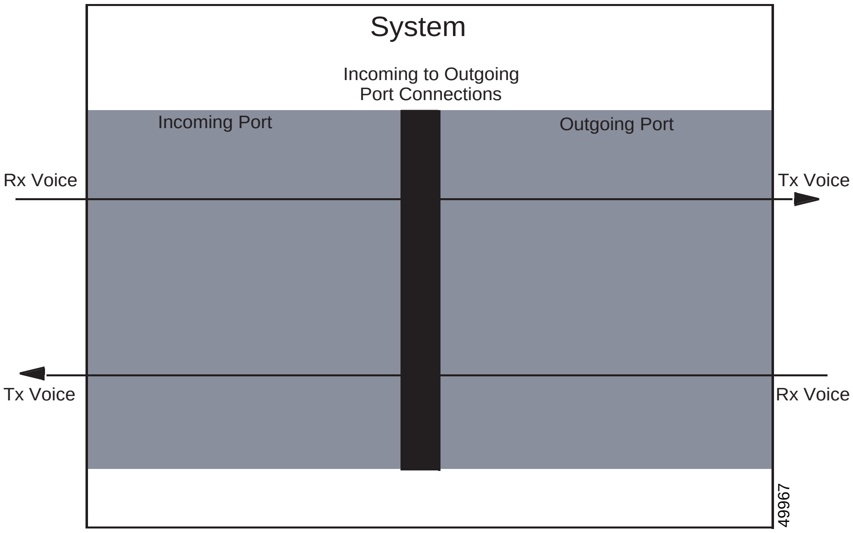
CHAPTER 5

Conferencing Examples

The four examples that follow illustrate several uses for the system conferencing feature. Each example contains two diagrams—one that illustrates the voice paths established in the conference, and a second that represents the port-to-port relations of the conference structure. The conference structure diagram indicates the number of conference ports required to satisfy the scenario. The command to start and tear down (where necessary) each conference is included in the text.

Information for the Conference Control ($6D) command and other commands is contained in the *Cisco VCO/4K Extended Programming Reference* and *Cisco VCO/4K Standard Programming Reference*. Voice path diagrams are based on the general format shown in Figure 5-1.

Figure 5-1 General Voice Path Format



Two-Party Call With Monitor

A supervisor at a catalog sales company decides to evaluate the performance of a phone-order attendant. Because the supervisor does not want the attendant to be aware of the monitoring, the supervisor arranges a one-way, receive-only connection. The text that follows describes the processing that could take place during this scenario.

Assume a stable call between a caller at address $21 and an attendant station at address $56 is in progress (both ports CP_STAB).

A supervisor with a station at address $52 wants to monitor the attendant's performance (port is CP_IDLE with COS = A2). Based on input from the supervisor's computer terminal, the host determines that the stable call above should be placed into a conference in which the ports at addresses $21 and $56 are two-way parties and the port at address $52 is a one-way party (supervisor functioning as a monitor). The following command sets up the conference:

```
6D  00  40  03  00  00  21  00  00  55  00  80  52
```

All three ports are now in CP_SETUP state. Return All was specified in the Network Control byte of the Network Header; the system returned a conference number of 06 to the host. Figure 5-2 illustrates a two-party call with monitor voice paths. The monitor conference structure is illustrated in Figure 5-3.

The supervisor decides to go on to another activity and signals the host to end the monitor. The following command tears down the conference and leaves the original two parties in CP_SETUP state. The monitor port is forced to CP_IDLE and remains off hook.

```
6D  06  20  02  00  00  21  00  00  56
```

The host must now use an Outgoing Port Control ($69) command to reconnect the original call.

```
69  00  21  C0  00  56  00  00
```

Both ports are already marked as being off hook, so a stable call is immediately established.

Figure 5-2 shows a two-party call with monitor voice paths.

