

# VoFR Encapsulation and Fragmentation

Document ID: 5733

---

## Introduction

### Prerequisites

- Requirements
- Components Used
- Conventions

### Guidelines for Configuring VoFR Encapsulation and Fragmentation

- Configuration Chart
- VoIP over Frame Relay Configuration
- VoFR Configurations

### Encapsulation

- Encapsulation for Data Traffic (IETF and Cisco)
- Encapsulation for Voice and Voice and Data

### Fragmentation

- FRF.11 Annex C Fragmentation
- Cisco Proprietary Fragmentation
- FRF.12 End-to-End Fragmentation

### Related Information

---

## Introduction

This document describes the encapsulation and fragmentation types of Voice over Frame Relay (VoFR).

## Prerequisites

### Requirements

This document requires a basic knowledge of the Frame Relay protocol, the dial-peer concepts, VoFR, and the different steps involved in a call setup. For information on VoFR configuration, refer to *Configuring Voice over Frame Relay*.

### Components Used

The configurations discussed in this document are implemented on these hardware devices:

- Cisco 3640 multiservice routers used as spoke routers
- Cisco MC3810 used as spoke routers
- Cisco 2500 series router used as a Frame Relay switch.

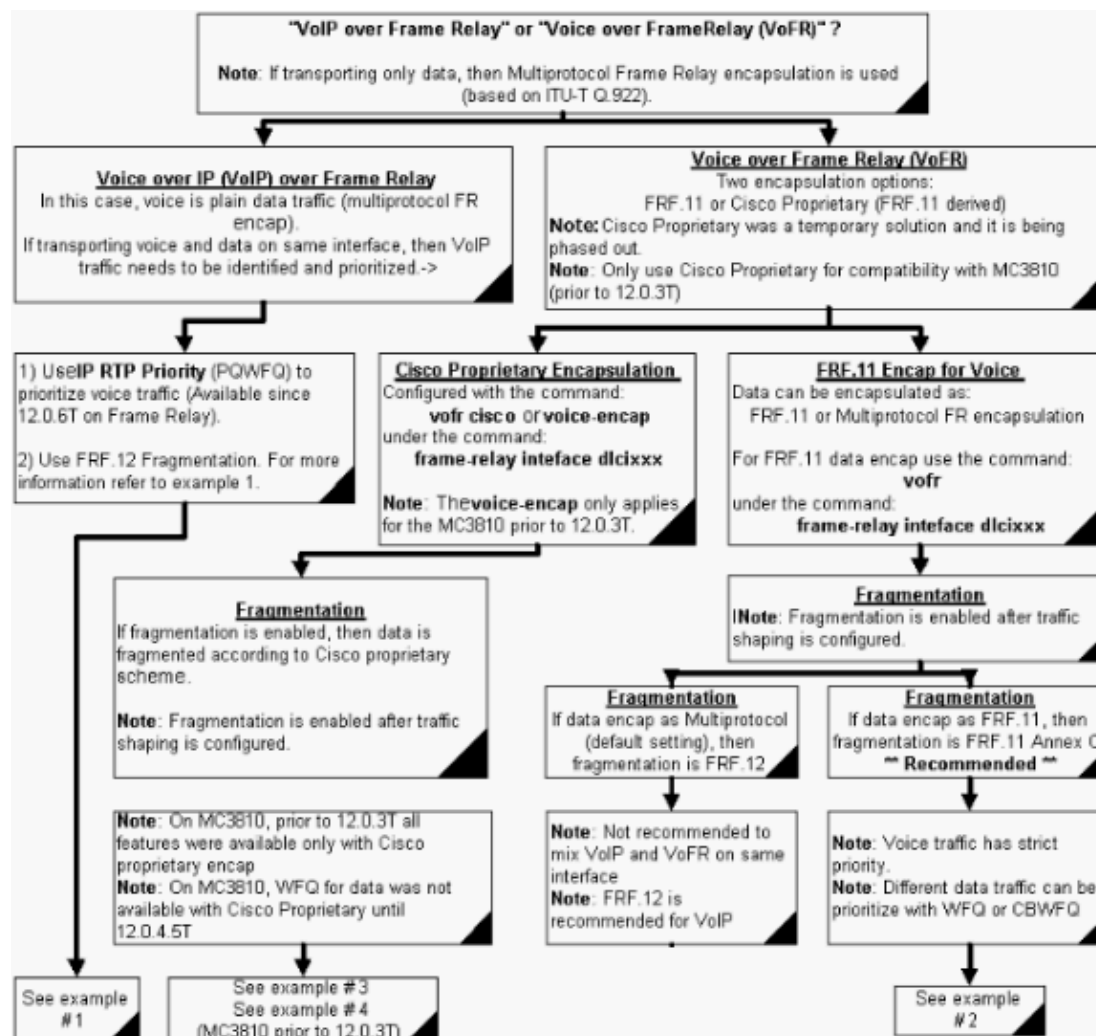
The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

### Conventions

For more information on document conventions, refer to the *Cisco Technical Tips Conventions*.

# Guidelines for Configuring VoFR Encapsulation and Fragmentation

## Configuration Chart



For more information on Traffic Shaping, refer to Frame Relay Traffic Shaping for Voice over IP (VoIP) and VoFR.

For more information on fragmentation, see the Frame Relay Fragmentation for Voice section of this document.

## VoIP over Frame Relay Configuration

This section includes the various configuration examples on VoFR.

**Note:** Only relevant output is displayed.

Example 1 displays the configuration required for VoIP over Frame Relay.

