AlliedWare

How To | Configure ISDN Calls on Allied Telesis Routers

Introduction

With increased functionality and features comes increased complexity and flexibility. The following notes are intended as a guide to configuring Allied Telesis routers for both optimal performance and “clean” operation.

Due to the flexibility and sophistication that can be achieved using ISDN services, the command interfaces for ISDN products are invariably very extensive. Allied Telesis routers lead the field in exploiting the capabilities of ISDN. As a result, the commands for configuring an ISDN call on the Allied Telesis router have more parameters and options than just about any other in the router’s entire command set. Hence, using these commands can seem to be a daunting task.

This document is intended to lead the reader through the most common commands in the ISDN set. Just getting the call to open, though, is only half the story. A further consideration is whether you have made optimum use of all the available authentication and call-handling features. Given the large number of available features, it is difficult to know whether or not you have configured the call in the manner most optimal for your requirements.

The purpose of this document is to examine all the ISDN configuration options available on Allied Telesis routers, explain their purpose in practical terms, and give examples of how they might be used

What information will you find in this document?

This How To Note describes:

- "Getting started" on page 3 - configuring a basic ISDN call between two routers.
- "The next step" on page 6 - identifying incoming calls for security and destination in a real application.
- "How is call identification achieved?" on page 7 - setting the identification parameters on an ISDN call.

- "Callback, retries, tenacity and precedence" on page 12 - understanding the parameters in the **add isdn call** command that control the dialling process.
- "Configuring for dynamic creation of higher-level interfaces" on page 14

Related How To Notes

You also may find the following How To Notes useful:

- How To Troubleshoot ISDN Connections
- How To Configure Common ISDN Access Concentration With The Firewall

How To Notes are available from www.alliedtelesis.com/resources/literature/howto.aspx

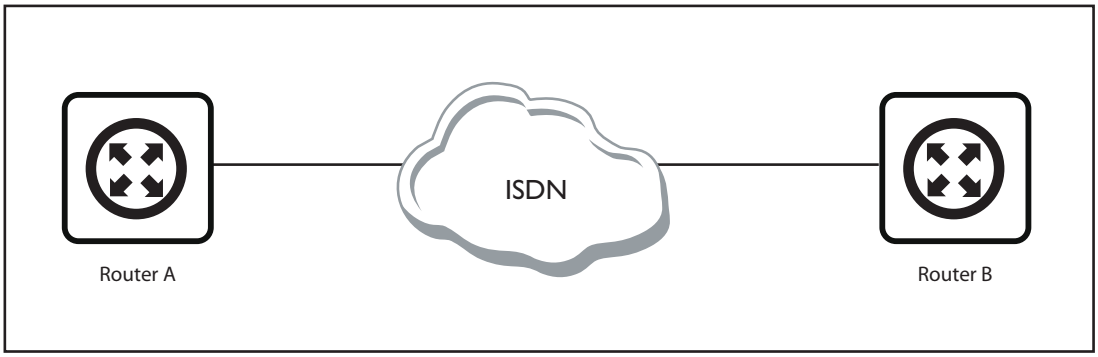
Which products and software version does it apply to?

This How To Note applies to:

- AR410, AR440S, AR441S, AR442S, AR415, AR750, and AR770 series routers
- Rapier series switches
- AlliedWare software versions: ALL

Getting started

For most users, the first thing they want to do is get an ISDN call open, just to be sure that it actually works. So, let's examine the minimum configuration required in order to open a call between two routers, as shown in the basic ISDN configuration diagram below.



Setting the Q.931 profile

The ISDN protocol, like all WAN protocols, consists of a number of layers. Most of the layers involved in ISDN are defined by internationally applicable standards. Some layers, however, are defined by standards that vary from country to country. In particular, the Q.931 layer has been implemented differently in different countries. Hence, the first step in setting your routers up for using ISDN is to ensure that the Q.931 profile has been set correctly for the country you are operating in.

