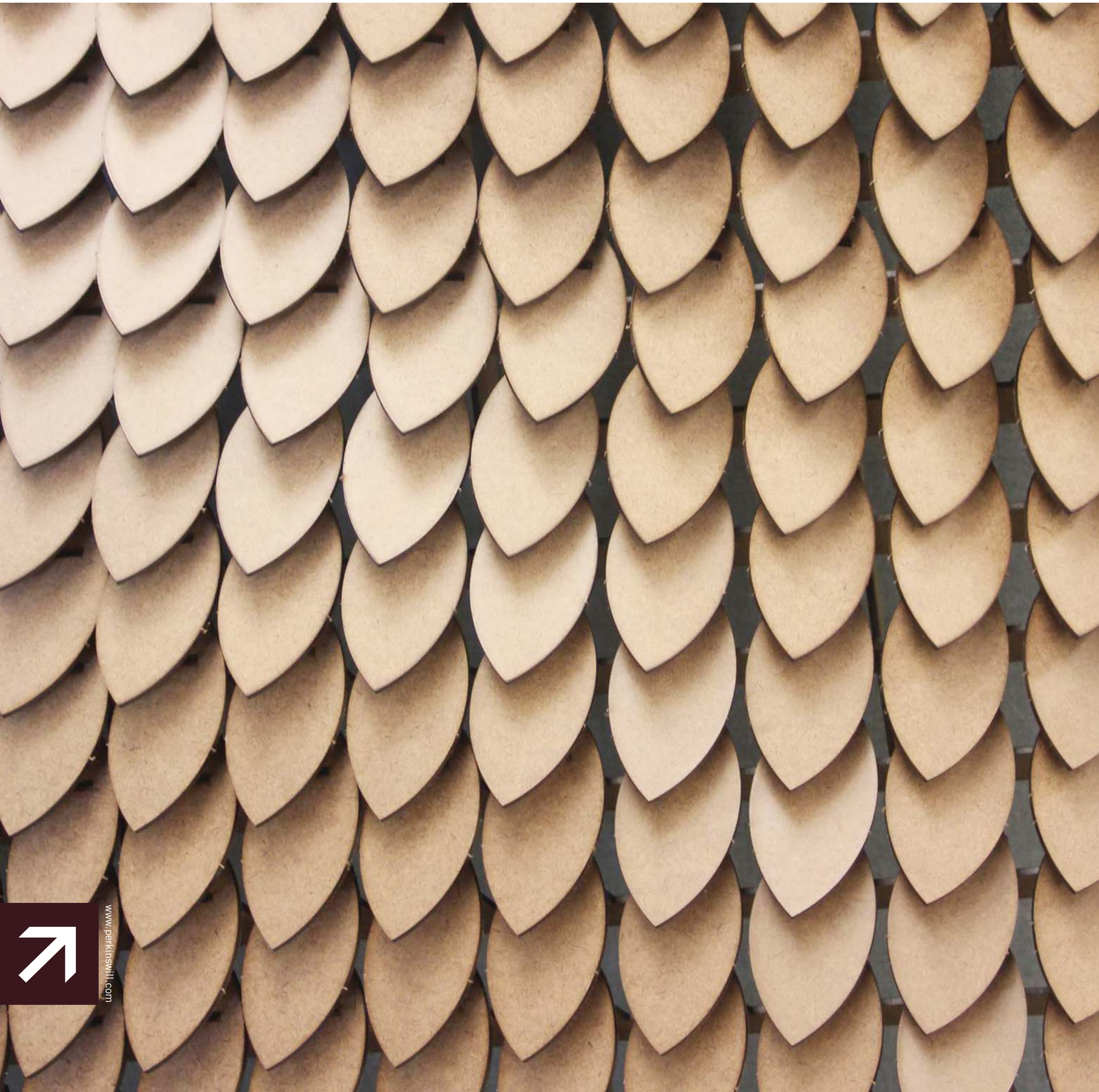
PERKINS+WILL

Research Journal

2011 / VOL 03.02



www.perkinswill.com

05.

BIM ON THE WAN:

Autodesk's Revit and the Wide Area Design Problem

Victor Okhoya, Associate AIA, *victor.okhoya@perkinswill.com*

ABSTRACT

As architects computerize their design practices they are faced with unique networking challenges. One of these is the "wide area design" problem. This is the problem of collaborating on large architectural projects from geographically dispersed locations. Revit, in particular, presents acute collaboration challenges owing to its large monolithic file sizes and its rigid synchronous database architecture.

