



Forensic Protocol and Application Analysis

Health Care Organization

October, 2000

Prepared by
Jeffrey J. Sicuranza, Principle Consultant
Applied Methodologies, Inc.

This report is an unpublished work containing proprietary information. It is not to be disclosed in whole or in part without the express written authorization of Health Care Co. and Applied Methodologies, Inc.

AMI would like to thank the following HEALTH CARE CO. personnel for their assistance during this project

Names not listed.

Table of Contents

EXECUTIVE SUMMARY 1

INTRODUCTION 3

PROFESSIONAL SUMMARY 4

SYSTEM BEHAVIORAL ILLNESSES 8

TERMINAL SERVER ISSUES 8

SQL SERVER ISSUES 8

NETWORK INFRASTRUCTURE AND PROTOCOL ISSUES 9

THIN CLIENT ISSUES 10

McKesson/HBOC Pathways Homecare Application Issues 10

EXECUTIVE RECOMMENDATIONS 12

Tactical Approach 12

Strategic Approach 13

ANALYSIS DETAILS 16

Terminal Server Analysis 16

SQL Servers Analysis 21

Microsoft SQL Server 23

SQL TDS Analysis 24

Network Analysis 41

Thin Client Analysis 47

Pathways Analysis 48

APPENDICES: ANALYSIS STATISTICS, GRAPHS AND SUPPLEMENTAL INFORMATION .. 62

Appendix I - Terminal Server Information 62

Appendix II - SQL Server Information 62

Appendix III - Network Statistics and Graphs 62

Appendix IV - Thin Client Information 65

Appendix V - Pathway Application Information 65

Executive Summary

HEALTH CARE CO. is currently experiencing significant performance problems relating to its Pathways application system. This system is built on a myriad of components that provides access to 700 plus users to process claim and visit transactions. An impact in performance and reliability does have a profound impact on HEALTH CARE CO.'s revenue generating capabilities. In light of Newsday's 10/17/00 article of a HEALTH CARE CO. net loss in its second quarter, having a system currently performing in a less than optimal manner does not help HEALTH CARE CO. achieve optimal revenue returns in future quarters.

There is not any one component that is causing the system to respond in a sluggish manner but the culmination of many parts reaching their limits at this current scale. The Pathways application was never intentionally designed to support this many users. Since HEALTH CARE CO. is HBOC's largest customer this behavior is a natural progression of growth. The application was not designed to handle this volume of traffic and user activity as evident with the frequent runtime and time-out errors occurring during the use of many applications functions, some examples are Billing and Intake FastPad. Reporting takes over four to sixteen hours at times for reports to be processed. The user interface is inconsistent leading users to believe there is an application time-out just because of the lack of a system busy indicator in some functions. The application also has functions that can execute a "runaway" query that keeps the user's session busy for over five minutes and results in additional overhead on the server components. The application, based on the number of executable code modules, indicates that it was designed to run locally on PC or from a local server in each branch. However, with HEALTH CARE CO.'s user population this footprint would prove too costly of an approach. Since this application has limited distributed scaling capabilities the use of Terminal Servers and Thin Clients must be used to provide access to the application.

To mask the inherent problems with the Pathways application HEALTH CARE CO. must surround the application with server hardware, software and network bandwidth just to provide a basic level of access to the end users. This is a classic example of architecting around an application as opposed to the preferable approach of the application, network and server components scaling in tandem.

Since the application uses several protocols to communicate (one efficient and one a legacy based protocol that is inefficient) with servers, and requires the loading and unloading of many separate application modules, the application is very difficult to scale. Also, the application was written in a programming language mostly suited to prototyping and smaller niche applications thus making it inflexible. The application requires the use of a standard LAN and departmental level grade server and operating system platform to distribute it's functions to end users. However, the volume metrics of transactions noted currently shows that HEALTH CARE CO. is experiencing mission critical level volume. HEALTH CARE CO.'s claim/visit volume is mission critical caliber but the Pathways application system does not scale to that level.

The application does provide the business logic and functionality necessary for HEALTH CARE CO. to function and be competitive in their industry. The users like the application and it has helped HEALTH CARE CO. to manage its business. However, this functionality combined with the transaction volume and growth scale plus the inherent system's platform liabilities is causing a ceiling of acceptable operating behavior to be reached much sooner than expected.

The Terminal Servers are experiencing high levels of utilization from the execution of multiple copies of Pathways and this results in bottlenecks at these servers thus, resulting in the users poor response time. The SQL servers are over burdened with disk activity and database operations. The network component is not an issue and adding any additional bandwidth cannot help mask these issues. The server and operating system components are not rated for mission critical level reliability and flexibility. Frequent hard disk problems have plagued the replication

server during AMI's visit causing a system process to be turned off and handled manually during non-production periods.

HEALTH CARE CO. is currently in a quandary, how to scale as the transaction volume grows. For this reason HEALTH CARE CO. must continually scale horizontally, by adding more Terminal and SQL servers, and scale vertically by increasing the CPU, BUS and DISK I/O performance for each server to mask the application's processing overhead.

A tactical position would be to continue in this matter if the cost of this activity, which could occur annually, depending on HEALTH CARE CO.'s revenue goals, does not increase the cost of doing business to the point where revenue is absorbed by any associated outage and scaling costs or lost by limited capability to handle additional volumes of claims/visits.

A strategic position would require HEALTH CARE CO. to examine a different application platform utilizing mission critical caliber technologies. This would position HEALTH CARE CO. for increased flexibility in terms of service levels for users, system/data integrity and claim/visit volume growth. A financial analysis should be performed to forecast the cost of performing ongoing tactical activities versus the overall cost to position HEALTH CARE CO. for a strategic solution. Considering that the application is roughly three years old will the current vendor scale the application to HEALTH CARE CO.'s needs? Such questions need to be answered, for the longer this situation continues the current platform may be deemed legacy in several years and at that point HEALTH CARE CO. would have no choice but to implement a strategic solution anyway. Additional details regarding tactical and strategic recommendations are provided in the section titled Executive Recommendations.

Please refer to the Professional Summary for additional details of the issues discussed here and the detailed sections on statistics and trends.

Introduction

Health Care Corp. (HEALTH CARE CO.) utilizes a proprietary application system across its WAN and LAN to service the organization and its customers. This application system, called Pathways, is an encompassing system that includes payroll, general accounting and the specific functions for medical patient claims processing. The application requires the service of a Microsoft SQL Server back-end database and is thus considered a two-tier Client Server system. The second tier, the client component, is distributed to remote end users via Microsoft Windows Terminal Server and Thin Client hardware technology. All of the server components are located at the client's headquarters in Lake Success New York. There are 700+ users connected to Pathways daily and the application is a critical component of HEALTH CARE CO.'s business operations. The client has been experiencing intermittent performance issues in the Pathway system. These performance issues have various symptoms ranging from sporadic slowdown of application functions to all users and all application functions responding in a sluggish manner.

HEALTH CARE CO.'s Information Systems department has requested Applied Methodologies, Inc. (AMI) to perform a forensic network and application traffic analysis to uncover any network or application related issues that may provide some direction toward redressing the Pathway performance behavior. This analysis will cover many components required to transport Pathway data to and from a user. Network traffic and application protocols will be reviewed for their performance considerations. Application and Terminal Servers will be analyzed for their sizing structure in relation to overall response time. It is AMI's goal to identify the areas across the different technical disciplines that could be improved in terms of better application performance. AMI will attempt to uncover as much tangible information so an answer(s) or a direction towards a resolution can be reared. HEALTH CARE CO. can utilize this information in regards to tactical and strategic decisions regarding the Pathway application. Certain application functions and business areas or relationships of the application presence may not be covered due to time constraints such as printing, reporting, NCFE and HBOC.

AMI performed the following tasks over the past three weeks, starting on October 1, 2000:

1. Conduct interviews with key support personnel to understand the functional components and symptoms of the Pathway system.
2. Perform a basic traffic analysis and baseline of the Headquarters LAN (Local Area Network) and WAN (Wide Area Network) segments.
3. Investigate general health of the Network and its infrastructure components.
4. Investigate MS Terminal Server performance and operating characteristics.
5. Investigate the Thin Client performance options and general operation.
6. Investigate SQL server data transport operation and efficiency through forensic protocol analysis and server inspection.
7. Perform packet traces to understand the mechanics of the application's use over a network and to uncover any areas for possible improvement in any of the following technical disciplines that make up this system: the network, database, Terminal Server, Thin Client and application system.

Professional Summary

AMI has reviewed the components of the Pathways system as a whole and the analysis of the information reviewed provided some new information and confirmed other facts of the behavior of the entire system. When AMI states "system" it is referring to all of the elements that comprise the system to deliver the Pathway application to end-users.

- Microsoft Terminal Servers
- Microsoft SQL Servers
- Network Infrastructure and protocols
- Thin Clients
- Pathway Application modules

To understand some of the illnesses of this system the reader must have a basic understanding of HEALTH CARE CO.'s system architecture and understand the basic flow of a typical transaction. Below is a summarized view of the HEALTH CARE CO. system architecture.

HEALTH CARE CO. utilizes the components listed above to provide access to the Pathways application modules so end users can process patient claim and visit information in greater volume and ease.

The Microsoft SQL server is the component that utilizes a standard Relational Database Management System (RDBMS) that contains all of a patient's claim and visit information in various tables consisting of rows and columns of specific data for each patient. All of the critical data that calculations are performed against, accounting, General Ledger, Payroll and patient data reside in these tables. There are three primary SQL servers called SBDBMS 1, 2 and 3 that the end users, via the Pathways application modules running on Terminal Servers, obtain their data from.

Another SQL server called SBDBMSCONS or CONS receives data replicated from the other SBDBMS servers. CONS is a critical component that ties all of the relational data from the other three SQL servers and is used as a back up of data, data integrity, reporting and inter application query engine when applicable.

The Terminal Servers establish a connection with the SQL servers for data retrieval to the Pathway application module user sessions. Queries built from the use of Pathway application modules are sent to the SQL sever for processing. The SQL server will respond to the Terminal Server and a user's Pathway application module with the appropriate data. The Terminal Server will then process and present (send) the screen result of the application module to the user's Thin Client. All of the Pathway application modules are executed on the Terminal Server. Multiple copies of the Pathway application are executed via profiles for user connections. As each user from a Thin Client logs into a Terminal Server he/she is allowed access to the applications that meet their user ID profile. For example, a typical user will have access to the Pathway application and maybe MS Word or Excel. As of this writing there are 15 Terminal Servers in use with an average of 40-55 Thin Client connections to each one. The Terminal Servers are configured to automatically balance the number of Thin Client user connections attaching to them. An upper limit of 100 Thin Client user connections per server at any one time is allowed.

Thin Client technology enables end users to have access to applications and resources remotely without the size of a full desktop PC. The Thin Client is essentially a scaled down PC without a hard drive. It utilizes a 486-based processor, standard RAM, VGA video and Ethernet network controller to communicate to a remote Terminal Server. The Thin Client utilizes a scaled down operating system with the capability to connect to a Terminal Server. The server can be located locally in the user's office or remotely in a different office location altogether. The user will log into a remote Terminal Server, establish a session, and access the Windows Operating System (OS) and applications just as if he/she were actually at the Terminal Server or had the OS and applications loaded locally on a PC. The applications will utilize the processor, memory and disk capacity on the Terminal Server. The user only receives a screen copy of what is happening on the server. Each user is allocated memory from a pool on the Terminal Server. The disk and processors of the Terminal Server are shared amongst all Thin Clients. Multiple Thin Clients can run multiple copies of the same application repeatedly.

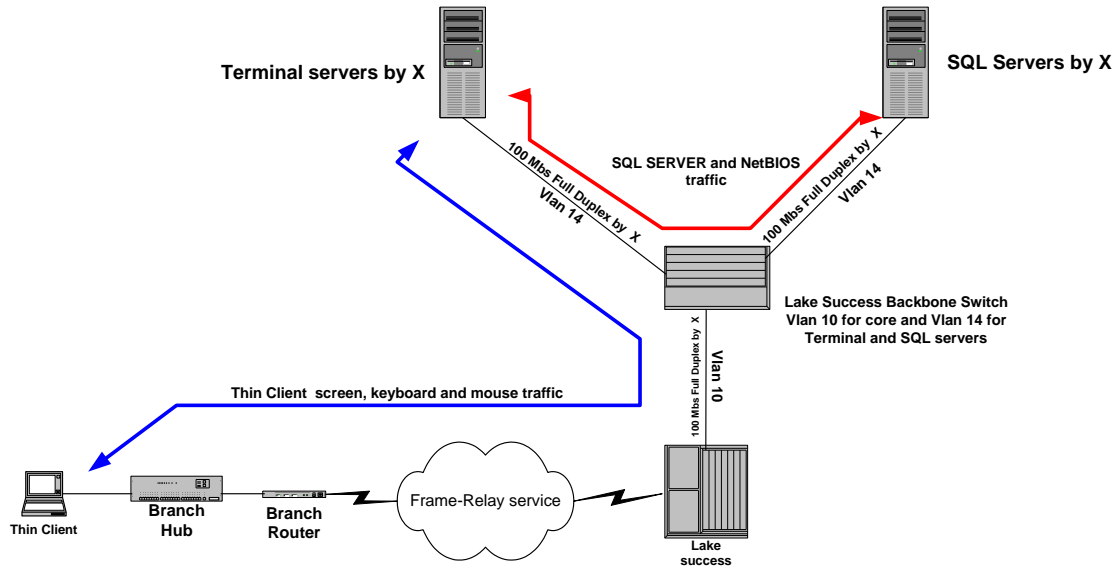
The use of the Thin Client reduces the overall cost of support and remote administration. Thin Client PCs are small in footprint and can be easily swapped out. The Thin Client will automatically discover Terminal Servers and establish a session when the user is ready to work. The Thin Client and Terminal Server architecture basically provides legacy host type of communication and deployment of an application to a large scale of users. The Thin Client architecture efficiently utilizes bandwidth over slower speed WAN links to remote offices by sending only screen information in small data packets and simplifies the distribution and changes of application software, for only changes or updates to the application are made at the Terminal Server.

In retrospect one would ask why not load the Pathways application on a PC or "Fat Client"? A Fat Client is a fully configured PC with a hard disk included and Windows Operating system installed locally that requires daily support, larger, more costly, WAN links to support the transfer of the application data, application or OS updates, driver files and remote control services for service personnel to assist users. The Pathway application is designed to run on a Fat Client but for a significantly smaller user base than HEALTH CARE CO. supports. The answer to the question posted earlier is due to Pathways many different application modules and support files to load, maintain and upgrade. Utilizing Fat Clients in every office location would prove too costly and cumbersome from an Information Systems administrative approach.

Whether each local Fat Client had the Pathways modules loaded, or loaded them from a local or remote server would increase the cost of hardware, software, network bandwidth and infrastructure support significantly to what it is currently.

The network infrastructure components that tie all of the above server components mentioned thus far comprises of high speed Ethernet switches in the Lake Success home office to link the Terminal Servers and the SQL servers together. The Terminal and SQL servers utilize a 100Mbps full duplex, non-shared pipe to and from the Ethernet switch. The Switch has a backbone of 3.2 Gigabits of bandwidth available for each of the 100Mbps Terminal or SQL servers to communicate between. WAN routers with multiple Frame-Relay links ranging from 56 to 256kbs of bandwidth provide the circuits to each remote office. Each remote office has a router to complete the Frame Relay link and a hub that supports shared 10Mbps of bandwidth to anywhere from 3 to 33 Thin Client users. The following diagram outlines the basic architecture for the transaction flow:

**Basic Health Care co.
Network Infrastructure flow
diagram for Pathways**



The following list of steps below illustrates the basic mechanics of a typical Pathways transaction.

When a user at a remote location wants to use Pathways he/she will:

1. Turn on their Thin Client.
2. The user selects the published application.
3. All of the Terminal Servers shall respond.
4. The first one to respond, ideally the least utilized, to the Thin Client is the one that the Thin Client will use. In this example Terminal Server # 1 will be used.
5. The Thin Client user will be presented with a Windows NT log on screen. Remember, the log on screen is actually running at the Terminal Server and not the Thin Client.
6. The Thin Client user will enter their User ID and password.
7. The Thin Client user will then be presented with a Windows NT based desktop with icons for the Pathway application pointing to one of the three production SQL servers.
8. The user clicks on the Pathway Icon for SBDBMS1(a Production server), for example, and is then presented with the Pathway log-on screen. Remember that Pathways is being loaded on the Terminal Server. Only the application screens are sent over the WAN to the Thin Client user.
9. The user is authenticated and is presented with the Pathway toolbar.

Using Pathway to obtain some information patient information:

1. The user will, for example, select Intake Fast Pad to find a patient.
2. The Terminal Server will load up the Intake Fast Pad application module and send the screen to the user.
3. The user will put in some information to query or search for all patients named "Smith".

When the user presses the search button the following will occur:

- The Fast Pad application will create the query and send it out of the Terminal Server over the switched network to the SQL server.
- The SQL server will parse the query, look up the information and return the data back to the user's session application module on the Terminal Server.
- The Terminal Server will process and send the screen of information to the Thin Client user.

Therein lies the problem users are experiencing. There are many different points in this system where a transaction can become delayed. These delays result in the symptoms that users are experiencing. Some of the symptoms include:

1. **Poor response time from application users receiving an hourglass and waiting too often for simple tasks to complete.**
2. **Sluggish navigation through Pathway screens.**
3. **Run time and overflow errors.**
4. **Long report and Gen-transaction generation cycles.**
5. **Users unable to determine application behavior.**
6. **Server hardware errors, thus causing loss of capability to some of the application functions.**
7. **Other, normal operating behavior being perceived as a problem.**

The symptoms listed above are present because this is a classic result of an application designed for a smaller number of users and was never meant to scale to this level of users.

The following section will provide a summarized view of the illness pertaining to each component. Details of each component's behavior and findings are in Analysis Details section following this next section.

System Behavioral Illnesses

Terminal Server Issues

Remember the basic transaction mechanics outlined in the previous section? Picture that transaction and many thousands of others in a variety of complexity occurring incessantly with all of the work the current SQL servers and Terminal Servers have to perform. The Terminal Servers have the following responsibilities:

- Handling Terminal Server tasks
- Handling Thin Client connections
- SQL client to server communication ODBC overhead
- Printing
- Application hosting and executing multiple copies of the same Pathway application modules per user

All of this occurring in and out of one network interface. The ironic fact in this situation is that the network performance is exceptional and bandwidth is plentiful.

The delays users are experiencing is from the high CPU utilization resulting from constant Pathway application modules and support files being loaded. A direct correlation to high CPU utilization and application load was noted during the analysis. The asymmetrical operation of the four processors is also evident. The CPUs frequently peak at 100% and can average 60% during the business day. The processors are not utilized in tandem, meaning that one processor can spike at 100% while another is at 70% as opposed to all processors running at 50%. This activity may be due to Windows NT's method of managing multiple processors efficiently.

AMI, in conjunction with HEALTH CARE CO. IT staff conducted several experiments to qualify and clarify the performance issues. Please refer to the Terminal Server Analysis subsection under the Analysis Details section of this report for detailed findings and recommendations.

SQL Server Issues

The SQL servers have the following responsibilities:

- SQL Server engine and query processing
- SQL server connections to Terminal Servers and ODBC overhead
- Replication to other servers - CONS
- SMB files copies/reads to and from the server
- Execution of stored procedures and repeated cursors
- Backups

Again, all of this is occurring out of one network interface. But the network performance and available bandwidth to handle the data to and from other servers is exceptional.

The delays the users are experiencing are also the result of the SQL servers, SBDBMS 1 through 3, and the CONS server experiencing high utilization levels as the transaction volume increases. AMI has noted that SQL Server engine activity does correlates to high CPU utilization. The Disk I/O read/write times are very high and also correlate to high utilization levels. Network I/O does spike to high levels but there is no regular correlation to CPU and Disk usage levels.