The syntax for this command is as follows (please select as appropriate to your situation):

```
set q931=0 profile=etb
```

(for basic rate ISDN in Europe or Asia), or

```
set q931=0 profile=aub
```

(for basic rate ISDN in Australia), or

```
set q931=0 profile=nzp
```

(for primary rate ISDN in New Zealand)

Note: *If you are in doubt, use the default—etb/etp as this is the most universal profile.*

Adding the ISDN call definitions

Before the router can make or accept ISDN calls, at least one call definition must be configured. A call definition informs the router how it is expected to make and/or receive ISDN calls.

The command for creating ISDN call definitions is the **add isdn call** command. There are several optional parameters on this command that relate to the process of deciding which incoming calls the router will answer, and to how outgoing calls will identify themselves to receiving routers. However, in the minimum possible configuration, you forget about all this call identification business, and simply configure a call that will answer an incoming data call (as against a voice call).

On router A:

```
add ISDN call=A_to_B number=1234 precedence=in inany=true
```

On router B:

```
add ISDN call=B_to_A number=5678 precedence=out inany=true
```

Done!

Now lets look at each of the parameters in these commands and understand what they mean.

- The terms **call=A_to_B / B_to_A** gives the calls a name.
- As one might expect, **number** is the ISDN phone number of the device at the other end of the call.
- The parameter **precedence** is used to decide which call ought to take precedence in the case that the router is attempting to establish an incoming and an outgoing call simultaneously. It is not too important which end of the call is configured with **precedence=in** and which is configured with **precedence=out**, just as long as the two ends of the call are configured with complementary values for precedence.
- **Inany=true** means that this call will come open upon receipt of any incoming data call.

Creating the PPP layer

The ISDN protocol is a Physical Layer protocol. In order to establish an effective exchange of data over an ISDN call, it is necessary to configure a Data Link Layer protocol over the call. The protocol most commonly used over ISDN is Point to Point Protocol (PPP). It is possible to use other protocols like Frame Relay or X25 over an ISDN call, but we need not discuss them here.

The commands to configure the PPP link over the ISDN call are:

On router A:

```
create PPP=0 over=ISDN-A_to_B idle=on
```

On router B:

```
create PPP=0 over=ISDN-B_to_A idle=on
```

Over the PPP link it is then possible to configure Network Layer protocols like IP or IPX.

However, just configuring the PPP layer is enough to test your ISDN calls. To activate the call on router A, you would simply type:

```
activate isdn call=A_to_B
```

To see if the call connected, use the command **show isdn call**–this outputs a list of the currently configured calls, followed by a list showing which of them is currently active. If the call A_to_B appears in the list of currently active calls, and its state is shown as **ON**, then the call has successfully connected. The command **show ppp** should also show the PPP link to have come open.

```
show ISDN call

ISDN call details
Name             Number       Remote call     State              Precedence
-------------------------------------------------------------------------
A_to_B            1234                         (E) IN & OUT           IN
-------------------------------------------------------------------------

ISDN active calls
Index   Name               Interface   User       State   Prec
-----------------------------------------------------------
0        A_to_B            BRI0       03-00       ON      No
-----------------------------------------------------------

show PPP

Name             Enabled  ifIndex  Over              CP             State
-------------------------------------------------------------------------
ppp1             YES      13                         IPCP           OPENED
                          isdn-A_to_B                LCP            OPENED
-------------------------------------------------------------------------
```

The next step

Identifying incoming calls

The setup described above is OK for testing a connection, but in a real application you invariably will not want your routers to have just a single, all answering, call configured on them. There are really three reasons for this:

1. **Security**

   Answering any old incoming call does open you up to the possibility of hackers gaining access to your network. By configuring the router to only accept calls that fulfil certain identification criteria, you significantly enhance your network security.