In this article we describe the unique networking challenges faced by design professionals. We explain the factors that impact network performance and using these factors, we describe and evaluate two possible solutions to the wide area design problem currently under consideration at Perkins+Will – Remote Desktop and Revit Server.

We end with some testing results of Revit WAN performance using both Remote Desktop and Revit Server.

KEYWORDS: networking, WAN, collaboration, remote desktop, revit server

1.0 INTRODUCTION

In the last ten years a revolution has been gathering momentum in the realm of building design practice with the introduction of building information modeling software applications. At the same time, the "network effect" has been sweeping through both the consumer and business worlds. Cheaper access to computing resources and advancement in network speeds and technologies mean that today, business processes are, to a great extent, conducted on-line, whether through the cloud or by software as a service or through remote access to computing resources.

This growing reliance on, and expectation of, remote computing access has presented a special challenge to the design community and architects in particular. Architectural software has tended to lag in terms of general development due to the specialized and niche market nature of architectural business needs and computer processing requirements tend to be steep because of the graphical nature of design. In fact, it is only in the last few years that online project collaboration on the same design model has begun to be a possibility for building designers.

As usual, the hype has tended to shadow the realities. Early attempts to solve the wide area network problem for designers were based on various wide area optimization strategies and were met with much enthusiasm. Technologies such as wide area network accelerators like Riverbed's Steelheads or wide area file services (WAFS) from Globalscape and Cisco were thought to be the panacea until it was realized that they are only part of the solution. Despite thousands of dollars of information technology (IT) investments, many leading architectural practices found that while these technologies improved IT performance, they did not solve the Revit collaboration issue in a satisfactory way. This is because of the unique character of architectural practice when compared to other industries that are serviced by IT infrastructure for which these technologies may have proved adequate.

This is why it is important for architectural IT practitioners to understand the underlying networking issues as they evaluate different alternatives to the wide area design problem. Unfortunately, despite the acute computational challenges faced by designers, many IT professionals do not understand design or design software. This means that there is less research into solutions and less reliable information to inform strategic decision making. Even when information is available it tends to be too technical, intended for academics or IT professionals or too distorted intended for marketing. This article is a contribution to bridging this information gap in a fashion that is accessible to the architectural IT practitioners.

The article is structured in this following manner: we begin by describing the unique wide area network challenges faced by architectural practices and then explain the factors that impact WAN performance. Using these as a base reference, we describe and evaluate the performance profile of Remote Desktop and Revit Server. We conclude with the results of a quantitative test of Revit performance over the WAN.

2.0 THE UNIQUE WAN CHALLENGE FOR ARCHITECTURAL PRACTICES

Although most architectural design practices in North America are now fully computerized, the digitization of the design process has come with some unique challenges. First, visualization of design data is by definition graphical and this means it commands large amounts of resources in terms of processing capabilities, memory and storage requirements. As an example, the recommended specification for Revit Architecture is dual core processors running at above 3.0 GHz clock cycles, 4GB of RAM and file sizes on large projects easily surpass 200MB. Three years ago these were considered top of the line specifications for standard laptop computers. At that time, it was difficult to run Revit projects on a laptop, whereas most other business needs were adequately catered for.

As a consequence, the network requirements for design applications are demanding on a local area network (Figure 1). Gigabit technology is recommended for adapters, Ethernet and switches. On a WAN it has not been conceivable to attempt collaborating synchronously on centralized Revit projects from dispersed locations until very recently.

The requirements of current business processes demand that designers must now conform to the need to share and access design data across large distances in real time. Large organizations with branch offices need to leverage resources across their different locations; projects need to be done collaboratively between different organizations with one organization providing specialized design expertise from a remote location and another providing the actual project management at the location of the project. These situations need design data to be shared concurrently over wide area networks and the internet in general and this has proved to be a steep challenge.

A second challenge unique to design applications is their cumbersome software architecture that has not been designed or implemented for optimum network deployment. Where most enterprise level applications

- Microsoft Windows 7 32-bit Enterprise, Ultimate, Professional, or Home Premium edition, Microsoft Windows Vista 32-bit (SP2 or later) Enterprise, Ultimate, Business, or Home Premium edition, or Microsoft Windows XP (SP2 or later) Professional or Home edition
- For Microsoft Windows 7 32-bit or Microsoft Windows Vista 32-bit: Intel Pentium 4 or AMD Athlon dual core processor, 3.0 GHz (or higher) with SSE2 technology
- For Microsoft Windows XP: Intel Pentium 4 or AMD Athlon dual core, 1.6 GHz (or higher) with SSE2 technology
- 4 GB RAM
- 5 GB free disk space
- 1,280 x 1,024 monitor with true color
- Display adapter capable of 24-bit color for basic graphics, 256 MB DirectX 10-capable graphics card with Shader Model 3 for advanced graphics Microsoft Internet Explorer 7.0 (or later)
- Microsoft Mouse-compliant pointing device
- Download or installation from DVD
- Internet connectivity for license registration

Figure 1: Revit 2012 recommended specifications from Autodesk.