The extremely large tables are not properly normalized which results in delays from long unneeded table searches. Replication functions and file copy/reads to other servers during a transaction add to the overall delay in a completed transaction. Transaction dependencies from a

SBDBMS (1-3) server to the CONS server also result in a delay if there is a problem in any of the components in between. CONS has been experiencing disk I/O problems over the past week resulting in replication services to be turned off and reports to be delayed.

AMI, in conjunction with HEALTH CARE CO. IT staff analyzed the traffic patterns and conducted several experiments to qualify and clarify the performance issues. Please refer to the SQL Server Analysis subsection under the Analysis Details section of this report for detailed findings and recommendations.

Network Infrastructure And Protocol issues

In light of the Terminal and SQL server's obvious illnesses the network infrastructure as a whole is running exceptionally well. This is evident in the sub second transmission of packets between SQL and Terminal Servers and Terminal Servers and Thin Clients.

The effective use of a sliding window in TCP/IP is present. For example when the Terminal Server sends a query for some data, SQL server will send 5 packets in a row, stuffed to their maximum size, with data before the Terminal will send an acknowledgment. This is in contrast to other networks that use an inefficient ping-pong method; that for every packet sent an acknowledgment must be seen before another packet can be sent. The ping-pong affect is not apparent on the HEALTH CARE CO. network backbone that hosts the SQL and Terminal Servers. These servers are utilizing large packets and windowing properly.

The WAN links also are very efficient. The packet response times are sub second. Since only screen information is sent, the behavior is quite different and results in the use of a ping-pong method. For every mouse move and keystroke on the Thin Client, traffic will be generated. These packets sizes are very small and the transmissions behavior is similar to Legacy SNA type of sessions. The utilization of the router and switches that make up the communication components of this system all report extremely low levels of utilization and latency if any at times. Bandwidth is plentiful in the home office, remote sites and on the WAN links. A periodic spike in utilization may appear on a WAN circuit, especially a smaller one (32Kbs), however this does not happen very often.

There was one issue noted regarding the presence of Zero Window activity on the SQL/Terminal Server backbone segment. The Zero Window is a result of a server or workstation advertising back to its communication peer that it does not have any additional buffer space to receive any packets. This function essentially throttles back any excessive transmission by a single peer. This activity was noted heavily from the entire Terminal Server farm, thus indicating that they were being over burdened with data volume from the SQL servers. The ramification here is that the Zero Window processing can cause unneeded CPU processing on the Terminal Servers and may result in extra network traffic. The network components can handle it but the servers cannot. During the second week of AMI's visit a configuration change was applied to all of the Servers to increase the window buffer size. Since that change very little if any Zero Window activity has been detected.

The encapsulated NetBIOS protocol is present on the SQL/Terminal Servers and is used by both types of servers for file copying and reading. The Pathway application does utilize this protocol as well for some of its functions. NetBIOS is an older, less efficient protocol, and does not lend itself well to larger networks with tremendous volume metrics of data. More details regarding this issue are covered in the Analysis Details section of this report

Please refer to the diagram at the beginning of the Professional Summary section regarding the network infrastructure when necessary.

The network Core segment that links the SQL/Terminal Server segment to the segment with the WAN router for remote office communication maintains very low latency and utilization levels.

The network operates in a resilient manner that is flexible and conducive to the needs of the Pathways application system. There are some additional tuning options from a networking standpoint on the Terminal and SQL servers that can be utilized. However, these options may marginally help and just further mask the overall problem of which is the servers acting as the bottleneck points in this system.

For specific details of HEALTH CARE CO. network components, other findings, and operational statistics and tuning options please refer to the Network Analysis subsection under the Analysis Details section of this report for detailed findings and recommendations.

Thin Client Issues

AMI has not noted any significant issues relating to Thin Client operation other than the occasional system failure that result in the Thin Client being replaced with a spare in the remote office. The Performance and network analysis show very efficient use of bandwidth and excellent response times over the WAN between Thin Clients and Terminal Servers. The use of a smaller receive window (1K) was observed but this is appropriate due to the smaller sizes of the packets transmitted and received from the Terminal Servers. There are tuning options on the Thin Client to enhance network performance. However these enhancements are not currently needed.

For specific details of HEALTH CARE CO. Thin Client component utilization statistics and tuning options please refer to the Thin Client Analysis subsection under the Analysis Details section of this report for detailed findings and recommendations.

McKesson/HBOC Pathways Homecare Application Issues

The Pathway application may have never intentionally been designed for this scale of users. Interviews conducted with IS personnel revealed that HEALTH CARE CO. is the largest customer using Pathways. Many of the other customers are much smaller in user population and claim processing or visit volume. To achieve some form of usability for 700+ personnel, multiple servers are required to handle the transaction manipulation and the presentation services to end users.

If necessary, please refer to the diagram at the beginning of the Professional Summary section regarding the network infrastructure and application flow when necessary.

AMI has experience first hand performance problems impacting users ability to complete their jobs. These problems were mainly the result of the sheer volume of transactions combined with multiple bottlenecks at the SQL Servers and Terminal Servers. The application also does not efficiently utilize SMP based systems, network bandwidth and user interface controls. Internal sources of HEALTH CARE CO. have stated that this application is written in Visual Basic. The application utilizes ODBC on a Microsoft NT 4.0 platform for the Database and presentation services. A combination of efficient SQL Server transport protocol and inefficient use of encapsulated NetBIOS protocols add to the general illness encountered. If the internal sources are correct, this may be an issue. The Visual Basic language is traditionally used as an application prototyping tool or to develop small niche applications. It is also used in all Microsoft products for inter object and application scripting/automation. For example MS Word and Excel use Visual Basic as their scripting languages. The end users experience cryptic "runtime" and "overflow" error messages sent down to them from the Terminal Servers.

Runaway queries can be executed by accident to cause the user to sit and wait for the completion or an overflow error to appear. Sometimes the user will not even get an hourglass depicting wait time and the user perceives the unresponsive action at that time as the system being frozen. This

results in the user opening yet an additional instance of the Pathways application thus causing additional CPU overhead on the Terminal Servers.

HEALTH CARE CO. transactions volumes are mission critical caliber, thus requiring an application system utilizing a programming language, hardware and software platform to support such current and future volume metrics.

The application does provide the business logic and functionality necessary for HEALTH CARE CO. to function and be competitive in their industry. The users like the application and it can help HEALTH CARE CO. to grow its business. However, this functionality combined with the transaction volume and growth scale and inherent system's platform liabilities is causing a ceiling of acceptable operating behavior to be reached much sooner than expected.

Experiments and analysis conducted by AMI and HEALTH CARE CO. staff confirms the behavior of the application. Details of the findings and recommendation regarding these experiments are included in the Pathway Analysis subsection of the Analysis Details section.

The use of Terminal Servers and Thin Clients by HEALTH CARE CO. is a sound approach and may be the only viable way to provide access to the Pathway application today. However, the application system, as a whole, is not sized to handle the volume of claim/visit processing today. This is evident in the daily reports from users with complaints of poor response times and IS activities to keep the system running. When performing the analysis AMI frequently received the following statement from different users and support personnel regarding Pathways ***"It isn't just one (application) function that is slow, it seems to be everywhere"***. The application system as a whole comprises of the following elements, SQL server 7 RDBMS servers, Terminal Servers, Thin Client user computers, Network Infrastructure and the Pathway application modules. AMI has experienced the same performance behavior during it's testing of some of the system's elements. Direct correlations have been found to the loading of Pathway modules, increased disk I/O, excessive query activity and general volume activity to degrade the CPU performance on the SQL and Terminal Servers thus resulting in delays and performance issues that affect the users.

The operating system platform itself is another example. HEALTH CARE CO. has grown to mission critical levels of transaction volume. The Windows NT 4.0 operating system, which is only four years old, and the current class of IBM Netfinity servers do not comprise a mission critical quality compared to other systems of similar transaction volume and importance that AMI has experienced in use at similar sized companies. With the advent of Windows 2000, NT version 4 will soon become of legacy status and support options may decrease.

The above statements were confirmed by conducting several experiments that are outlined in more detailed in the Analysis Details section following this one.

Couple the inherent hardware sizing issues with the questionable RDBMS table management, Cursor use, stored procedure issues, plus Pathway application code errors and the use of different protocols some efficient and others inefficient for transaction completion, comprise a system that will behave in a sluggish manner, thus making it difficult for HEALTH CARE CO. IT staff to enforce any quality based Service Levels Agreements, maintain, administer, support and most importantly scale this system.

Overall the, Server components, Operating system and Pathway application modules do not lend themselves very gracefully to this scale of users and mission critical status. Unfortunately it is the Pathway application that dictates the direction of components to be used. The result is a system with limited application flexibility and scalability options.

The following section outlines the major overall recommendations that HEALTH CARE CO. should consider.

Executive Recommendations

HEALTH CARE CO. has two major directions it can take in moving forward, tactical and strategic

Tactical Approach

The tactical direction is comprised of two parts that can be used together:

- 1). Use of the recommendations outlined in the remaining sections of this report to provide any relief possible.
- 2). Continue to scale this system by adding more hardware and software to it.

The second part of the tactical direction is one that HEALTH CARE CO. must heavily consider for it may have a significant financial impact and burden on the organization depending on the organizations future growth plans.

By scaling AMI infers the method of scaling in a horizontal manner by adding more terminal and SQL servers and vertically by increasing the number of CPUs, speed of the CPUs, size and speed of the disk storage subsystem, and overall server hardware and operating system. Basically, what this means is to add more hardware to support the number of users and transaction volumes or to mask or redress the issues listed earlier by throwing hardware and software technology at the overall system to make it appear that it can be supported.

HEALTH CARE CO. can add additional Terminal Servers and reduce the number of users per Terminal Servers to provide some level of consistent response time to the end user. Adding additional SQL servers may help as well but may also increase the complexity of the SQL transaction dependencies and add more reliance on the replication server. For HEALTH CARE CO. to determine what is an acceptable level of performance with room to grow some tests must be conducted.

For example HEALTH CARE CO. can take one new or existing Terminal Server and start it with 10 users. Have those users only run on that server for a week and note their performance results. If positive, add another 5 or 10 and repeat the test for another week. HEALTH CARE CO. can continue this test until the users start to notice a slight degradation in performance that is consistent and conclusive. Let's say that the performance degradation is at 30 users. HEALTH CARE CO. can then size the Terminal Server hardware/software platform to optimally support 30 and scale to a maximum of 40 users per Terminal Server. For 700 users this would mean 23 Terminal Servers.

The network currently has the bandwidth to support this approach but additional options such as adding a second network adapter in each Terminal Server, to handle the Thin Client Traffic and another to handle the SQL server traffic may reduce the CPU burden on packet processing, especially if these network interfaces have integrated processors.

The same approach would apply to the SQL servers as well. However, the use of a Storage area Network (SAN) for optimized data retrieval, management and recovery should be considered. A SAN can off load some of the SQL servers burden of Disk I/O and provide improved reliability and scalability then what is available at HEALTH CARE CO. today. The SAN must be able to integrate with the Windows NT 4.0 platform for this to be effective.

The scaling of the servers could possibly include the migration to Windows 2000 data-center and SQL server 2000. These platforms may provide improved performance and reliability. However, these options must be thoroughly tested, for both platforms are less than a year old.

Remember scaling horizontally and vertically will not address any of the Pathway application code issues or RDBMS table or query issues. This approach just masks the underlying platform based problems.

HEALTH CARE CO. can use this approach if they know for certain that they will only reach a static amount of claim/visit volume per year or can forecast their business growth by claim/visit volume in three to five years. For example if HEALTH CARE CO. knows that today it is handling 10 million claim/visits and that equates to X amount of revenue per year and expect to double that each year for two years and then they expect the volume to stay, let's say 60 million claim/visits, at a consistent level for three years following the doubling years, then this tactical approach can be used to provide the service levels to support such growth. However if HEALTH CARE CO. cannot predict, forecast or just wants to obtain as much growth as possible in terms of handling claims/visits then this approach is very costly, for HEALTH CARE CO. may be upgrading servers, operating systems and other components every year to maintain a decent level of service to the end users.

This direction provides the following benefits:

- **Provides some quick relief to the end users.**
- **Utilizes known platform elements.**
- **Depending on claim/visit to revenue futures - may just require a one-time cost for a single scale horizontally and vertically.**
- **Easy to implement.**

This direction provides the following negatives:

- **Extremely costly depending on how far and how often HEALTH CARE CO. is required to scale before losing revenue due to inability to handle claim/visit volume.**
- **Increases the complexity of the system especially from the SQL server, data integrity and replication perspective.**
- **Difficult to manage and may become overwhelming.**
- **Additional reliance on non-mission critical rated elements.**
- **Begins to expand exponentially and can reach a point of no return for cost and administration burden.**

For HEALTH CARE CO. to scale such a limited system they must surround the application with many expensive server hardware and operating systems components to compensate for the scalability limitations.

Strategic Approach

The strategic direction involves additional planning and a clear understanding of the company direction in processing additional claims/visits to rear higher revenues.

If for example the company's goal is to reach one billion claim/visits in 5 years and maintain that level of processing while experiencing revenue growth and industry prominence then HEALTH CARE CO. should consider moving this system to a platform more conducive to this goal.

This may mean having the application re-written or developed in-house using languages that are traditionally used in mission critical applications such as C++. This will provide HEALTH CARE CO. the flexibility to scale this application in any manner and provide platform independence. The application can be compiled for a UNIX or NT environment. Also, the re-coding can provide the flexibility to be WEB based or traditional three-tier client server.

Another possibility is the use of an Enterprise Resource Planning system (ERP) to handle the accounting, human resources and payroll functions. This may be possible right out of the box with Oracle or SAP products and then code only the unique clinical claim/visit transaction modules to interface.

The use of a mission critical level back-end RDBMS platform, such as Oracle, to provide TPC-C quality level of performance with superior fault tolerance and scaling options should be considered.

Moving the system to a UNIX platform will provide mission critical data handling and program execution resilience required for such volume. Utilizing UNIX based hardware platforms from IBM, SUN or HP will provide the disk and physical resiliency of the components plus achieve true SMP performance and scalability.

The use of a SAN to provide high levels of tolerance and flexibility for data stored outside of the servers physical chassis should also be researched.

The actions listed above better position HEALTH CARE CO. for the future. The initial expenditure may be exorbitant for the hardware and software, man hours of labor, plus the annual cost may increase for additional personnel (programmers if written in house, UNIX administrators and Oracle DBAs) but this may pale in comparison of the expense of maintaining the current system for the next five years or the possibility of the lost revenue due to the inability to handle claim/visit volume over a certain threshold.

Another approach is the possibility of outsourcing the application. However this direction requires careful research and is beyond the scope of this report.

The benefits of this approach is:

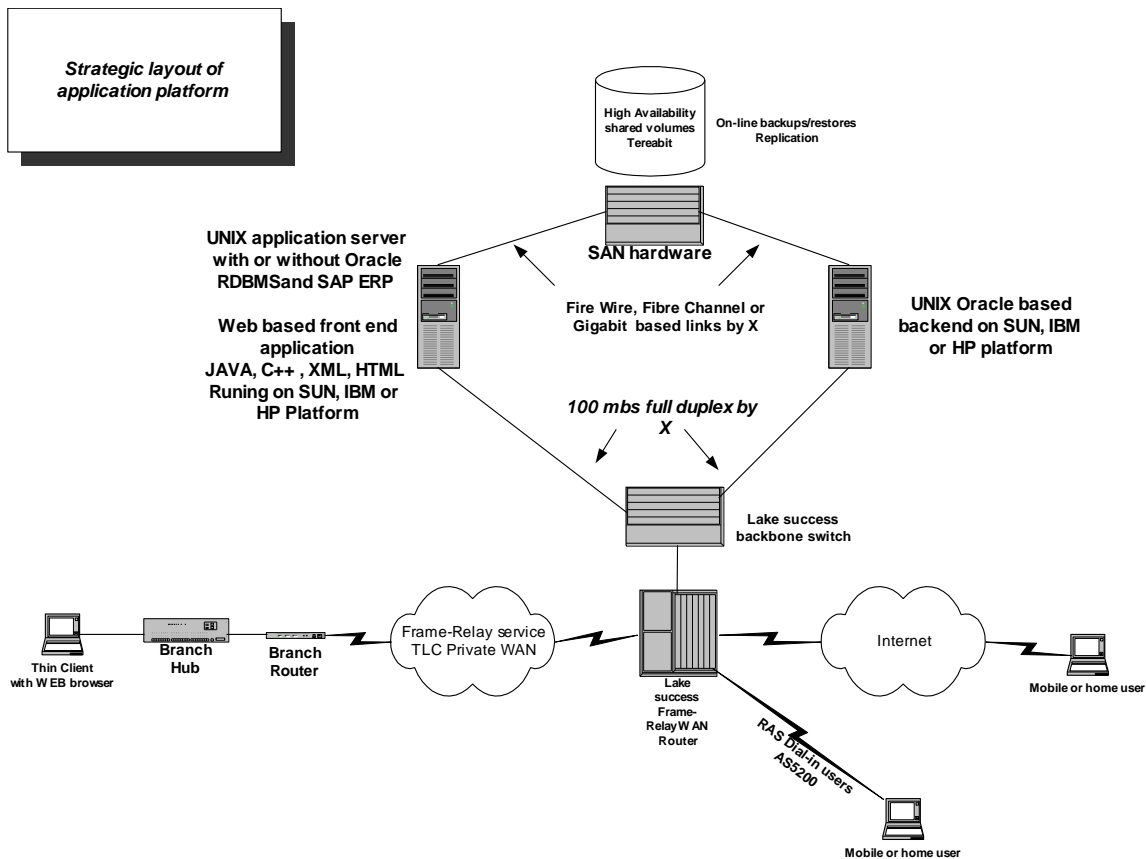
- **Positions HEALTH CARE CO. for future volume and revenue.**
- **Increases IT asset valuations for newer, mainstream, more adaptable technology is now in house.**
- **Provides HEALTH CARE CO. the ability to scale easily.**
- **Reduces complexity based on the development cycle and may possibility lead to a system that is easier to manage and administer.**
- **Increased flexibility and resiliency is available.**
- **Consistent Service Levels can now be maintained for end users.**
- **Can be Internet ready.**

The negatives of this approach is:

- If HEALTH CARE CO. sees no further growth in the next five years or has no plans to grow significantly then the cost of this approach will outweigh it's benefits quickly.
- HEALTH CARE CO. may be able to "get by" on a tactical solution if the solution is not too costly.

It is recommended that HEALTH CARE CO. partake in the exercise of just pricing out what a strategic solution would cost to get HEALTH CARE CO. into that position and compare it annually to the tactical costs to determine a point of no return from a tactical cost perspective and start applying funds to a strategic approach.

Below is a basic diagram depicting a strategic architecture



Analysis Details

This section will outline the specific findings and recommendations for each finding. The recommendations can be implemented independently or in conjunction with each other. It is advised that if a recommendation is applied to the current system only one recommendation at a time be executed to easily determine a change in operational behavior.

Terminal Server Analysis

The operational behavior of the Terminal Servers is adequate to support anywhere from 1 to 40 Thin Clients users on average. However, as stated in the Professional Summary the Terminal Servers do experience high levels of utilization due to the loading and execution of Pathway application code. This is especially obvious when the user count reaches into the fifties.

The Standard HEALTH CARE CO. Terminal Server Platform consists of IBM Netfinity 7000 servers comprising of four Intel Pentium II 400Mhz Xeon microprocessors, 2 gigabytes of Ram and an 8 gigabyte hard disk partitioned into two volumes. The servers are utilizing Windows NT 4.0 with Service Pack 5. The Pathway application modules, published application profiles, Citrix Metframe software and Windows Terminal Server applications all reside on each Terminal Server. The 2 gigabytes of RAM is used in an open pool for each Thin Client user session and applications are executed under each session. A pair of Compaq PL1600R 6/500 based Terminal Servers with a similar configuration are also in place but these servers handle fewer Thin Clients than their IBM counterparts. As of this writing HEALTH CARE CO. utilizes 14 such servers to support their Thin Client population. These servers are backed up every evening and transactions logs are backed up every three hours. As of this writing the Terminal Server information was gathered during the week of 10/9. Additional IBM based Terminal Servers have been added since with increased overall processing capacity. AMI did not review these servers. Since the majority of the servers are of the IBM Netfinity 7000 quad 400Mhz class, all experiments conducted and trends reviewed were performed against these units.

