

Chapter 9

Healing the Platforms

Everyone has a doctor in him or her; we just have to help it in its work. The natural healing force within each one of us is the greatest force in getting well. Our food should be our medicine. Our medicine should be our food. But to eat when you are sick is to feed your sickness.

—Hippocrates (460 BC–377 BC)

Fixing problems with computers is big business and expensive. Because of the many possible problems a computer can run into, getting the right people and knowhow at the right place at the right time can be a problem. The costs increase when one also includes expense of the downtime of an employee. Intel® Active Management Technology (Intel AMT) makes it easier for a wider array of problems to be fixed remotely.

In general, organizations who want to perform remote repairs should start by installing software solutions that specialize in doing just that. Often this type of software will allow a remote administrator to see and even take control of the user's computer remotely, upload files, download files, apply patches, and much more. Intel AMT does not replace such software; it enhances it.

Repair software operates within the operating system it is trying to fix. This is okay if the problem is minor, but is of no help when the operating system itself is the problem, or worse, the problem is completely outside the operating system, such as a wrong BIOS setting that boots up the wrong hard disk. This is where Intel AMT becomes essential.

In this chapter we cover how Intel AMT can assist in making remote repairs. The two main features we discuss are remote IDE and Serial-over-LAN. We show how these two features can be used separately or together to remotely diagnose and repair the systems.

Remoted IDE (IDE-R)

Simply put, this feature allows a remote administrator to install or “mount” a virtual disk drive on an Intel AMT computer and optionally instruct the computer to boot into this remote drive. It could be argued that IDE-R is the most powerful feature of Intel AMT, since it allows a remote administrator to take over a computer completely, regardless of its state. Before getting into what can be done with IDE-R, it is necessary to talk about how IDE-R works in detail to better understand its power and limitations.

Intel AMT IDE-R is a simple switch that can be turned on and off. By default it is off, but when turned on, Intel AMT creates two new plug-and-play (PnP) virtual devices that are visible by the operating system. These two devices are a standard floppy disk drive and a standard CD-ROM drive, as shown in Figure 9.1.

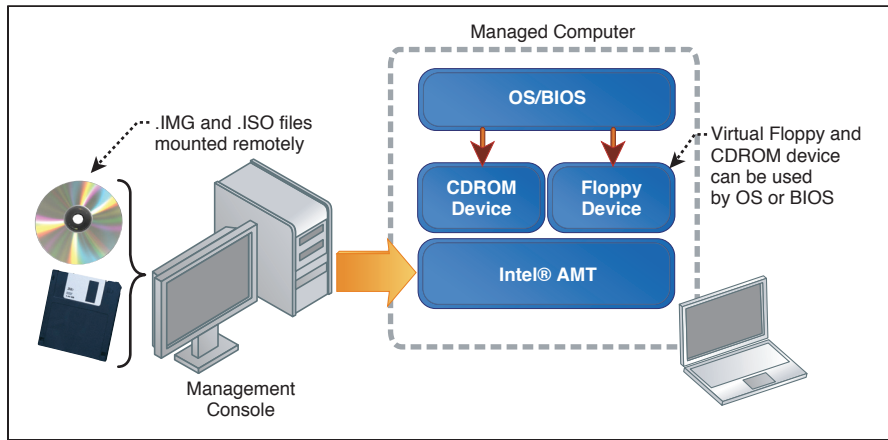


Figure 9.1 IDE-Redirection in Intel® AMT

The virtual floppy device, within Microsoft DOS or Microsoft Windows, will be visible as A: or if such a drive already exists B:. The floppy disk device is generally a read/write device, but may be read only in some cases. This device is generally 1.44 megabytes in size, but by using a small trick, it can be made much larger, up to 1 gigabyte or more.

The virtual CD-ROM device in Microsoft operating systems takes on the next available drive letter (for example, E: or F:). This device is always read only and has a maximum size of a standard CD-ROM, about 700 megabytes.

When IDE-R is activated, Intel AMT will present both of these new PnP devices to the operating system. There is no way to present just one or the other; both devices are either enabled or disabled even if the administrator will only be using one of them. Also, once activated, all IDE operations on both these devices are relayed through the network to the management console. The management console will then handle all of the IDE commands just like a hard disk does when it's connected to an IDE or SATA port within a computer.