Figure 5-2 Two-Party Call with Monitor Voice Paths

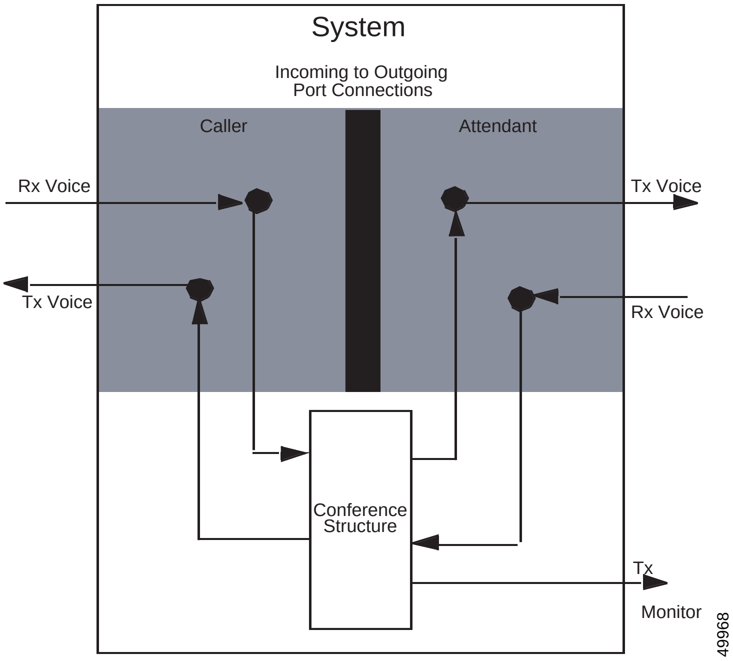
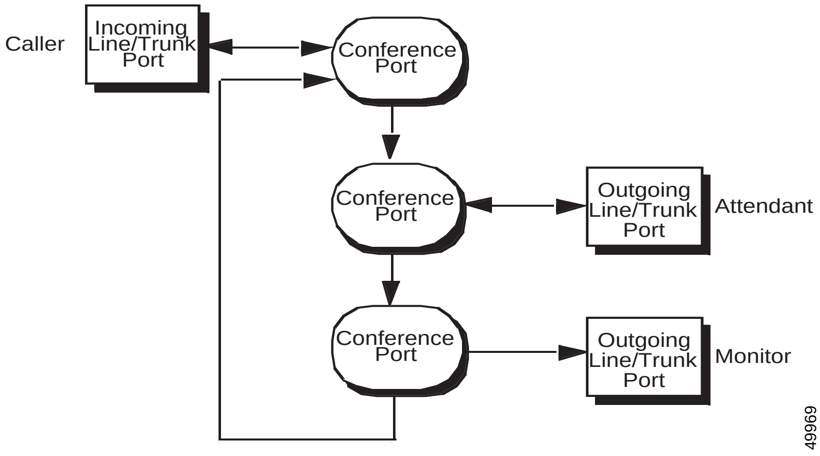


Figure 5-3 shows a two-party call with monitor conference structure.

Figure 5-3 Two-Party Call with Monitor Conference Structure



Two-Party Call With Consultant

This example is based on the scenario used in the previous example. During a phone order, a disagreement arises between the caller and attendant over an item's current price. The attendant brings the supervisor into the call to resolve the issue. At some point during the discussion the supervisor needs to confer privately with the attendant, after which the supervisor informs the customer of the problem's solution. The text that follows describes the processing that could take place during this scenario.

Assume a stable call between a caller at address $23 and an attendant station at address $51 is in progress (both ports CP_STAB).

It is determined that a consultant at address $60 should be added to the call (consultant port is CP_IDLE with COS = A2). Based on input from the attendant's computer terminal, the host determines that the stable call above should be placed into a conference in which all three ports are two-way parties. The following command sets up the conference.

```
6D 00 40 03 00 00 23 00 00 51 00 00 60
```

All three ports are now in CP_SETUP state. Return All was specified in the Network Control byte of the Network Header; the system returned a conference number of 08 to the host.