The Terminal Servers utilize Network interface adapters form IBM and Intel. Some of the drivers are at different versions on the Terminal Servers:

- Terminal Servers with IBM EtherJet 10/100 PCI interface MAC OUI 00-04-AC

SBTSE 1,2,4,5,6,7,8,9,10,13,15 all interfaces are utilizing an NDIS 4 compliant driver IBMFENT.SYS and the version is 3.36.10 except for SBTSE 1 and 4 their driver would not be displayed in the Network Neighborhood properties indicating a different version.
- Terminal Servers with Intel Pro 100 interface MAC OUI 00-90-27

SBTSE 3

Terminal Servers with NetFlex 3 Net Intelligent Dual 10/100 interfaces MAC OUI 00-80-5F
Driver version 2.26.00

SBTSE 12
Driver version 4.25

Terminal Servers with NetFlex 3 Net Intelligent Dual 10/100 interfaces MAC OUI 00-80-C7

SBTSE 11
Driver version 4.25

The ODBC driver is setup for a connection pooling time-out of 60.

As stated previously in the Executive Summary these Terminal Servers have many responsibilities:

- Handling Terminal Server tasks.
- Handling Thin Client connections.
- SQL client to server communication ODBC overhead.
- Printing.
- Application hosting and executing multiple copies of the same Pathway application modules per user.

The following trends have been observed during the analysis period and are consistent across all Terminal Servers.

AMI used the PERFMON utility to gather a basic set of statistics and by using the utility's graphing capability was able to notice trends in real time. Due to the busy nature of the servers and the production status no logging of the statistics was performed. Only the trends observed in real time, on the screen, were noted at different intervals over the course of a week.

The PERFMON objects used were as follows:

% Processor - Amount of time the processor was busy
% User - User mode for application code executing. The time the processor spent handling code execution
% Disk Time - The time the disk was busy servicing read/write requests
% Committed Bytes in use - Percentage of memory in use from the overall pool
Bytes Total/Sec - Total Network Send and Receive traffic

It was often observed that the quad 400Mhz Xeon processor do not operate in a fully symmetrical manner. One processor may reach 100% utilization while another is at 70% and the remaining two at 50%. This may be due to Windows NT 4.0 ability to efficiently distribute the processor load evenly. There were also instances when all processors would be synchronized. When AMI refers to CPU utilization below it is referring to all four CPUs being used.

These Terminal Servers are experiencing high levels of CPU utilization for access at the direct console of a server is sluggish at times.

AMI observed that the Disk wait and I/O time does not correlate to high levels of CPU utilization. Or, High CPU utilization and poor/sluggish performance is not a result of disk I/O activity.

The Network traffic utilization infrequently does correlate with the processor utilization levels but this is not often enough to alone cause the high CPU utilization observed. Network traffic may peak at 8mbs a second into and from the interface but at 100 Mbs full duplex, this activity can be seen as 8% maximum utilization in any given direction. The average network I/O is around 2Mbs per Terminal Server.

The memory utilization usually sits at around 40%. Meaning that 40% of the 2 gigabytes is allocated on average to support the Terminal functions listed earlier. There was no correlation to high CPU utilization and increased memory usage.

The amount of open files remains constant and does not fluctuate. There was no correlation to open files and high CPU utilization as well.

There was a direct correlation to the CPU utilization and the execution of user programs. User programs are application code being executed, such as the Pathways.exe and PWHC.exe plus associated RPT, INP, OCX and any DLLs. This shows us that all four CPUs are very busy just running each Thin Client user's application code. The usage will cause all four processor to sit at 100% utilization for several seconds and sometimes all four processors are averaging 70%. This behavior results in the users complaints of poor performance and sluggish application navigation especially for module loading and screen presentation to the Thin Client user.

Several experiments were performed to validate that the high levels of utilization are a direct result of Pathways application code execution.

A test was performed using a Fat Client to access the patient data from the SQL server SBDBMS3. The Fat Client and SBDBMS3 server resided on the same segment. The Fat Client was used to eliminate the Terminal Server from the transaction path and see if a major difference in performance was noted. Also, to actually see the Pathway application modules run in an isolated single session manner. The Pathways software was loaded on the Fat Client thus eliminating the use of the Terminal Server. A side-by-side comparison was performed between this Fat Client and a Thin Client setup in the home office. The Fat Client would sometimes outperform the Thin Client but other times the Thin Client would present screens faster. Just loading the Pathway modules on the Fat Client resulted in the Fat Client reaching over 70% CPU utilization.

Sometimes the delay was due to the loading of modules and sometimes to dependent SQL activity required before the module is loaded. Otherwise any wait on the Fat Client was from the database server. When the Thin Client did out perform the Fat Client on module loading this may have been because the Fat Client is a basic Pentium class machine with only 64 Mb of RAM and the Terminal Server is utilizing 4 400Mhz processors with 2 Gbs of RAM. The Terminal Server hosting the test Thin Client's session may have been slightly less utilized at the moment of the comparison and edged out the Fat Client. This shows that even executed locally the Pathway application modules have a significant impact to performance without any Terminal Server overhead. Now scale this performance behavior to the Terminal Server thus acting like a major Fat Client but utilizing more processors, memory and disk and running the same instance 40 to 60 times for 60 Thin Clients.

Another test conducted was to further qualify the impact application code execution has on a Terminal Server. An early morning test was conducted against the Terminal Server to determine the impact. A production caliber Terminal Server was isolated to one Thin Client user connection. The Thin Client initiated one session of Pathways and some basic transactions were executed. Loading of the Billing module alone resulted in a sustained 30% CPU utilization for all four processors for three seconds. Other functions ranged in the 25 to 40 percent range and the processor was usually tied up for over 3 seconds. This is just one user on a Terminal Server and the response was barely adequate. We executed the blank Intake FastPad, which is an application quirk that results in a full table scan of every patient's name returned if the search field is left blank. Many users execute this by accident thus causing unneeded processing on both the Terminal and SQL servers. The results were all four processors did sustain 40% utilization for over 5 seconds. When we terminated the function it was noted that that we were exited from Pathways application but the traffic still remained on the network from the SQL server.

Performing a search in the Billing Process again alone caused the four processors to hit 40% utilization for an average of 5 seconds.

Performing a search in Customer Maintenance process caused only one processor to spike to 30% utilization for 3 seconds.

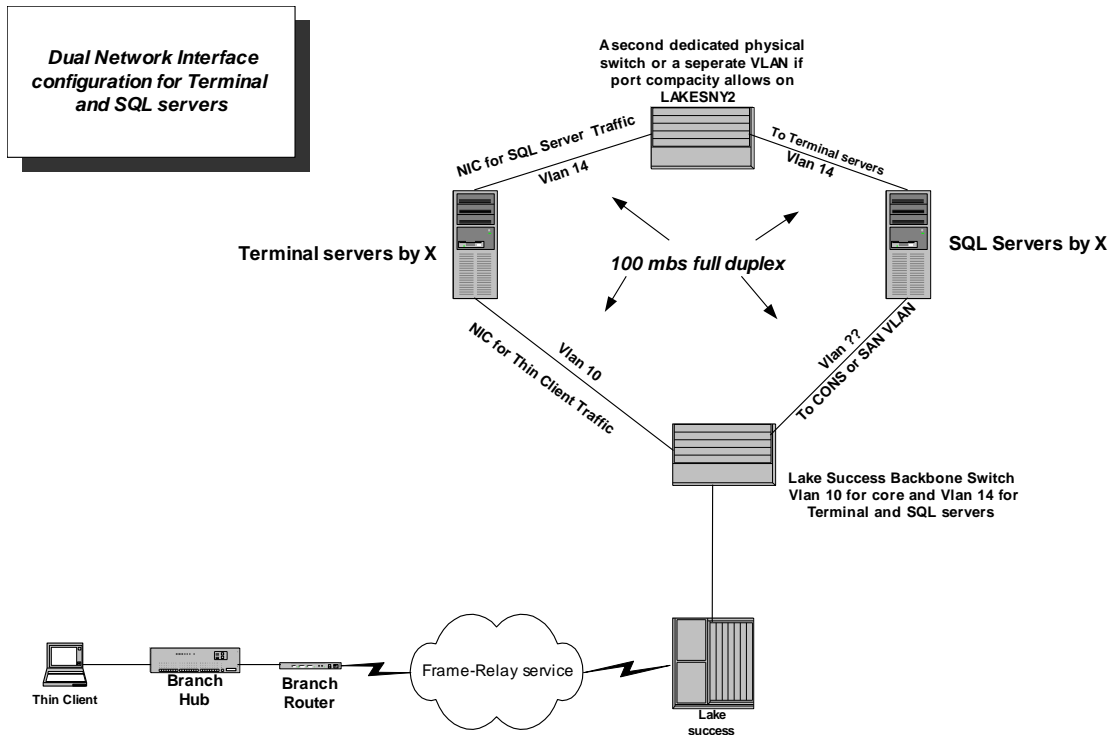
This final test was not really a planned test but just a confirmation of the volume and sizing issues currently experienced with this system. The Columbus Day and Jewish Holiday on October 9th.

Resulted in less users on the system altogether due to the holidays. The overall response of the system improved and users were reporting, "The system was flying today". This event alone shows that with fewer users per Terminal Server the system performs to a level of user satisfaction.

Recommendation: Reduce the number of users on each Terminal Server. Please refer to the Executive Recommendations section for details.

Recommendation: Integrate NICs with on board CPUs to lighten the burden of the four Xeon processors when network traffic needs to be processed. Even though the correlation to network traffic and high CPU utilization were infrequent, the advent of the Zero Window issue, outlined in the next section, and just the processing requirements of the IP stack from the sheer volume of data the Terminal Servers must handle, off loading or mitigating any CPU cycles to a separate processor on the NIC could provide some relief.

Recommendation: Investigate the use of adding a second NIC. By adding two NICs in each server one to handle the Thin Client Terminal traffic and another to handle the SQL server traffic may slightly improve overall response performance on the Terminal Server. Since the network bandwidth is not an issue no changes are required here. This recommendation may provide some relief but mask the overall issue of too many instances of application code being executed per Terminal Server. The use of routing between the NICs or other methods of directing or balancing the specific traffic to a specific NIC must also be considered. The use of the previous recommendation should also be considered. See the network diagram below for an example:



Recommendation: Bring all network drivers up to the same level. This is a minor issue and will be resolved with the advent of new adapters or new servers if those options are exercised.

Recommendation: Experiment and possibly implement the following Network adapter tuning options. Some of these tuning options may be used to reduce the CPU time required to process packets. It may result in less efficient use of the NIC, but if the NIC is handling the traffic well the

reduced processor overhead may provide a little relief. These tuning options pertain to the IBM EtherJet interfaces only.

Adaptive Performance Tuning

Sets the number of frames the adapter receives before triggering an interrupt. Under normal operation, the adapter generates an interrupt every time a frame is received. Reducing the number of interrupts improves CPU utilization.

Move the slider to Max Adapter Bandwidth to generate one interrupt per frame. This increases adapter bandwidth, but may reduce CPU utilization, slowing your computer. Move the slider toward Max CPU Utilization to increase the number of frames the adapter receives before generating an interrupt. This improves CPU utilization, but may reduce adapter bandwidth. The default setting works well for most computers.

Adaptive Technology

Recommended value = On

This parameter either enables or disables the Adaptive Technology performance enhancement feature. To enable the feature, click ON. To disable the feature, click OFF. For best performance and maximum CPU utilization, leave this parameter ON.

Adaptive Transmit Threshold

Recommended value = 16

Specifies the number of bytes before the PCI adapter empties its internal transmit FIFO onto the wire. The value is multiplied by 8 to produce the number of bytes. For example, if Transmit Threshold = 200, the number of bytes is 1600. This is greater than the maximum packet size for Ethernet. Consequently, the adapter won't attempt early transmits. Although this is the safest setting, the best performance is achieved when the threshold parameter is as low as possible (without producing under runs).

To experiment, set the parameter to 16 and then incrementally increase it if performance drops significantly.

Note: Don't set the transmit threshold parameter below 200 for computers with multiple bus mastering cards, or computers with otherwise high latency.

Coalesce Buffers

Recommended value = 8

Specifies the number of memory buffers available to the driver in case the driver runs out of available Map Registers. This buffer area is also used when a packet consists of many fragments. If no coalesce buffers or map registers are available, the driver will be forced to queue the packet for later transmission. The preferred method of transmitting data is to use map registers since it's the most efficient method.

Coalesce buffers range: 1-32

PCI Bus Efficiency

When enabled, causes all transmit packets to be coalesced into a single buffer before being sent to the network card. Because the entire frame requires only one PCI transaction, the PCI bus is more efficient but transmit time is slightly longer.

When disabled, the packets are not coalesced, and each packet requires several PCI transactions. The PCI Bus is less efficient, but transmit time is faster.

Receive Buffers

Recommended value = 32

Specifies the number of buffers used by the driver when copying data to the protocol memory.

In high network load situations, increasing receive buffers can increase performance. The tradeoff is that this also increases the amount of system memory used by the driver.

If too few receive buffers are used, performance will suffer. If too many receive buffers are used, the driver will unnecessarily consume memory resources.

Receive Buffers range: 1-1024

Transmit Control Blocks

Recommended value = 16

Specifies how many transmit control blocks the driver allocates for adapter use. This directly corresponds to how many outstanding packets the driver can have in its "send" queue.

If too few transmit control blocks are used, performance will suffer. If too many transmit control blocks are used, the driver will unnecessarily consume memory resources.

Transmit Control Blocks range: 1-80

Recommendation: ODBC retry wait time. Experiment with this option to see if longer or shorter waiting periods affect overall performance in any way.

SQL Servers Analysis

The SQL servers analyzed were SBDBMS 1, 2, 3 and SBDBMS CONS, for these servers provide all of the company and patient data to the Terminal Servers, which ultimately gets sent to the Thin Client users. These servers also utilize Window NT 4.0 with service pack 5 as the operating platform and Microsoft SQL server 7.0 as the RDBMS engine. These Servers are also IBM Netfinity Class similar to the Terminal Servers but with larger disk and memory capacity. The servers also employ the use of IBM EtherJet 10/100 PCI interfaces. However, there are different versions of the driver installed.

SBSBMS1: Driver information not recovered from network neighborhood properties tab. This suggests an older driver.

SBDBMS2: IBMFENT.SYS version 3.36.10

SBDBMSCONS: IBMFENT.SYS version 4.02.27

SBDBMS3: Uses an Intel Pro 100 Intelligent Server interface. Driver version 2.26

It was observed during the week of 10/9 that there were some noticeable trends and correlations regarding the operational behavior of these servers. All of the servers with the exception of CONS exhibit similar trends. CONS is excluded because it has and still is as of this writing experiencing

disk failures so replication has been turned off thus changing the landscape of it's normal operating behavior.

AMI used the PERFMON utility to gather a basic set of statistics and by using the utility's graphing capability was able to notice trends in real time. Due to the busy nature of the servers and the production status no logging of the statistics was performed. Only the trends observed in real time on the screen were noted at different intervals over the course of a week.

The PERFMON objects used were as follows:

% Processor - Amount of time the processor was busy
% User - User mode for application code executing. The time the processor spent handling code execution
% Disk Time - The time the disk was busy servicing read/write requests
% Committed Bytes in use - Percentage of memory in use from the overall pool
Bytes Total/Sec - Total Network Send and Receive traffic

It was often observed that the quad 400Mhz Xeon processor do not operate in a fully symmetrical manner. One processor may reach 100% utilization while another is at 70% and the remaining two at 50%. This may be due to Windows NT 4.0 ability to efficiently distribute the processor load evenly. There were also instances when all processors would be synchronized. When AMI refers to CPU utilization below it is referring to all four CPUs being used.

SBDBMS 1,2 and 3 all maintain on average 40% of the 2.3 gigabytes of RAM is allocated to the Operating system and SQL Server engine functions.

CPU, disk and application server code all correlate when high CPU utilization levels are experienced. The application code represented is the SQL Server engine. When the SQL engine is busy the utilization will also increase. The Disk I/O also does cause the CPU to reach very high levels of utilization. However, the majority of the correlation to high CPU utilization is derived from the SQL server engine with the disk I/O coming in a close second. The Network processing does not correlate in any significant way to higher CPU utilization. The Network traffic will spike and a slight correlation with the CPU will appear but this does not happen often enough to cause a severe performance issue. A majority of the time the Network traffic is low when the CPU is reaching high levels of utilization.

SBDBMS1 Trends

The CPU and SQL Server Engine code always correlate to high utilization. Averages were noted around 50% with sustained rates over 5 seconds to 80%. The disk activity was low compared to the CPU and did not constitute any significant increase to CPU utilization. The Network activity also is low and does not correlate to any high CPU activity. The maximum I/O of network traffic noted was 1mbs per second or 1% of the 100 MB full duplex link in any direction.

SBDBMS2

The CPU and SQL Server Engine code always correlate to high utilization. Averages were noted around 70% with sustained rates over 10 seconds to 100%. The disk activity was high compared to the other SQL servers. The disk busy time did correlate to CPU at times and on it's own without any correlation to the CPU would average 48% and reach a sustained maximum of 100% for over 10 seconds.

The Network activity also is low and does not correlate to any high CPU activity. The maximum I/O of network traffic noted was 2mbs per second or 2% of the 100 MB full duplex link in any direction.

SBDBMS3

The CPU and SQL Server Engine code always correlate to high utilization. Averages were noted around 50% with sustained rates over 5 seconds to 80%. The disk activity was low compared to the CPU and did not constitute any significant increase to CPU utilization. The Network activity also is low and does not correlate to any high CPU activity. The maximum I/O of network traffic noted was 2mbs per second or 2% of the 100 MB full duplex link in any direction.

Microsoft SQL Server

The servers run SQL Server 7.00.0721 and are running on service pack 1. It was disclosed to AMI from HEALTH CARE CO. personnel that service pack 2 could not be used with the current version of the application. There were many questions still open regarding the operating characteristics of the servers for the HEALTH CARE CO. staff does have some access to the SQL servers but not full DBA level access. There is no officially defined HEALTH CARE CO. DBA to answer questions relating to the operation behavior of the database. What was derived from the HEALTH CARE CO. personnel is as follows:

The table sizes are very large and not efficiently normalized plus there are many inefficient Join operations for manually produced reports.

The largest and most popular tables in use and slowest to scan are as follows:

TRASVC 2+million entries set to VARCHAR 250 columns

TRAART

SVC

TRABIL

TRAPAY

The use of Stored procedures (Transact SQL), triggers and constraints are employed to offload application logic from the application.

Cursors are used often causing considerable overhead on the SQL Server Engine. Whether these were intentional in the application design or an afterthought remains to be answered.

The application function's buttons, LOV, and keystrokes by the end user build queries dynamically.

There are some distributed queries for NCFE but that information is beyond the scope of this current paper.

The Indices types, Clustered, Non-clustered and Unique could not be characterized as to which types are in use and are considered by HEALTH CARE CO. personnel as not used efficiently.

The use of an array size function could not be answered. An array size is a tuning function that causes the database engine to stuff as much into a packet as possible before sending a packet of data thus making efficient use of the network medium. The traffic characteristics outlined in the Network section of this report does show an efficient use of the network medium from the SQL servers. However an array size function could be referenced in an application on a transaction basis and be changed dynamically making the application behave better or worse depending on the function.

ODBC was recommended by HBOC as opposed to RPC or Native TDS access.