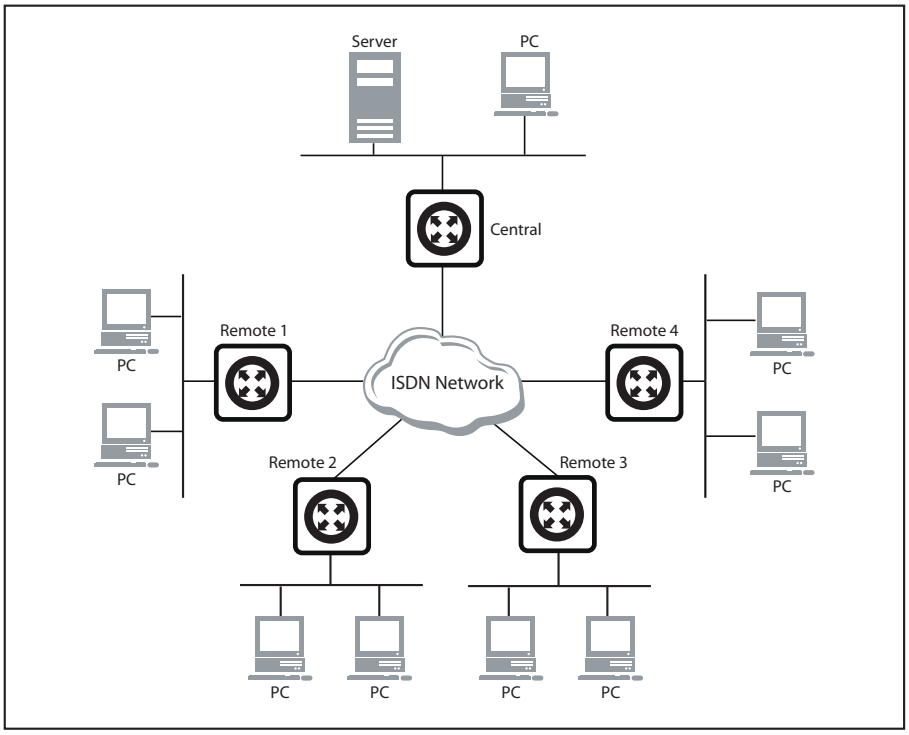
2. **Choosing between logical links**

   Consider a setup in which several remote routers are set up to all call into a single central router (as shown in the network diagram below).

   When any particular remote site calls in to the central site, the central router needs to be able to determine just which remote router is calling. This is so that it can then route the correct traffic down the call. Obviously, it is no good routing data destined for devices on the LAN at site 2 down an ISDN call connected to site 3, for instance.

3. **Choosing between devices on an S-Bus**

   An S-bus is a set of sockets that can be attached to an ISDN NTU to enable more than one device use the ISDN circuit. When a call comes in on the circuit, it must be possible for each device on the S-bus to decide whether or not the call is actually destined for it.



How is call identification achieved?

An ISDN call is always initiated by sending out a "call setup" packet. This packet contains various pieces of information (known as information elements) like the number being called, the type of call (data or voice) etc. There are three particular information elements in the call setup packet that can be used for call identification: **subaddresses**, **calling number** and **user-user** data. The basic idea is that the router making the call puts a certain string or number into one or more of these information elements. The router receiving the call setup packets extracts the contents of the information elements and uses them to identify the call.

Let us now look in detail at the parameters that can be set on the ISDN call definition to carry out this identification.

Searching and checking

As there is more than one field in the ISDN call setup packet that can be used for identifying a call, it is possible to have two layers of identification. The first level has been dubbed "searching" and the second level "checking". Looking at the syntax of the **add isdn call** command, you will see that there are three search parameters (**searchcli**, s**earchsub**, and **searchuser**) and three check parameters (**checkcli**, **checksub**, and **checkuser**). On any given call you can specify multiple search parameters and multiple check parameter. The logic of the call identification process works as follows:

When the call setup packet is received, each call that has been configured with one or more search parameters is looked at in turn, its search criteria are checked against the content of the relevant information elements of the call setup packet. The first call that comes up with a match is chosen to be the matching call. If, in addition, this call is configured with check parameter(s), its check criterion is also checked against the content of the relevant incoming information element. If the check fails, the incoming call is rejected. If the check passes, the incoming call is accepted.