At some point in the call, the attendant and consultant need to confer without the caller. A Voice Path Control ($66) command is used to set the caller to listen to quiet tone. This action has no effect on the conference structure, since the $66 command does not alter any links or resource chains already in place.

```
66 01 00 23 04 C0
```

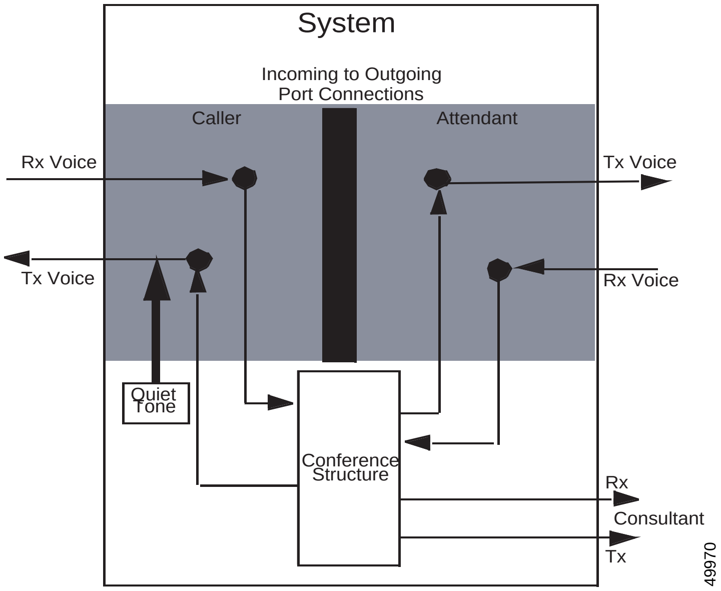
To return the caller to the conference, the following $66 command is used:

```
66 00 00 23 04 C0
```

The conference ends when all three parties go on hook (hang up). All ports are returned to CP_IDLE. No host command is required.

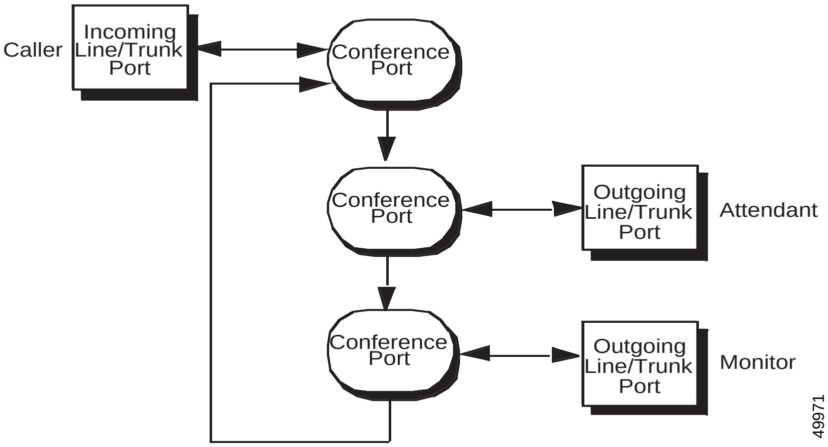
Figure 5-4 shows a two-party call with consultant voice paths.

Figure 5-4 Two-Party Call With Consultant Voice Paths



The conference structure is shown in Figure 5-5.

Figure 5-5 Two-Party Call With Consultant Conference Structure



Operator Services

The example that follows details a operator service scenario. An incoming caller wants to make a collect call. The caller, operator, and called (third) party enter into a conference until the called party accepts responsibility for the charges, at which point a stable call is established between caller (incoming port) and called party (outgoing port).

