TECH EXCHANGE No. 4 – Suffering From Internet Buzzword Overload? (part 1)

When reading reviews of software and hardware in the computer related press you often come across a deluge of buzzwords and acronyms. You soon find yourself rubbing shoulders with **protocols, ports, HTML, XML, RSS, POP3** and other unfamiliar buzzwords. To the newcomer this can be pretty intimidating. The best approach is to take things one bite (or maybe that should be byte) at a time. This will also be a two-part article. This part describes the communication side of the Internet and the second part covers the technologies used within the web browser. First in reply to the emails on how to set a restore point in Windows XP - select the following options start button->All Programs->Accessories->System Tools->System Restore.

First lets get one definition under our belt. A **protocol** is a set of well-defined commands and responses, which allow the **client** (your computer) to communicate with the **server** (the computer at the other end of the communication line). A lot of the standard protocols started life in the **Unix** (an alternative operating system to windows) world - here the philosophy was that a program should just do one job and do it well. A task would be completed by using the output of one program as the input to another program and so on. This is part of the reason for the reliability of Unix systems, where the uptime of a computer is measured in days, rather then the hours of a windows system. Two of the main protocols you will come across are **HTTP** (Hypertext Transfer Protocol - i.e. web browsing) and **FTP** (File Transfer Protocol). These protocols run on top of the **TCP** (Transmission Control Protocol) which runs on top of the **IP** (Internet Protocol). You can get an Internet protocols roadmap poster from http://www.javvin.com.

When you receive email from your ISP's server, you are using the **POP3** (Post Office Protocol Version 3) to communicate with the POP3 mail server. Similarly when you send email you are using the **SMTP** (Simple Mail Transfer Protocol) to communicate with the SMTP server. For a more in depth treatment of these protocols refer to 'Data Communications, Computer Networks and Open Systems' by Fred Halsall published by Addison-Wesley. There are also many sources of reference to be found on the Internet.

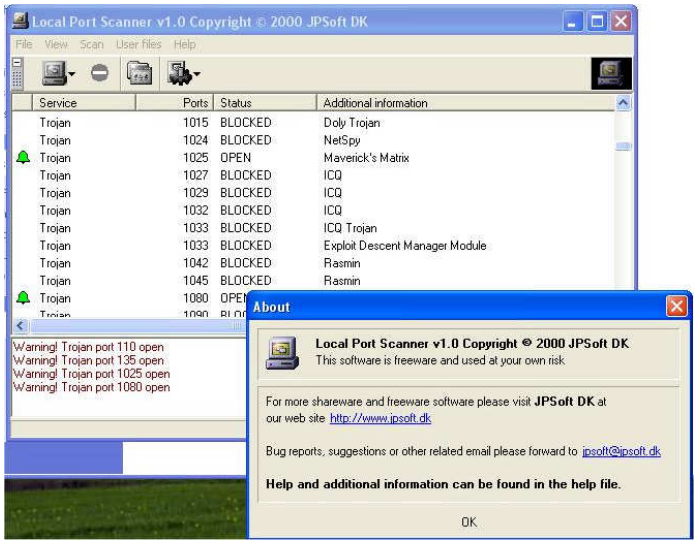
The **OSI** (Open System Interconnection) model is the blueprint for the Internet. It defines seven distinct layers and introduces the concept of protocol layers (or stack). Layer 1 is closest to the hardware and level 7 is closest to the application program that you are running (e.g. Internet Explorer). You can think of it like the layers of an onion. Keep in mind that the whole idea is to shift data from one computer to another. Each layer imposes its own format on the data packets it processes. Data is sent from the application program you are running down through each layer (7 to 1) and then transmitted over the communication line as **packets** (discrete blocks of data) to the server computer. There it travels up through each layer (1 to 7) on the server to its application program for processing. Data is then returned to the client in a similar fashion.

Layer 1 the physical Layer, defines the hardware connection. Next up is layer 2 known as the Data Link Layer. One of its jobs is to handle errors, which occur in the Physical Layer. This layer also determines how the computer gains access to the physical network and transmits/receives data. Layer 3 the Network Layer, is responsible for assembling the data packets in the correct order. This layer implements the Internet protocol (IP). Layer 4 the Transport Layer, implements the **TCP** (Transport Control Protocol) and **UDP** (User Datagram Protocol). The last three layers are often grouped together and are known as the Session, Presentation and Application Layers respectively. This is where the higher level protocols (FTP, POP3, SMTP, etc.) reside, as they rely on the functionality provided by the TCP and IP protocols. The grouping of the top three layers together results in an alternative to the OSI 7-layer model known as the TCP/IP Protocol Suite.

TCP is a connection-orientated protocol. This means a virtual connection is made for the duration of the conversation. Errors are flagged and the data packets are guaranteed to be delivered unless there is a hardware error. The downside is the need to add routing control data to the packet increasing the network overheads and hence the transmission time. On the other hand the UDP protocol does not use connections and is not reliable. You might never know you have not received a data packet. So what use is it? One main advantage is the lack of network overheads meaning it's more suitable to real-time applications like chatting. For the inquisitive and

technically minded, take a look at the PowerPoint presentations at http://www.si.umich.edu/classes/540/. In the Windows operating system the above is implemented in the WinSock (Windows Sockets) subsystem and can be found in **WSOCK32.DLL** or later **WS2_32.DLL** in your windows system directory.

For the Internet protocols that are connection oriented (see above) the logical connections at either end are defined as **sockets** (think of an electrical socket). A socket consists of two parts, the Internet address of the host machine on which it resides (for example **56.87.215.8**) and the **port** number. The port numbers are part of a pre-defined standard, for example, **port 25 is SMTP, port 110 is POP3 and port 80 is HTTP**. Port numbers make it possible to have multiple servers on the same machine. Your web browser will open and listen at port 80 for web pages you request. In a default configuration some ports are left open (enabled) that you do not normally use. It is advisable to close all unused ports via the Windows Security Centre, which can be run from the 'Control Panel'. Select the 'Windows Firewall' link and the 'Exceptions' tab (see my earlier article). This prevents virus entering your system via this method. Also download a copy of 'Local Port Scanner' from http://www.jpsoft.dk, it's also free to use (see screen shot).



This week's recommended site is jEdit (http://www.jedit.org). jEdit is a programmer's text editor written in Java, being developed by Slava Pestov and others. It has an easy to use interface that resembles that of many other Windows and MacOS text editors. It is also highly customisable, and contains a "plugin" architecture that allows its features to be extended by additional programs. That's it for this week, please send any questions or comments to techexchange@endhousesoftware.com.

By Gavin Baker.