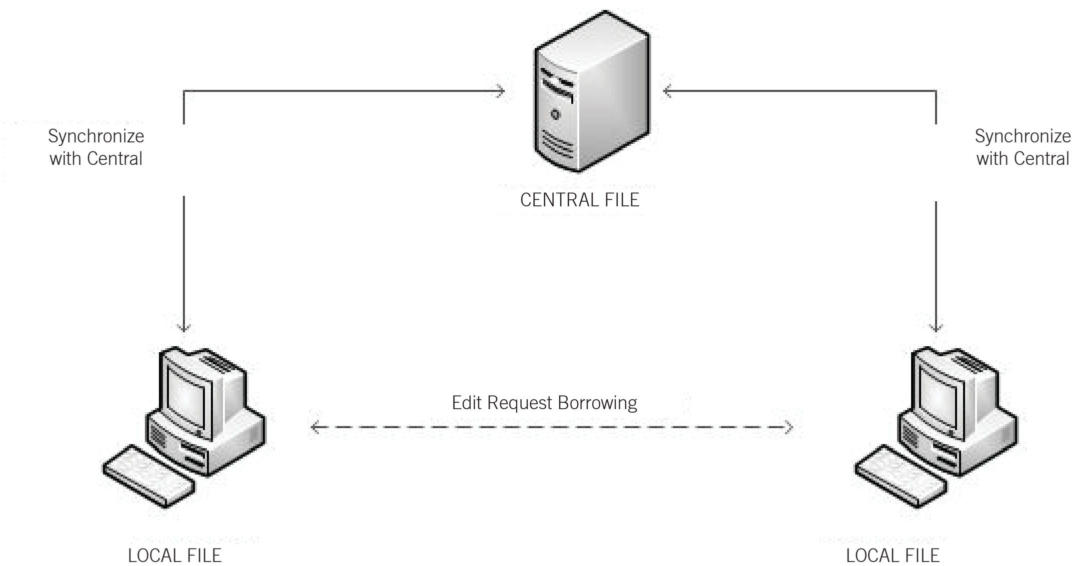


Figure 2: Revit collaboration architecture over a LAN

have boasted distributed software architecture that is easy to deploy on networks large and small, architectural design software has tended towards desktop centric, file-based processing.

Revit, in particular, is built upon a monolithic, proprietary database format that generates large project files (typically over 200MB) and is doubly challenged by processing graphics as well as parametric database relationships. Revit's native solution to the project collaboration question is a two-tier, database replication architecture with a server-based central file and client-based local files that are exact duplicates of the central file linked over a network. The central file performs record level locking coordination and permits manually triggered synchronizations by the local files (Figure 2).

Although this central-local file approach may sound like a network-based solution, it suffers from the defect that the synchronization procedure transfers a large amount of data in one process call. It is true that the process is optimized to transmit only changes to the file since the last transmission. On large projects this can still be significant enough to last several seconds or a few minutes over slow connections.

Simple edits can also be affected by the record locking mechanisms due to the possibility of multiple parametric relationships with objects that are not being edited. Even though only one object may be chosen for editing by the user, Revit must check the edit state of several objects that may be in relationships with the edited object. This slows down performance and in the event of an edit lock to any of the related objects, Revit throws an exception that requires manual intervention through an edit request.

To add to this are some pragmatic factors working against the architect. Architectural firms tend to be comparatively small organizations and will typically not be capitalized to invest heavily in IT and network infrastructure. Design software is also a specialized niche market with relatively few vendors active in providing solutions to design professionals. This means that design applications are not highly optimized for underlying technologies and indeed IT infrastructure is not designed with design applications at the forefront of considerations. This makes it harder to deploy such applications in environments like the internet where the long range network capabilities are still far behind the processing capability of the desktop or the LAN.

For Revit, this has meant that projects have had to be undertaken by teams at a single location. If at all a project was to be attempted by teams at more than one location, then the very project had to be split into more than one chunk and these chunks treated as separate projects and only assembled into a single whole periodically for coordination purposes. Needless to say, this approach is suboptimal and defeats the very reason for a single file database solution.