VoIP over Frame Relay (Example 1)
!
version 12.3

```

interface serial0
  encapsulation frame-relay
  frame-relay traffic-shaping
  bandwidth 32
  frame-relay ip rtp header-compression
!
interface s0.1 point-to-point
  ip address 192.168.1.1 255.255.255.0
  frame-relay interface-dlci 100
  class frf12

!---- The class name "frf12" was randomly selected.

!
map-class frame-relay frf12
  no frame-relay adaptive-shaping

!---- True CIR must be here.

  frame-relay cir 32000
  frame-relay bc 1000
  frame-relay be 0
  frame-relay mincir 32000
  frame-relay fragment 40
  frame-relay fair-queue 64 256
  frame-relay ip rtp priority 16384 16383 100
!
dial-peer voice 1 voip

  destination-pattern 9.....

  session target ipv4:192.168.1.2

dial-peer voice 2 pots
  destination-pattern 88888888
  port 3/0/0

```

For more information on VoIP over Frame Relay, refer to VoIP over Frame Relay with Quality of Service (Fragmentation, Traffic Shaping, LLQ / IP RTP Priority).

## VoFR Configurations

**Note:** Only relevant output is displayed.

This example displays the configuration required when FRF.11 encapsulation is used for Voice with standard encapsulation for data:

### Frame Relay Forum (FRF).11 for Voice Standard Encapsulation for Data

```

!
version 12.3

interface S0
  encapsulation frame-relay
  frame-relay traffic-shaping
!
interface S0.1 point-to-point
ip address 192.168.1.1 255.255.255.0
  frame-relay interface-dlci 100
  class frf11

```

```

!--- The class name "frf11" was randomly selected.

vofr cisco
!

!--- For information on the vofr cisco command please refer to vofr

map-class frame-relay frf11
no frame-relay adaptive-shaping

!--- True CIR must be here.

frame-relay cir 32000
frame-relay bc 1000
frame-relay be 0
frame-relay mincir 32000
frame-relay fair-queue 64 256 2 600
frame-relay voice bandwidth 20000
frame-relay fragment 40
!
dial-peer voice 1 vofr
destination-pattern 9.....
session target serial0 100

dial-peer voice 2 pots
destination-pattern 88888888
port 3/0/0

```

For a detailed explanation on all the above configurations, refer to Frame Relay Traffic Shaping for VoIP and VoFR.

Example2 displays the configuration required when FRF.11 encapsulation is used for Voice and Data with FRF.11 Annex C fragmentation.

#### FRF.11 for Voice, FRF.11 for Data + Traffic Shaping & FRF.11 Annex C Fragmentation **\*\*\*(Example 2)\*\*\***

```

interface S0
encapsulation frame-relay
frame-relay traffic-shaping
!
interface S0.1 point-to-point
ip address 192.168.1.1 255.255.255.0
frame-relay interface-dlci 100
vofr data 4 call-control 5

!--- For information on the vofr data command please refer to vofr

class frf11

!--- The class name "frf11"
!--- was randomly selected.

!
Dial-peer voice 1 vofr
destination-pattern 9.....
session target serial0 100
!
dial-peer voice 2 pots

```

```

destination-pattern 88888888
port 3/0/0

map-class frame-relay frf11
  no frame-relay adaptive-shaping
  frame-relay voice bandwidth 48000
  frame-relay cir 64000
  frame-relay BC 1000
  frame-relay be 0
  frame-relay mincir 64000
  frame-relay fragment 40

```

Example3 displays the configuration required when Cisco Proprietary encapsulation is used for Voice and Data with traffic shaping and fragmentation.

### Cisco Proprietary Encapsulation for Voice and Data + Traffic Shaping & Fragmentation **\*\*\*(Example 3)\*\*\***

```

interface S0
  encapsulation frame-relay
  frame-relay traffic-shaping
!
interface S0.1 point-to-point
ip address 192.168.1.1 255.255.255.0
  frame-relay interface-dlci 100
  vofr cisco

!--- For information on the vofr cisco command please refer to vofr

  class vofr_cisco

!--- The class name "vofr_cisco"
!--- was randomly selected.

!
Dial-peer voice 1 vofr
  destination-pattern 9.....
  session target serial0 100

!
dial-peer voice 2 pots
  destination-pattern 88888888
  port 3/0/0
!
map-class frame-relay vofr_cisco
  no frame-relay adaptive-shaping
  frame-relay voice bandwidth 48000
  frame-relay cir 64000
  frame-relay BC 1000
  frame-relay be 0
  frame-relay mincir 64000
  frame-relay fragment 40

```

### MC3810 – prior to Cisco IOS® Software Release 12.0.3T

```

interface S0
  encapsulation frame-relay
!
interface S0.1 point-to-point
ip address 192.168.1.1 255.255.255.0
  frame-relay interface-dlci 100
!
dial-peer voice 1 vofr

```

```
destination-pattern 9.....
session target interface s0.1 100
```

### Cisco Proprietary Encapsulation for Voice and Data + Traffic Shaping & Fragmentation MC3810– prior to Cisco IOS Software Release 12.0.3T **\*\*\*(Example 4)\*\*\***

```
interface S0
  encapsulation frame-relay
  frame-relay traffic-shaping
!
interface S0.1 point-to-point
ip address 192.168.1.1 255.255.255.0
  frame-relay interface-dlci 100
  voice-encap 40
  class vofr_cisco

!--- The class name "vofr_cisco"
!--- was randomly selected.

!
Dial-peer voice 1 vofr
  destination-pattern 9.....
  session target interface s0.1 100
!
map-class frame-relay vofr_cisco
  no frame-relay adaptive-shaping
  frame-relay voice bandwidth 48000
  frame-relay cir 64000
  frame-relay BC 1000
  frame-relay be 0
  frame-relay mincir 64000
```

**Note:** For more information on all the above commands, refer to the Command Lookup tool ( registered customers only) .

## Encapsulation

### Encapsulation for Data Traffic (IETF and Cisco)

Before discussing voice encapsulation, it is important to look at multiprotocol encapsulation for data. With VoFR implementations, data can be FRF.11 encapsulated or multiprotocol encapsulated.

Frame Relay protocol is based on the International Telecommunication Union (ITU)–T Q.922 Annex A standard, or ANSI T1.618 in the United States. It provides a minimal set of switching functions to forward variable–sized data payloads through a network.

