# Post 6

Thursday, January 24, 2019 11:24 PM

# FRAME RELAY P2

**Access Rate/Committed Info Rate**

SP's build FR networks using very powerful switches: Devices only see switch int of SP

- From customer POV: FR single int config w/1/more PVCs
- Customers buy FR services from SP

| | |
|---|---|
| **Access rate** | Capacity of local loop: Refers to port speed<br>Customer POV: SP provides serial connect/access link to FR network over leased line<br>• Rate at which access circuits join FR network<br>• May be 56 kb/s/T1 [1.544 Mb/s]/Fractional T1 [multiple 56 kb/s/64 kb/s]<br>• Clocked on FR switch: Not possible to send data higher than it |
| **CIR** | **Committed Info Rate:** Capacity through local loop guaranteed by provider<br>Customers negotiate CIRs w/SP for each PVC:<br>• Amt of data network receives from access circuit<br>• SP guarantees customer can send data at CIR<br>• All frames received at/below CIR accepted |

CIR specifies max avg data rate network undertakes to deliver under normal conditions

- When subscribing to FR service: Local access rate is specified
- Typically, provider asks customer to specify CIR for each DLCI

If customer sends info faster than CIR on a DLCI: Network marks some frames w/DE: Discard Eligibility bit

- Network does its best to deliver all packets: Discards DE packets 1st if congestion

**Customer pays for 3 FR cost components as follows:**

| | |
|---|---|
| **Access rate** | Cost of access line from DTE to DCE (customer to SP): Line charged based on port speed neg/install |
| **PVC** | Cost based on PVCs: After PVC established: Addl cost to increase CIR can be done |
| **CIR** | Customers normally choose CIR lower than access rate: Allows take advantage bursts |

**Oversubscription**

- SP's sell more capacity than they have on assumption not everyone will demand their capacity all the time

**Bursting:** Any network capacity being unused is made avail/shared w/customers at no extra charge

- Allows customers to burst over CIR as bonus

**Various terms used to describe burst rates:**

- **Bc: Committed Burst Size**
- **Be: Excess Burst Size**

**Bc is a negotiated rate above CIR that customer can use to transmit for short burst**

- Represents max allowed traffic under normal working conditions
- Allows traffic to burst to higher speeds as avail bw permits
- cannot exceed access rate of link
- device can burst up to Bc/still expect the data to get through
- If long bursts persist: Higher CIR should be purchased

**Be describes bw avail above CIR up to access rate of link**

- Unlike Bc: Not negotiated
- Frames may be transmitted at this lvl but are most likely dropped

**FR Flow Control**

**Congestion-notification mech:**

- **FECN: Fwd Explicit Congestion Notification**
- **BECN: Backward Explicit Congestion Notification**

**FECN/BECN each controlled by single bit contained in frame header**

- Let rtr know there is congestion: Should stop trans until condition reversed
- When DCE sets BECN bit to 1: Notifies devices in dir of source (upstream) there is congestion on

network
- When DCE sets FECN bit to 1: Notifies devices in dir of dest (downstream) there is congestion on network

**Frame header:** Also contains DE bit: ID's less impt traffic that can be dropped during periods of congestion
- DTE devices can set value of DE bit to 1: Indicates frame has lower imptance than others
- When network congested: DCE devices discard frames w/DE bit set to 1 before discarding those that don't
- Reduces likelihood of critical data being dropped during periods of congestion

**Congestion?** SP's FR switch applies following logic rules to each incoming frame based on whether CIR exceeded:
- If incoming frame doesn't exceed Bc: Frame passed
- If incoming frame exceeds Bc: Marked DE
- If incoming frame exceeds Bc/Be: Discarded