The replication services from SBDBMS 1,2 and 3 to CONS are Transactional based. Meaning for each transaction to a SBDBMS 1, 2 or 3 server a two phase commit type of transaction will occur to write the data for integrity purposes to the CONS server, thus making CONS the single

repository for data integrity and history. The payroll and customer reports tables also reside on CONS plus hold other important tables for joins and queries from the other SBDBMS servers. There could be a performance issue especially in relation to sizing and scaling if the transaction from an SBDBMS server is perceived slower due to the processing at CONS. This delay could be felt all the way back to the Thin Client user. This activity however was never verified for CONS has been experiencing disk problems during AMI's visit and replication was turned off.

It was disclosed to AMI from HEALTH CARE CO. personnel that the row level locking is not dynamic. Page level locking is not used.

HEALTH CARE CO. could not answer if the use of Multiphase Query optimizer is employed on the SQL servers.

As you can see HEALTH CARE CO. has their hands tied in their ability to effectively manage, scale or tune the SQL server engine to optimize its performance. Some of the performance issues could be remedied by the proper normalization of tables, tighter stored procedure code and optimizing of cursors and queries. HBOC technical personnel handle these responsibilities currently.

SQL TDS analysis

SQL Server utilizes a protocol called TDS for Tabular Data Stream protocol and is used by SQL server to efficiently manage streams of data returned from table scans over a network. The Agilent Advisor was used to monitor TDS traffic on Vlan 14 to characterize its transmission behavior.

Tabular Data Stream (TDS) is the data stream protocol used by SQL Server to transfer requests and responses between the client and the server. TDS is a logical data stream protocol and must be supported by a physical network interprocess communication mechanism (IPC). The Net-Library architecture provides a method of sending TDS across a physical network connection, and it provides a transparent interface to DB-Library for C.

The Agilent Advisor's analysis of the SQL traffic uncovered the following:

The TDS (Tabular Data Stream): Excessive Data Rate from Server message is a WARNING event. System administrators want to know when a large amount of data is being served by their databases. This is seen because of the sheer volume of transactions occurring on Vlan 14. The Excessive Data Rate from Server message occurs when a number of bytes that exceeds a user-defined threshold value are transmitted from the MS SQL/Sybase server during a sample period.

The TDS (Tabular Data Stream): Excessive Transaction Rate message is a WARNING event. System administrators want to know when a large number of requests are being made of their servers. Again, this is seen because of the sheer volume of transactions occurring on Vlan 14. The Excessive Transaction Rate message occurs when a number of transactions that exceeds a user-defined threshold value are made of a MS SQL/Sybase server during a sample period.

The TDS (Tabular Data Stream): Excessive Transaction Rate message is a WARNING event. System administrators want to know when a large number of requests are being made of their servers. The Excessive Transaction Rate message occurs when a number of transactions that exceeds a user-defined threshold value are made of a MS SQL/Sybase server during a sample period.

The above warning events are of informational nature. Use this information to determine how to upgrade/optimize your networked MS SQL/Sybase applications and servers based on the volume metrics of data being transmitted.

The TDS (Tabular Data Stream): Excessive Urgent Data Percentage message is a WARNING event. A TDS frame that contains information that is considered urgent could be interesting to a database administrator (D.B.A.) or network administrator.

The Excessive Urgent Data Percentage message occurs when a TDS frame with the TDS_BUF_URGENT bits set in the MsgType field. Determine the nature of the data contained within the TDS packet marked urgent and determine what the cause of the urgency is. These Urgent messages are used often and may be causing unneeded SQL server engine overhead resulting in higher CPU utilizations. This could be caused by the priority these TDS messages have over normal messages. AMI did not have the time to quantify and categorize which application functions are using the Urgent Data messages to determine if some function do or do not need to utilize it. However some of the more common messages from one Terminal Server to a SQL serve were repeated often is as follows:

IP: Source address = 129.1.14.10
IP: Destination address = 129.1.14.31

----- TDS Header -----
TDS: Message Type = Packet Contains An Urgent Tokenized TDS Request (16)
TDS: Status = 0x01
TDS: 0... No Event Notification
TDS:0.. No Attention Request
TDS:0. No Attention Acknowledge
TDS:1 Last Buf In Req Or Resp
TDS: Buffer Length = 0x0103
TDS: Channel = 0x0000
TDS: Packet = 1
TDS: Window = 0
TDS: Byte Order Unknown; Using LE
TDS: TDS Token Type = Unknown (0xFB)
TDS: 251 Bytes Of Data

fb 00 00 00 53 00 42 00 54 00 53 00 45 00 32 00S.B. T.S.E.2.
50 00 57 00 48 00 43 00 55 00 53 00 45 00 52 00 P.W.H.C. U.S.E.R.
80 a5 51 a5 20 a5 b1 a5 41 a5 41 a5 f1 a5 52 00 ..Q. ... A.A...R.
65 00 66 00 65 00 72 00 65 00 6e 00 63 00 65 00 e.f.e.r. e.n.c.e.
45 00 64 00 69 00 74 00 6f 00 72 00 73 00 62 00 E.d.i.t. o.r.s.b.
64 00 62 00 6d 00 73 00 32 00 4f 00 44 00 42 00 d.b.m.s. 2.O.D.B.
43 00 70 00 77 00 68 00 63 00 70 00 72 00 6f 00 C.p.w.h. c.p.r.o.
64 00 30 00 32 00 4e 54 4c 4d 53 53 50 00 01 00 d.0.2.NT LMSPP...
00 00 07 b2 00 00 0d 00 0d 00 26 00 0 00 06 00&.....
06 00 20 00 00 00 53 42 54 53 45 3F 53 54 41 46SB HEALTH CO
46 42 55 49 4c 44 45 52 53 11 4f f4 6b S.O.k

Other TDS Urgent messages between other Terminal and SQL servers contained:

01 00 00 53 00 42 00 54 00 53 00 45 00 33 00S.B. T.S.E.3.
50 00 57 00 48 00 43 00 55 00 53 00 45 00 52 00 P.W.H.C. U.S.E.R.
80 a5 51 a5 20 a5 b1 a5 41 a5 41 a5 f1 a5 50 00 ..Q. ... A.A...P.
77 00 48 00 63 00 20 00 2d 00 20 00 50 00 77 00 w.H.c. . - .P.w.
48 00 63 00 2e 00 45 00 58 00 45 00 20 00 28 00 H.c...E. X.E. .(.
50 00 61 00 74 00 68 00 77 00 61 00 79 00 73 00 P.a.t.h. w.a.y.s.
20 00 48 00 6f 00 6d 00 65 00 63 00 61 00 72 00 .H.o.m. e.c.c.a.r.
65 00 20 00 4d 00 65 00 6e 00 75 00 29 00 53 00 e. .M.e. n.u.)S.
42 00 44 00 42 00 4d 00 53 00 32 00 4f 00 44 00 B.D.B.M. S.2.O.D.
42 00 43 00 4e 54 4c 4d 53 53 50 00 01 00 00 00 B.C.NTLM SSP.....
07 b2 00 00 0d 00 0d 00 26 00 00 00 06 00 06 00 &.....
20 00 00 00 53 42 54 53 4F 33 53 54 31 46 46 42 ...SBTS E3HELTHO
55 49 4c 44 45 52 53 26 11 e7 8a CORPS& ...

And:

```

f7 00 00 00 53 00 42 00      54 00 53 00 45 00 35 00      ....S.B. T.S.E.5.
50 00 57 00 48 00 43 00      55 00 53 00 45 00 52 00      P.W.H.C. U.S.E.R.
80 a5 51 a5 20 a5 b1 a5      41 a5 41 a5 f1 a5 50 00      ..Q. ... A.A...P.
57 00 48 00 43 00 20 00      2d 00 20 00 50 00 61 00      W.H.C. . -. .P.a.
74 00 53 00 75 00 6d 00      53 00 42 00 44 00 42 00      t.S.u.m. S.B.D.B.
4d 00 53 00 35 00 4f 00      44 00 42 00 43 00 70 00      M.S.5.O. D.B.C.p.
77 00 68 00 63 00 70 00      72 00 6f 00 64 00 30 00      w.h.c.p. r.o.d.0.
33 00 4e 54 4c 4d 53 53      50 00 01 00 00 00 07 b2      3.NTLMSS P.....
00 00 0d 00 0d 00 26 00      00 00 F6 00 06 00 20 00      .....&.....
00 00 53 42 54 53 45 35      53 34 41 46 42 42 55 49      ..SBTSE5 HELTHCO.
 4c 44 45 52 53 7c 52 5b      35                                LDERS|R[ 5
    
```

Large queries are often used for application functions. The performance affect that these queries have on the parsing engine remains to be categorized. The queries are sometimes continued in three to five packets and an example is shown below.

Also, it was noted that the use of a period delimiter in the packets is present in all of the query packets and sometimes in the data packets. This period delimiter may not be required and is wasting precious space in a packet especially for the larger queries where the SQL server has to spend time parsing the queries. If eliminating the periods can reduce a typical query packet from 3 packets to 2 some efficiency is gained. This gain can be exponential when the number of transactions increases. The network does have the capacity to handle the current situation easily, however the focus here is less work to be done on the SQL servers. If the periods are required for application functionality then no action is required. An example of a query and response is shown

The following is a query from Intake Fast Pad search, for customer Smith sent from a Terminal Server session:

```

Packet #1:
 00 90 27 11 51 1b 00 80      5f bd b7 ca 08 00 45 00      ...'.Q... _.....E.
 05 dc f8 81 40 00 80 06      dd 82 81 01 0e f5 81 01      ....@... ..
 0e 20 06 f1 05 99 20 83      bb 4e 37 ef f6 e7 50 10      . .... .N7....P.
 1d 63 ca b0 00 00 03 00      10 00 00 00 01 00 0d 00      .c.....
 73 00 70 00 5f 00 63 00      75 00 72 00 73 00 6f 00      s.p._.c. u.r.s.o.
 72 00 6f 00 70 00 65 00      6e 00 00 00 00 01 38 00      r.o.p.e. n.....8.
 00 00 00 00 00 e7 0e 18      0e 18 53 00 45 00 4c 00      .....S.E.L.
 45 00 43 00 54 00 20 00      45 00 6e 00 74 00 49 00      E.C.T. . E.n.t.I.
 44 00 2e 00 4e 00 61 00      6d 00 53 00 74 00 72 00      D...N.a. m.S.t.r.
 2c 00 20 00 45 00 6e 00      74 00 49 00 44 00 2e 00      ,. .E.n. t.I.D...
 4c 00 73 00 74 00 4e 00      61 00 6d 00 2c 00 20 00      L.s.t.N. a.m.,.
 45 00 6e 00 74 00 49 00      44 00 2e 00 46 00 73 00      E.n.t.I. D...F.s.
 74 00 4e 00 61 00 6d 00      2c 00 20 00 45 00 6e 00      t.N.a.m. ,. .E.n.
 74 00 49 00 44 00 2e 00      49 00 44 00 2c 00 20 00      t.I.D... I.D.,.
 45 00 6e 00 74 00 49 00      44 00 2e 00 43 00 61 00      E.n.t.I. D...C.a.
 74 00 2c 00 20 00 45 00      6e 00 74 00 49 00 44 00      t.,. .E. n.t.I.D.
 2e 00 45 00 78 00 74 00      49 00 44 00 2c 00 20 00      ..E.x.t. I.D.,.
 45 00 6e 00 74 00 49 00      44 00 2e 00 4f 00 72 00      E.n.t.I. D...O.r.
 67 00 53 00 79 00 73 00      2c 00 20 00 45 00 6e 00      g.S.y.s. ,. .E.n.
 74 00 49 00 44 00 2e 00      45 00 6e 00 74 00 53 00      t.I.D... E.n.t.S.
 79 00 73 00 2c 00 20 00      45 00 6e 00 74 00 44 00      y.s.,. . E.n.t.D.
 65 00 6d 00 2e 00 53 00      73 00 6e 00 2c 00 20 00      e.m...S. s.n.,.
 45 00 6e 00 74 00 44 00      65 00 6d 00 2e 00 44 00      E.n.t.D. e.m...D.
 6f 00 62 00 2c 00 20 00      45 00 6e 00 74 00 44 00      o.b.,. . E.n.t.D.
 65 00 6d 00 2e 00 53 00      65 00 78 00 20 00 46 00      e.m...S. e.x. .F.
 52 00 4f 00 4d 00 20 00      45 00 6e 00 74 00 49 00      R.O.M. . E.n.t.I.
 44 00 20 00 4c 00 45 00      46 00 54 00 20 00 4a 00      D. .L.E. F.T. .J.
 4f 00 49 00 4e 00 20 00      45 00 6e 00 74 00 44 00      O.I.N. . E.n.t.D.
 65 00 6d 00 20 00 4f 00      4e 00 20 00 45 00 6e 00      e.m. .O. N. .E.n.
 74 00 49 00 44 00 2e 00      45 00 6e 00 74 00 53 00      t.I.D... E.n.t.S.
 79 00 73 00 20 00 3d 00      20 00 45 00 6e 00 74 00      y.s. =. .E.n.t.
 44 00 65 00 6d 00 2e 00      45 00 6e 00 74 00 53 00      D.e.m... E.n.t.S.
 79 00 73 00 20 00 57 00      48 00 45 00 52 00 45 00      y.s. .W. H.E.R.E.
 20 00 45 00 6e 00 74 00      49 00 44 00 2e 00 54 00      .E.n.t. I.D...T.
    
```

| | | |
|-------------------------|-------------------------|-------------------|
| 79 00 70 00 20 00 3d 00 | 20 00 27 00 50 00 27 00 | y.p. .=. .'.P.' |
| 20 00 41 00 4e 00 44 00 | 20 00 45 00 6e 00 74 00 | .A.N.D. .E.n.t. |
| 49 00 44 00 2e 00 4f 00 | 72 00 67 00 53 00 79 00 | I.D...O. r.g.S.y. |
| 73 00 20 00 49 00 4e 00 | 20 00 28 00 6A 00 2c 00 | s. .I.N. .(.1.,. |
| 33 00 2c 00 34 00 2c 00 | 35 00 2c 00 36 00 2c 00 | 3.,.4.,. 5.,.6.,. |
| 32 00 30 00 2c 00 32 00 | 6A 00 2c 00 32 00 32 00 | 2.0.,.2. 1.,.2.2. |
| 2c 00 32 00 33 00 2c 00 | 32 00 34 00 2c 00 32 00 | ,.2.3.,. 2.4.,.2. |
| 35 00 2c 00 32 00 36 00 | 2c 00 32 00 37 00 2c 00 | 1.,.2.6. ,.2.7.,. |
| 32 00 38 00 2c 00 32 00 | 39 00 2c 00 33 00 30 00 | 2.8.,.2. 9.,.3.0. |
| 2c 00 33 00 6A 00 2c 00 | 33 00 32 00 2c 00 33 00 | ,.3.1.,. 1.2.,.3. |
| 33 00 2c 00 33 00 34 00 | 2c 00 33 00 35 00 2c 00 | 3.,.3.4. ,.3.5.,. |
| 33 00 36 00 2c 00 33 00 | 37 00 2c 00 33 00 38 00 | 3.6.,.3. 7.,.3.8. |
| 2c 00 33 00 39 00 2c 00 | 34 00 30 00 2c 00 34 00 | ,.3.9.,. 4.0.,.4. |
| 6A 00 2c 00 34 00 32 00 | 2c 00 34 00 33 00 2c 00 | 1.,.4.2. ,.4.3.,. |
| 34 00 34 00 2c 00 34 00 | 35 00 2c 00 34 00 36 00 | 4.4.,.4. 5.,.4.6. |
| 2c 00 34 00 37 00 2c 00 | 34 00 38 00 2c 00 34 00 | ,.4.7.,. 4.8.,.4. |
| 39 00 2c 00 35 00 30 00 | 2c 00 35 00 6A 00 2c 00 | 9.,.5.0. ,.5.1.,. |
| 35 00 32 00 2c 00 35 00 | 33 00 2c 00 35 00 34 00 | 5.2.,.4. 3.,.5.4. |
| 2c 00 35 00 35 00 2c 00 | 35 00 36 00 2c 00 35 00 | ,.5.5.,. 5.6.,.5. |
| 37 00 2c 00 35 00 38 00 | 2c 00 35 00 39 00 2c 00 | 7.,.5.8. ,.5.9.,. |
| 36 00 30 00 2c 00 36 00 | 6A 00 2c 00 36 00 32 00 | 6.0.,.6. 1.,.6.2. |
| 2c 00 36 00 33 00 2c 00 | 36 00 34 00 2c 00 36 00 | ,.6.3.,. 6.4.,.6. |
| 35 00 38 00 36 00 36 00 | 2c 00 36 00 37 00 2c 00 | 5.,.6.6. ,.6.7.,. |
| 36 00 38 00 2c 00 36 00 | 39 00 2c 00 37 00 30 00 | 6.8.,.6. 9.,.7.0. |
| 2c 00 37 00 6A 00 2c 00 | 37 00 32 00 2c 00 37 00 | ,.7.1.,. 7.2.,.7. |
| 33 00 2c 00 37 00 34 00 | 2c 00 37 00 35 00 2c 00 | 3.,.7.4. ,.7.5.,. |
| 37 00 36 00 2c 00 37 00 | 37 00 2c 00 37 00 38 00 | 7.6.,.7. 7.,.7.8. |
| 2c 00 37 00 39 00 2c 00 | 38 00 30 00 2c 00 38 00 | ,.7.9.,. 8.0.,.8. |
| 6A 00 2c 00 38 00 32 00 | 2c 00 38 00 33 00 2c 00 | 1.,.8.2. ,.8.3.,. |
| 38 00 34 00 2c 00 38 00 | 35 00 2c 00 38 00 36 00 | 8.4.,.8. 8.,.8.6. |
| 2c 00 38 00 37 00 2c 00 | 38 00 38 00 2c 00 38 00 | ,.8.7.,. 8.8.,.8. |
| 39 00 2c 00 39 00 30 00 | 2c 00 39 00 6A 00 2c 00 | 0.,.9.0. ,.9.1.,. |
| 39 00 32 00 2c 00 39 00 | 33 00 2c 00 39 00 34 00 | 9.2.,.9. 3.,.9.4. |
| 2c 00 39 00 35 00 2c 00 | 39 00 36 00 2c 00 39 00 | ,.9.5.,. 9.6.,.9. |
| 37 00 2c 00 39 00 38 00 | 2c 00 39 00 39 00 2c 00 | 7.,.9.8. ,.9.9.,. |
| 6A 00 30 00 30 00 2c 00 | 31 00 2F 00 31 00 2c 00 | 1.0.0.,. 1.0.1.,. |
| 31 00 2F 00 32 00 2c 00 | 31 00 2F 00 33 00 2c 00 | 1.0.2.,. 1.0.3.,. |
| 31 00 2F 00 34 00 2c 00 | 31 00 2F 00 35 00 2c 00 | 1.0.4.,. 1.0.5.,. |
| 31 00 2F 00 37 00 2c 00 | 31 00 2F 00 38 00 2c 00 | 1.0.7.,. 1.0.8.,. |
| 31 00 2F 00 39 00 2c 00 | 31 00 31 00 2F 00 2c 00 | 1.0.9.,. 1.1.0.,. |
| 31 00 31 00 31 00 2c 00 | 31 00 31 00 32 00 2c 00 | 1.1.1.,. 1.1.2.,. |
| 31 00 31 00 33 00 2c 00 | 31 00 31 00 34 00 2c 00 | 1.1.3.,. 1.1.4.,. |
| 31 00 31 00 35 00 2c 00 | 31 00 31 00 36 00 2c 00 | 1.1.5.,. 1.1.6.,. |
| 31 00 31 00 37 00 2c 00 | 31 00 31 00 38 00 2c 00 | 1.1.7.,. 1.1.8.,. |
| 31 00 31 00 39 00 2c 00 | 31 00 32 00 2F 00 2c 00 | 1.1.9.,. 1.2.0.,. |
| 31 00 32 00 31 00 2c 00 | 31 00 32 00 32 00 2c 00 | 1.2.1.,. 1.2.2.,. |
| 31 00 32 00 33 00 2c 00 | 31 00 32 00 34 00 2c 00 | 1.2.3.,. 1.2.4.,. |
| 31 00 32 00 35 00 2c 00 | 31 00 32 00 36 00 2c 00 | 1.2.5.,. 1.2.6.,. |
| 31 00 32 00 37 00 2c 00 | 31 00 32 00 38 00 2c 00 | 1.2.7.,. 1.2.8.,. |
| 31 00 32 00 39 00 2c 00 | 31 00 33 00 2F 00 2c 00 | 1.2.9.,. 1.3.0.,. |
| 31 00 33 00 31 00 2c 00 | 31 00 33 00 32 00 2c 00 | 1.3.1.,. 1.3.2.,. |
| 31 00 33 00 33 00 2c 00 | 31 00 33 00 34 00 2c 00 | 1.3.3.,. 1.3.4.,. |
| 31 00 33 00 35 00 2c 00 | 31 00 33 00 36 00 2c 00 | 1.3.5.,. 1.3.6.,. |
| 31 00 33 00 37 00 2c 00 | 31 00 33 00 38 00 2c 00 | 1.3.7.,. 1.3.8.,. |
| 31 00 33 00 39 00 2c 00 | 31 00 34 00 2F 00 2c 00 | 1.3.9.,. 1.4.0.,. |
| 31 00 34 00 31 00 2c 00 | 31 00 34 00 32 00 2c 00 | 1.4.1.,. 1.4.2.,. |
| 31 00 34 00 33 00 2c 00 | 31 00 34 00 34 00 2c 00 | 1.4.3.,. 1.4.4.,. |
| 31 00 34 00 35 00 2c 00 | 31 00 34 00 36 00 2c 00 | 1.4.5.,. 1.4.6.,. |
| 31 00 34 00 37 00 2c 00 | 31 00 34 00 38 00 2c 00 | 1.4.7.,. 1.4.8.,. |
| 31 00 34 00 39 00 2c 00 | 31 00 35 00 2F 00 2c 00 | 1.4.9.,. 1.5.0.,. |
| 31 00 35 00 31 00 2c 00 | 31 00 95 d6 ca 3b | 1.5.1.,. 1.....; |