Note: *If there is no check parameter configured on the first call to match the search criterion, the incoming call is accepted there and then. Also—if no call has a search criterion that matches the incoming call, then the call will be rejected, unless the router has been configured with a call with* **inany** *set to true.*

The SEARCH and CHECK criteria

In the description above of the call identification logic, there is reference made to the search and check criteria, but what exactly are these criteria, i.e. just exactly how do you specify the strings that you are looking for in the relevant information elements?

In the cases of the subaddress and user-user information elements there are two possible choices of the string to look for—the local call name (the name of the call itself) or the remote call name (the value given to the parameter **remote**, typically the call name

configured on the router calling in). In the case of CLI (Calling Line Identification), things are a little different. A list of phone numbers (called a CLI list) can be created, and the CLI value in the incoming call setup can be compared against all the items in the list.

This makes it possible to have more than one valid number from which to call in. In addition, you can see from the syntax of the **add isdn call** command that there are the options **present** and **required**. The former means that the CLI number will be checked if there happens to be one in the incoming call setup packet, the latter means that the call cannot be accepted unless there is an CLI number in the incoming call setup packet, AND that this number must match a member of the clilist.

Put the right data into an outgoing call setup packet

So far we have only looked at the process of identifying incoming calls. Now let us look at how to configure a call to put the desired string(s) into the desired information element(s) when calling out.

There are three parameters on the **add isdn call** command which enable you to specify the contents of the three possible information elements: **outsub**, **outuser**, and **remote**.

The parameters **outsub** and **outuser** can be set to either local (the actual call name) or remote (the value given to the parameter; **remote** - is usually the call name configured on the router at the other end of the call).

If, for example, **outsub** is set to **local**, then the call name will be put into the subaddress field of the call setup packet when an outgoing call is made.

For **outcli** there are three options:

- `outcli=calling`

  indicates that the value given to the **callingnumber** parameter will be entered into the CLI information element.

- `outcli=interface`

  indicates the number configured on the interface out which the call is being made will be put into the CLI information element. This is particularly for the case where a router possesses more than one ISDN interface, and you cannot be sure which interface the call will go out through. You can specify the number on each ISDN interface using the **set q931** command.

- `outcli=nonumber`

  indicates an empty CLI information element will be resent in the call setup packet, and that the ISDN network provider's switch will fill in the correct calling number when the packet arrives at the switch.