Once enabled, a user can go to the device manager in Microsoft Windows and click on “Scan Plug-and-Plug devices” to have both devices show up in the Microsoft Windows file explorer, at which point both drives work like any other drive. As we will see later, the administrator can force a reboot of the computer with Intel AMT with a forced boot onto either one of the IDE-R devices.

IDE-R Protocol

Unlike most of the usual communications between a management console and Intel AMT, which use HTTP requests and responses on TCP port 16992 (or 16994 if TLS is used), IDE-R uses a proprietary binary protocol on port 16993 (or 16995 if TLS is used). The details of the protocol are not published by Intel, but Intel does provide a library called “IMRSDK” that implements the binary protocol and all its functionality.

Only one management console can connect and activate an IDE-R session at any given time and the binary protocol takes care of authentication and possibly encryption if TLS is used. The same credentials used for other Intel AMT operations are also used for IDE-R. It's also important to note that unlike HTTP, when using a username and password with IDE-R, both the username and password are sent in the clear over the network unless TLS is

used. As a result of this, it is highly recommended that Intel AMT be used in TLS mode before an IDE-R session be attempted.

IDE-R Speed

On a fast network, both the IDE-R floppy and CD-ROM will operate at about CDRom 2x to 4x speeds or about 300 to 600kb/sec. These numbers vary depending on the network and Intel AMT version. Because IDE-R has many round trips, the speed of IDE-R goes down quickly as network latency goes up. Using IDE-R over a slower network connection such as a DSL modem could cause a remote OS boot to go from a few minutes on a fast network to hours. This is because while IDE-R is very powerful, it's also very simple and not compressed or optimized for slow network connections.

Booting a Recovery OS

A recovery OS is an operating system that is intended to only be booted up when there is a problem with the computer. Many recovery operating systems are available online based on Linux, and more recently Microsoft has started offering Microsoft WinPE (Microsoft Windows Pre-boot Environment) and Microsoft WinRE (Microsoft Windows Recovery Environment). All of these operating systems are lightweight and can be booted up remotely using IDE-R.

Often administrators will customize their own version of a recovery OS to automatically call back home and run a set of diagnostics on the computer. Complete backup and restore solutions like Norton[†] Ghost can also be booted up using IDE-R. Administrators should customize recovery OS boot images so that when booted up, they perform a set of recovery operations automatically. If interactions are required between the recovery OS and the administrator, these can be handled in a limited way using serial-over-LAN covered in the following section.

Serial-over-LAN (SOL)

The serial-over-LAN port allows a management console to send and receive data over the network to a virtual serial port on the computer with Intel AMT. Serial-over-LAN is one of the most interesting features of Intel AMT. This section describes what this serial port is and how it works.

When purchasing a computer with Intel AMT, users will notice that the computer comes with an extra serial port, as shown in Figure 9.2. This port is always enabled and there is no external connector for it.

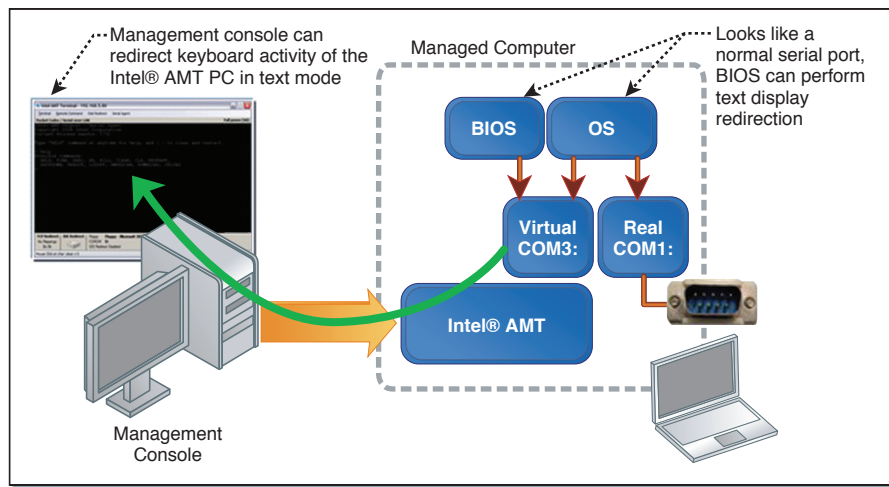


Figure 9.2 Serial-over-LAN (SOL)

In Microsoft Windows, this extra COM port requires a separate device driver. Actually, it is a small .INF file that instructs Microsoft Windows to use its own standard serial driver for this PCI serial device, as shown in Figure 9.3. Outside of the OS, the BIOS can use this port and no driver is required.