Packet #2

| | | |
|-------------------------|-------------------------|-------------------|
| 00 90 27 11 51 1b 00 80 | 5f bd b7 ca 08 00 45 00 | ..'.Q... _.....E. |
| 05 dc f9 81 40 00 80 06 | dc 82 81 01 0e f5 81 01 |@... |
| 0e 20 06 f1 05 99 20 83 | c1 02 37 ef f6 e7 50 10 |7...P. |

| | | |
|-------------------------|-------------------------|--------------------|
| 1d 63 79 31 00 00 35 00 | 32 00 2c 00 31 00 35 00 | .cyl1..5. 2.,.1.5. |
| 33 00 2c 00 31 00 35 00 | 34 00 2c 00 31 00 35 00 | 3.,.1.5. 4.,.1.5. |
| 35 00 2c 00 31 00 35 00 | 36 00 2c 00 31 00 35 00 | 5.,.1.5. 6.,.1.5. |
| 37 00 2c 00 31 00 35 00 | 38 00 2c 00 31 00 35 00 | 7.,.1.5. 8.,.1.5. |
| 39 00 2c 00 31 00 36 00 | 7B 00 2c 00 31 00 36 00 | 9.,.1.6. 0.,.1.6. |
| 31 00 2c 00 31 00 36 00 | 32 00 2c 00 31 00 36 00 | 1.,.1.6. 2.,.1.6. |
| 33 00 2c 00 31 00 36 00 | 34 00 2c 00 31 00 36 00 | 3.,.1.6. 4.,.1.6. |
| 36 00 2c 00 31 00 37 00 | 7B 00 2c 00 31 00 37 00 | 6.,.1.7. 0.,.1.7. |
| 31 00 2c 00 31 00 37 00 | 32 00 2c 00 31 00 37 00 | 1.,.1.7. 2.,.1.7. |
| 33 00 2c 00 31 00 37 00 | 34 00 2c 00 31 00 37 00 | 3.,.1.7. 4.,.1.7. |
| 35 00 2c 00 31 00 37 00 | 36 00 2c 00 31 00 37 00 | 5.,.1.7. 6.,.1.7. |
| 37 00 2c 00 31 00 37 00 | 38 00 2c 00 31 00 37 00 | 7.,.1.7. 8.,.1.7. |
| 39 00 2c 00 31 00 38 00 | 7B 00 2c 00 31 00 38 00 | 9.,.1.8. 0.,.1.8. |
| 31 00 2c 00 31 00 38 00 | 32 00 2c 00 31 00 38 00 | 1.,.1.8. 2.,.1.8. |
| 33 00 2c 00 31 00 38 00 | 34 00 2c 00 31 00 38 00 | 3.,.1.8. 4.,.1.8. |
| 35 00 2c 00 31 00 38 00 | 36 00 2c 00 31 00 38 00 | 5.,.1.8. 6.,.1.8. |
| 37 00 2c 00 31 00 38 00 | 38 00 2c 00 31 00 38 00 | 7.,.1.8. 8.,.1.8. |
| 39 00 2c 00 31 00 39 00 | 7B 00 2c 00 31 00 39 00 | 9.,.1.9. 0.,.1.9. |
| 31 00 2c 00 31 00 39 00 | 32 00 2c 00 31 00 39 00 | 1.,.1.9. 2.,.1.9. |
| 33 00 2c 00 31 00 39 00 | 34 00 2c 00 31 00 39 00 | 3.,.1.9. 4.,.1.9. |
| 35 00 2c 00 31 00 39 00 | 36 00 2c 00 31 00 39 00 | 5.,.1.9. 6.,.1.9. |
| 37 00 2c 00 31 00 39 00 | 38 00 2c 00 31 00 39 00 | 7.,.1.9. 8.,.1.9. |
| 39 00 2c 00 32 00 7B 00 | 7B 00 2c 00 32 00 7B 00 | 9.,.2.0. 0.,.2.0. |
| 31 00 2c 00 32 00 7B 00 | 32 00 2c 00 32 00 7B 00 | 1.,.2.0. 2.,.2.0. |
| 33 00 2c 00 32 00 7B 00 | 34 00 2c 00 32 00 7B 00 | 3.,.2.0. 4.,.2.0. |
| 35 00 2c 00 32 00 7B 00 | 36 00 2c 00 32 00 7B 00 | 5.,.2.0. 6.,.2.0. |
| 37 00 2c 00 32 00 7B 00 | 38 00 2c 00 32 00 7B 00 | 7.,.2.0. 8.,.2.0. |
| 39 00 2c 00 32 00 31 00 | 7B 00 2c 00 32 00 31 00 | 9.,.2.1. 0.,.2.1. |
| 31 00 2c 00 32 00 31 00 | 32 00 2c 00 32 00 31 00 | 1.,.2.1. 2.,.2.1. |
| 33 00 2c 00 32 00 31 00 | 34 00 2c 00 32 00 31 00 | 3.,.2.1. 4.,.2.1. |
| 35 00 2c 00 32 00 31 00 | 36 00 2c 00 32 00 31 00 | 5.,.2.1. 6.,.2.1. |
| 37 00 2c 00 32 00 31 00 | 38 00 2c 00 32 00 31 00 | 7.,.2.1. 8.,.2.1. |
| 39 00 2c 00 32 00 32 00 | 7B 00 2c 00 32 00 32 00 | 9.,.2.2. 0.,.2.2. |
| 31 00 2c 00 32 00 32 00 | 32 00 2c 00 32 00 32 00 | 1.,.2.2. 2.,.2.2. |
| 33 00 2c 00 32 00 32 00 | 34 00 2c 00 32 00 32 00 | 3.,.2.2. 4.,.2.2. |
| 35 00 2c 00 32 00 32 00 | 36 00 2c 00 32 00 32 00 | 5.,.2.2. 6.,.2.2. |
| 37 00 2c 00 32 00 32 00 | 38 00 2c 00 32 00 32 00 | 7.,.2.2. 8.,.2.2. |
| 39 00 2c 00 32 00 33 00 | 7B 00 2c 00 32 00 33 00 | 9.,.2.3. 0.,.2.3. |
| 31 00 2c 00 32 00 33 00 | 32 00 2c 00 32 00 33 00 | 1.,.2.3. 2.,.2.3. |
| 33 00 2c 00 32 00 33 00 | 34 00 2c 00 32 00 33 00 | 3.,.2.3. 4.,.2.3. |
| 35 00 2c 00 32 00 33 00 | 36 00 2c 00 32 00 33 00 | 5.,.2.3. 6.,.2.3. |
| 37 00 2c 00 32 00 33 00 | 38 00 2c 00 32 00 33 00 | 7.,.2.3. 8.,.2.3. |
| 39 00 2c 00 32 00 34 00 | 7B 00 2c 00 32 00 34 00 | 9.,.2.4. 0.,.2.4. |
| 31 00 2c 00 32 00 34 00 | 32 00 2c 00 32 00 34 00 | 1.,.2.4. 2.,.2.4. |
| 33 00 2c 00 32 00 34 00 | 34 00 2c 00 32 00 34 00 | 3.,.2.4. 4.,.2.4. |
| 35 00 2c 00 32 00 34 00 | 36 00 2c 00 32 00 34 00 | 5.,.2.4. 6.,.2.4. |
| 37 00 2c 00 32 00 34 00 | 38 00 2c 00 32 00 34 00 | 7.,.2.4. 8.,.2.4. |
| 39 00 2c 00 32 00 35 00 | 7B 00 2c 00 32 00 35 00 | 9.,.2.5. 0.,.2.5. |
| 31 00 2c 00 32 00 35 00 | 32 00 2c 00 32 00 35 00 | 1.,.2.5. 2.,.2.5. |
| 33 00 2c 00 32 00 35 00 | 34 00 2c 00 32 00 35 00 | 3.,.2.5. 4.,.2.5. |
| 35 00 2c 00 32 00 35 00 | 36 00 2c 00 32 00 35 00 | 5.,.2.5. 6.,.2.5. |
| 37 00 2c 00 32 00 35 00 | 38 00 2c 00 32 00 35 00 | 7.,.2.5. 8.,.2.5. |
| 39 00 2c 00 32 00 36 00 | 7B 00 2c 00 32 00 36 00 | 9.,.2.6. 0.,.2.6. |
| 31 00 2c 00 32 00 36 00 | 32 00 2c 00 32 00 36 00 | 1.,.2.6. 2.,.2.6. |
| 33 00 2c 00 32 00 36 00 | 34 00 2c 00 32 00 36 00 | 3.,.2.6. 4.,.2.6. |
| 35 00 2c 00 32 00 36 00 | 36 00 2c 00 32 00 36 00 | 5.,.2.6. 6.,.2.6. |
| 37 00 2c 00 32 00 36 00 | 38 00 2c 00 32 00 36 00 | 7.,.2.6. 8.,.2.6. |
| 39 00 2c 00 32 00 37 00 | 7B 00 2c 00 32 00 37 00 | 9.,.2.7. 0.,.2.7. |
| 31 00 2c 00 32 00 37 00 | 32 00 2c 00 32 00 37 00 | 1.,.2.7. 2.,.2.7. |
| 33 00 2c 00 32 00 37 00 | 34 00 2c 00 32 00 37 00 | 3.,.2.7. 4.,.2.7. |
| 35 00 2c 00 32 00 37 00 | 36 00 2c 00 32 00 37 00 | 5.,.2.7. 6.,.2.7. |
| 37 00 2c 00 32 00 37 00 | 38 00 2c 00 32 00 37 00 | 7.,.2.7. 8.,.2.7. |
| 39 00 2c 00 32 00 38 00 | 7B 00 2c 00 32 00 38 00 | 9.,.2.8. 0.,.2.8. |
| 31 00 2c 00 32 00 38 00 | 32 00 2c 00 32 00 38 00 | 1.,.2.8. 2.,.2.8. |
| 33 00 2c 00 32 00 38 00 | 34 00 2c 00 32 00 38 00 | 3.,.2.8. 4.,.2.8. |
| 35 00 2c 00 32 00 38 00 | 36 00 2c 00 32 00 38 00 | 5.,.2.8. 6.,.2.8. |

| | | |
|-------------------------|-------------------------|-------------------|
| 37 00 2c 00 32 00 38 00 | 38 00 2c 00 32 00 38 00 | 7.,.2.8. 8.,.2.8. |
| 39 00 2c 00 32 00 39 00 | 7B 00 2c 00 32 00 39 00 | 9.,.2.9. 0.,.2.9. |
| 31 00 2c 00 32 00 39 00 | 32 00 2c 00 32 00 39 00 | 1.,.2.9. 2.,.2.9. |
| 33 00 2c 00 32 00 39 00 | 34 00 2c 00 32 00 39 00 | 3.,.2.9. 4.,.2.9. |
| 35 00 2c 00 32 00 39 00 | 36 00 2c 00 32 00 39 00 | 5.,.2.9. 6.,.2.9. |
| 37 00 2c 00 32 00 39 00 | 38 00 2c 00 32 00 39 00 | 7.,.2.9. 8.,.2.9. |
| 39 00 2c 00 33 00 7B 00 | 7B 00 2c 00 33 00 7B 00 | 9.,.3.0. 0.,.3.0. |
| 31 00 2c 00 33 00 7B 00 | 32 00 2c 00 33 00 7B 00 | 1.,.3.0. 2.,.3.0. |
| 33 00 2c 00 33 00 7B 00 | 34 00 2c 00 33 00 7B 00 | 3.,.3.0. 4.,.3.0. |
| 35 00 2c 00 33 00 7B 00 | 36 00 2c 00 33 00 7B 00 | 5.,.3.0. 6.,.3.0. |
| 37 00 2c 00 33 00 7B 00 | 38 00 2c 00 33 00 7B 00 | 7.,.3.0. 8.,.3.0. |
| 39 00 2c 00 33 00 31 00 | 7B 00 2c 00 33 00 31 00 | 9.,.3.1. 0.,.3.1. |
| 31 00 2c 00 33 00 31 00 | 32 00 2c 00 33 00 31 00 | 1.,.3.1. 2.,.3.1. |
| 33 00 2c 00 33 00 31 00 | 34 00 2c 00 33 00 31 00 | 3.,.3.1. 4.,.3.1. |
| 35 00 2c 00 33 00 31 00 | 36 00 2c 00 33 00 31 00 | 5.,.3.1. 6.,.3.1. |
| 37 00 2c 00 33 00 31 00 | 38 00 2c 00 33 00 31 00 | 7.,.3.1. 8.,.3.1. |
| 39 00 2c 00 33 00 32 00 | 7B 00 2c 00 33 00 32 00 | 9.,.3.2. 0.,.3.2. |
| 31 00 2c 00 33 00 32 00 | 32 00 2c 00 33 00 32 00 | 1.,.3.2. 2.,.3.2. |
| 33 00 2c 00 33 00 32 00 | 34 00 2c 00 33 00 32 00 | 3.,.3.2. 4.,.3.2. |
| 35 00 2c 00 33 00 32 00 | 36 00 2c 00 33 00 32 00 | 5.,.3.2. 6.,.3.2. |
| 37 00 2c 00 33 00 32 00 | 38 00 2c 00 33 00 32 00 | 7.,.3.2. 8.,.3.2. |
| 39 00 2c 00 33 00 33 00 | 7B 00 2c 00 33 00 33 00 | 9.,.3.3. 0.,.3.3. |
| 31 00 2c 00 33 00 33 00 | 32 00 2c 00 33 00 33 00 | 1.,.3.3. 2.,.3.3. |
| 33 00 2c 00 33 00 33 00 | 34 00 2c 00 33 00 33 00 | 3.,.3.3. 4.,.3.3. |
| 35 00 2c 00 33 00 33 00 | 36 00 2c 00 33 00 33 00 | 5.,.3.3. 6.,.3.3. |
| 37 00 2c 00 33 00 33 00 | 38 00 a8 91 d8 64 | 7.,.3.3. 8.....d |

Packet #3

| | | |
|-------------------------|-------------------------|-------------------|
| 00 90 27 11 51 1b 00 80 | 5f bd b7 ca 08 00 45 00 | ..'.Q... _.....E. |
| 04 c0 fa 81 40 00 80 06 | dc 9e 81 01 0e f5 81 01 |@... |
| 0e 20 06 f1 05 99 20 83 | c6 b6 37 ef f6 e7 50 18 | 7...P. |
| 1d 63 d3 ab 00 00 2c 00 | 33 00 33 00 39 00 2c 00 | .c..... 3.3.9.,. |
| 33 00 34 00 7B 00 2c 00 | 33 00 34 00 31 00 2c 00 | 3.4.0.,. 3.4.1.,. |
| 33 00 34 00 32 00 2c 00 | 33 00 34 00 33 00 2c 00 | 3.4.2.,. 3.4.3.,. |
| 33 00 34 00 34 00 2c 00 | 33 00 34 00 35 00 2c 00 | 3.4.4.,. 3.4.5.,. |
| 33 00 34 00 36 00 2c 00 | 33 00 34 00 37 00 2c 00 | 3.4.6.,. 3.4.7.,. |
| 33 00 34 00 38 00 2c 00 | 33 00 34 00 39 00 2c 00 | 3.4.8.,. 3.4.9.,. |
| 33 00 35 00 7B 00 2c 00 | 33 00 35 00 31 00 2c 00 | 3.5.0.,. 3.5.1.,. |
| 33 00 35 00 32 00 2c 00 | 33 00 35 00 33 00 2c 00 | 3.5.2.,. 3.5.3.,. |
| 33 00 35 00 34 00 2c 00 | 33 00 35 00 35 00 2c 00 | 3.5.4.,. 3.5.5.,. |
| 33 00 35 00 36 00 2c 00 | 33 00 35 00 37 00 2c 00 | 3.5.6.,. 3.5.7.,. |
| 33 00 35 00 38 00 2c 00 | 33 00 35 00 39 00 2c 00 | 3.5.8.,. 3.5.9.,. |
| 33 00 36 00 7B 00 2c 00 | 33 00 36 00 31 00 2c 00 | 3.6.0.,. 3.6.1.,. |
| 33 00 36 00 32 00 2c 00 | 33 00 36 00 33 00 2c 00 | 3.6.2.,. 3.6.3.,. |
| 33 00 36 00 34 00 2c 00 | 33 00 36 00 35 00 2c 00 | 3.6.4.,. 3.6.5.,. |
| 33 00 36 00 36 00 2c 00 | 33 00 36 00 37 00 2c 00 | 3.6.6.,. 3.6.7.,. |
| 33 00 36 00 38 00 2c 00 | 33 00 36 00 39 00 2c 00 | 3.6.8.,. 3.6.9.,. |
| 33 00 37 00 7B 00 2c 00 | 33 00 37 00 31 00 2c 00 | 3.7.0.,. 3.7.1.,. |
| 33 00 37 00 32 00 2c 00 | 33 00 37 00 33 00 2c 00 | 3.7.2.,. 3.7.3.,. |
| 33 00 37 00 34 00 2c 00 | 33 00 37 00 35 00 2c 00 | 3.7.4.,. 3.7.5.,. |
| 33 00 37 00 36 00 2c 00 | 33 00 37 00 37 00 2c 00 | 3.7.6.,. 3.7.7.,. |
| 33 00 37 00 38 00 2c 00 | 33 00 37 00 39 00 2c 00 | 3.7.8.,. 3.7.9.,. |
| 33 00 38 00 7B 00 2c 00 | 33 00 38 00 31 00 2c 00 | 3.8.0.,. 3.8.1.,. |
| 33 00 38 00 32 00 2c 00 | 33 00 38 00 33 00 2c 00 | 3.8.2.,. 3.8.3.,. |
| 33 00 38 00 34 00 2c 00 | 33 00 38 00 35 00 2c 00 | 3.8.4.,. 3.8.5.,. |
| 33 00 38 00 36 00 2c 00 | 33 00 38 00 37 00 2c 00 | 3.8.6.,. 3.8.7.,. |
| 33 00 38 00 38 00 2c 00 | 33 00 38 00 39 00 2c 00 | 3.8.8.,. 3.8.9.,. |
| 33 00 39 00 7B 00 2c 00 | 33 00 39 00 31 00 2c 00 | 3.9.0.,. 3.9.1.,. |
| 33 00 39 00 32 00 2c 00 | 33 00 39 00 33 00 2c 00 | 3.9.2.,. 3.9.3.,. |
| 33 00 39 00 34 00 2c 00 | 33 00 39 00 35 00 2c 00 | 3.9.4.,. 3.9.5.,. |
| 33 00 39 00 36 00 2c 00 | 33 00 39 00 37 00 2c 00 | 3.9.6.,. 3.9.7.,. |
| 33 00 39 00 38 00 2c 00 | 33 00 39 00 39 00 2c 00 | 3.9.8.,. 3.9.9.,. |
| 34 00 7B 00 7B 00 2c 00 | 34 00 7B 00 31 00 2c 00 | 4.0.0.,. 4.0.1.,. |
| 34 00 7B 00 32 00 2c 00 | 34 00 7B 00 33 00 2c 00 | 4.0.2.,. 4.0.3.,. |
| 34 00 7B 00 34 00 2c 00 | 34 00 7B 00 35 00 2c 00 | 4.0.4.,. 4.0.5.,. |
| 34 00 7B 00 36 00 2c 00 | 34 00 7B 00 37 00 2c 00 | 4.0.6.,. 4.0.7.,. |