First, Frame Relay framing is another High–Level Data Link Control (HDLC)–like framing called Link Access Protocol for Frame bearer services (LAPF), and the basic structure as defined in Q.922 is laid out here:

#### Frame Relay Frame

<b>01111110 Flag (one byte)</b>	<b>Address Field (one, two or four bytes)</b>	<b>Data (variable)</b>	<b>Cyclic Redundancy Check (CRC) (two bytes)</b>	<b>01111110 Flag (one byte)</b>
---	---	----------------------------	--	---

The **Address** field contains these fields:

- **Data Link Channel/Connection Identifier (DLCI)** As the name suggests, it identifies the Virtual Circuit (VC).
- **Discard Eligible (DE)** If set, it indicates the frame can be discarded first if congestion is experienced.
- **Forward Explicit Congestion Notification (FECN)** The network experiences congestion in the direction of the frame flow. This notification is then intended for the receiver. The idea behind it is that the receiver can delay their acknowledgements. The common implementation is "Don't care".
- **Backward Explicit Congestion Notification (BECN)** The network experiences congestion in the opposite direction of the frame flow. This notification is then intended for the sender, which might slow down the speed of transfer and then avoid retransmission.
- **Command/Response (C/R) bit** Used for control and management frames.
- **Address field extension (EA)** Used to see if the size of the address field is two, three or four bytes.

The **Flag** is used to delimit the beginning and end of the frame. The protocol makes sure that six contiguous "1"s can only be seen in the flags. This is achieved by placing a "0" after five contiguous "1" in any other field.

The Internet Engineering Task Force (IETF) created RFC 1490 to ease implementation of data encapsulation and de-encapsulation. This RFC specifies that the Data field is used as described here:

#### Frame Relay Frame: Data Field Format

<b>Control UI</b> <b>0x03</b>	<b>Optional</b> <b>Pad 0x00</b>	<b>NLPID</b> <b>(one byte)</b>	<b>Encapsulated</b> <b>Upper-Layer Data</b>
----------------------------------	------------------------------------	-----------------------------------	--

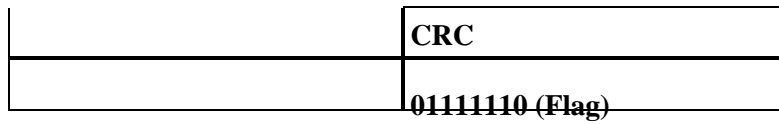
- **Control Unnumbered Information (UI)** This can be safely ignored, as it does not have any significance.
- **Optional Padding** The one-byte padding is added to adjust the size of the frame to an even number.
- **Network Level Protocol Identifier (NLPID)** This byte identifies which Layer 3 protocol the data corresponds to. NLPIDs are defined by the ISO TR 9577.

**Note:** The NLPID is only one byte long, so there are few possibilities.

**Note:** RFC 1490 specifies a two-byte Address Field (this implies DLCI values from 0 to 1023).

In summary, an IP packet IETF-encapsulated into a Frame Relay frame looks like this:

<b>Frame Relay Header</b>	<b>01111110 (Flag)</b>
	<b>Address Field...</b>
	<b>...Address Field</b>
<b>RFC 1490 Header</b>	<b>Control UI (0x03)</b>
	<b>Optional Pad (0x00)</b>
	<b>NLPID</b>
	<b>...</b>
	<b>Data (Variable Size)</b>
	<b>...</b>
<b>Frame Relay Header</b>	<b>CRC</b>



**Note:** In the above diagram, each box represents one byte.

If you configure Cisco encapsulation (Frame Relay encapsulation Cisco (default)), the frame has no RFC 1490 protocol–header and has essentially the DLCI plus an Ether type (two bytes). In this document, this encapsulation type is referred to as multiprotocol encapsulation.

**Note:** The basic Frame Relay protocol augmented by IETF RFC and additional agreements enables successful support for data applications such as LAN bridging, IP routing, and Systems Network Architecture (SNA) (FRF.1.1, FRF.1.2, FRF.3.1, FRF.9).

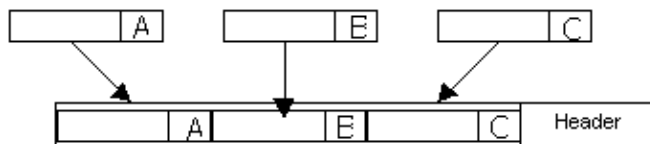
## Encapsulation for Voice and Voice and Data

To extend Frame Relay application support to transport digital voice payloads, a different encapsulation technique is required. The FRF.11 Implementation Agreement describes frame formats and procedures required for voice transport. Initial proprietary VoFR implementation on Cisco routers was FRF.11–derived. Both are described in this document.