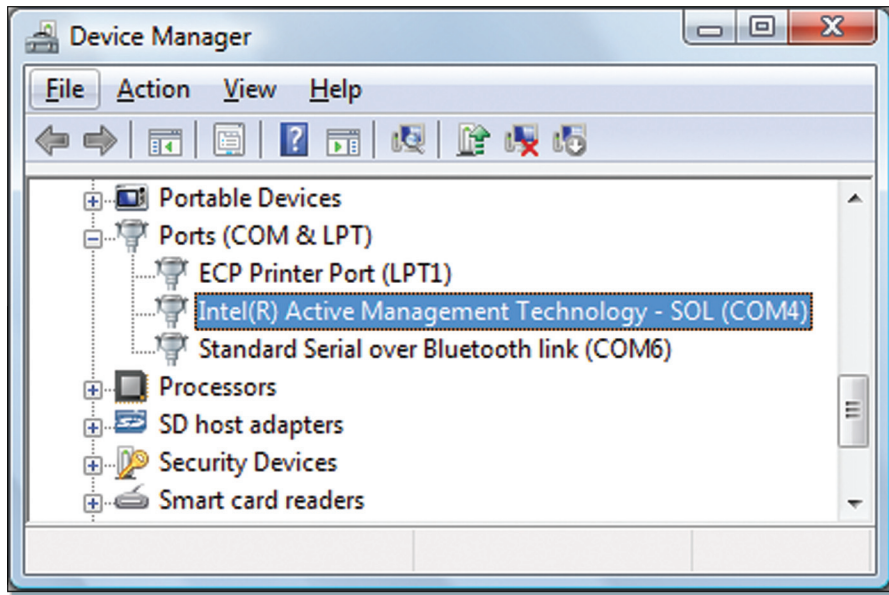


Figure 9.3 The SOL Device Driver in Microsoft Windows

Like any serial port, applications can open the port and send data to it. Intel AMT will forward the data to the connected management console. At most one management console can connect to the serial-over-LAN feature and if no console is connected sent data is simply ignored.

Serial-over-LAN Protocol

Much like IDE-R, Serial-over-LAN communication occurs using a proprietary binary protocol and with the same ports as IDE-R: 16993 or 16995 if TLS is used. Like IDE-R, specifics of this protocol are not public. A negotiation is performed at the start of the connection telling Intel AMT if this is going to be a serial-over-LAN or IDE-R session. At most one serial-over-LAN and one IDE-R session can occur, but they don't need to come from the same management console.

Like IDE-R, SOL is authenticated using the same username and password or Kerberos credentials used to authenticate with Intel AMT for

other operations. The same security warning also applies here: serial-over-LAN sessions that do not use TLS will send the username and password in plain text on the network. It is therefore recommended to always make sure TLS is in use before attempting to open a serial-over-LAN session.

Serial-over-LAN Speed

Serial-over-LAN is mostly limited by the speed of the serial port. The standard COM port setting for SOL is: “115200, N, 8, 1” or 115200 bits/sec, no parity, 8 bit and 1 stop bit. These settings are set in Intel AMT and can rarely be changed. Applications making use of this virtual port should make sure to use these settings.

With the exception of dial-up, 115kb/sec is much slower than most networks. Even if Intel AMT can handle much faster data rates, the serial port speed limits the data rate between Intel AMT and the BIOS or OS.

BIOS Using Serial-over-LAN

There are two possible users for this serial port, applications that run in the OS and the BIOS itself. On computers with Intel AMT, a serial-over-LAN redirection switch is present in the BIOS and is turned off by default. When turned on, the BIOS attempts to redirect text characters into the serial port and take characters received from the serial port and emulate keyboard input, as shown in Figure 9.4.