| | | |
|-------------------------|-------------------------|-------------------|
| 34 00 7B 00 38 00 2c 00 | 34 00 7B 00 39 00 2c 00 | 4.0.8.,, 4.0.9.,, |
| 34 00 31 00 7B 00 2c 00 | 34 00 31 00 31 00 2c 00 | 4.1.0.,, 4.1.1.,, |
| 34 00 31 00 32 00 2c 00 | 34 00 31 00 33 00 2c 00 | 4.1.2.,, 4.1.3.,, |
| 34 00 31 00 34 00 2c 00 | 34 00 31 00 35 00 2c 00 | 4.1.4.,, 4.1.5.,, |
| 34 00 31 00 36 00 2c 00 | 34 00 31 00 37 00 2c 00 | 4.1.6.,, 4.1.7.,, |
| 34 00 31 00 38 00 2c 00 | 34 00 31 00 39 00 2c 00 | 4.1.8.,, 4.1.9.,, |
| 34 00 32 00 7B 00 2c 00 | 34 00 32 00 31 00 2c 00 | 4.2.0.,, 4.2.1.,, |
| 34 00 32 00 32 00 2c 00 | 34 00 32 00 33 00 2c 00 | 4.2.2.,, 4.2.3.,, |
| 34 00 32 00 34 00 2c 00 | 34 00 32 00 35 00 2c 00 | 4.2.4.,, 4.2.5.,, |
| 34 00 32 00 36 00 2c 00 | 34 00 32 00 37 00 2c 00 | 4.2.6.,, 4.2.7.,, |
| 34 00 32 00 38 00 2c 00 | 34 00 32 00 39 00 2c 00 | 4.2.8.,, 4.2.9.,, |
| 34 00 33 00 7B 00 2c 00 | 34 00 33 00 31 00 2c 00 | 4.3.0.,, 4.3.1.,, |
| 34 00 33 00 32 00 2c 00 | 34 00 33 00 33 00 2c 00 | 4.3.2.,, 4.3.3.,, |
| 34 00 33 00 34 00 2c 00 | 34 00 33 00 35 00 2c 00 | 4.3.4.,, 4.3.5.,, |
| 34 00 33 00 36 00 2c 00 | 34 00 33 00 37 00 2c 00 | 4.3.6.,, 4.3.7.,, |
| 34 00 33 00 38 00 2c 00 | 34 00 33 00 39 00 2c 00 | 4.3.8.,, 4.3.9.,, |
| 34 00 34 00 7B 00 2c 00 | 34 00 34 00 31 00 2c 00 | 4.4.0.,, 4.4.1.,, |
| 34 00 34 00 32 00 2c 00 | 34 00 35 00 7B 00 2c 00 | 4.4.2.,, 4.5.0.,, |
| 34 00 35 00 31 00 2c 00 | 34 00 35 00 32 00 2c 00 | 4.5.1.,, 4.5.2.,, |
| 34 00 35 00 33 00 2c 00 | 34 00 35 00 34 00 2c 00 | 4.5.3.,, 4.5.4.,, |
| 34 00 35 00 35 00 2c 00 | 34 00 35 00 36 00 2c 00 | 4.5.5.,, 4.5.6.,, |
| 34 00 35 00 37 00 2c 00 | 34 00 35 00 38 00 2c 00 | 4.5.7.,, 4.5.8.,, |
| 34 00 35 00 39 00 2c 00 | 34 00 36 00 7B 00 2c 00 | 4.5.9.,, 4.6.0.,, |
| 34 00 36 00 31 00 2c 00 | 34 00 36 00 32 00 2c 00 | 4.6.1.,, 4.6.2.,, |
| 34 00 36 00 33 00 2c 00 | 34 00 36 00 34 00 2c 00 | 4.6.3.,, 4.6.4.,, |
| 34 00 36 00 35 00 2c 00 | 34 00 36 00 36 00 2c 00 | 4.6.5.,, 4.6.6.,, |
| 34 00 36 00 37 00 2c 00 | 34 00 36 00 38 00 2c 00 | 4.6.7.,, 4.6.8.,, |
| 34 00 36 00 39 00 2c 00 | 34 00 37 00 7B 00 2c 00 | 4.6.9.,, 4.7.0.,, |
| 34 00 37 00 31 00 2c 00 | 34 00 37 00 32 00 2c 00 | 4.7.1.,, 4.7.2.,, |
| 34 00 37 00 33 00 2c 00 | 34 00 37 00 34 00 2c 00 | 4.7.3.,, 4.7.4.,, |
| 34 00 37 00 35 00 2c 00 | 34 00 37 00 36 00 2c 00 | 4.7.5.,, 4.7.6.,, |
| 34 00 37 00 37 00 2c 00 | 34 00 37 00 38 00 2c 00 | 4.7.7.,, 4.7.8.,, |
| 34 00 37 00 39 00 2c 00 | 34 00 38 00 7B 00 2c 00 | 4.7.9.,, 4.8.0.,, |
| 34 00 38 00 31 00 2c 00 | 34 00 38 00 32 00 2c 00 | 4.8.1.,, 4.8.2.,, |
| 34 00 38 00 33 00 2c 00 | 34 00 38 00 34 00 2c 00 | 4.8.3.,, 4.8.4.,, |
| 34 00 38 00 35 00 2c 00 | 34 00 38 00 36 00 2c 00 | 4.8.5.,, 4.8.6.,, |
| 34 00 38 00 37 00 2c 00 | 34 00 38 00 38 00 2c 00 | 4.8.7.,, 4.8.8.,, |
| 34 00 38 00 39 00 2c 00 | 34 00 39 00 7B 00 2c 00 | 4.8.9.,, 4.9.0.,, |
| 34 00 39 00 31 00 2c 00 | 34 00 39 00 32 00 8e f0 | 4.9.1.,, 4.9.2.,, |
| 1f 04 | | .. |

Packet #4

| | | |
|-------------------------|-------------------------|-------------------|
| 00 90 27 11 51 1b 00 80 | 5f bd b7 ca 08 00 45 00 | ..'.Q... _.....E. |
| 05 dc fb 81 40 00 80 06 | da 82 81 01 0e f5 81 01 |@... |
| 0e 20 06 f1 05 99 20 83 | cb 4e 37 ef f6 e7 50 10 |N7...P. |
| 1d 63 42 85 00 00 03 01 | 08 5f 00 00 02 00 2c 00 | .cB..... _..... |
| 34 00 39 00 33 00 2c 00 | 34 00 39 00 34 00 2c 00 | 4.9.3.,, 4.9.4.,, |
| 34 00 39 00 35 00 2c 00 | 34 00 39 00 36 00 2c 00 | 4.9.5.,, 4.9.6.,, |
| 34 00 39 00 37 00 2c 00 | 34 00 39 00 38 00 2c 00 | 4.9.7.,, 4.9.8.,, |
| 34 00 39 00 39 00 2c 00 | 35 00 7B 00 7B 00 2c 00 | 4.9.9.,, 5.0.0.,, |
| 35 00 7B 00 31 00 2c 00 | 35 00 7B 00 32 00 2c 00 | 5.0.1.,, 5.0.2.,, |
| 35 00 7B 00 33 00 2c 00 | 35 00 7B 00 34 00 2c 00 | 5.0.3.,, 5.0.4.,, |
| 35 00 7B 00 35 00 2c 00 | 35 00 7B 00 36 00 2c 00 | 5.0.5.,, 5.0.6.,, |
| 35 00 7B 00 37 00 2c 00 | 35 00 7B 00 38 00 2c 00 | 5.0.7.,, 5.0.8.,, |
| 35 00 7B 00 39 00 2c 00 | 35 00 31 00 7B 00 2c 00 | 5.0.9.,, 5.1.0.,, |
| 35 00 31 00 31 00 2c 00 | 35 00 31 00 32 00 2c 00 | 5.1.1.,, 5.1.2.,, |
| 35 00 31 00 33 00 2c 00 | 35 00 31 00 34 00 2c 00 | 5.1.3.,, 5.1.4.,, |
| 35 00 31 00 35 00 2c 00 | 35 00 31 00 36 00 2c 00 | 5.1.5.,, 5.1.6.,, |
| 35 00 31 00 37 00 2c 00 | 35 00 31 00 38 00 2c 00 | 5.1.7.,, 5.1.8.,, |
| 35 00 31 00 39 00 2c 00 | 35 00 32 00 7B 00 2c 00 | 5.1.9.,, 5.2.0.,, |
| 35 00 32 00 31 00 2c 00 | 35 00 32 00 32 00 2c 00 | 5.2.1.,, 5.2.2.,, |
| 35 00 32 00 33 00 2c 00 | 35 00 32 00 34 00 2c 00 | 5.2.3.,, 5.2.4.,, |
| 35 00 32 00 35 00 2c 00 | 35 00 32 00 36 00 2c 00 | 5.2.5.,, 5.2.6.,, |
| 35 00 32 00 37 00 2c 00 | 35 00 32 00 38 00 2c 00 | 5.2.7.,, 5.2.8.,, |
| 35 00 32 00 39 00 2c 00 | 35 00 33 00 7B 00 2c 00 | 5.2.9.,, 5.3.0.,, |
| 35 00 33 00 31 00 2c 00 | 35 00 33 00 32 00 2c 00 | 5.3.1.,, 5.3.2.,, |

| | | |
|-------------------------|-------------------------|-------------------|
| 35 00 33 00 33 00 2c 00 | 35 00 33 00 34 00 2c 00 | 5.3.3.,, 5.3.4.,, |
| 35 00 33 00 35 00 2c 00 | 35 00 33 00 36 00 2c 00 | 5.3.5.,, 5.3.6.,, |
| 35 00 33 00 37 00 2c 00 | 35 00 33 00 38 00 2c 00 | 5.3.7.,, 5.3.8.,, |
| 35 00 39 00 39 00 2c 00 | 35 00 34 00 7B 00 2c 00 | 5.3.9.,, 5.4.0.,, |
| 35 00 34 00 31 00 2c 00 | 35 00 34 00 32 00 2c 00 | 5.4.1.,, 5.4.2.,, |
| 35 00 34 00 33 00 2c 00 | 35 00 34 00 34 00 2c 00 | 5.4.3.,, 5.4.4.,, |
| 35 00 34 00 35 00 2c 00 | 35 00 34 00 36 00 2c 00 | 5.4.5.,, 5.4.6.,, |
| 35 00 34 00 37 00 2c 00 | 35 00 34 00 38 00 2c 00 | 5.4.7.,, 5.4.8.,, |
| 35 00 34 00 39 00 2c 00 | 35 00 35 00 7B 00 2c 00 | 5.4.9.,, 5.5.0.,, |
| 35 00 35 00 31 00 2c 00 | 35 00 35 00 32 00 2c 00 | 5.5.1.,, 5.5.2.,, |
| 35 00 35 00 33 00 2c 00 | 35 00 35 00 34 00 2c 00 | 5.5.3.,, 5.5.4.,, |
| 35 00 35 00 35 00 2c 00 | 35 00 35 00 36 00 2c 00 | 5.5.5.,, 5.5.6.,, |
| 35 00 35 00 37 00 2c 00 | 35 00 35 00 38 00 2c 00 | 5.5.7.,, 5.5.8.,, |
| 35 00 35 00 39 00 2c 00 | 35 00 36 00 7B 00 2c 00 | 5.5.9.,, 5.6.0.,, |
| 35 00 36 00 31 00 2c 00 | 35 00 36 00 32 00 2c 00 | 5.6.1.,, 5.6.2.,, |
| 35 00 36 00 33 00 2c 00 | 35 00 36 00 34 00 2c 00 | 5.6.3.,, 5.6.4.,, |
| 35 00 36 00 35 00 2c 00 | 35 00 36 00 36 00 2c 00 | 5.6.5.,, 5.6.6.,, |
| 35 00 36 00 37 00 2c 00 | 35 00 36 00 38 00 2c 00 | 5.6.7.,, 5.6.8.,, |
| 35 00 36 00 39 00 2c 00 | 35 00 37 00 7B 00 2c 00 | 5.6.9.,, 5.7.0.,, |
| 35 00 37 00 31 00 2c 00 | 35 00 37 00 32 00 2c 00 | 5.7.1.,, 5.7.2.,, |
| 35 00 37 00 33 00 2c 00 | 35 00 37 00 34 00 2c 00 | 5.7.3.,, 5.7.4.,, |
| 35 00 37 00 35 00 2c 00 | 35 00 37 00 36 00 2c 00 | 5.7.5.,, 5.7.6.,, |
| 35 00 37 00 37 00 2c 00 | 35 00 37 00 38 00 2c 00 | 5.7.7.,, 5.7.8.,, |
| 35 00 37 00 39 00 2c 00 | 35 00 38 00 7B 00 2c 00 | 5.7.9.,, 5.8.0.,, |
| 35 00 38 00 31 00 2c 00 | 35 00 38 00 32 00 2c 00 | 5.8.1.,, 5.8.2.,, |
| 35 00 38 00 33 00 2c 00 | 35 00 38 00 34 00 2c 00 | 5.8.3.,, 5.8.4.,, |
| 35 00 38 00 35 00 2c 00 | 35 00 38 00 36 00 2c 00 | 5.8.5.,, 5.8.6.,, |
| 31 00 7B 00 7B 00 36 00 | 31 00 31 00 2c 00 31 00 | 1.0.0.6. 1.1.,.1. |
| 7B 00 7B 00 36 00 31 00 | 33 00 2c 00 31 00 7B 00 | 0.0.6.1. 3.,.1.0. |
| 7B 00 36 00 31 00 34 00 | 2c 00 31 00 7B 00 7B 00 | 0.6.1.4. ,.1.0.0. |
| 36 00 31 00 35 00 2c 00 | 31 00 7B 00 7B 00 36 00 | 6.1.5.,, 1.0.0.6. |
| 32 00 31 00 2c 00 31 00 | 7B 00 7B 00 36 00 32 00 | 2.1.,.1. 0.0.6.2. |
| 32 00 2c 00 31 00 7B 00 | 7B 00 36 00 32 00 33 00 | 2.,.1.0. 0.6.2.3. |
| 2c 00 31 00 7B 00 7B 00 | 36 00 32 00 34 00 2c 00 | ,.1.0.0. 6.2.4.,, |
| 31 00 7B 00 7B 00 36 00 | 34 00 37 00 2c 00 31 00 | 1.0.0.6. 4.7.,.1. |
| 7B 00 7B 00 36 00 34 00 | 38 00 2c 00 31 00 7B 00 | 0.0.6.4. 8.,.1.0. |
| 7B 00 36 00 34 00 39 00 | 2c 00 31 00 7B 00 7B 00 | 0.6.4.9. ,.1.0.0. |
| 36 00 35 00 7B 00 2c 00 | 31 00 7B 00 7B 00 36 00 | 6.5.0.,, 1.0.0.6. |
| 35 00 39 00 2c 00 31 00 | 7B 00 7B 00 36 00 36 00 | 5.9.,.1. 0.0.6.6. |
| 7B 00 2c 00 31 00 7B 00 | 7B 00 36 00 36 00 31 00 | 0.,.1.0. 0.6.6.1. |
| 2c 00 31 00 7B 00 7B 00 | 36 00 38 00 31 00 2c 00 | ,.1.0.0. 6.8.1.,, |
| 31 00 7B 00 7B 00 36 00 | 38 00 32 00 2c 00 31 00 | 1.0.0.6. 8.2.,.1. |
| 7B 00 36 00 36 00 38 00 | 33 00 2c 00 31 00 7B 00 | 0.0.6.8. 3.,.1.0. |
| 7B 00 36 00 38 00 34 00 | 2c 00 31 00 7B 00 7B 00 | 0.6.8.4. ,.1.0.0. |
| 36 00 38 00 35 00 2c 00 | 31 00 7B 00 7B 00 36 00 | 6.8.5.,, 1.0.0.6. |
| 38 00 36 00 2c 00 31 00 | 7B 00 7B 00 36 00 38 00 | 8.6.,.1. 0.0.6.8. |
| 37 00 2c 00 31 00 7B 00 | 7B 00 36 00 38 00 38 00 | 7.,.1.0. 0.6.8.8. |
| 2c 00 31 00 7B 00 7B 00 | 36 00 38 00 39 00 2c 00 | ,.1.0.0. 6.8.9.,, |
| 31 00 7B 00 7B 00 36 00 | 39 00 7B 00 2c 00 31 00 | 1.0.0.6. 9.0.,.1. |
| 7B 00 7B 00 36 00 39 00 | 31 00 2c 00 31 00 7B 00 | 0.0.6.9. 1.,.1.0. |
| 7B 00 36 00 39 00 32 00 | 2c 00 31 00 7B 00 7B 00 | 0.6.9.2. ,.1.0.0. |
| 36 00 39 00 34 00 2c 00 | 31 00 7B 00 7B 00 36 00 | 6.9.4.,, 1.0.0.6. |
| 39 00 35 00 2c 00 31 00 | 7B 00 7B 00 36 00 39 00 | 9.5.,.1. 0.0.6.9. |
| 36 00 2c 00 31 00 7B 00 | 7B 00 36 00 39 00 37 00 | 6.,.1.0. 0.6.9.7. |
| 2c 00 31 00 7B 00 7B 00 | 36 00 39 00 38 00 2c 00 | ,.1.0.0. 6.9.8.,, |
| 31 00 7B 00 7B 00 36 00 | 39 00 39 00 2c 00 31 00 | 1.0.0.6. 9.9.,.1. |
| 7B 00 7B 00 37 00 7B 00 | 7B 00 2c 00 31 00 7B 00 | 0.0.7.0. 0.,.1.0. |
| 7B 00 37 00 7B 00 31 00 | 2c 00 31 00 7B 00 7B 00 | 0.7.0.1. ,.1.0.0. |
| 37 00 7B 00 32 00 2c 00 | 31 00 7B 00 7B 00 37 00 | 7.0.2.,, 1.0.0.7. |
| 7B 00 33 00 2c 00 31 00 | 7B 00 7B 00 37 00 7B 00 | 0.3.,.1. 0.0.7.0. |
| 34 00 2c 00 31 00 7B 00 | 7B 00 37 00 7B 00 35 00 | 4.,.1.0. 0.7.0.5. |
| 2c 00 31 00 7B 00 7B 00 | 37 00 7B 00 36 00 2c 00 | ,.1.0.0. 7.0.6.,, |
| 31 00 7B 00 7B 00 37 00 | 7B 00 39 00 2c 00 31 00 | 1.0.0.7. 0.9.,.1. |
| 7B 00 7B 00 37 00 31 00 | 7B 00 2c 00 31 00 7B 00 | 0.0.7.1. 0.,.1.0. |
| 7B 00 37 00 31 00 31 00 | 2c 00 31 00 7B 00 7B 00 | 0.7.1.1. ,.1.0.0. |
| 37 00 31 00 32 00 2c 00 | 31 00 7B 00 7B 00 37 00 | 7.1.2.,, 1.0.0.7. |

| | | |
|-------------------------|-------------------------|-------------------|
| 31 00 33 00 2c 00 31 00 | 7B 00 7B 00 37 00 31 00 | 1.3.,.1. 0.0.7.1. |
| 34 00 2c 00 31 00 7B 00 | 7B 00 37 00 31 00 35 00 | 4.,.1.0. 0.7.1.5. |
| 2c 00 31 00 7B 00 7B 00 | 37 00 31 00 36 00 2c 00 | ,.1.0.0. 7.1.6.,. |
| 31 00 7B 00 7B 00 37 00 | 31 00 37 00 2c 00 31 00 | 1.0.0.7. 1.7.,.1. |
| 7B 00 7B 00 37 00 32 00 | 37 00 20 49 6f 88 | 0.0.7.2. 7. Io. |