### FRF.11 VoFR Encapsulation and Procedures

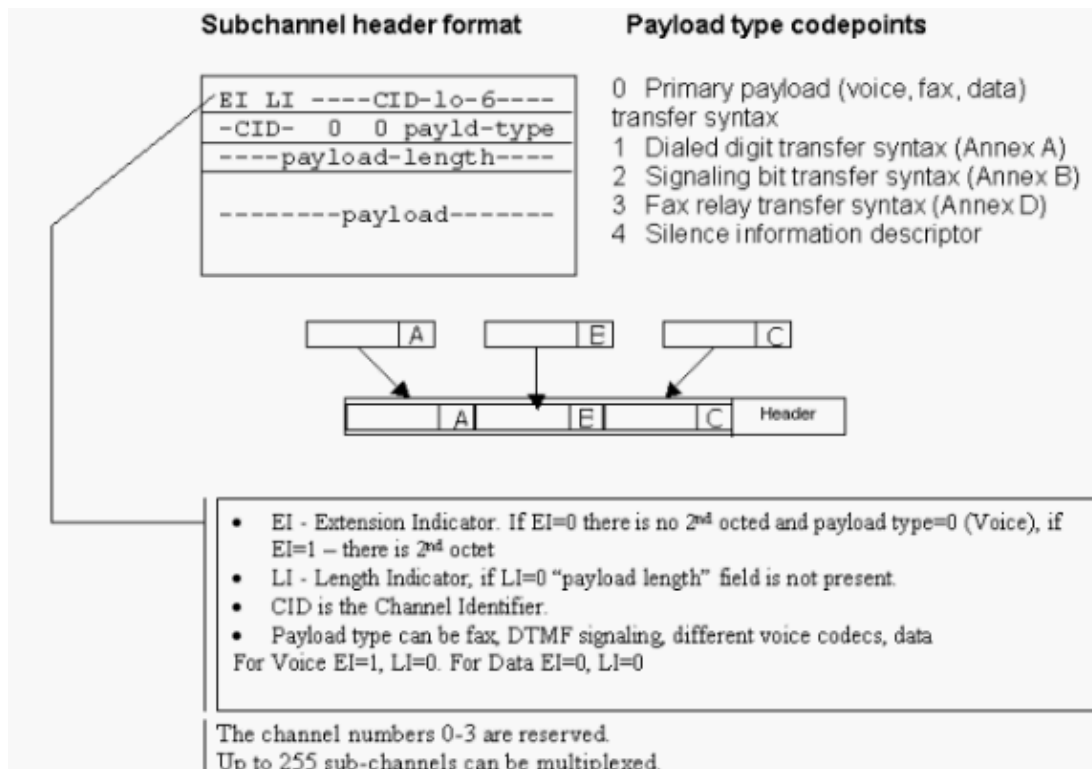
#### Frame Format

The FRF.11 Implementation defines frame formats and procedures to transfer voice traffic compressed with different codecs, fax, signaling information, dialed digits and data over Frame Relay circuits. For this, FRF.11 defines a frame format that supports sub–channel multiplexing on a single virtual circuit.



For instance, one channel can be used for G.729 compressed voice, one for signaling, and one for data.

Each sub–channel payload type is defined by the sub–channel header. At least one sub–channel is present in each frame.



- Channel IDs 0–3 are reserved.
- Up to 255 sub–channels can be multiplexed.
- Data can be configured for either multiprotocol encapsulation or FRF.11–encapsulation in the data channel.
- Cisco implementation does not mix different payload types in one frame, but can accept such frames if sent from another vendor Voice Over Frame Relay Access Device (VFRAD).

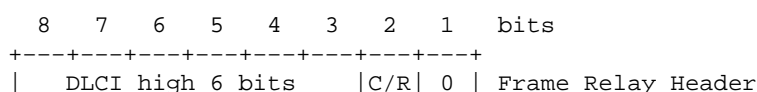
### Transfer Syntax

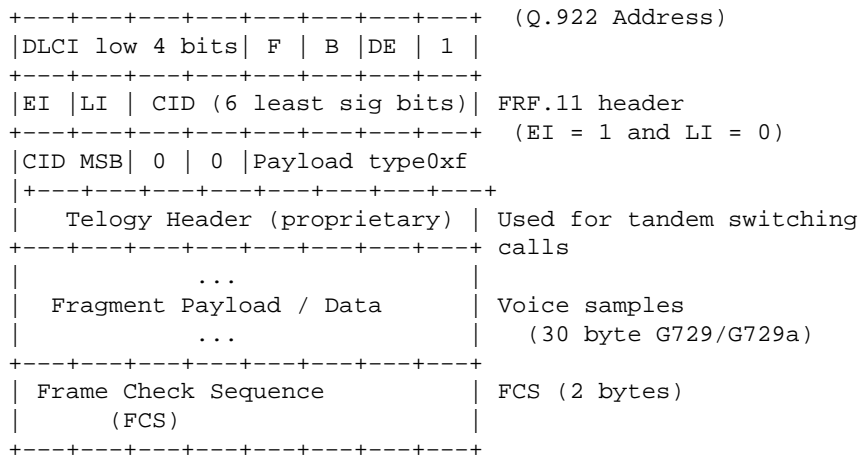
- Different payload types (different codecs, fax, dual tone multifrequency (DTMF) signaling) have different needs.
- The payload format calls out a sequence number, a codec type, and a voice payload. On some codecs, this byte is optional. For instance, for code excited linear prediction compression (CELP) codecs, the sequence number and the codec type are optional. However, these bytes are required with adaptive differential pulse code modulation (ADPCM) or pulse code modulation (PCM) codecs.
- Based on channel ID, specify in the frame what kind of payload it is.

The unique needs of different voice compression algorithms are reflected in different transfer syntax definitions. Each transfer syntax defines different frame formats and procedures, and are described in one of the annexes to FRF.11 standard.

- Annex A: Dialed digits transfer syntax (recommended for use with high compression codecs).
- Annex B: Signaling bit transfer syntax.
- Annex C: Data transfer syntax (including fragmentation which is discussed later).
- Annex D: Fax relay transfer syntax.
- Annex E–I: Voice transfer syntax.

For G.729 compressed voice, the frame looks like this:

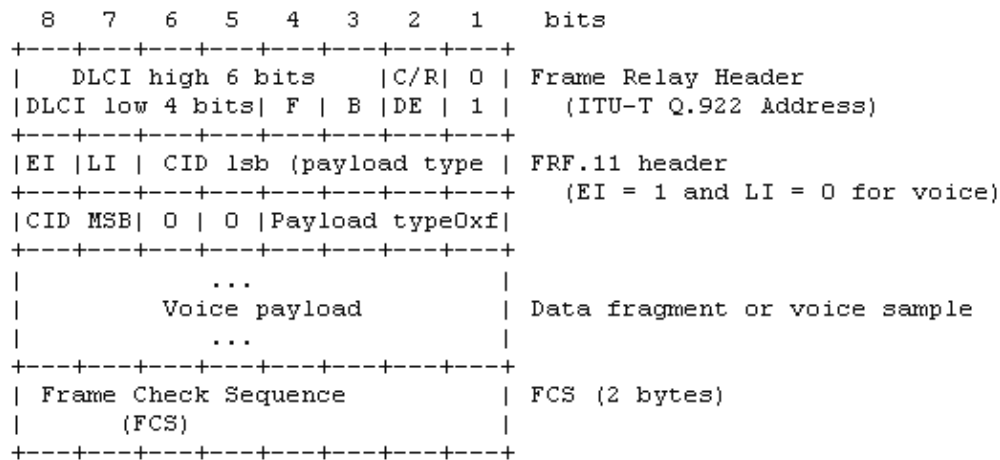




CID MSB Channel ID most significant bits.