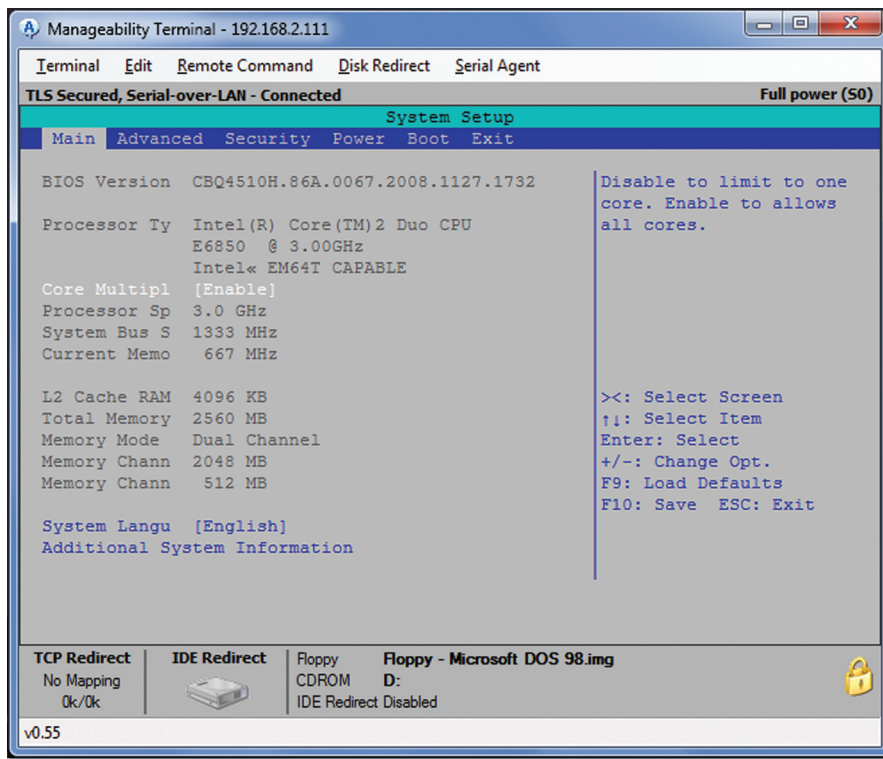


Figure 9.4 BIOS Text Display Redirection

This feature only works when the display is set to 80 by 25 text mode. This is the display mode that is used when first starting the computer and when entering the BIOS setup screens or booting up Microsoft DOS. The exact way the text display is captured and sent to the serial port varies from one BIOS vendor to another, but in general VT100 encoding is used to send cursor position and color attributes in the serial port. Some BIOS vendors have BIOS settings to allow the user to change how the display is encoded to the serial port.

We talked about the BIOS display redirection switch, but have not talked about how to turn it on or off. This switch is controlled by the RemoteControl method of the remote control EOI/SOAP service, and can only be turned on

upon remote reboot of the computer with Intel AMT. The switch is turned back off by default upon the next reboot.

Examples:

Remote reset without BIOS text redirection.

```
RemoteControl(0x10, 0x157, 0x00, 0x0000, 0x00, 0x00);
```

Remote reset with BIOS text redirection.

```
RemoteControl(0x10, 0x157, 0x00, 0x0000, 0x00, 0x01);
```

Remote reset to BIOS setup with text redirection.

```
RemoteControl(0x10, 0x157, 0xC1, 0x0008, 0x00, 0x01);
```

Remote reset to IDE-R floppy with text redirection.

```
RemoteControl(0x10, 0x157, 0xC1, 0x0001, 0x00, 0x01);
```

Remote reset to IDE-R CDROM with text redirection.

```
RemoteControl(0x10, 0x157, 0xC1, 0x0101, 0x00, 0x01);
```

As we can see from these examples, BIOS text redirection can be used in many situations. The last two examples combine using BIOS text redirection with remote IDE-R. This is especially interesting if the administrative console is remotely booting a Microsoft DOS floppy disk and wants the BIOS to redirect the command prompt back to the administrator, as shown in Figure 9.5.