The operator at address $48 (COS = A) answers an incoming call from a caller at address $16. The caller gives the phone number for the third party. An outgoing port ($57) is hunted from a resource group and attached to a virtual call generation port by the following $6A command:

```
6A  00  00  00  FE  E0  00  08  CO  57  00  00
```

Return All was specified in the Network Control byte of the Network Header of the $6A command. The system returns the outgoing port address to the host. Based on input from the attendant's computer terminal, the host determines that the three parties should be placed into a conference in which all three ports are two-way parties. The following command sets up the conference:

```
6D  00  40  03  00  00  16  00  00  48  00  00  57
```

All three ports are now in CP_SETUP state. Return All was specified in the Network Control byte of the Network Header. The system returned a conference number of 02 to the host.

When the third party answers, the operator asks the party if they will accept the charges. If the third party accepts the charges, the host idles the operator station (removing it from the conference) using the following $6D command:

```
6D  02  20  02  00  00  16  00  00  57
```

The host then creates a stable call between the caller and the third party with the following $69 command:

```
69  00  16  D0  57  00  00
```

If the third party refuses to accept the call charges, the host idles the outgoing port (removing it from the conference) using the following $6D command:

```
6D  02  20  02  00  00  48  00 00  16
```

The host then reconnects the operator and caller with the following $69 command:

```
69  00  16  D0  48  00  00
```

Once a stable call is established (between either the caller and third party, or the caller and operator), if any party goes on hook, the call ends.

Figure 5-6 shows operator services voice paths.

Figure 5-6 Operator Services Voice Paths

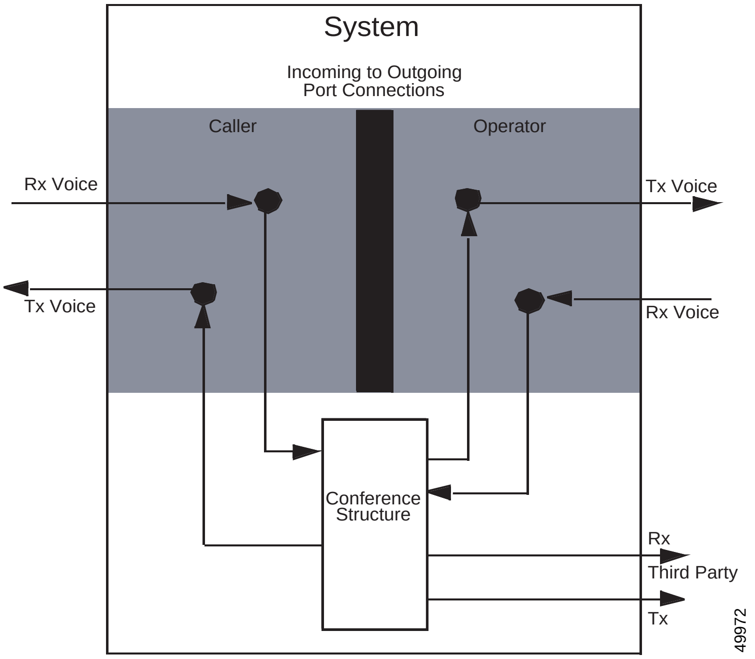
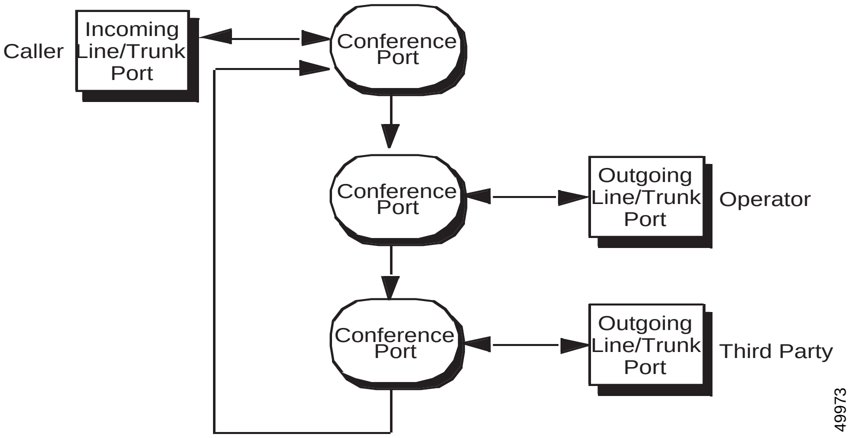


Figure 5-7 shows operator services conference structure.