### Cisco VoFR Proprietary Encapsulation and Procedures

Procedures here are the same as for FRF.11, but the frame format has some differences.



Frame is identified as voice if CID is not 4 or 5.

From a functionality point of view, Cisco proprietary and FRF.11 solutions are equivalent. The proprietary implementation was a temporary solution and is being phased out.

### Cisco Extensions

In addition to the FRF.11 implementation agreement, Cisco implementation has a number of extensions such as:

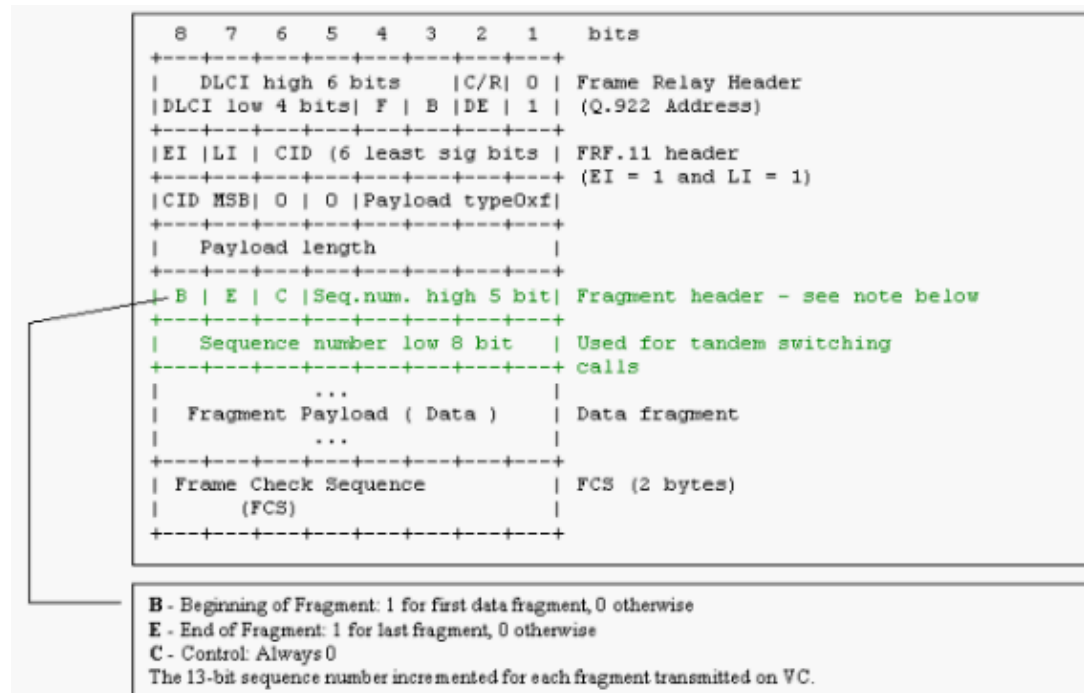
- Intelligent call setup protocol.
- Per-hop keep-alive.
- Dial-plan support.
- Hunting.
- Runs over VoFR, VoATM, and VoHDLC.
- Master-slave.
- Fragmentation.

# Fragmentation

## FRF.11 Annex C Fragmentation

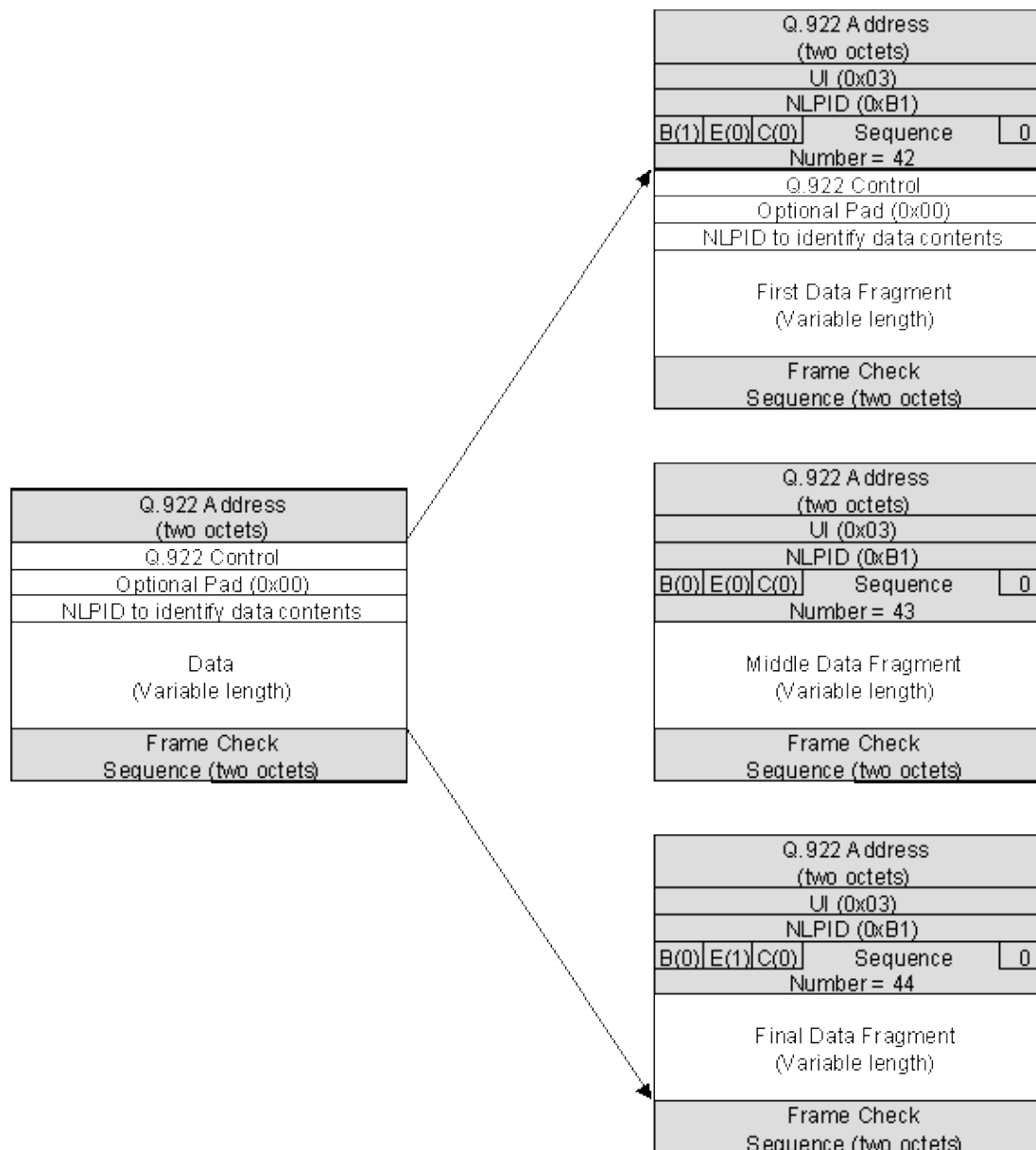
If FRF.11 is configured for data encapsulation, data can be transported in any channel (default is Channel 4).