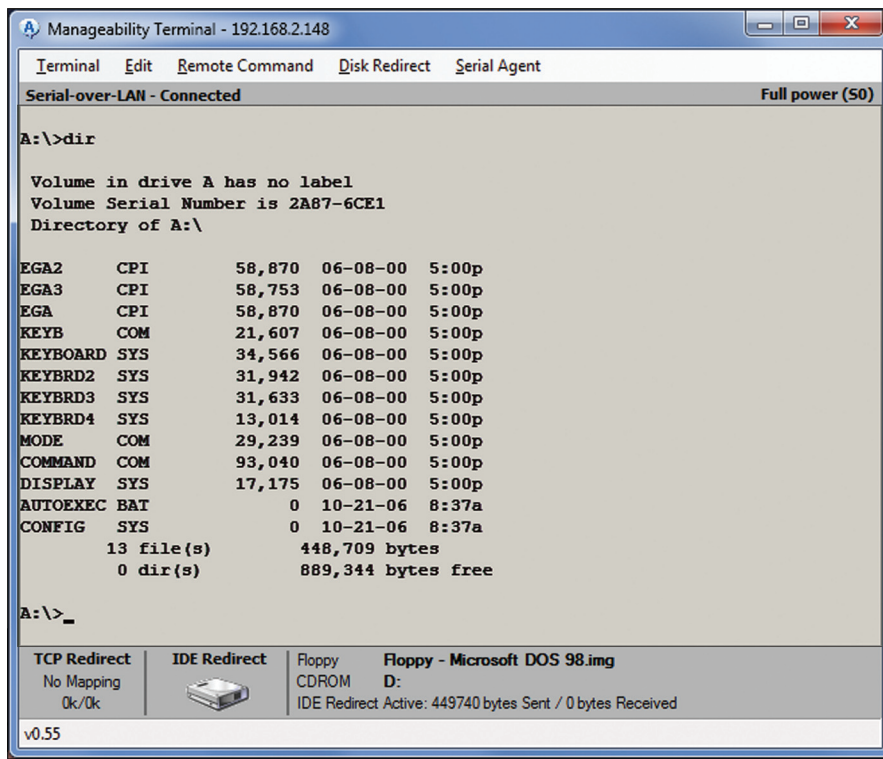


Figure 9.5 Redirected Microsoft MS-DOS Screen

OS Applications Using Serial-over-LAN

The second possible use of the serial-over-LAN port after the BIOS is OS level applications. Only one application may open the serial-over-LAN port at any given time and send data to the management console. It is generally accepted that such applications send VT100 encoded serial data just like the BIOS. This allows the remote administrator to use the same terminal software for both BIOS display redirection and controlling serial applications.

So why would an OS level application want to communicate with a management console using a slow serial port when network sockets would be much faster? Because this serial port bypasses the OS network stack and communicates with the management console using the Intel AMT network

stack instead. If the OS network stack is not working or completely disabled, this serial port will still work. Also, an application using this serial port can be assured to be communicating with an authorized administrator since Intel AMT will perform the authentication before letting a management console send and receive data to this serial port.

When running a variation of Linux, it is common for administrators to run a new shell and pipe the input and output to the serial-over-LAN port. This way, once the OS is running an administrator can log into a terminal session using serial-over-LAN even if the OS network stack is not working. In Microsoft Windows, software like Intel AMT Outpost that is part of the Manageability DTK will offer the administrator many remote services through the serial port.

One other possible use of this serial port is for kernel debuggers that can route debugging data to a serial port. Using Intel AMT as a serial port for this type of usage is beyond network management, but a perfectly acceptable usage.

When building an application that runs on an Intel AMT computer and binds to the Intel AMT SOL serial port, it is not possible to know when a management console connects or disconnects from the serial port.

Also, there is no way for an application to tell which Intel AMT user is currently connected to the serial port. If needed, applications running in the OS must perform their own access control over the serial port. All an application can be assured of is that one of the users authorized to connect to Intel AMT may be connected to the serial port, but you cannot know which one.

Building a Serial-over-LAN Terminal

We talked about how the BIOS and applications can send VT100 data to a management terminal using the Intel AMT serial port. This section covers the art of building and using a terminal on the console side. It is an art because so many things can and do go wrong.