However, collaborating synchronously over a WAN is far from easy and has been virtually impossible without network enhancements like WAN optimizations and technologies that build on these optimizations such as Remote Desktop and Revit Server. To understand why and to better understand the strategies for solving the problem, we first need to understand the factors that impact network performance over the WAN.

3.0 WAN PERFORMANCE FACTORS

The following factors directly impact the performance of applications being deployed over a WAN: bandwidth, latency and throughput. In addition, protocols that are used by applications also contribute to WAN performance degradation. Finally, factors not inherent to the network like workstation hardware or operating systems also play a role, but we will not consider these factors in our analysis since they tend to hold constant across the WAN performance approach being considered[1,2].

3.1 Bandwidth

Bandwidth is given by the number of bits that can be transmitted over the network in a certain period of time (Figure 3). For example, a network might have 10Mbps bandwidth meaning it can deliver 10 million bits every second. In a LAN environment the available bandwidth is generally higher than the requirements of two communicating computers. Hence, remote desktop computing or thin clients is an attractive solution over today's high speed LANs.

In a WAN environment, however, points of over-subscription or points of aggregation are often encountered. These occur where several incoming links have to contend for fewer outgoing links through a switch or a router. This means that the switch or router must queue traffic, which causes delays. Further, WAN links will have different bandwidth capacities. Thus, several higher speed links may be in contention for a low speed link that only compounds the delay (Figure 3).

In addition, network protocols also introduce a significant amount of processing overhead, thus reducing the effective throughput of the transmissions. Transport protocols like TCP will add overhead in the form of segmenting, window management and acknowledgements. Network and data link protocols like IP and Ethernet add overhead due to packeting and framing. All these impact the effectiveness of WAN performance. This performance degradation is particularly acute for the designer who is typically transacting large volumes of data.

It is important to note that the 'b' in Mbps is a small 'b'. eight bits make a byte and, therefore, when one talks about streaming a 200 MB Revit file across a WAN connection, they are essentially speaking about streaming 1600 Mb or 1.6 Gb.

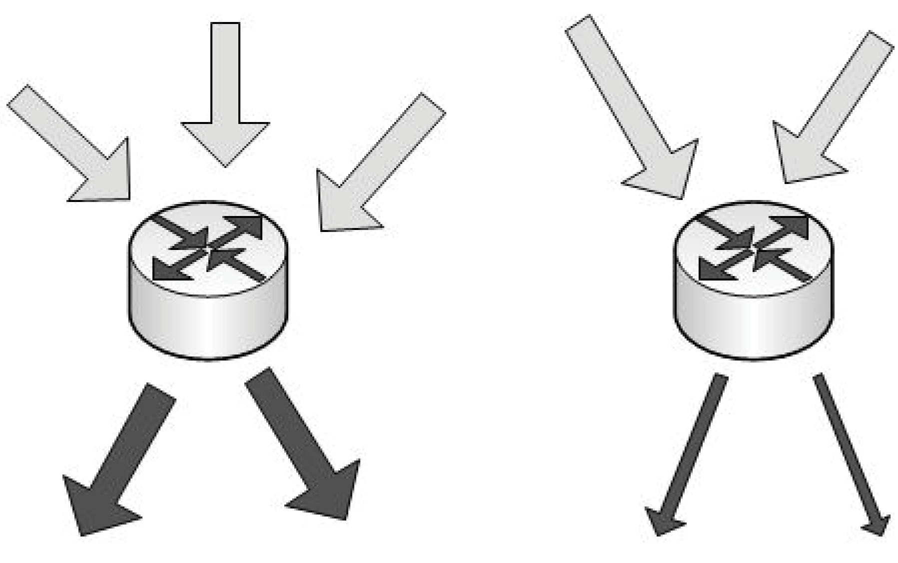


Figure 3: The bandwidth problem.

It is also worth noting that most non-specialists think of bandwidth when they think of network performance. After throwing hardware at a problem (i.e., buying new computers), the next step in solving performance issues is usually to try to increase bandwidth. Unfortunately, bandwidth is only one of a number of factors and taken by itself may not make very much of a difference. Both Remote Desktop and Revit Server act to mitigate bandwidth as a factor in network performance, but they do so in very different ways as we shall see.

3.2 Latency

Latency corresponds to how long it takes a transmission to travel from one end of a network to the other (Figure 4). Latency is measured in terms of time and could be one-way latency, the time taken from one end of a network to the other or it could be round trip time, which is the time to send a transmission to the other end of the network and back again.

Latency has three main components:

- Propagation: has to do with the fact that nothing travels faster than the speed of light
- Transmission: is a function of the bandwidth and packet size
- Queuing: packets need to be stored and processed in the network before transmission.