The data frame encapsulated in the FRF.11 Annex C fragmented structure is described in this section.



The difference between a FRF.11 Annex C frame and a FRF.11 frame is that it has an extra two bytes fragmentation header.

This diagram depicts a fragmentation procedure for a FRF.11-encapsulated data frame:



Fragmentation is done after the frame is dequeued from the Weighted-Fair queue (after traffic shaping). After fragmentation, the first fragment is transmitted. The rest are interleaved with real-time frames (voice).

VoFR frames are never fragmented, regardless of size.

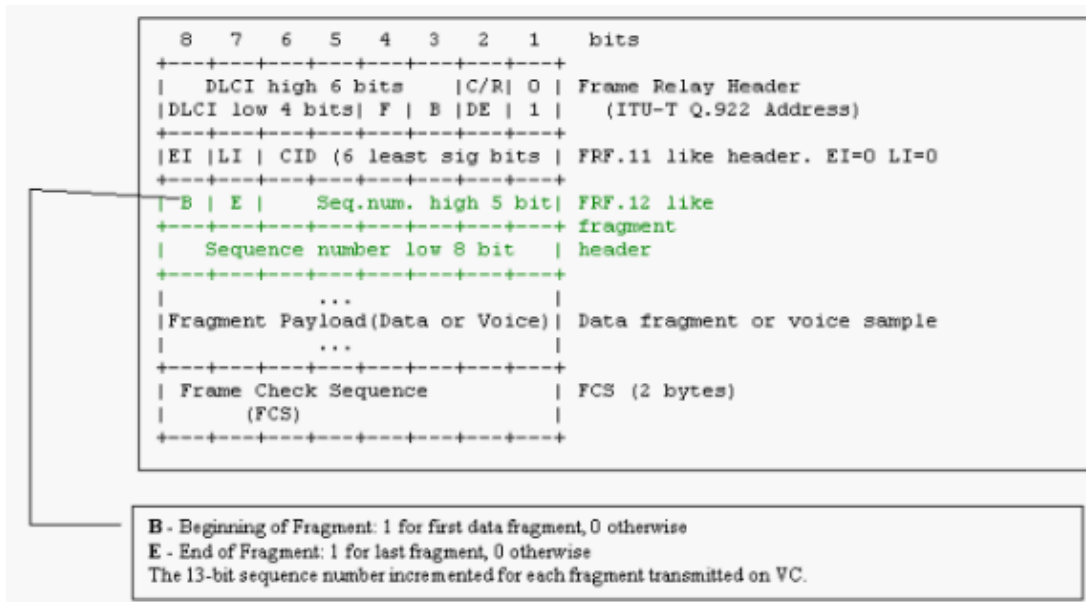
Packets are dropped if fragments arrive out of sequence.

**Note:** It is not recommended to mix VoIP and VoFR traffic on the same interface. With FRF.11, VoIP packets are treated as data packets. Therefore, each frame has at least two extra bytes per packet overhead and may be fragmented. For this reason, FRF.12 end-to-end fragmentation is recommended for VoIP over Frame Relay.