**Basic FR Config**

| **Required Tasks** | **Optional** |
|---|---|
| Enable FR encapsulation on int | Config LMI |
| Config dynamic/static addr mapping | Config FR SVC's |
| | Config FR traffic shaping |
| | Customize FR for network |
| | Monitor/maintain FR connections |

```
R1(config)# int s0/0/1
R1(config-if)# bandwidth 64
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:café:1::1/64
R1(config-if)#  ipv6 address fe80::1 link-local
R1(config-if)# encapsulation frame-relay
```

```
R1(config)# int s0/0/1
R1(config-if)# bandwidth 64
R1(config-if)# ip address 10.1.1.2 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:café:1::2/64
R1(config-if)#  ipv6 address fe80::2 link-local
R1(config-if)# encapsulation frame-relay
```

**Steps:**
1. Set IP on int: IOS: FR most commonly supported on sync serial ints
2. Config Encapsulation: **encapsulation frame-relay [cisco | ietf]**
   1. Enables FR encapsulation/allows processing on supported int
   2. Cisco: Default: 4-byte header/2 bytes to ID DLCI/2 bytes to ID packet type
   3. ietf encapsulation complies w/RFC 1490/RFC 2427: Non-Cisco rtr
3. Set BW
4. Set LMI Type (optional): Cisco/ANSI Annex D/Q933-A Annex A

**Config Static FR Map**

```
R1(config-if)# frame-relay map protocol protocol-addr dlci [broadcast]
```

| | |
|---|---|
| **Protocol** | Defines supported protocol/bridging/logical link control:<br>• ip/ipv6/AppleTalk/decnet/dlsw/ipx/llc2/rsrb/vines/xns |
| **Protocol-addr** | Defines network layer addr of dest rtr int |
| **Dlci** | Local DLCI used to connect to remote protocol addr |
| **Broadcast** | Optional: Allows broadcasts/multicasts over VC<br>Permits use of dynamic r-protocols over VC |

```
R1(config)# int s0/0/1
R1(config-if)# bandwidth 64
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:café:1::1/64
R1(config-if)# ipv6 address fe80::1 link-local
```

```
R1(config-if)# encapsulation frame-relay
R1(config-if)# frame-relay map ipv6 2001:db8:café:1::2 102
R1(config-if)# frame-relay map ipv6 fe80::2 102 broadcast

R2(config)# int s0/0/1
R1(config-if)# bandwidth 64
R1(config-if)# ip address 10.1.1.2 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:café:1::2/64
R1(config-if)# ipv6 address fe80::2 link-local
R1(config-if)# encapsulation frame-relay
R1(config-if)# frame-relay map ip 10.1.1.1 201 broadcast
R1(config-if)# frame-relay map ipv6 2001:db8:café:1::1 201
R1(config-if)# frame-relay map ipv6 fe80::1 201 broadcast
```

**Cisco rtrs**: Support all network layer protocols over FR [IPv4/6/IPX/AppleTalk]

- Address-to-DLCI mapping can be done dynamically/statically

Dynamic mapping: Performed by I.ARP: Default enabled: No addl cmd req to config dynamic mapping on int

**Static mapping:** Manual: To map bet next hop protocol addr/DLCI dest add:

**frame-relay map protocol protocol-address dlci [broadcast]**

FR/ATM/X.25: Non-broadcast multi-access NBMA networks

- NBMA allow only data transfer from 1 computer to another over VC/across switching device
- Don't support multicast/broadcast traffic: Single packet can't reach all destinations
- Reqs you to replicate packets manually to all dest

**broadcast allows IPv4 broadcasts/multicasts to be propagated to all nodes**

- Also allows IPv6 multicasts over PVC
- When enabled: Rtr converts broadcast/multicast traffic into unicast so other nodes receive updates
- Some protocols may require addl config options: RIP/EIGRP/OSPF

**Verify Static FR Map || R1# show frame-relay map**

**Reachability Issues:** Default: Most FR networks provide NBMA connectivity: Using hub-and-spoke topology bet sites

- NBMA FR topology: When single multipoint int must be used to interconnect multiple sites
- R-update reachability issues may result
- With distance vectors: May result from split horizon/multicast/broadcast replication
- With link states: Issues w/DR/BDR election