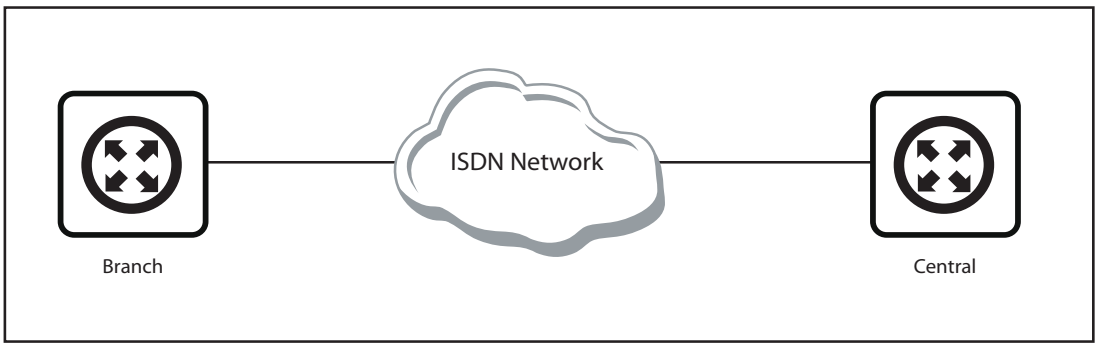
Deciding which method to use

There are the three information elements to choose from—cli, subaddress and user-user data. How do you decide which information element(s) you will use? Often you will find that there is actually no choice, as a lot of ISDN network providers do not pass all the information elements through. In Europe it is common to pass CLI, but not subadresses. A good many ISDN networks do not pass through the user-user data field. So, even if your router creates a call setup packet which contains a particular information element, when the packet arrives at the other end, that information element will have been stripped off if the network is not set up to pass it through. With some ISDN networks you can pay an extra fee to have various information elements passed through for calls to and from your connection.

However, if you do have the choice, the most secure option is to search on one element and check another. The hardest element to spoof is CLI, so if that is available it would certainly be best to search on cli, and check either subaddress or user-user data.

Example 1

Just to illustrate, let's consider a simple case of two routers calling each other, using subadresses for call identification.



On the router "branch"

```
add isdn call=central remote=branch num=1234 prec=in
  outsub=remote searchsub=local
```

On the router "central"

```
add isdn call=branch remote=central num=5678 prec=out
  outsub=remote searchsub=local
```

The idea here is that the name given to the ISDN call on each router is the name of the router to which the call is being made. The remote name configured on each call is the name given to the call on the other router.

So, if, for example, the Branch router makes a call, it will send out the string **branch** (as this is the remote name configured on the call, and **outsub** is set to **remote**) as the subaddress in the call setup packet. The Central router will receive this packet and compare this subaddress against the call name (as **searchsub** is set to **local**), the strings match, and the call is accepted.

Example 2

If the two routers were to use CLI for all call identification and subaddress for checking, the configuration would be:

On the router 'branch'

```
add ISDN clilist=0 number=12345

add ISDN clilist=0 number=23456

add ISDN call=central remote=branch number=12345 precedence=in
   outsub=remote callingnumber=56789 outcli=calling searchcli=0
   checksub=local
```

On the router 'central'

```
add ISDN clilist=0 number=56789

add ISDN clilist=0 number=67890

add ISDN call=branch remote=central number=56789 precedence=out
   outsub=remote callingnumber=12345 outcli=calling searchcli=0
   checksub=local
```

In this case:

When a call is made from the 'branch' router, the identifying fields in the call setup packet will contain the following values:

CLI: 56789

subaddress: branch

When a call is made from the 'central' router, the corresponding fields in the call setup packet will be:

CLI: 12345

subaddress: central

When the branch router receives a call, it will take the CLI from the incoming call setup packet and compare it against the content of clilist 0 (12345 23456); then, if a match is found, it will compare the subaddress field in the incoming packet against its own call name ('central').

Similarly, the 'central' router will compare the incoming CLI value against the list 56789, 67890 and then check that the incoming subaddress matches the callname 'branch'.

Example 3

In the example above, the routers filled the CLI field in the outgoing call setup packet by using the **callingnumber** parameter. Alternatively the call can be configured to use the phone number on the outgoing interface. So, for example, if the routers each had two BRI interfaces, and the CLI field was to be filled in by using the number configured on the interface out which the call happened to go, then the configuration would be:

On the router 'branch'

```
set Q931=0 num1=56789
set Q931=1 num1=67890
add ISDN clilist=0 number=12345
add ISDN clilist=0 number=23456
add ISDN call=central remote=branch number=12345 precedence=in
  outsub=remote outcli=interface searchcli=0 checksub=local
```

On the router 'central'

```
set Q931=0 num1=12345
set Q931=1 num1=23456
add ISDN clilist=0 number=56789
add ISDN clilist=0 number=67890
add ISDN call=branch remote=central number=56789 precedence=out
  outsub=remote outcli=interface searchcli=0 checksub=local
```

Note: *The complexity of the* ***add isdn call*** *command is intended to cater for a bewildering array of possible call identification schemes. The best approach is to decide which ONE call identification scheme works best in your network—then STICK TO THIS ONE SCHEME, and forget about all the other parameters.*

Callback, retries, tenacity and precedence

There are a number of parameters in the **add isdn call** command that can be generally regarded as controlling the dialling process.