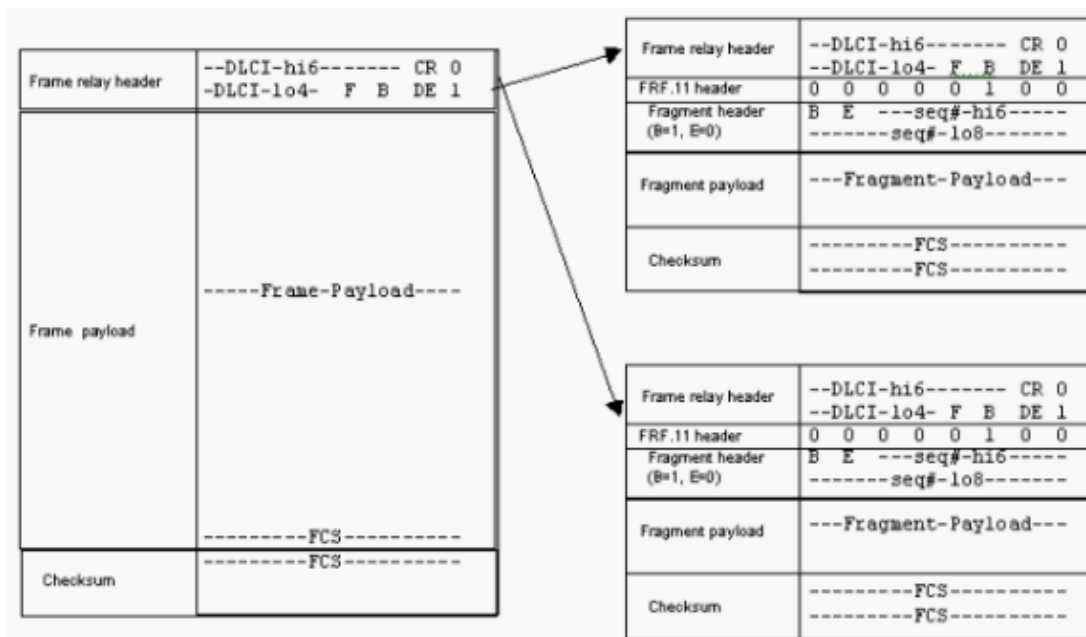
## Cisco Proprietary Fragmentation

For Cisco proprietary encapsulation, fragmentation fields are part of the frame header and procedures are the same as for FRF.12. As previously mentioned, the FRF.11 configuration is recommended unless you need to interoperate with a Cisco MC3810 (prior to Cisco IOS Software Release 12.0.3T). In addition, with the MC3810 **voice-encap** command there is Cisco bug ID CSCdp77029 (registered customers only).

The frame format for data is depicted here:



The Cisco proprietary fragmentation procedure is depicted here:



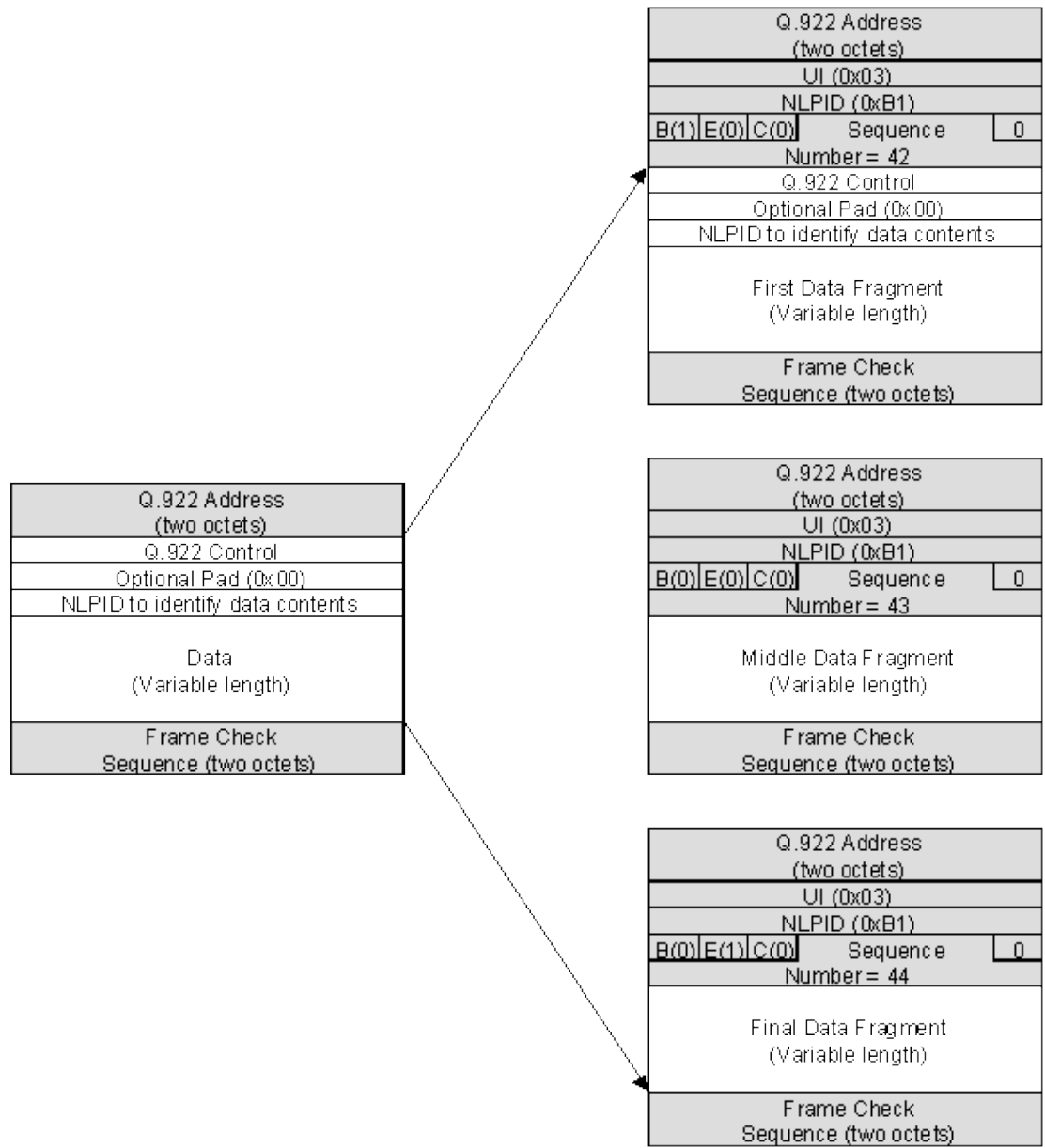
## FRF.12 End-to-End Fragmentation

FRF.12 is designed for Frame Relay multiprotocol encapsulated data packets for use with FRF.11. FRF.12 frame formats and procedures are used when voice is FRF.11-encapsulated and data has multiprotocol encapsulation.