**Split Horizon:** A loop prevention mech for distance vector routing protocols [EIGRP/RIP]

- Not applicable to link-state r-protocols
- Reduces r-loops by preventing an update that is received on int from being fwded out same int
- Not an issue if only 1 PVC config on phys int

**Multicast/Broadcast Replication:**

- Due to split horizon: When rtr supports multipoint connections over single int:
  - Must replicate broadcast/multicast packets
- In case of r-updates: Must be replicated/sent on each PVC to remote rtrs
- These packets consume BW/cause latency vars in usr traffic
- Amt of broadcast traffic/# of VCs terminating at each rtr should be evaluated during design phase of FR
- Overhead traffic [r-updates] Can affect delivery of critical usr data
  - Especially when delivery path contains low-BW [56 kb/s] links

**Neighbor Discovery: DR/BDR**

- Link-state r-protocols [OSPF] doesn't use split horizon rule for loop prevention
- Reachability issues can arise with DR/BDR
- OSPF over NBMA networks works in non-broadcast network mode by default
- Neighbors are not auto discovered
- Neighbors can be statically config: Ensure hub rtr becomes DR
- NBMA network behaves like Ethernet: A DR needed to exchange r-info bet all rtrs on segment
- Only hub rtr can act as DR: Only rtr that has PVCs w/all other rtrs

**Solving Reachability Issues**

| | |
|---|---|
| **Disable split horizon** | Method for solving reachability issues produced by split horizon<br>• Disabling split horizon increases chances of loops<br>• Only IP allows ability to disable split horizon: IPX/AppleTalk don't |
| | |

| | |
|---|---|
| **Full meshed topology** | Method is to use: Increases costs |
| **Subints** | Hub-and-spoke FR topology: Hub rtr can be config w/logically assigned ints called subints |

**FR Subints:** Can partition phys int into multiple virtual ints called subints

**Subint Logical int directly associated w/phys int**

- To enable fwding of broadcast r-updates in FR: Can config rtr w/logically assigned subints
- Each VC can be config as p-t-p
- Partially meshed: Can be divided into # of smaller/fully meshed/p-t-p networks
- Each subnetwork can be assigned unique address
- Allows each subint to act similarly to leased line
- Allows packets received on 1 subint to be sent out another: Even though packets being fwded out same phys int

**FR subints can be config in either point-to-point/multipoint:**

| | |
|---|---|
| **Point-to-point** | Single p-t-p subint establishes 1 PVC connection to phys int/subint on remote rtr<br>• Each pair of point-to-point rtrs is on its own subnet<br>• Each point-to-point subint has single DLCI<br>In p-t-p env:<br>• Each subint acting like p-t-p int: For each VC: Separate subnet<br>• Routing update traffic not subject to split horizon rule |
| **Multipoint** | Single m/p subint establishes multiple PVC connects to multiple phys ints/subint on remote rtrs<br>• All participating ints are in same subnet<br>• Subint acts like NBMA FR int<br>• Routing update traffic subject to split horizon rule<br>• All multipoint VCs belong to same subnet |

**When config subints: encapsulation frame-relay** assigned to phys int

- All other config items: Network layer addr/DLCIs assigned to subint

**Multipoint subint configs can be used to conserve addrs**

- Helpful if **VLSM: Var Length Subnet Masking** not being used
- Config may not work properly given broadcast traffic/split horizons
- P-t-p subint created to avoid these issues

**Config Point-to-Point Subints**

```
R1(config-if)# int s0/0/0 number.subint {multipoint | point-to-point}
```

| | |
|---|---|
| **Multipoint** | All rtrs exist in same subnet |
| **Point-to-Point** | So each pair of p-t-p rtrs have own subnet<br>• Normally mask of 255.255.255.252 |

**Assigning a DLCI:**

```
R1(config-subif)# frame-relay interface dlci dlci-#
```

| | |
|---|---|
| **dlci-number** | Defines local DLCI # being linked to subint<br>• Only way to link LMI-derived DLCI to a subint<br>• B/C LMI doesn't know about subints |