Callback

Setting callback to **on** simply means that if this call is chosen to accept an incoming call (its call identification criteria match the contents of the incoming call setup packet), then it will immediately clear the incoming call and call back to the number that is configured on the call.

There are two prime reasons for using callback:

Security–if someone has found a clever means of spoofing CLI so as to make it look as though they are calling from a particular number, callback can counter this. Unless the intending intruder is also able to trick the ISDN switch call to route calls to the spoofed number to them, your call back will not get to them, it will go to the authentic device that they are trying to impersonate.

Billing–if, for example, your router is set up to provide an ISDN link to an Internet service provider, but the ISP wants all ISDN charges to be billed to you, even if the traffic is initiated from their end, then by setting callback to **on**, you ensure that all the ISDN calls are billed to your connection, not theirs.

Retries

Retries can be made in up to 10 groups, with up to 5 retries in each group. The parameters **RN1** and **RN2** specify the number of retries per group, and the number of groups, respectively. The parameters **RT1** and **RT2** specify the intervals between retries within a group and the intervals between groups, respectively.

Keepup (tenacity)

In addition to the retries, there is the **keepup** parameter. This is a sort of "super retry" i.e. if the call goes down for any reason other than a specific deactivation from a manager command, or by a higher layer module, then keep on retrying ad infinitum.

This is typically used for 'semi-permanent' ISDN links, where the intention is that the call be open constantly. If some event within the ISDN network causes the call to go down, then the intention is that the router try at all costs to get the call re-established.

Precedence

Although a reasonably rare occurrence, it is possible that a call can be dialling out at exactly the same moment as an incoming call arrives that matches it.

In such a case, known as call collision, the router is faced with something of a dilemma—does it reject the incoming call and continue establishing the outgoing call, or does it accept the incoming call and give up making the outgoing call?

The **precedence** parameter indicates to the router which option it should take in this situation. If precedence is set to **out**, the incoming call is rejected. If **precedence** is set to **in**, the outgoing call is abandoned.

Typically, ISDN calls are configured in pairs—one on each of the routers at the two ends of the call. In order to ensure consistent behaviour in call collisions, it is intended that the precedence on one end of the call is set to **in**, and that on the other end is set to **out**.

Alternate number

If all retries on the main number fail, then this alternate number will be tried once. If **keepup** is set **on**, then the full set of retries will be made on the alternate number as well as on the main number.

Direction

A call can be set to only accept incoming calls, or only make outgoing calls, or to do both. Setting **direction** to `out`, for instance, can be a useful security measure if you are using the router to enable you to call out to the Internet or some other service, but want to ensure that having your router connected to a public ISDN network is not making you vulnerable to attack by intruders dialling in.

On the other hand, if the router is being used to enable remote users to gain access to some service, and you want to ensure that no strange routing quirk is going to cause your router to activate ISDN calls, and incur charges, then setting the **direction** to **in** on all ISDN calls will ensure that no outgoing calls will be activated.

Required or preferred interface

If a router has more than one ISDN interface, you may wish to specify that particular outgoing calls must be made via a particular physical interface. This might be to ensure that the call goes out with a particular CLI number, or that it goes out via a connection for which a special billing agreement has been made with the ISDN network provider or to ensure that it goes out over a BRI interface rather than a PRI interface or for whatever reason.

The **intreq** parameter specifies a mandatory interface for the outgoing call— if that interface has no free channels available, then the call fails. The **intpref** parameter simply

indicates a preference for using a particular interface—if the interface has no available channels, then the call will be sent out another interface that does have channels available.

