Frequently Asked Questions

1. Q: What is the Network Data Tunnel?
   A: Network Data Tunnel (NDT) is a software-based solution that accelerates data transfer in point-to-point or point-to-multipoint network architectures. NDT uses a patent-pending combination of User Diagram Protocol (UDP) and Transmission Control Protocol (TCP) to accelerate data transfer. NDT leverages its proprietary technology to overcome performance limitations of standard TCP-based data transfer. NDT also provides enhanced delivery services including real-time payload encryption, compression and verification. The NDT software is plug and play – you can install it directly into your existing network without the need for additional appliances, devices or hardware to be able to deploy the solution. The result is fast data transfer between remote locations with exceptionally high throughput, even areas where network latency and packet loss previously existed. In addition, because of its software-based nature, it can leverage cloud and mobile applications quickly and effectively.

2. Q: Is NDT a software or a hardware solution?
   A: NDT is an entirely software-based data transfer acceleration solution.

3. Q: How does NDT work?
   A: When a protocol such as TCP is used to transfer data, the data transfer depends on several parameters, such as the TCP window size and the round trip latency. These parameters can be used to compute the maximum data throughput between any two points. The throughput is independent on the actual bandwidth. For example, consider an environment that uses a 1 Gbps fiber network link from San Francisco to Singapore, with a round trip latency of 200 milliseconds. If we attempt to transfer a large data file from a server in San Francisco to a server in Singapore, we can estimate the best effective data throughput that can be achieved. To estimate this, we first convert the TCP window size from bytes to bits. In this instance we are using the standard 64KB TCP window size of a Windows machine, which can be expressed as 65,536 Bytes or 65,536 * 8 = 524,288 bits. If the latency is 200 milliseconds, then the maximum data throughput is

   542,288 bits / 0.200 sec = 2,621,440 bits / sec = 2,621,440 / (1,024*1,024) Mbps = 2.5 Mbps

This means that, even though the fiber network link may support a 1 Gbps bandwidth, we would be unable to utilize more than 2.5 Mbps of that bandwidth. In this example, the effective bandwidth is reduced by a factor of approximately 400.

The TCP window size is often controlled by the operating system. In order to reduce the window size, another protocol, such as UDP, can be used, which, unlike TCP, does not require acknowledgement of packets received. In accordance with an embodiment, UDP can be used to build a data protocol that resembles TCP in every other aspect, but utilizes a different window size, wherein the window size can be made adjustable.

Another technique that is utilized to increase data throughput is to use multiple TCP connections. TCP congestion can limit the number of multiple data connections that can be used, i.e. the maximum number of TCP connections is limited by the data loss that is incurred as a result of data congestion. For example, 8 to 10 multiple TCP connections will typically speed up the data transfer considerably.

In addition to the techniques above, very often, lossless data compression is capable of reducing the amount of data that needs to be transferred. Lossless data compression allows original data to be exactly reconstructed from compressed data. Typically, a lossless compression algorithm generates a statistical model for the data, and then uses this model to map input data to bit sequences in such a way that more frequently encountered data will require smaller output than less frequent data sequences.

NDT utilizes lossless compression to reduce the amount of data that needs to be transferred.

4. Q: Why is the NDT UDP protocol better than regular FTP, HTTP, or other methods of data transfer?
A: The inherent characteristics of TCP make it highly susceptible to network latency and packet loss. Even on a relatively stable network, TCP throughput is always lower than the actual available line speed. For example, on a T3 network (45 Mbps) with packet loss of 0.1% and a delay of 10 ms, FTP transfers can peak at only 30 Mbps. In sharp contrast, NDT yields throughput of 44 Mbps- only slightly less than maximum available line speed. When network conditions deteriorate to 2% packet loss and a delay of 150 ms, FTP transfers can be expected to perform at 450 Kbps, or

1% of the actual available bandwidth. NDT maintains its 44Mbps throughput.

NDT's proprietary UDP-based protocol is a highly efficient retransmission and congestion control mechanism that adds a reliability layer to UDP. The flow of data can achieve full line speed with an amazingly low 0.25% overhead. File transfer solutions that use UDP for transport lack the speed and scalability of NDT. Due to the nature of UDP itself, packet transmission is not inherently guaranteed. All UDP-based solutions must track individual data packets to ensure delivery. If a packet is lost, a message must be sent back to the sender to request retransmission. If this process is not handled efficiently, the performance will degrade substantially when packet loss is high. NDT's patent-pending technology ensures a constant flow of data, has built-in flags to ensure data is received and is always retransmitting concurrently with new data.