Packet #5

| | | |
|-------------------------|-------------------------|-------------------|
| 00 90 27 11 51 1b 00 80 | 5f bd b7 ca 08 00 45 00 | ..'.Q... _.....E. |
| 02 d3 fc 81 40 00 80 06 | dc 8b 81 01 0e f5 81 01 |@... |
| 0e 20 06 f1 05 99 20 83 | d1 02 37 ef f6 e7 50 18 |7...P. |
| 1d 63 98 16 00 00 2c 00 | 31 00 7B 00 7B 00 37 00 | .c....., 1.0.0.7. |
| 32 00 38 00 2c 00 31 00 | 7B 00 7B 00 37 00 32 00 | 2.8.,.1. 0.0.7.2. |
| 39 00 2c 00 31 00 7B 00 | 7B 00 37 00 33 00 7B 00 | 9.,.1.0. 0.7.3.0. |
| 2c 00 31 00 7B 00 7B 00 | 37 00 33 00 31 00 2c 00 | ,.1.0.0. 7.3.1.,. |
| 31 00 7B 00 7B 00 37 00 | 33 00 32 00 2c 00 31 00 | 1.0.0.7. 3.2.,.1. |
| 7B 00 7B 00 37 00 33 00 | 33 00 2c 00 31 00 7B 00 | 0.0.7.3. 3.,.1.0. |
| 7B 00 37 00 33 00 34 00 | 2c 00 31 00 7B 00 7B 00 | 0.7.3.4. ,.1.0.0. |
| 37 00 33 00 35 00 2c 00 | 31 00 7B 00 7B 00 37 00 | 7.3.5.,. 1.0.0.7. |
| 33 00 36 00 2c 00 31 00 | 7B 00 7B 00 37 00 33 00 | 3.6.,.1. 0.0.7.3. |
| 37 00 2c 00 31 00 7B 00 | 7B 00 37 00 33 00 38 00 | 7.,.1.0. 0.7.3.8. |
| 2c 00 31 00 7B 00 7B 00 | 37 00 33 00 39 00 2c 00 | ,.1.0.0. 7.3.9.,. |
| 31 00 7B 00 7B 00 37 00 | 34 00 7B 00 2c 00 31 00 | 1.0.0.7. 4.0.,.1. |
| 7B 00 7B 00 37 00 34 00 | 32 00 2c 00 31 00 7B 00 | 0.0.7.4. 2.,.1.0. |
| 7B 00 37 00 34 00 33 00 | 2c 00 31 00 7B 00 7B 00 | 0.7.4.3. ,.1.0.0. |
| 37 00 34 00 34 00 2c 00 | 31 00 7B 00 7B 00 37 00 | 7.4.4.,. 1.0.0.7. |
| 34 00 35 00 2c 00 31 00 | 7B 00 7B 00 37 00 34 00 | 4.5.,.1. 0.0.7.4. |
| 36 00 2c 00 31 00 7B 00 | 7B 00 37 00 34 00 37 00 | 6.,.1.0. 0.7.4.7. |
| 2c 00 31 00 7B 00 7B 00 | 37 00 34 00 38 00 2c 00 | ,.1.0.0. 7.4.8.,. |
| 31 00 7B 00 7B 00 37 00 | 34 00 39 00 2c 00 31 00 | 1.0.0.7. 4.9.,.1. |
| 7B 00 7B 00 37 00 35 00 | 7B 00 2c 00 31 00 7B 00 | 0.0.7.5. 0.,.1.0. |
| 7B 00 37 00 35 00 31 00 | 2c 00 31 00 7B 00 7B 00 | 0.7.5.1. ,.1.0.0. |
| 37 00 35 00 32 00 2c 00 | 31 00 7B 00 7B 00 37 00 | 7.5.2.,. 1.0.0.7. |
| 35 00 33 00 2c 00 31 00 | 7B 00 7B 00 37 00 35 00 | 5.3.,.1. 0.0.7.5. |
| 34 00 2c 00 31 00 7B 00 | 7B 00 37 00 35 00 35 00 | 4.,.1.0. 0.7.5.5. |
| 2c 00 31 00 7B 00 7B 00 | 37 00 35 00 36 00 2c 00 | ,.1.0.0. 7.5.6.,. |
| 31 00 7B 00 7B 00 37 00 | 35 00 37 00 2c 00 31 00 | 1.0.0.7. 5.7.,.1. |
| 7B 00 7B 00 37 00 35 00 | 38 00 2c 00 31 00 7B 00 | 0.0.7.5. 8.,.1.0. |
| 7B 00 37 00 35 00 39 00 | 2c 00 31 00 7B 00 7B 00 | 0.7.5.9. ,.1.0.0. |
| 37 00 36 00 7B 00 2c 00 | 31 00 7B 00 7B 00 37 00 | 7.6.0.,. 1.0.0.7. |
| 36 00 31 00 2c 00 31 00 | 7B 00 7B 00 37 00 36 00 | 6.1.,.1. 0.0.7.6. |
| 32 00 2c 00 31 00 7B 00 | 7B 00 37 00 36 00 33 00 | 2.,.1.0. 0.7.6.3. |
| 2c 00 31 00 7B 00 7B 00 | 37 00 36 00 34 00 2c 00 | ,.1.0.0. 7.6.4.,. |
| 31 00 7B 00 7B 00 37 00 | 36 00 35 00 2c 00 31 00 | 1.0.0.7. 6.5.,.1. |
| 7B 00 7B 00 37 00 36 00 | 36 00 29 00 20 00 41 00 | 0.0.7.6. 6.) .A. |
| 4e 00 44 00 20 00 45 00 | 6e 00 74 00 49 00 44 00 | N.D. .E. n.t.I.D. |
| 2e 00 4c 00 73 00 74 00 | 4e 00 61 00 6d 00 20 00 | ..L.s.t. N.a.m. . |
| 6c 00 69 00 6b 00 65 00 | 20 00 27 00 73 00 6d 00 | l.i.k.e. .'.s.m. |
| 69 00 74 00 68 00 25 00 | 27 00 20 00 4f 00 52 00 | i.t.h.%. '.O.R. |
| 44 00 45 00 52 00 20 00 | 42 00 59 00 20 00 45 00 | D.E.R. . B.Y. .E. |
| 6e 00 74 00 49 00 44 00 | 2e 00 4e 00 61 00 6d 00 | n.t.I.D. ..N.a.m. |
| 53 00 74 00 72 00 2c 00 | 20 00 45 00 6e 00 74 00 | S.t.r.,. .E.n.t. |
| 49 00 44 00 2e 00 49 00 | 44 00 20 00 00 01 38 08 | I.D...I. D. ...8. |
| 00 00 00 00 01 38 01 20 | 00 00 00 01 38 00 00 00 |8.8... |

The SQL server will respond with the column headings or table description information:

| | | |
|-------------------------|-------------------------|-------------------|
| 00 80 5f bd b7 ca 00 90 | 27 11 51 1b 08 00 45 00 | .._..... '.Q...E. |
| 01 97 d8 b4 40 00 80 06 | 01 95 81 01 0e 20 81 01 |@... |
| 0e f5 05 99 06 f1 37 ef | f6 e7 20 83 d3 ad 50 18 |7.P. |
| 7f ff 20 be 00 00 04 01 | 01 6f 00 00 00 00 81 0c |o..... |
| 00 00 00 09 00 a7 82 00 | 06 4e 00 61 00 6d 00 53 |N.a.m.S |
| 00 74 00 72 00 00 00 09 | 00 a7 23 00 06 4c 00 73 | .t.r.... .#..L.s |
| 00 74 00 4e 00 61 00 6d | 00 00 00 09 00 a7 23 00 | .t.N.a.m#. |
| 06 46 00 73 00 74 00 4e | 00 61 00 6d 00 00 00 09 | .F.s.t.N .a.m.... |
| 00 a7 14 00 02 49 00 44 | 00 00 00 09 00 a7 0a 00 |I.D |
| 03 43 00 61 00 74 00 00 | 00 09 00 a7 14 00 05 45 | .C.a.t..E |

```

00 78 00 74 00 49 00 44      00 00 00 09 00 26 04 06      .x.t.I.D .....&..
4f 00 72 00 67 00 53 00      79 00 73 00 00 00 08 00      O.r.g.S. y.s.....
38 06 45 00 6e 00 74 00      53 00 79 00 73 00 00 00      8.E.n.t. S.y.s....
09 00 a7 11 00 03 53 00      73 00 6e 00 00 00 09 00      .....S. s.n.....
6f 08 03 44 00 6f 00 62      00 00 00 09 00 a7 02 00      o..D.o.b .....
03 53 00 65 00 78 00 00      00 00 00 38 07 72 00 6f      .S.e.x.. ...8.r.o
00 77 00 73 00 74 00 61      00 74 00 a4 1a 00 05 00      .w.s.t.a t.....
45 00 6e 00 74 00 49 00      44 00 06 00 45 00 6e 00      E.n.t.I. D...E.n.
74 00 44 00 65 00 6d 00      a5 24 00 01 01 00 02 01      t.D.e.m. .$.....
00 03 01 00 04 01 00 05      01 00 06 01 00 07 01 00      .....
08 01 00 09 02 00

```

The Terminal Server will acknowledge and send a fetch command

```

00 90 27 11 51 1b 00 80      5f bd b7 ca 08 00 45 00      ..'.Q..._.....E.
00 6c fd 81 40 00 80 06      dd f2 81 01 0e f5 81 01      .l..@... .....
0e 20 06 f1 05 99 20 83      d3 ad 37 ef f8 56 50 18      . .... .7..VP.
22 38 ac e2 00 00 03 01      00 44 00 00 01 00 0e 00      "8..... .D.....
73 00 70 00 5f 00 63 00      75 00 72 00 73 00 6f 00      s.p._.c. u.r.s.o.
72 00 66 00 65 00 74 00      63 00 68 00 02 00 00 00      r.f.e.t. c.h.....
38 5c 40 3d 7c 00 00 38      01 00 00 00 00 00 38 00      8\@=|..8 .....8.
00 00 00 00 00 38 64 00      00 00 69 38 1c 5f          .....8d. ..i8._

```

Then the SQL server will return the data in as many as five sequential packets stuffed to 1518 bytes for a very efficient transmission (note the data is returned without the period delimiter):

One packet shown only:

```

00 80 5f bd b7 ca 00 90      27 11 51 1b 08 00 45 00      .._..... '.Q...E.
05 dc d9 b4 40 00 80 06      fc 4f 81 01 0e 20 81 01      ....@... .O... ..
0e f5 05 99 06 f1 37 ef      f8 56 20 83 d3 f1 50 10      .....7. .V ...P.
7f bb 87 fa 00 00 04 00      10 00 00 00 00 00 d1 13      .....
00 53 4d 49 54 48 20 48      4f 55 53 45 20 52 45 53      .SMART H OUSE RES
2e 2c 20 20 12 00 53 4d      49 54 48 20 48 4f 55 53      ., ..SM ITH HOUS
45 20 52 45 53 2e 20 20      0b 00 20 20 20 20 20 20      E RES. ..
20 20 20 20 20 08 00 30      30 34 30 36 36 35 32 ff      ..0 0406652.
ff ff ff 04 37 00 00 00      7c 34 06 00 00 00 00 00      ....7... |4.....
00 01 00 00 00 d1 0b 00      53 4d 49 54 48 2c 20 41      ..... SMART, A
44 41 20 12 00 53 4d 49      54 48 20 20 20 20 20 20      DA ..SMI TH
20 20 20 20 20 20 20 0b      00 41 44 41 20 20 20 20      . .ADA
20 20 20 20 08 00 30 30      33 33 37 39 39 33 ff ff      ..00 337993..
ff ff 04 21 00 00 00 49      28 05 00 09 00 33 36 34      ...!....I (...364
33 34 33 38 39 38 08 48      1b 00 00 00 00 00 00 01      343898.H .....
00 46 01 00 00 00 d1 0f      00 53 4d 49 54 48 2c 20      .F..... .SMART,
41 44 41 4c 49 4e 45 20      12 00 53 4d 49 54 48 20      ADALINE ..SMART
20 20 20 20 20 20 20 20      20 20 20 20 07 00 41 44      ..AD
41 4c 49 4e 45 08 00 30      30 33 33 37 39 39 34 ff      ALINE..0 0337994.
ff ff ff 04 4f 00 00 00      4a 28 05 00 09 00 31 35      ....O... J(...15
38 32 32 37 37 30 35 08      60 09 00 00 00 00 00 00      8227705. `.....
01 00 46 01 00 00 00 d1      0e 00 53 4d 49 54 48 2c      ..F..... .SMART,
20 41 44 52 45 4e 41 20      12 00 53 4d 49 54 48 20      ADRENA ..SMART
20 20 20 20 20 20 20 20      20 20 20 20 06 00 41 44      ..AD
52 45 4e 41 08 00 30 30      33 33 37 39 39 35 ff ff      RENA..00 337995..
ff ff 04 3d 00 00 00 4b      28 05 00 09 00 32 35 30      ...=...K (...250
37 32 37 30 39 30 08 9e      42 00 00 00 00 00 00 01      727090.. B.....
00 46 01 00 00 00 d1 0f      00 53 4d 49 54 48 2c 20      .F..... .SMART,
41 44 52 45 4e 49 41 20      12 00 53 4d 49 54 48 20      ADRENIA ..SMART
20 20 20 20 20 20 20 20      20 20 20 20 0b 00 41 44      ..AD
52 45 4e 49 41 20 20 20      20 08 00 30 30 33 33 37      RENIA ..00337
39 39 36 ff ff ff ff 04      3d 00 00 00 4c 28 05 00      996..... =...L(..
09 00 30 30 30 30 30 30      30 30 30 08 60 42 00 00      ..000000 000.`B..
00 00 00 00 01 00 46 01      00 00 00 d1 0d 00 53 4d      .....F. ....SM
49 54 48 2c 20 41 4c 49      43 45 20 12 00 53 4d 49      ITH, ALI CE ..SMI

```

| | | | |
|-------------------------|-------------------------|------------|-----------------|
| 54 48 20 20 20 20 20 20 | 20 20 20 20 20 20 20 05 | TH | . |
| 00 41 4c 49 43 45 08 00 | 30 30 33 36 32 30 34 33 | .ALICE.. | 00362043 |
| 00 00 00 00 04 44 00 00 | 00 3b 86 05 00 09 00 30 |D.. | .;.....0 |
| 32 39 32 34 34 32 36 31 | 08 06 1e 00 00 00 00 00 | 29244261 | |
| 00 01 00 46 01 00 00 00 | d1 0f 00 53 4d 49 54 48 | ...F.... | ...SMART |
| 2c 20 41 4c 49 53 54 45 | 4e 20 12 00 53 4d 49 54 | , ALISTE N | ..SMIT |
| 48 20 20 20 20 20 20 20 | 20 20 20 20 20 20 07 00 | H | .. |
| 41 4c 49 53 54 45 4e 08 | 00 30 30 32 30 33 37 30 | ALISTEN. | .0020370 |
| 35 ff ff ff ff 04 24 00 | 00 00 b9 1b 03 00 09 00 | 5.....\$. | |
| 33 34 35 31 34 32 39 31 | 37 08 08 12 00 00 00 00 | 34514291 | 7..... |
| 00 00 01 00 46 01 00 00 | 00 d1 0c 00 53 4d 49 54 |F... |SMIT |
| 48 2c 20 41 4c 4d 41 20 | 12 00 53 4d 49 54 48 20 | H, ALMA | ..SMART |
| 20 20 20 20 20 20 20 20 | 20 20 20 20 04 00 41 4c | | ..AL |
| 4d 41 08 00 30 30 32 30 | 33 37 30 36 ff ff ff ff | MA..0020 | 3706.... |
| 04 24 00 00 00 ba 1b 03 | 00 09 00 32 35 36 34 36 | .\$..... | ...25646 |
| 33 31 39 34 08 12 36 00 | 00 00 00 00 00 01 00 46 | 3194..6. |F |
| 01 00 00 00 d1 0c 00 53 | 4d 49 54 48 2c 20 41 4c |S | MITH, AL |
| 4d 41 20 12 00 53 4d 49 | 54 48 20 20 20 20 20 20 | MA ..SMI | TH |
| 20 20 20 20 20 20 20 04 | 00 41 4c 4d 41 08 00 30 | | . ALMA..0 |
| 30 32 30 33 37 30 37 ff | ff ff ff 04 24 00 00 00 | 0203707. |\$... |
| bb 1b 03 00 09 00 33 35 | 33 33 30 30 39 39 32 08 |35 | 3300992. |
| 12 36 00 00 00 00 00 00 | 01 00 46 01 00 00 00 d1 | .6..... | ..F..... |
| 0e 00 53 4d 49 54 48 2c | 20 41 4e 47 45 4c 41 20 | ..SMART, | ANGELA |
| 12 00 53 4d 49 54 48 20 | 20 20 20 20 20 20 20 20 | ..SMART | |
| 20 20 20 20 06 00 41 4e | 47 45 4c 41 08 00 30 30 | | ..AN GELA..00 |
| 33 33 37 39 39 37 ff ff | ff ff 04 2b 00 00 00 4d | 337997.. | ...+...M |
| 28 05 00 09 00 30 30 30 | 30 30 30 30 30 30 08 9d | (....000 | 000000.. |
| 66 00 00 00 00 00 00 01 | 00 46 01 00 00 00 d1 0d | f..... | .F..... |
| 00 53 4d 49 54 48 2c 20 | 41 4e 49 54 41 20 12 00 | .SMART, | ANITA .. |
| 53 4d 49 54 48 20 20 20 | 20 20 20 20 20 20 20 20 | SMART | |
| 20 20 0b 00 41 4e 49 54 | 41 20 20 20 20 20 08 | | ..ANIT A |
| 00 30 30 33 38 34 31 31 | 38 00 00 00 00 04 24 00 | .0038411 | 8.....\$. |
| 00 00 76 dc 05 00 09 00 | 33 35 35 32 34 38 32 30 | ..v..... | 35524820 |
| 36 08 ec 20 00 00 00 00 | 00 00 01 00 46 01 00 00 | 6.. |F... |
| 00 d1 0b 00 53 4d 49 54 | 48 2c 20 41 4e 4e 20 12 |SMIT | H, ANN . |
| 00 53 4d 49 54 48 20 20 | 20 20 20 20 20 20 20 20 | .SMART | |
| 20 20 20 0b 00 41 4e 4e | 20 20 20 20 20 20 20 20 | | ..ANN |
| 08 00 30 30 33 39 35 32 | 32 30 ff ff ff ff 04 3d | ..003952 | 20.....= |
| 00 00 00 d4 07 06 00 09 | 00 30 32 38 33 32 33 39 | | .0283239 |
| 31 37 08 57 3f 00 00 00 | 00 00 00 01 00 46 01 00 | 17.W?... |F.. |
| 00 00 d1 0c 00 53 4d 49 | 54 48 2c 20 41 4e 4e 41 |SMI | TH, ANNA |
| 20 12 00 53 4d 49 54 48 | 20 20 20 20 20 20 20 20 | ..SMART | |
| 20 20 20 20 20 0b 00 41 | 4e 4e 41 20 20 20 20 20 | | ..A NNA |
| 20 20 08 00 30 30 34 31 | 33 30 37 38 ff ff ff ff | | ..0041 3078.... |
| 04 4f 00 00 00 96 4d 06 | 00 09 00 31 38 31 30 35 | .O....M. | ...18105 |
| 39 31 30 31 08 1a 14 00 | 00 00 00 00 00 01 00 46 | 9101.... |F |
| 01 00 00 00 d1 0c 00 53 | 4d 49 54 48 2c 20 41 4e |S | MITH, AN |
| 4e 45 20 12 00 53 4d 49 | 54 48 20 20 20 20 20 20 | NE ..SMI | TH |
| 20 20 20 20 20 20 20 04 | 00 41 4e 4e 45 08 00 30 | | . ANNE..0 |
| 30 32 30 33 37 30 38 ff | ff ff ff 04 24 00 00 00 | 0203708. |\$... |
| bc 1b 03 00 09 00 33 32 | 33 30 31 38 34 32 32 08 |32 | 3018422. |
| e4 11 00 00 00 00 00 00 | 01 00 46 01 00 00 00 d1 | | ..F..... |
| 0c 00 53 4d 49 54 48 2c | 20 41 4e 4e 45 20 12 00 | ..SMART, | ANNE .. |
| 53 4d 49 54 48 20 20 20 | 20 20 20 20 20 20 20 20 | SMART | |
| 20 20 0b 00 41 4e 4e 45 | 20 20 20 20 20 20 08 | | ..ANNE |
| 00 30 30 33 33 37 39 39 | 38 ff ff ff ff 04 3d 00 | .0033799 | 8.....=. |
| 00 00 4e 28 05 00 09 00 | 30 33 30 31 36 36 38 37 | ..N(.... | 03016687 |
| 33 08 46 1f 00 00 00 00 | 00 00 01 00 46 01 00 00 | 3.F..... |F... |
| 00 d1 12 00 53 4d 49 54 | 48 2c 20 41 4e 4e 4d 41 |SMIT | H, ANNMA |
| 52 49 45 20 4d 20 12 00 | 53 4d fd 36 64 82 | RIE M .. | SM.6d. |

Record #16 Captured on 10.18.00 at 17:21:05.927439097 Length = 1518

Recommendation SQL SERVER 2000 beta testing for increased performance and flexibility. However, with limited DBA level access to the current SQL engine and no official HEALTH CARE CO. DBA this option may not have any benefit. If HEALTH CARE CO. cannot qualify and correct the SQL server performance issues before moving them over to SQL Server 2000 some issues may be corrected but not any of the coding related ones.