Configuring for dynamic creation of higher-level interfaces

There are two ways that higher-level (PPP, Frame Relay, etc.) interfaces can be attached to ISDN calls—**statically** and **dynamically**. When statically configuring an interface over an ISDN call, you create the ISDN call, and then explicitly create a PPP or whatever interface over the call, the particular logical PPP interface is then tied to that call. So, when the call comes open, it is that particular logical interface that is activated. In the case of dynamic creation of logical interfaces, there is no interface explicitly configured over the ISDN call. The configuration parameters on the call are required to contain sufficient information to enable the router to dynamically create a logical interface when the call comes open. Currently, only PPP interfaces can be dynamically created, and they can only be created over incoming calls.

There are two different mechanisms by which the router can create dynamic PPP interfaces over ISDN calls: using information contained in the call setup packet or using information contained in a PPP PAP response. Let us look at the details of each of these mechanisms.

Using the call setup information

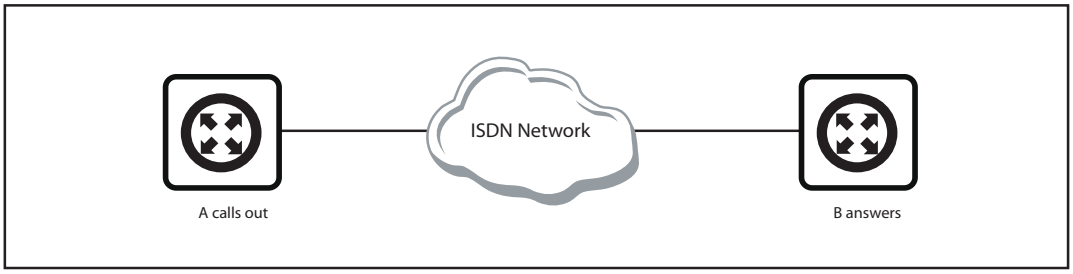
In order to correctly identify and authenticate the user calling in, it is necessary to obtain a user name and a password from the incoming call setup packet. As with the call identification process described above, the three information elements in the call setup packet that can be used are the CLI, subaddress and user-user data fields. So, the idea is to put the user ID in one of these information elements and the password in another. Having decided which information element will contain which identifier, it is then necessary to configure the answering call accordingly. The parameters which indicate where the user ID and password are deemed to be found are simply called **username** and **password**. Each can be set to any of the three information elements, or to "name", i.e. the call name can be used as the user ID or password.

Having extracted the user ID and password, it is then necessary to pass them off to some process that will authenticate them and pass back the IP address and netmask to be used on the IP interface which is to be dynamically created over the call. There are two choices for this process—either a RADIUS server or the router's own user database. The process to be used is specified by the parameter **login**, which may take on the values **user** or **radius**.

Finally, it is required that you specify the type of dynamic interface to be created over the call. In reality, the only valid choice is **PPP**.

An example

Consider a case where it has been decided to put the user ID in the subaddress information element, and the password in the CLI information element, and use a RADIUS server to authenticate the calls.



On router A, the router making the calls, the ISDN call would be configured as follows:

```
add isdn call=user1 num=1234 callingnumber=5678 dir=out prec=out
   outsub=local outcli=calling
```

and typically a static PPP interface would be configured over the call:

```
create ppp=0 over=isdn-user1 idle=on
```

On router B, which answers and creates the dynamic interface, the ISDN call would be configured:

```
add isdn call=answering num=0 dir=in prec=in user=ppp
   login=radius username=calledsub password=cli inany=true
```

It is important to note that **inany** must be set to **true** on the answering call, as the call falls outside the usual search process—it initially will accept any call, extract the necessary authenticating information, and then finally accept or reject the call on the basis of the response received from the authenticating process.

Note that on the answering router there is no need to create a PPP interface; one will be created automatically when a call is accepted. It should also be noted that several PPP interfaces can be in existence simultaneously over this one call—there is not the usual "one active interface at a time over one call" rule that is true for statically configured interfaces. In this dynamic case, the ISDN call is effectively a vehicle for extracting user IDs and passwords, and launching PPP interfaces for valid users.