Figure 5-7 Operator Services Conference Structure



Programmable Attenuation Device (PAD)

Assume a stable call between an incoming port at address $20 and an outgoing port at address $50 is in progress (both ports in CP_STAB state).

The host determines that the input from the incoming port must be increased. The command below starts a conference and raises the incoming port input level by 1.5 dB.

```
6D 00 40 02 F0 00 20 00 00 50
```

Both ports are now in CP_SETUP state. Return All was specified in the Network Control byte of the Network Header. The system returned a conference number of 06 to the host.

The conference ends when both ends go on hook (hang up). Both ports are returned to CP_IDLE. No host command is required.

Figure 5-8 shows programmable attenuation device voice paths.

Figure 5-8 Programmable Attenuation Device Voice Paths

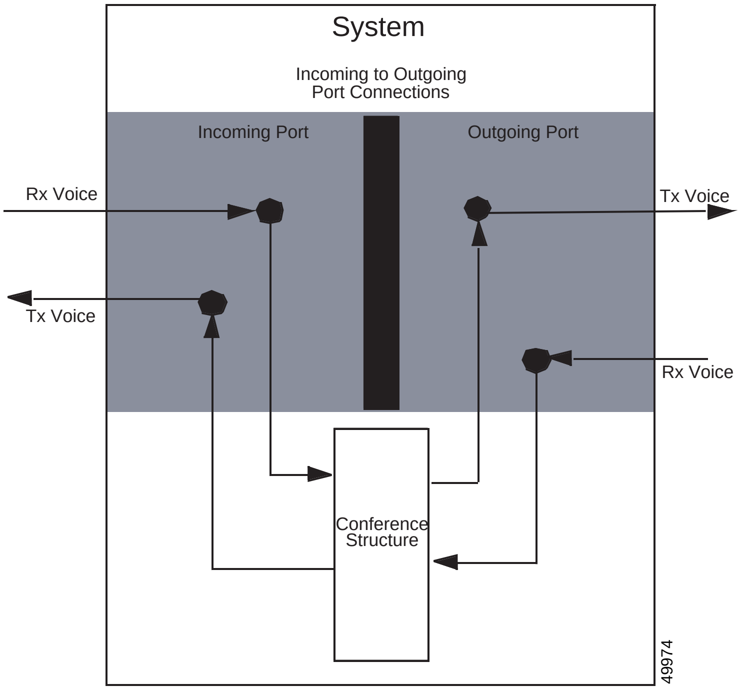
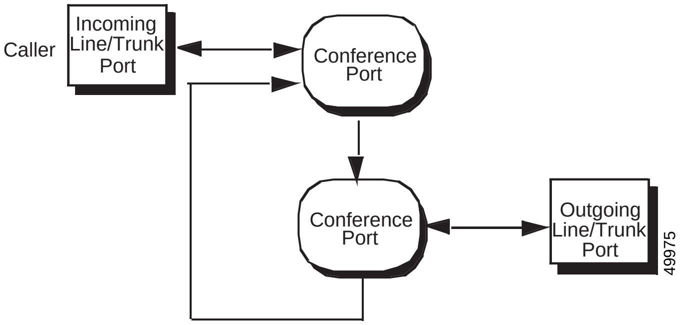


Figure 5-9 shows programmable attenuation device conference structure.

Figure 5-9 Programmable Attenuation Device Conference Structure


Caller
Incoming Line/Trunk Port
Conference Port
Conference Port
Outgoing Line/Trunk Port
49975