Apart from geographical distance apart, some factors impacting latency include serialization delays, which is the amount of time for a network device to extract data from one queue and package it onto the next network for transmission; processing delays, which are the amount of time spent within the network node such as router, switch or firewall, determining how to handle a piece of data based on set rules; and forwarding delays, which is the amount of time to determine where to forward a piece of data.

Another major factor affecting latency is the use of the Transmission Control Protocol (TCP). First, TCP must establish a connection, which involves the exchange of synchronization and acknowledgement responses. Second, TCP provides guaranteed service, which involves acknowledgment of successful receipt of data and a number of integrity checks, all of which increase the transmission delay.

Latency is one of the main factors impacting Remote Desktop and Revit Server. For Remote Desktop, even though a comparatively small amount of data is being transmitted across the WAN, this data still experiences latency related delays. In the case of Revit Server, high latency between the central server and the localized servers can impact the user experience for specific synchronization operations.

Other factors that affect network performance are throughput and the choice and design of network protocols. Throughput, or the net effective data transmission rate, is impacted by capacity, latency and packet loss. Protocols can act as a barrier to WAN performance if, for example, they were designed for a LAN environment and do not scale very well to the WAN. In this article we focus on bandwidth and latency as the major factors affecting WAN performance.

4.0 THE REMOTE DESKTOP PROTOCOL

Remote Desktop Protocol (RDP) is a proprietary protocol developed by Microsoft, which provides a user with a graphical interface to another computer. Formerly

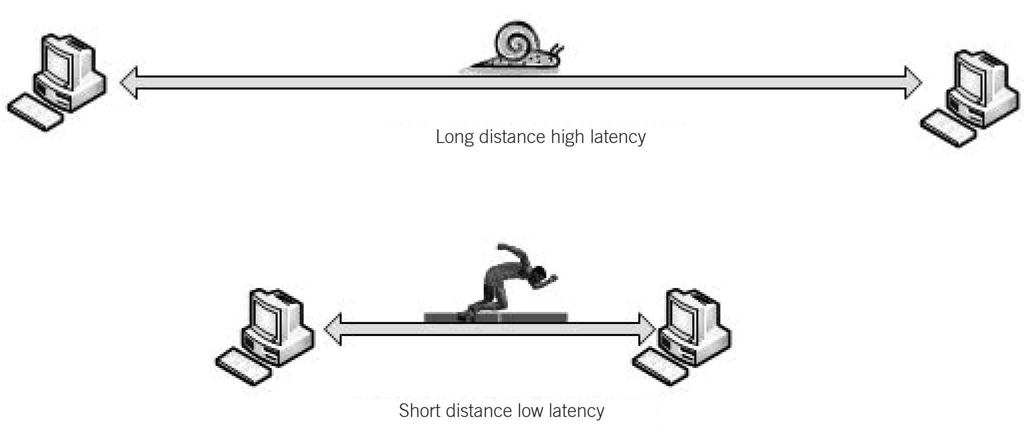


Figure 4: The latency problem.

known as Terminal Services, it is the protocol that drives Microsoft's version of a thin client solution.

In a Remote Desktop session, all computing is done on the remote computer (Figure 5). The local computer connects to the remote computer over the intervening network and only sends keyboard and mouse input over the network. The remote computer, in turn, sends back graphic display updates after processing the requested tasks. Application data does not stream over the network and this keeps bandwidth requirements low, thus enhancing performance.

In considering the performance of RDP as a solution to the Revit WAN collaboration issue, we need to ask two questions. First, is thin client computing truly a viable solution to the wide area design problem? Second, is RDP a good thin client solution for graphical applications?

Although a thin client solution effectively circumvents the WAN bandwidth problem by removing the need to stream application data, it does nothing about the latency problem since the reduced RDP signals still have to travel the full length of the network path from the client to the remote server.

This means that, regardless of the bandwidth, if a WAN has a high latency (say 1000 milliseconds), then every operation that would take a second on a local computer will appear to take two seconds using RDP over the WAN. This means tasks will appear to have a lag and depending on the severity of this lag, the end user experience can become intolerable. In reality, continental latency times fluctuate, but are in the order of 100 milliseconds in North America[3]. This value, though small, has the cumulative effect of making the remote application appear to run significantly slower to the end user.

The second question is whether Microsoft's Remote Desktop is actually a good thin client solution for graphic applications. According to Nieh and Yang, we can measure thin client applications according to four characteristics that influence their performance: display encoding, encoding compression, display update policy and client caching[4].