First, the management console terminal must have a display size of 80 characters wide and 25 characters high. Terminals intended to be used with modems often have 24 characters high, which causes problems.

Second, different BIOS vendors have different key mappings for F1 to F12 keys. Since these keys are often used in the BIOS screens and the mapping varies, a terminal should allow the administrator to quickly change from one mapping to another.

Table 9.1 The Three Common Fx Key Mappings that Serial-over-LAN Terminals Should Support

Key	Mapping 1	Mapping 2	Mapping 3
F1	ESC + [+ O + P	ESC + 1	ESC + O + P
F2	ESC + [+ O + Q	ESC + 2	ESC + O + Q
F3	ESC + [+ O + w	ESC + 3	ESC + O + R
F4	ESC + [+ O + x	ESC + 4	ESC + O + S
F5	ESC + [+ O + t	ESC + 5	ESC + O + T
F6	ESC + [+ O + u	ESC + 6	ESC + O + U
F7	ESC + [+ O + q	ESC + 7	ESC + O + V
F8	ESC + [+ O + r	ESC + 8	ESC + O + W
F9	ESC + [+ O + p	ESC + 9	ESC + O + X
F10	ESC + [+ O + M	ESC + 0	ESC + O + Y
F11	Not Defined	ESC + !	ESC + O + Z
F12	Not Defined	ESC + @	ESC + O + [

This first mapping is often use by Intel BIOS and because F11 and F12 keys are not defined in this mapping, administrators can't start the boot device selection menu when booting; this would require the F12 key.

The second mapping is used by HP BIOS. It is interesting because most users can type hit the Escape key followed by 1 for F1, so explicit terminal support is not required but recommended.

The third mapping seems to be supported by many other BIOS vendors because freely available terminals such as Putty¹ support this mapping by default.

Because an administrator may be managing many computers from different vendors with different BIOS support, an ideal serial-over-LAN terminal should support all three mappings and allow switching between them.

¹ Putty is freely available for download at: <http://www.putty.org>.

Lastly, a good serial-over-LAN terminal must support all of the possible VT100 escape codes correctly. Not implementing this correctly will likely result in the screen not looking quite right in some situations.

Table 9.2 List of Common VT100 Escape Sequences

Sequence	Description
ESC + [+ 2 + J	Clear Screen
ESC + [+ (y) + ; + (x) + H	Move cursor to (x), (y) coordinates. Upper left of the screen is (1,1). X and Y should never be zero.
ESC + [+ (a1) + ; + (a2) + m	Set display attributes. 0 = Reset attributes 1 = Bright, 2 = Dim 7 = Reverse, 27 = No reverse 30 to 39 = Foreground colors 40 to 49 = Background colors 90 to 99 = Bright foreground colors 100 to 109 = Bright background colors
ESC + [+ (c) + K	c = 0 or omitted, start at cursor and erase rest of line. c = 1, erase from start of line until cursor c = 2, erase entire line
ESC + [+ h	Enable line wrap
ESC + [+ l	Disable line wrap

If a developer wants to use a standard terminal like Putty or Microsoft Windows XP HyperTerminal, these terminals can't connect directly to Intel AMT. The protocol is proprietary and authentication is required. Using these standard terminals is not recommended, but someone can use this by building a small serial relay application and using the raw TCP feature that both Putty and HyperTerminal support, as shown in Figure 9.6

```
A:\>dir

Volume in drive A has no label
Volume Serial Number is 2A87-6CE1
Directory of A:\

EGA2      CPI           58,870   06-08-00   5:00p
EGA3      CPI           58,753   06-08-00   5:00p
EGA       CPI           58,870   06-08-00   5:00p
KEYB      COM           21,607   06-08-00   5:00p
KEYBOARD  SYS           34,566   06-08-00   5:00p
KEYBRD2   SYS           31,942   06-08-00   5:00p
KEYBRD3   SYS           31,633   06-08-00   5:00p
KEYBRD4   SYS           13,014   06-08-00   5:00p
MODE      COM          29,239   06-08-00   5:00p
COMMAND   COM           93,040   06-08-00   5:00p
DISPLAY   SYS           17,175   06-08-00   5:00p
AUTOEXEC  BAT              0   10-21-06   8:37a
CONFIG    SYS              0   10-21-06   8:37a
          13 file(s)         448,709 bytes
          0 dir(s)         889,344 bytes free

A:\>
```

Figure 9.6 Using a Standard Terminal

Of course, the best way to go is to use a terminal that was built from the ground up to support serial-over-LAN, such as IAmTerm.exe that comes with source code in the Manageability DTK. Such a terminal also has the benefit of having support power state monitoring, remote reboot, and IDE redirection all on the same user interface.

