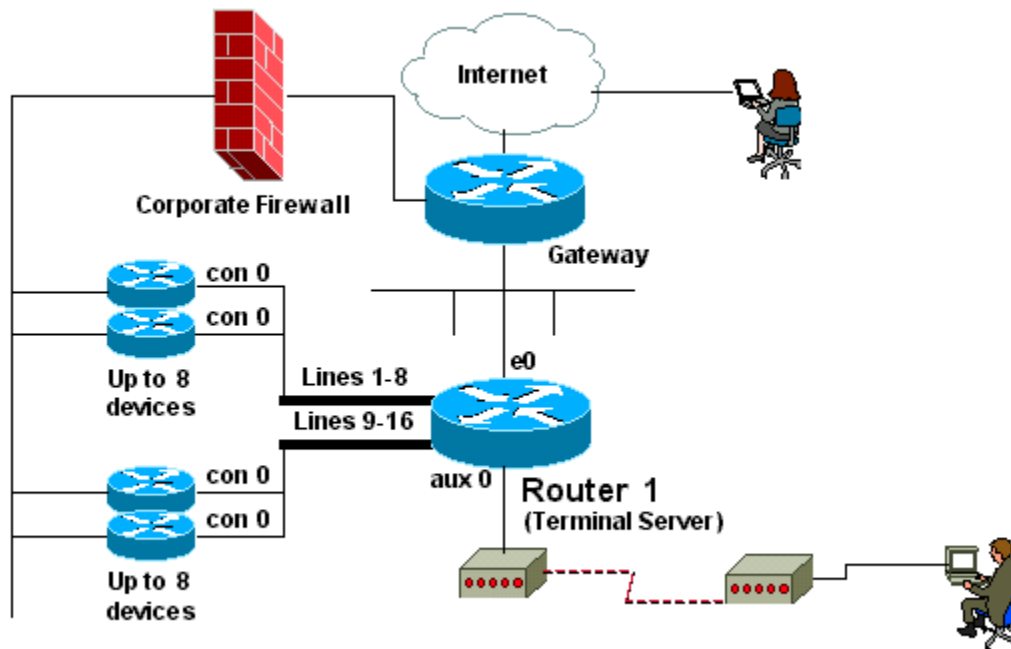


What is a Terminal Server Anyway?

An Access or **Terminal Server (TS)** is a device that allows you to access multiple devices from a single management station.

In a Lab environment, a TS will allow you to use a single computer with a single Terminal Emulation window (Hyper Terminal, Tera Term, etc.) to **access ALL the devices in your Lab!**

That way **you won't have to switch Console Cables** to access each individual device. As shown in the following diagram, when using a TS we have a **permanent connection from the Console port to the respective TS ports**. A special "Octal cable" is needed for this purpose.



The diagram shows a user, sitting at his/her computer, with a connection to the Terminal Server, **accessing and controlling multiple devices**. The connection to the TS could be either **Remote** (known as *out-of-band access*) or simply a **Local** connection through either the **Console Port** (most common case scenario) or through the **LAN** (Telnet session through the Ethernet port).

If your Lab has only a couple of devices, then switching Console cables back and forth may not be a big deal. But as the number of devices in your Lab grows, then this back and forth Console cable switching usually becomes a burden!

So adding a Terminal Server to your Lab is a very good idea! It will save you time and it will make your entire Lab experience much more enjoyable!

Our Terminal Servers are shipped pre-configured, so they are completely Plug-and Play!

For a complete description of how Terminal Servers are configured, please visit the following link from the Cisco website:

http://www.cisco.com/en/US/tech/tk801/tk36/technologies_configuration_example09186a008014f8e7.shtml

This document is also included in the Lab Disk as a PDF file; folder:

Lab Kit Setup Instructions\Terminal Server Configuration

The configuration is done using a mechanism known as **Reverse Telnet**. Reverse Telnet allows you to Telnet out from a device you are Telneting from, but on a different interface.

[Cabling](#)

Our Cisco Terminal Servers use a **Breakout Cable** known as **Octal Cable** (part CAB-OCTAL-ASYNC) which has **8 segments** of “legs”. The other side of the Cable has a special **SCSI connector** (68-pin connector) which must be connected to the respective Port in the Terminal Server.



Each Octal Cable segment is terminated with an **RJ-45 connector**. These segments must be connected **to the console port of a device**.



Our Terminal Servers are implemented with a **Cisco 2600** or **Cisco 2800** Router with an **NM-16A** or an **NM-32A** network module.

The NM-16A has two SCSI connector, so you may connect two Octal Cables, which will provide access **up to 16 devices**.

The NM-32A has four SCSI connectors, so you may connect four Octal Cables, which will provide access **up to 32 devices**.



Implementing a Terminal Server this way is a far more reliable method than using the older 2509 - 2512 series Routers.