**Subints address limitations of FR by providing way to subdivide partially meshed FR:**

- Into # of smaller/fully meshed/point-to-point: Subnetworks
- Each subnetwork assigned own network #/appears to protocols as if it were reachable through separate int

**To create subint: Global:**

int serial | physical port # | period | subint #

**R1(config-if)# int s0/0/0.103 point-to-point** Creates a point-to-point subint for PVC 103 to R3

- If subint config as point-to-point: Local DLCI for subint must also be config to distinguish from phys int
- DLCI also req for multipoint subints for which I.ARP enabled for IPv4
- Not req for multipoint subints config w/static route maps

**R1(config-subif)# frame-relay interface-dlci** Configs local DLCI on subint

- FR SP assigns DLCI #'s: Range from 16-992: Usually have only local sig: Range depends on LMI used

**Config Point-to-Point Subints**

```
R1(config)# int s0/0/1
R1(config-if)# encapsulation frame-relay
R1(config-if)# no shut
```

```
R1(config)# int s0/0/1.102 point-to-point
R1(config-subif)# ip address 10.1.1.1 255.255.255.252
R1(config-subif)# bandwidth 64
R1(config-fr-dlci)# frame-relay interface-dlci 102
R1(config-subif)# exit
R1(config)# int s0/0/1.103 point-to-point
R1(config-subif)# ip address 10.1.1.5 255.255.255.252
R1(config-subif)# bandwidth 64
R1(config-subif)# frame-relay interface-dlci 103
```

**Config subints on phys int:**

1. Rem any network layer addr assigned to physical int: If phys int addr: Frames not received by local subints
2. Config FR encapsulation on phys int using **encapsulation frame-relay**
3. For each defined PVCs: Create logical subint: Specify port # followed by period [.] and subint #
   - Subint # should match DLCI #
4. Config IP for int/set BW
5. Config local DLCI on subint using **frame-relay interface-dlci** FR SP assigns DLCI#'s

**R1# show interfaces** Verify FR is operating correctly on int

**Verify FR: LMI**

**R1# show frame-relay lmi** Displays sample output that shows # of status msgs exchanged bet local rtr/FR switch

- Ensure counters bet status msgs being sent/received are incrementing
- Validates there is active comm bet DTE/DCE
- Look for any non-0 Invalid items: Helps isolate problem of FR comm bet carrier's switch/client rtr

**Verify FR: PVC Status**

**R1# show frame-relay pvc [interface interface] [dlci]** View PVC/traffic statistics

- Useful for # of BECN/FECN packets received by rtr
- Status can be active/inactive/del

**R1# show frame-relay pvc** Displays status of all PVCs config on rtr: Can also specify particular PVC

**R1#  clear counters** Reset statistics counters: Wait 5/10min after clearing counters before issuing sh cmds again

**Verify FR: I.ARP**

**R1# clear frame-relay inarp** Clear dynamically created FR maps created using I.ARP

**R1# frame-relay inverse-arp** Confirm whether cmd  resolved remote IPv4 to local DLCI

**When I.ARP req made: Rtr updates map table w/3 possible PVC connection states:**

| | |
|---|---|
| **ACTIVE** | Successful end-to-end circuit [DTE to DTE] |
| **INACTIVE** | Successful connection to switch [DTE to DCE] w/out DTE detected on other end of PVC<br>• Misconfig on switch |
| **DELETED** | DTE is config for DLCI that switch doesn't recognize as valid for that int |

**R1# show frame-relay map** Display current map entries

**R1# debug frame-relay lmi** Determine whether rtr/FR switch are sending/receiving LMI packets properly

**Possible values of status fields:**

| | |
|---|---|
| **0x0** | Switch has DLCI programmed: But it's not usable<br>• Possibly other end of PVC down |
| **0x2** | FR switch has DLCI: Everything operational |
| **0x4** | FR switch doesn't have DLCI programmed for rtr:<br>• Was programmed at some point in past<br>• Could be caused by DLCIs being reversed on router<br>• PVC being del by SP in FR cloud |