Display encoding refers to the basic data type used to transmit screen updates such as a pixel. Encoding compression is the type of compression applied to the graphic data before transmission. Update policy is the policy for determining when screen updates are sent from the remote computer to the client, while the client cache is a cache for display data types that then do not have to be resent from the remote computer.

Yang et al. performed tests to measure the performance of six popular thin-client platforms running over a wide range of network access bandwidths[5]. They studied the behavior of these platforms when downloading web pages as well as when streaming video. Since graphics rendering is an approximation of video rendering at lower frame refresh rates, we present their findings with respect to video performance. We also restrict the discussion to the 1.5 to 10 Mbps bandwidth range as being most representative of current WAN bandwidths that Revit users encounter.

The remote desktop clients that were tested included Citrix MetaFrame, Microsoft Remote Desktop (Terminal Services), AT&T Virtual Network Computing, Sun's Tarantella, Oracle's Sun Ray and Apple's X. According to

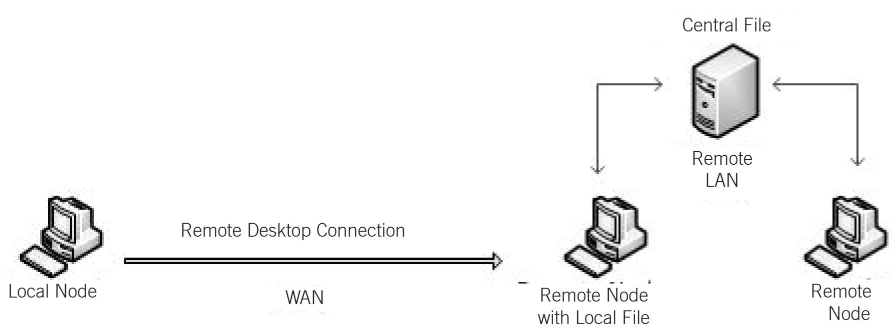


Figure 5: : Remote desktop architecture.

the research, when all platforms were tested with default settings, video quality in Citrix was better than RDP at 1.5 Mbps and Citrix, RDP and Tarantella were tied at 10 Mbps. However, when testing for other remote access display factors (by turning off capabilities and isolating the factor being tested), RDP did not stand out. For its display encoding system, RDP produced lower video quality than Tarantella and Sun Ray. For compression, VNC performed better while for caching, Tarantella performed better. This means that although RDP supports all the features of remote access display mentioned above, it does not provide the strongest implementation of these features for graphic related applications.

Before we leave our evaluation of RDP, it should be pointed out that despite any shortcomings it has the benefit of easy deployment and management as it is bundled with the Windows operating system and requires no special installation. While RDP is proprietary and will not function across some platforms, it still has the benefit, like other thin clients, of being broadly useful unlike Revit Server, which is purpose-built for Revit. Also RDP is ubiquitous, tried and tested technology while Revit Server, is not.

5.0 REVIT SERVER

Revit Server is Autodesk's solution to Revit's wide area design challenges. It comprises three main components: a central server, a local server and local files (Figure 6). The central server hosts the Revit central file for all locations on the WAN, the local server is a mirror of the central server at each physical location and the local files are mirrors of the local server at the end user's workstation.

Once all components are in place, end users periodically synchronize their local files to the local server. Once this synchronization is complete, the local server synchronizes the changes with the central server, which will then propagate them to other local servers and then down to end user local files at the different physical locations.

How does Revit Server deal with the WAN performance issues of bandwidth and latency? By introducing a local server, Revit Server essentially localizes the user experience to be the same as if they were working on a central file over their LAN. Except for specific operations noted below, Revit Server gets around the bandwidth and latency issues by making them transparent to the end user.

In principle, this sounds like a good fix. In practice, however, several Revit operations require not only communication between the local file and the local server, but also between the local server and the central server to complete. Depending on the frequency and complexity of these operations, the overall impact on both the network and the end user experience could be degrading. Some of these operations are discussed briefly[6].

