Early Detection gateways for Congestion Avoidance http://www-nrg.ee.lbl.gov/papers/red/red.html
7. [FR1] Walter Goralski, Frame Relay for High-Speed Networks John Wiley & Sons Inc. 1999, ISBN 0-471-31274-6
8. [FR2] Uyless Black, Frame Relay Networks: Implementations and Specifications, Second Edition, McGraw-Hill, Inc. 1995, ISBN 0-07-005590-4
9. [FR3] RFC 2427, Multiprotocol Interconnect Over Frame Relay
10. [FR4] RFC 1586, Guidelines for Running OSPF Over Frame Relay Networks
11. [IP1] Radia Perlman, Interconnections, Second Edition, Addison Wesley Longman, Inc. 2000, ISBN 0-201-63448

[IP2] Alvaro Retana et al., Advanced IP Network Design Cisco Press 1999, ISBN 1-57870-097-3

Riverstone Networks, Inc.
5200 Great America Pkwy, Santa Clara, CA 95054 USA
(877) 778-9595, (408) 878-6500, www.riverstonenet.com

Version 1.0, 18 Dec 2001