Advanced Uses of Serial-over-LAN

Earlier in this chapter, we talked about how OS-resident applications could open and use the Intel AMT serial port to send and receive data from the administrative console and that it is generally recommended for applications to assume that the management console is running a VT100 terminal and so VT100 escape codes should be used.

If the same software developer builds console and agent software located on either side of the serial connection, extra nonstandard escape codes can be defined to greatly enhance the features offered by VT100. Of course, care should be taken to remain VT100-compatible in case one side or the other does not support these extra escape codes.

The Manageability DTK's terminal and agent software do exactly that to enhance the features supported over the serial port while remaining backward-compatible. Some of these features include uploading and downloading files, obtaining a list of running processes, starting and stopping processes, browsing the file system, enabling and disabling device drivers, and forwarding TCP connections over the serial port.

All of these extended features are very useful if the operating system's network stack is not working, or when you need to boot a recovery OS and want to make more tools available than the ones offered by a command prompt.

Probably the most powerful extended serial feature of the DTK is TCP connection forwarding over the serial port. This feature forwards a local console port to a remote port on the computer with Intel AMT, with the terminal and the agent acting as middle-men, relaying the data, as shown in Figure 9.7.

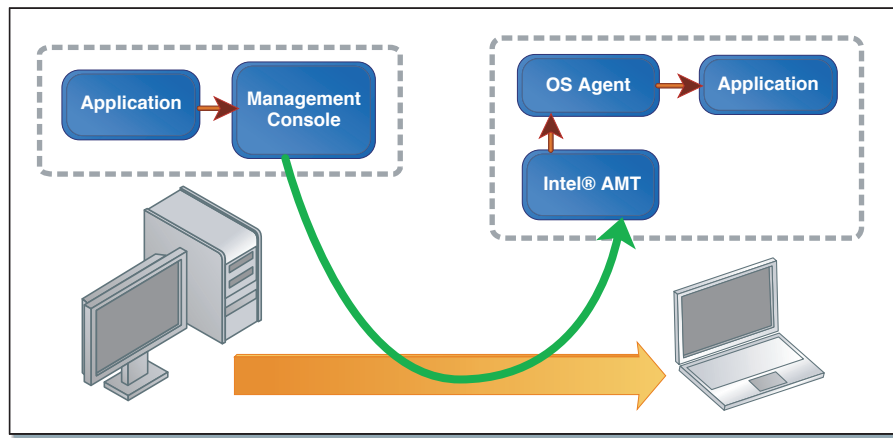


Figure 9.7 TCP-over-SOL, Using Intel® AMT Out-of-Band Channel to Carry TCP Traffic

Probably the most impressive demonstration of this feature involves performing a remote display session using VNC or Microsoft RDP over the serial port. The demonstration starts by disabling all of the network adapters on the computer with Intel AMT. Going to a command prompt and typing “IPCONFIG” proves that the operating system has no IP address. In reality,

the Intel AMT network stack is still running and a serial-over-LAN connection to the OS agent on the Intel AMT computer is still possible.

Using the TCP-over-SOL feature, the administrator can still take control over the computer's display and mouse remotely and even fix and re-enable the network adapter in the OS. Details on how to do this are explained in the Manageability DTK serial-over-LAN white paper and tutorial video.

Summary

This chapter covered IDE redirection and serial-over-LAN, two of the most powerful features of Intel AMT and two features that allow an administrator to take action over a remote computer in order to diagnose problems and fix them. IDE-R redirection allows an administrator to bring diagnostic and repair tools to a troubled computer, and serial-over-LAN allows the administrator to have control over these tools even if the OS network driver is not loaded or not functional. A proper combination of software and Intel AMT hardware support results in unprecedented new possibilities for remotely fixing problems that would otherwise have required hands-on access to the computer.