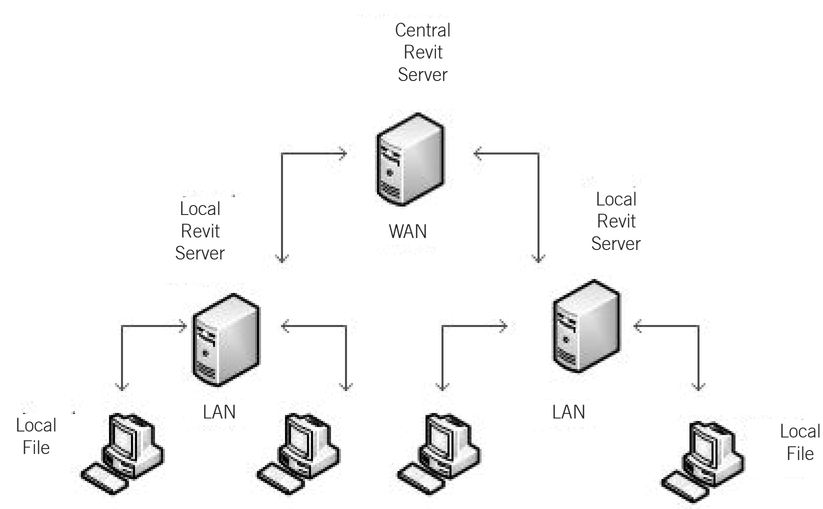


Figure 6: : Revit Server architecture.

Element borrowing requires local servers to communicate with the central server before they can grant permissions for an element to a local file. This is because permission states reside exclusively on the central server to prevent editing conflicts. On high latency networks, these permissions cannot be granted without the experience of a lag. In fact, on high latency networks, it is recommended that users explicitly check out worksets rather than rely on transparent element borrowing to avoid this permissions related lag.

Synchronizing to central by end users requires that changes are committed not only to the local server, but also to the central server for the operation to complete. On low bandwidth or high latency WANs, this can take significantly longer than synchronizing to central over a LAN. Moreover, a slow synchronize to central operation by a distant user will impact other users who cannot save to the locked central file. Teams, therefore, need to coordinate their synchronization times and this requires additional tools and management.

Depending on whether or not local server caches are up to date, the reload latest operation may also need to pull data down from the central server and in this case, the operation will slow down on high latency networks. This makes the operation unpredictable when using Revit Server whereas on a LAN, it is usually a fast one way data stream.

In addition, while localizing the user experience seems like a good way to get around the performance issues of bandwidth and latency, in the case of Revit Server this comes with substantial cost in administrative complexity[7]. Revit Server must be installed on Windows Server 2008 or later and on 64-bit systems. Microsoft.NET framework 3.5 SP1 or later is required. Further, an administrative install and configuration of IIS 7.0 or later is required.

Installing the central and local servers requires configuration of the server firewalls to allow ICMP requests. Then the servers are installed and configured to run as services whenever the server boots up. Permissions should be established for the Revit Server Administrator on both the central and local servers. Finally, the Revit Server extension should be installed on the end user machine and they can proceed to create local files from their local server as they normally would.

In order to execute synchronizations with the central file, users will need to establish communications by connecting to the Revit Server. Also managers are able to perform basic management tasks by using the web based Revit Server Administrative Console.

Clearly there is much more to configure and manage than in the standard installation of Revit over a LAN. Since this is all done to run just one application and given the persistent latency issues we discussed above, these negatives must counterbalance what seems a giant positive of localizing the Revit WAN experience to LAN performance.

6.0 COMPARATIVE ANALYSIS

The following test results were taken to investigate the impact of the WAN on Revit processes when using RDP as well as Revit Server. The test was made over the Perkins+Will WAN between Vancouver with 10Mbps download and upload speed and Chicago with 45Mbps download and upload speed. All testing was done on Perkins+Will computational nodes to keep the hardware as close to uniform as possible. The Revit test file was chosen as a 200MB single file project that would represent an average sized project in most of Perkins+Will offices. The tasks were chosen as being representative of typical procedures that a user would undertake on a Revit project, but that would clock sufficient cycles to be measured for comparison. The testing methodology involved performing each measurement twice with a third measurement to resolve any large discrepancy.
In performing such tests a distinction should be made between processing performance and response performance. Processing performance has to do with how long the computer takes to perform tasks. Response performance, on the other hand, has to do with how smooth the user perceives interaction with the computer to be.

Response performance is hard to measure quantitatively since it could involve aspects such as one second additional delay in cursor response time or a slower screen refresh rate as examples. It could, nonetheless, cumulatively cause the user experience to be as intolerable as processing delays.

Four different scenarios were tested (Table 1):

1. Revit Server with the central server in Chicago, the local server in Vancouver and the central Revit file in Chicago.
2. Remote Desktop with the local client in Vancouver, remote computer in Chicago and the Revit file in Chicago.
3. Direct access over the WAN with the local computer in Vancouver and the Revit file in Chicago.

Table 1: Revit WAN test results in minutes.