**Note:** Permanent virtual circuits (PVCs) that utilize the VoFR sub-frame data payload (FRF.11) for non-voice frames must use the Data Transfer Syntax Payload Format defined in Annex C of FRF.11, instead of formats indicated in FRF.12.

With FRF.12 fragmentation, Frame Relay fragments get an extra two bytes of fragmentation header after Frame Relay frame header.

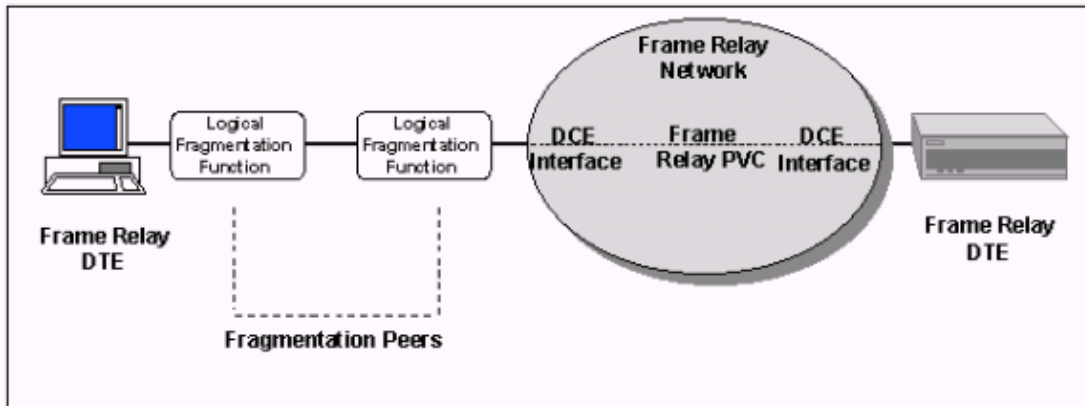
Only frames that require fragmentation (larger than defined limit, not configurable on Cisco Routers) get the fragmentation header.



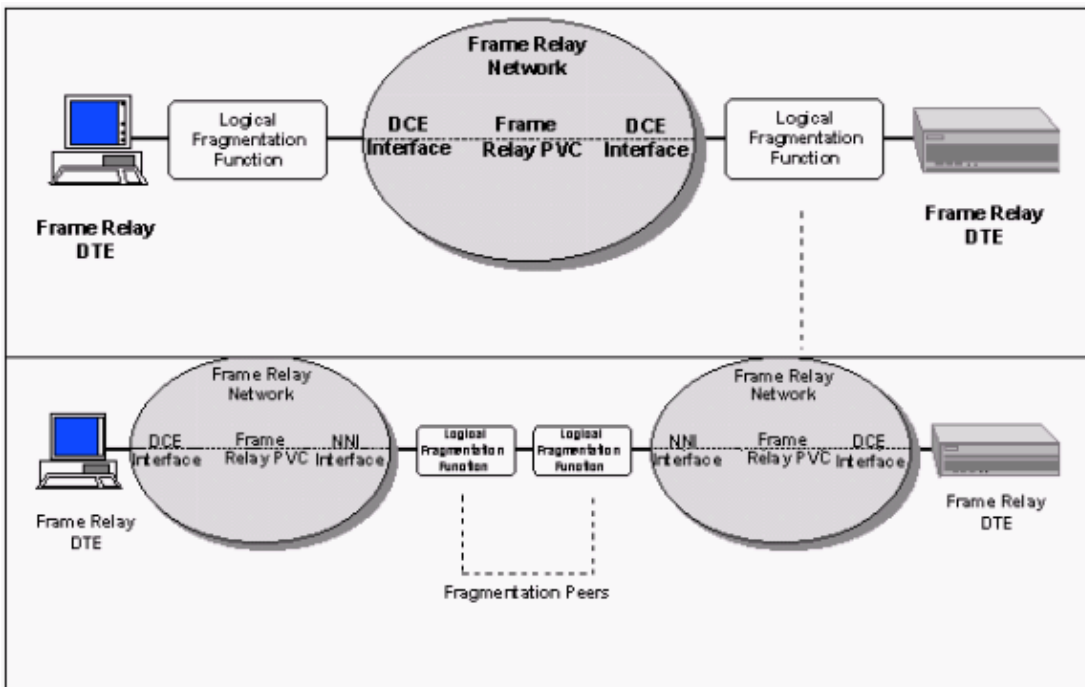
As an example, the FRF.12 fragmentation procedure for FRF 3.1–encapsulated data frame is depicted below.

**Note:** FRF.12 on switch PVCs support is only expected in Cisco IOS Software Release 12.1.2T.

For the sake of fairness, note that FRF.12 also defines user-to-network fragmentation or Interface Fragmentation where the Frame Relay frames get fragmented on the interface of a customer premises equipment (CPE) device and reassembled when they enter the Frame Relay network and network-to-network fragmentation. Frame format here is also different., as the fragmentation header here precedes the Frame Relay frame.



What is actually used is end-to-end fragmentation between peer data terminal equipment (DTE) devices.



Unlike User–Network Interface (UNI) and Network–to–Network Interface (NNI) fragmentation, which fragments all frames on an interface, end–to–end fragmentation is limited to fragmenting frames on selected PVCs.

When used between DTEs, as shown, the fragmentation procedure is transparent to Frame Relay networks between the transmitting and receiving DTEs. The transmitting Frame Relay DTEs fragment long frames into a sequence of shorter frames, which are then reassembled into the original frame by the receiving DTE.

## Related Information

- **VoIP over Frame Relay with Quality of Service (Fragmentation, Traffic Shaping, LLQ / IP RTP Priority)**
- **Setting up Cisco CallManager Traces for Cisco Technical Support**
- **Voice Technology Support**
- **Voice and Unified Communications Product Support**
- **Recommended Reading: Troubleshooting Cisco IP Telephony**
- **Technical Support & Documentation – Cisco Systems**