5. Q: Do I have to use NDT's client?
A: For the pure client, Yes. Because NDT is background software that will create a tunnel, you will need two points, a "forwarder" and a "receiver", to create the tunnel.

   For the NDT File Transfer, No. NDT comes packaged with versions of FileZilla Client & Server for users that would like to transfer files. Once the tunnel is created, users can use their own data transfer software as long as it can route its data through a SOCKS 5 proxy.

   NDT is literally plug and play. You can install NDT directly into your existing networks and use NDT with your existing scripts and processes. Installation, configuration and deployment is transparent, with no disruption to your current networking activities. While While Nuvel creates and enables the tunnel, it is important to note that Nuvel does not initiate any data transfer.

6. Q: How does "on-the-fly" compression help NDT?
A: On-the-fly compression allows digital files to be reduced in size as they are sent. What differentiates NDT is that compression occurs as the file is being transferred, saving precious preparation and restoration time. As the files reach the recipient, they are decompressed and automatically restored to their original formats. The time saved in preparation and restoration is maximized especially when thousands or more files are involved. Standard compression techniques interrupt the data flow, adding to the total time and making the file transfer appear slower. NDT compresses the data it sends over the network in real time, meaning there is no

time-consuming compression or decompression at the beginning or end of each transfer.

7. Q: What operating systems does NDT support?
A: Clients that use NDT can run on any operating system including Windows, Linux, Mac OS X, iOS and Android. NDT only requires that client software be capable of routing data via a SOCKS 5 proxy. The NDT server supports both Windows and Linux operating systems. For Windows, we recommend the use of Windows Server 2008 or 7, and for Linux, we recommend the Ubuntu 12 distribution.

8. Q: Can NDT work through a corporate proxy server?
A: Yes. The NDT server can be configured to allow you to create a data tunnel through a corporate proxy server. This should not be confused with the proxy mechanism that the NDT server uses to serve data transfer clients.

9. Q: How can I determine the latency of a connection?
A: Packet latency is the elapsed time required for a packet sent over a network to be received at the destination. A good method to measure latency between two points over the Internet is to use a common computer utility called ping. The ping command will report the round-trip-time (RTT) between any two IP addresses measured in milliseconds. Latency is approximately half the ping time measured.

10. Q: Can I use NDT for streaming content?
A: NDT can be used to accelerate data transfer for streaming content that relies on TCP-based protocols such as HTTP.

11. Q: Can NDT be used with my current broadband connection?
A: NDT allows users to choose between the traditional NDT UDP Protocol, which is optimized for maximum acceleration over high-bandwidth connections (greater than 5 Mb/s), and a multiplex TCP protocol optimized for slower connections (1 to 5 Mb/s). The multiplex TCP setting is for users with ADSL, cable modem, and T1/E1 connections, while the traditional UDP setting applies to users with Fast Ethernet, Gigabit Ethernet, DS-3/E3, OC-3/STM-1, or OC-12/STM-4 connections.

12. Q: How can I find out what my network connection's upload/download speeds are?
A: If you are connected to the Internet, you can get an approximation for your network connection's speeds using some common speed test web sites, like Speedtest.net. However, these results will only be an approximation and will vary if other users are also on your network. For a complete end-to-end test and for non-Internet-connected networks (LANs, private networks, etc.), you can use a common tool like **Iperf**. Note that your upload and download speeds may not be the same. The throughput you obtain with NDT will depend on the smaller of (a) your network connection and (b) the network connection at the far end of the link.

13. Q: Should I encrypt files before sending them using NDT?
A: NDT provides real-time payload encryption of content, thus alleviating the need to pre- or post-process the file.

14. Q: Should I compress my files before sending them over NDT?
A: NDT provides the option to compress files for data transfer. The performance benefits of using compression will depend on the type of data transmitted. Large log files, documents, database backups, mainframe extracts, and text data are highly suitable for compression in transit. Non-compressed images may also be compressed in real-time by NDT.

15. Q: What are the firewall requirements?
A: For the NDT server receiver, the port requirements are: 1080 (out) and 9080 (in). For the NDT client forwarder, the port requirements are: TCP port 1080, TCP Port 9080 to 90xx (depending on the number of connections), UDP Port 9080 to 90xx (depending on the concurrent number of connections) for in/out.