Task	Revit Server	Remote Desktop on the WAN	Remote Desktop on the LAN	Direct access over the WAN
Open file	2:26:30	1:08:90	0:28:70	1:00:40
Open file cached	2:09:80	0:39:40	0:28:30	0:40:70
Select all 3D objects	0:28:20	0:30:40	0:24:40	0:23:50
Swap out exterior walls	0:37:50	0:39:90	0:35:40	0:35:10
Create group array	0:47:40	0:45:90	0:44:70	0:47:10
Save the file as new central	3:45:20	1:09:90	1:07:20	1:54:50
Synchronize to central	1:47:30	0:16:80	0:14:60	0:25:20

4. A control Remote Desktop scenario over a 1Gbps LAN with the local computer in Vancouver, the remote computer in Vancouver and the Revit file in Vancouver.

The tasks for the test were:
1. Open the file detached from central.
2. Open the file a second time detached from central to account for caching.
3. Select all modeled objects in a 3D view.
4. Select and swap 95 instances of exterior wall to a different type.
5. Create an array of 20 grouped room suites.
6. Save the file as a new central file.
7. Delete the suites created in 6 above and synchronize the file to central.

From the results, we can conclude that over a comparatively high performance WAN such as that between Vancouver and Chicago:

- For tasks involving network calls such as opening and saving files, Revit Server consistently performs poorer than Remote Desktop. It even performs poorer than direct access over an optimized WAN. For tasks that do not involve network calls, performance is about even.
- Display latency delays are not significant in measuring processing performance over the WAN. However, this is not to say that they are not a factor in the response performance that contributes to the overall user experience.
- The overall performance of Revit is still primarily dependent on the power of the desktop rather than the underlying network infrastructure. Regardless of the networking approach, in-process application tasks run at about the same clock speeds. It is only when application data needs to be transferred across the network that performance differences are observed.

These results are probably quite different for lower bandwidth WANs. Also, it is worth pointing out that these tests were done under single user conditions. Under multiple user conditions Revit Server's coordination of synchronizing to central may well provide performance enhancements. However, these considerations are outside the scope of this article.

7.0 CONCLUSION

In this article we have undertaken to explain the unique networking challenges faced by architects. We have explained the performance factors that affect a WAN. We then used these factors to describe the performance of two approaches to the wide area design problem - Remote Desktop and Revit Server. We ended with test results of Revit performance under specified conditions.

From our discussion and testing, it is clear that under WAN conditions that are becoming commonplace, 10Mbps and above, Remote Desktop provides faster processing performance than Revit Server. At these bandwidths, latency related processing lags are negligible. Also, only network related tasks are impacted; locally processed tasks are minimally impacted by the underlying networking approach.

If Revit Server has a place it may be in the event of high numbers of dispersed users working simultaneously, or under low bandwidth conditions such as to small office locations, or under high latency situations such as over a transcontinental link. Under these conditions, performance degradation of the other approaches may leave Revit Server as the best alternative by attrition. More testing is required to determine if this is the case.

REFERENCES

[1] Peterson, L. and Davie, B., (2007). *Computer Networks: A Systems Approach,* San Francisco, CA: Elsevier.

[2] Grevers, T. and Christner, J., (2007). *Application Acceleration and WAN Optimization Fundamentals*, Indianapolis, IN: Cisco Press.

[3] Juniper Networks, (2005). Accelerating Application Performance across the WAN, White Paper, Retrieved on 9/2011 from http://www-05.ibm.com/uk/juniper/pdf/accelerating_application_performance_acros_the_wan_whitepaper.pdf.

[4] Nieh, J., and Yang, S., (2000). "Measuring the Multimedia Performance of Server-Based Computing," *Proceedings of the 10th International Workshop on Network and Operating System Support for Digital Audio and Video*, Retrieved on 9/2011 from http://www.sigmm.org/archive/NOSSDAV/NOSSDAV00.pdf

[5] Yang, S., Nieh, J., Selssky, M., and Tiwari, N., (2002). "The Performance of Remote Display Mechanisms for Thin-Client Computing," *Proceedings of the 2002 Usenix Annual Technical Conference,* Retrieved on 9/2011 from www.usenix.org/event/usenix02/full_papers/yang/yang_html/.

[6] DC CADD, (2010). Autodesk Revit Server Installation, Configuration and Workflow, White Paper, Retrieved on 9/2011 from http://www.dccadd.com/papers/Revit%20Server%20White%20Paper.pdf.

[7] Autodesk, (2011). Use of Revit Server on High Latency Networks, White Paper, Retrieved on 9/2011 from http://de0qmbqba3hfm.cloudfront.net/attachments/53504/0.