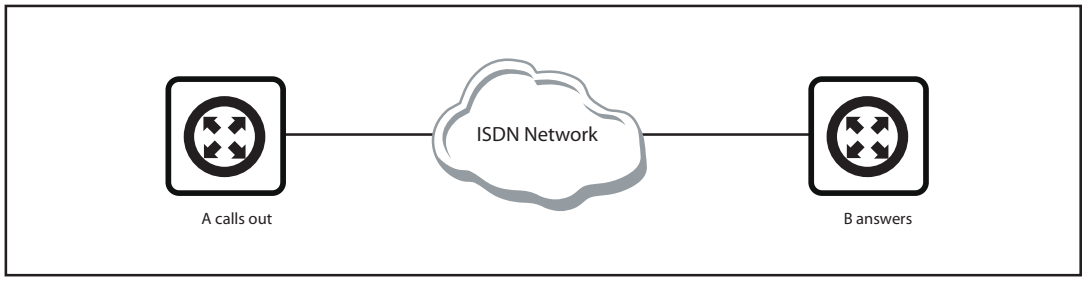
Using PAP information

When the call setup packet is used to provide the user information, the PPP interface is not created until after the call has been validated (in fact, the call is not even accepted until it has been validated). Now let's look at a second approach in which every incoming call is immediately accepted, and a dynamic PPP interface is immediately created. From then on control is passed off to the PPP layer. It is PPP that asks the calling device to identify itself using PAP, and sends the resulting user ID and password off to an authenticating process.

Since the user name and password do not have to be extracted out of the call setup packet, the ISDN call is only required to specify the authenticating process to be used, and the type of interface to be created over the call, i.e. the parameters **login** and **user**.

The **user** parameter, of course, has to be set to PPP. The **login** parameter can take on either of the values **PAP-TACACS** or **PAP-RADIUS**. In the former case, the router's user database is to be searched, and the user information only forwarded to the TACACS server if no match is found in the user database. If **PAP-RADIUS** is specified, the user database is not used, the information is forwarded straight to the RADIUS server.

An example



On router A, the router making the calls, the ISDN call would be configured with just the bare minimum of parameters:

```
add isdn call=user1 num=1234 dir=out prec=out
```

and typically a static PPP interface would be configured over the call:

```
create ppp=0 over=isdn-user1 idle=on password=password1
   user=user1
```

On router B, which answers and creates the dynamic interface, the ISDN call would be configured:

```
add isdn call=answer num=0 dir=in prec=in user=ppp
   login=papradius inany=true
```

Note, once again, that it is necessary to set **inany=true** on the answering call.

Pros and cons of the two methods

There are two quite different methods for using an ISDN call as a vehicle for authenticating incoming calls and creating dynamic PPP interfaces for those that are successfully validated. Why would you chose one method over the other?

The first method has one very significant advantage, namely that the ISDN call does not even come open unless a valid user name and password are received. This is very useful for putting potential intruders "off the scent". However, this approach does have the disadvantage that a number of ISDN networks just do not pass through enough information elements to enable a call setup packet to contain both a user ID and a password. What is more, even if the ISDN network does pass through all the information elements, there is the problem that a lot of ISDN devices that might be calling in simply cannot be configured to put particular strings into particular information elements of the call setup packet.

The latter approach is more standard—it is the method used by most other vendor's products, and it is not so likely to be hamstrung by shortcomings in either the ISDN network, or the dialling-in devices. However, it is not quite as secure, or as flexible.

USA Headquarters | 19800 North Creek Parkway | Suite 200 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
European Headquarters | Via Motta 24 | 6830 Chiasso | Switzerland | T: +41 91 69769.00 | F: +41 91 69769.11
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
www.alliedtelesis.com

C613-16081-00 REV C