One of the main advantages of Terminal Servers based on a 2600 or 2800 Router is the fact that **they support SSH**, which is NOT supported under any Cisco 2500 Router. (2500's do not support Crypto commands!)

Terminal Server Operation

As mentioned before, with a Terminal Server **you won't have to physically move the Console Cable** to each device that you want to access.

Instead, you simply connect each Octal Cable segment or leg to the Console Port of each device. Then you switch between “sessions” by pressing a **special combination of keyboard keys: 'Ctrl+Shift+6'** (all 3 keys at the same time) and **then 'X'**

Please notice that we used the word “sessions”. **But what exactly is a session?**

By session we mean a **Telnet session** that you create in the Terminal Server using the command **telnet**. To be technically precise, these sessions are actually **Reverse Telnet sessions!**

That is, normally you create a Telnet session using the command telnet with a specific IP Address. For instance: **telnet 192.168.1.1**

But in a Terminal Server you don't have IP addresses. What you have instead is a **Port Number**, which refers to a Logical Port that the NM Module creates in the Router.

To better explain these concepts, let's look at the actual Terminal Server's configuration, which looks like this:

```
interface Loopback0
  ip address 2.2.2.2 255.255.255.0
  !
ip host R1 2033 2.2.2.2
ip host R2 2034 2.2.2.2
ip host R3 2035 2.2.2.2
.....
ip host R15 2047 2.2.2.2
ip host R16 2048 2.2.2.2
  !
alias exec S1 telnet R1 2033
alias exec S2 telnet R2 2034
alias exec S3 telnet R3 2035
.....
alias exec S15 telnet R15 2047
alias exec S16 telnet R16 2048
```

This is the partial configuration of a 2600 Router with an **NM-16A** Module, which as you know has **16 Ports**.

Each Port is assigned a Telnet Port number that goes **from 2033 to 2048**

The first configuration command creates a Loopback Interface in the Router with **IP Address 2.2.2.2 255.255.255.0**

This IP could be anything actually, but the rest of the configuration is based on that IP. So if the Loopback IP changes, of course the rest of the config should also change!

Then the configuration includes the commands **ip host R1, ip host R2**, etc.

With these commands we create a Hostname that will be mapped to each Telnet Port and thus to each device that we want to access through the Term. Server.

So let's analyze the first command: **ip host R1 2033 2.2.2.2**

This command simply creates a **static mapping** between a Host (in this case defined by IP Address 2.2.2.2) and a TCP Port (in this case Port 2033). And it assigns a name to this mapping; in this case it assigns **R1**

So whenever the Router makes a reference to R1, it is actually making a reference to the **mapping 2033 2.2.2.2**

This would be enough to create the sessions that we need to access each device connected to the Terminal Server.

But in order to make things even easier, the configuration includes a series of **alias exec** commands, which are simply a sort of **Macro command**, which saves us some typing.

In other words, to create a Telnet session the first device (that is, to Port 2033) we would need to type this command: **telnet R1 2033**

But to make easier the configuration includes the Macro command:

alias exec S1 telnet R1 2033

So are basically creating the Macro command **S1**, which when executed, it will be translated to **telnet R1 2033**

In other words, to create a Telnet session for the first leg of the Octal Cable, at the Terminal Server's prompt you'll simply type:

TermServer#S1

This will create the session and you'll be automatically switched to that session!

So if the first leg of the Octal Cable is already connected to the Console Port of any device and if that that device is already powered up, then you'll be able to access that device and enter any commands as if you were directly connected with a Console Cable.

Of course is that device is not active (powered down), then you won't get any response from it. But if you power up that device while still in session S1, then you'll see the boot sequence appear on session S1.

Now, eventually you'll need to leave session S1 and move to another session. As mentioned above, to do you'll have to press a **special combination of keyboard keys: 'Ctrl+Shift+6'** (all 3 keys at the same time) and **then 'X'**

Once you have completed this sequence, you'll go back to the Terminal Server's prompt; in this case you'll go back to: **TermServer#**

To create another session you'll have to type S2, S3, and so on.

The best practice is to **create all sessions that you'll need right away** and then start moving around sessions.

In other words, you should type S1 and then go back to the TS's prompt. Then type S2 and go back to the TS's prompt; type S3 and so.

Now, once all sessions are created, in order to switch to a specific active session, **you don't have to type S1, or S2, etc.** Instead, you'll simply type the session number!

For instance, if you are at the TS's prompt and want to switch session #6, you'll simply do this:

TermServer#6

If you type S6 again, the Router will try to create a new session 6, **but it won't be able to** because that session already exists! So you'll get the following error message:

TermServer#S6

Trying R6 (2.2.2.2, 2038)...

% Connection refused by remote host

In other words, the Router won't able to connect to Port 2038, because there is another session already using that same Port!

That's it! If you have any questions, please write to us at:

sales@ciscoland.net