Recommendation: Determine if the period delimiter is an issue.

Recommendation: Discover which queries and application functions are setting the Urgent Message Bit and research of SQL server is actually assigning a higher priority over other packets. This will require a SQL server support expert or a DBA familiar with this level of detail.

Recommendation: Look into using alternative methods of replication if the transaction based becomes too processing intensive. Using the Snapshot or Merge method may change the overall behavior of the application system's logical flow of transaction processing and data uniqueness, but may increase integrity and uptime plus reduce processing overhead on the SQL servers.

Recommendation: Optimizing Server Performance Using Memory Configuration Options
The memory manager component of Microsoft SQL Server eliminates the need for manual management of the memory available to SQL Server. When SQL Server starts, it dynamically determines how much memory to allocate based on how much memory the operating system and other applications are currently using. As the load on the computer and SQL Server changes, so does the memory allocated. The following server configuration options can be used to configure memory usage and affect server performance:

- **min server memory**
- **max server memory**
- **max worker threads**
- **index create memory**
- **min memory per query**

The **min server memory** server configuration option can be used to ensure that SQL Server starts with at least the minimum amount of allocated memory and does not release memory below this value. This configuration option can be set to a specific value based on the size and activity of your SQL Server. Always set the **min server memory** server configuration option to some reasonable value to ensure that the operating system does not request too much memory from SQL Server, affecting SQL Server performance.

The **max server memory** server configuration option can be used to specify the maximum amount of memory SQL Server can allocate when it starts and while it runs. This configuration option can be set to a specific value if you know there are multiple applications running at the same time as SQL Server and you want to guarantee that these applications have sufficient memory to run. If these other applications, such as Web or e-mail servers, request memory only as needed, then do not set the **max server memory** server configuration option, because SQL Server will release memory to them as needed. However, applications often use whatever memory is available when they start and do not request more if needed. If an application that behaves in this manner runs on the same computer at the same time as SQL Server, set the **max server memory** server configuration option to a value that guarantees that the memory required by the application is not allocated by SQL Server.

Do not set **min server memory** and **max server memory** server configuration options to the same value, thereby fixing the amount of memory allocated to SQL Server. Dynamic memory allocation will give you the best overall performance over time.

The **max worker threads** server configuration option can be used to specify the number of threads used to support the users connected to SQL Server. The default setting of 255 can be slightly too high for some configurations, depending on the number of concurrent users. Because each worker thread is allocated, even if it is not being used (because there are fewer concurrent connections than allocated worker threads), memory resources that can be better utilized by other operations, such as the buffer cache, can be unused. Generally, this configuration value should be set to the number of concurrent connections, but cannot exceed 1,024.

The **index create memory** server configuration option controls the amount of memory used by sort operations during index creation. Creating an index on a production system is usually an infrequently performed task, often scheduled as a job to execute during off-peak time. Therefore, when creating indexes infrequently and during off-peak time, increasing this number can improve the performance of index creation. Keep the **min memory per query** configuration option at a lower number, however, so the index creation job will still start even if all the requested memory is not available.

The **min memory per query** server configuration option can be used to specify the minimum amount of memory that will be allocated for the execution of a query. When there are many queries executing concurrently in a system, increasing the value of the **min memory per query** can help improve the performance of memory-intensive queries, such as substantial sort and hash operations. However, do not set the **min memory per query** server configuration option too high, especially on very busy systems, because the query will have to wait until it can secure the minimum memory requested or until the value specified in the **query wait** server configuration option is exceeded. If more memory is available than the specified minimum value required to execute the query, the query is allowed to make use of the additional memory, provided that the memory can be used effectively by the query.

Recommendation: Optimizing Server Performance Using I/O Configuration Options. The following server configuration options can be used to configure I/O usage and affect server performance:

- **max async IO**
- **recovery interval**

The **max async IO** server configuration option can be used to configure the maximum number of outstanding asynchronous disk I/O requests that the computer can issue against a file. Generally, leave the **max async IO** server configuration option at the default value of 32. For very sophisticated I/O subsystems with many disks and controllers, the value can be increased to 64 or possibly higher. Monitor disk write activity using Windows NT Performance Monitor, and watch for spikes caused by the lazy writer. If spikes occur that cause the **LogicalDisk: %Disk Time** counter to peak, then lower the value of this configuration option. If your I/O activity is low, consider raising this value.

The **recovery interval** server configuration option controls when SQL Server issues a checkpoint in each database. By default, SQL Server determines the best time to perform checkpoint operations. However, to determine if this is the appropriate setting, monitor disk write activity on the database files using Windows NT Performance Monitor. Spikes of activity that cause disk utilization to reach 100 percent can affect performance. Changing this parameter to cause the checkpoint process to occur less often can improve overall performance in this situation. However, continue to monitor performance to determine if the new value has had a positive effect on performance.

Recommendation: Verifying the cursor types used and utilize cursor optimization options.

Cursor Implementations

SQL Server supports three cursor implementations:

Transact-SQL cursors

Are based on the DECLARE CURSOR syntax and are used mainly in Transact-SQL scripts, stored procedures, and triggers. Transact-SQL cursors are implemented on the server. Transact-SQL statements sent from the client to the server, or contained in batches, stored procedures, or triggers manage them.

Application programming interface (API) server cursors

Support the API cursor functions in OLE DB, ODBC and DB-Library. API server cursors are implemented on the server. Each time a client application calls an API cursor function, the SQL Server OLE DB provider, ODBC driver, or DB-Library dynamic-link library (DLL) transmits the request to the server for action against the API server cursor.

Client cursors

Are implemented internally by the SQL Server ODBC driver, the DB-Library DLL, and by the DLL that implements the ADO API. Client cursors are implemented by caching all the result set rows on the client. Each time a client application calls an API cursor function, the SQL Server ODBC driver, the DB-Library DLL, or the ADO DLL performs the cursor operation on the result set rows cached on the client.

Because Transact-SQL cursors and API server cursors are implemented on the server, they are referred to collectively as server cursors.

Do not mix the use of these various types of cursors. If you execute a DECLARE CURSOR and OPEN statement from an application, first set the API cursor attributes to their defaults. If you set API cursor attributes to something other than their defaults and then execute a DECLARE CURSOR and OPEN statement, you are asking SQL Server to map an API cursor over a Transact-SQL cursor. For example, do not set the ODBC attributes that call for mapping a keyset-driven cursor over a result set, and then use that statement handle to execute a DECLARE CURSOR and OPEN calling for an INSENSITIVE cursor.

A potential drawback of server cursors is that they currently do not support all Transact-SQL statements. Server cursors do not support Transact-SQL statements that generate multiple result sets; therefore, they cannot be used when the application executes a stored procedure or a batch that contain more than one SELECT statement. Server cursors also do not support SQL statements that contain the keywords COMPUTE, COMPUTE BY, FOR BROWSE, or INTO. Please refer to any of the traces in this document or reference the complete set of traces left by AMI to determine the cursor types used.

Server Cursors vs. Default Result Sets

Using a cursor is less efficient than using a default result set. In a default result set the only packet sent from the client to the server is the packet containing the statement to execute. When using a server cursor, each FETCH statement must be sent from the client to the server, where it must be parsed and compiled into an execution plan.

If a Transact-SQL statement will return a relatively small result set that can be cached in the memory available to the client application and you know before executing the statement that you must retrieve the entire result set, use a default result set. Use server cursors only when cursor operations are required to support the functionality of the application, or when only part of the result set is likely to be retrieved.

Server Cursors vs. Client Cursors

There are several advantages to using server cursors instead of client cursors:

- Performance

If you are going to access a fraction of the data in the cursor (typical of many browsing applications), using server cursors provides optimal performance because only fetched data is sent over the network. Client cursors cache the entire result set on the client.
- Additional cursor types

If the SQL Server ODBC driver used only client cursors, it could support only forward-only and static cursors. By using API server cursors the driver can also support keyset-driven and dynamic cursors. SQL Server also supports the full range of cursor concurrency attributes only through server cursors. Client cursors are limited in the functionality they support.
- More accurate positioned updates

Server cursors directly support positioned operations, such as the ODBC **SQLSetPos** function or UPDATE and DELETE statements with the WHERE CURRENT OF clause. Client cursors, on the other hand, simulate positioned cursor updates by generating a Transact-SQL searched UPDATE statement, which leads to unintended updates if more than one row matches the WHERE clause conditions of the UPDATE statement.
- Memory usage

When using server cursors, the client does not need to cache large amounts of data or maintain information about the cursor position because the server does that.
- Multiple active statements

When using server cursors, no results are left outstanding on the connection between cursor operations. This allows you to have multiple cursor-based statements active at the same time.

The operation of all server cursors, except static or insensitive cursors, depends on the schema of the underlying tables. Any schema changes to those tables after a cursor has been declared results in an error on any subsequent operation on that cursor.

Cursor Programming Details

Choosing the correct cursor options is an important part of developing a SQL Server application.

Using block cursors can reduce the number of network roundtrips between the client and SQL Server, thereby improving performance. SQL Server may implicitly convert a cursor type if you execute a Transact-SQL statement not supported by the cursor type you requested. SQL Server populates the keyset of a large keyset-driven cursor asynchronously, which shortens the time between when the cursor is opened and when you can fetch the first rows.

Fast Forward-only Cursors

SQL Server implements a performance optimization called a fast forward-only cursor. Fast forward-only cursors are supported in two environments:

- Transact-SQL cursors can specify the `FAST_FORWARD` clause on `DECLARE CURSOR`. This opens an optimized forward-only, read-only cursor.
- Applications using the Microsoft SQL Server ODBC driver can set the driver-specific statement attribute `SQL_SOPT_SS_CURSOR_OPTIONS` to `SQL_CO_FFO` or `SQL_CO_FFO_AF`. Setting `SQL_CO_FFO` requests that the cursor be opened with the same optimizations as the `FAST_FORWARD` clause on `DECLARE CURSOR`. `SQL_CO_FFO_AF` request that an autofetch option also be enabled.

Using the Autofetch Option

Although some performance improvements are realized by specifying `FAST_FORWARD` on `DECLARE CURSOR`, or by specifying `SQL_CO_FFO` in ODBC applications, the most important performance gain comes from specifying `SQL_CO_FFO_AF` in ODBC applications to enable the autofetch option. Autofetch enables two optimizations that can significantly reduce network traffic:

- When the cursor is opened, the first row or batch of rows is automatically fetched from the cursor. This saves having to send a fetch request across the network.
- When a fetch hits the end of the cursor, the cursor is automatically closed. This saves having to send a separate close request across the network.

The most dramatic improvement is seen when processing cursors with relatively small result sets that can be cached in the memory of an application. The fast forward-only cursor with autofetch enabled represents the most efficient method of getting a result set into an ODBC application. When the autofetch option is on for a cursor containing n rows, an ODBC application can:

- Specify a rowset size of $n+1$.
- Allocate arrays of $n+1$ variables to hold the data from the result set columns.
- Bind the result set columns to the arrays.
- Execute the SQL statement that generates the cursor.

When the SQL Server ODBC driver executes the statement, it requests that the cursor be opened. Because autofetch is enabled, the server fetches and sends back n rows. The server fits as many rows as possible into each network packet returned to the client. When the server attempts to fetch the row at $n+1$ it detects the end of the cursor and automatically closes the cursor. When the application then executes `SQLCloseCursor` or `SQLFreeStmt` the ODBC driver does not have to send any close request to the server. The entire operation is done with only one packet being sent from the client to the server, and a minimal number of packets being returned from the server to the client.

Implicit Conversion of Fast Forward-only Cursors

Fast forward-only cursors are implicitly converted to other cursor types when:

- If the `SELECT` statement joins one or more tables with triggers to tables without triggers, the cursor is converted to a static cursor.
- If the `SELECT` statement references `text`, `ntext`, or `image` columns the cursor is converted to a dynamic cursor if the OLE DB Provider for SQL Server or the SQL Server ODBC driver are used.
- If a fast forward-only cursor is not read-only, it is converted to a dynamic cursor.

- If the SELECT statement is a distributed query that references one or more remote tables on linked servers, the cursor is converted to a keyset-driven cursor

Recommendation: Reviewing the types of indices used and determine if the following index types can be applied to increase performance.

Using Clustered Indexes

A clustered index determines the storage order of data in a table. A clustered index is analogous to a telephone directory, which arranges data by last name. Because the clustered index dictates the physical storage order of the data in the table, a table can contain only one clustered index. However, the index can comprise multiple columns (a composite index), like the way a telephone directory is organized by last name and first name.

A clustered index is particularly efficient on columns often searched for ranges of values. Once the row with the first value is found using the clustered index, rows with subsequent indexed values are guaranteed to be physically adjacent. For example, if an application frequently executes a query to retrieve records between date ranges, a clustered index can quickly locate the row containing the beginning date, and then retrieve all adjacent rows in the table until the last date is reached. This can help increase the performance of this type of query. Also, if there is a column(s), which is used frequently to sort the data retrieved from a table, it can be advantageous to cluster (physically sort) the table on that column(s) to save the cost of a sort each time the column(s) is queried.

Clustered indexes are also efficient for finding a specific row when the indexed value is unique. For example, the fastest way to find a particular employee using the unique employee ID column **emp_id** would be to create a clustered index or PRIMARY KEY constraint on the **emp_id** column.

Using Nonclustered Indexes

A nonclustered index is analogous to an index in a textbook. The data is stored in one place, the index in another, with pointers to the storage location of the data. The items in the index are stored in the order of the index key values, but the information in the table is stored in a different order (which can be dictated by a clustered index). If no clustered index is created on the table, the rows are not guaranteed to be in any particular order.

Similar to the way you use an index in a book, SQL Server searches for a data value by searching the nonclustered index to find the location of the data value in the table and then retrieves the data directly from that location. This makes nonclustered indexes the optimal choice for exact match queries because the index will contain entries describing the exact location in the table of the data values being searched for in the queries. If the underlying table is sorted using a clustered index, the location is the clustering key value; otherwise, the location is the row ID (RID) comprised of the file number, page number, and slot number of the row. For example, to search for an employee ID (**emp_id**) in a table that has a nonclustered index on the **emp_id** column, SQL Server looks through the index to find an entry that lists the exact page and row in the table where the matching **emp_id** can be found, and then goes directly to that page and row.

Using Unique Indexes

A unique index ensures that the indexed column contains no duplicate values. In the case of multicolumn unique indexes, the index ensures that each combination of values in the indexed column is unique. For example, if a unique index **full_name** is created on a combination of **last_name**, **first_name**, and **middle_initial** columns, no two people could have the same full name in the table.

Both clustered and nonclustered indexes can be unique. Therefore, provided that the data in the column is unique, you can create both a unique, clustered index and multiple unique, nonclustered indexes on the same table.

If the columns in a WHERE clause use the equals operator (=) with leading columns in the index, the query optimizer uses the unique index to process the query. This is because few rows will fit the query, so using the index has a minimal fixed cost. For example, if the query searched for the **last_name** and **first_name** columns, the **full_name** index would be used even though the **middle_initial** column is not specified.

Considerations

Specifying a unique index makes sense only when uniqueness is a characteristic of the data itself. If uniqueness must be enforced to ensure data integrity, create a UNIQUE or PRIMARY KEY constraint on the column rather than a unique index. For example, if you plan to query frequently on the Social Security number (**ssn**) column in the employee table (where the primary key is **emp_id**), and you want to ensure that Social Security numbers are unique, create a unique constraint on **ssn**. If the user enters the same Social Security number for more than one employee, an error is displayed.

Fill Factor

When you create a clustered index, the data in the table is stored in the data pages of the database according to the order of the values in the indexed columns. When new rows of data are inserted into the table or the values in the indexed columns are changed, Microsoft® SQL Server™ may have to reorganize the storage of the data in the table to make room for the new row and maintain the ordered storage of the data. This also applies to nonclustered indexes; when data is added or changed, SQL Server may have to reorganize the storage of the data in the nonclustered index pages.

When adding a new row to a full index page, SQL Server will move approximately half the rows to a new page to make room for the new row, a reorganization known as a page split. This can impair performance and fragment the storage of the data in a table.

When creating an index, you can specify a fill factor to leave extra gaps and reserve a percentage of free space on each leaf level page of the index to accommodate future expansion in the storage of the table's data and reduce the potential for page splits. The fill factor value is a percentage from 0 to 100 that specifies how much to fill the data pages after the index is created. A value of 100 means the pages will be full and will take the least amount of storage space. This setting should be used only when there will be no changes to the data, for example, on a read-only table. A lower value leaves more empty space on the data pages, which reduces the need to split data pages as indexes grow but requires more storage space. This setting is more appropriate when there will be changes to the data in the table.

The fill factor option is provided for fine-tuning performance. However, the server-wide default fill factor, specified using the **sp_configure** system stored procedure, is the best choice in the majority of situations.

Note Even for a heavy INSERT/UPDATE oriented application, the number of database reads typically outnumber database writes by a factor of 5 to 10. Therefore, specifying a fill factor other than the default can degrade database read performance by an amount inversely proportional to the fill factor setting. For example, a fill factor value of 50 percent can cause database read performance to degrade by two times.

Network Analysis

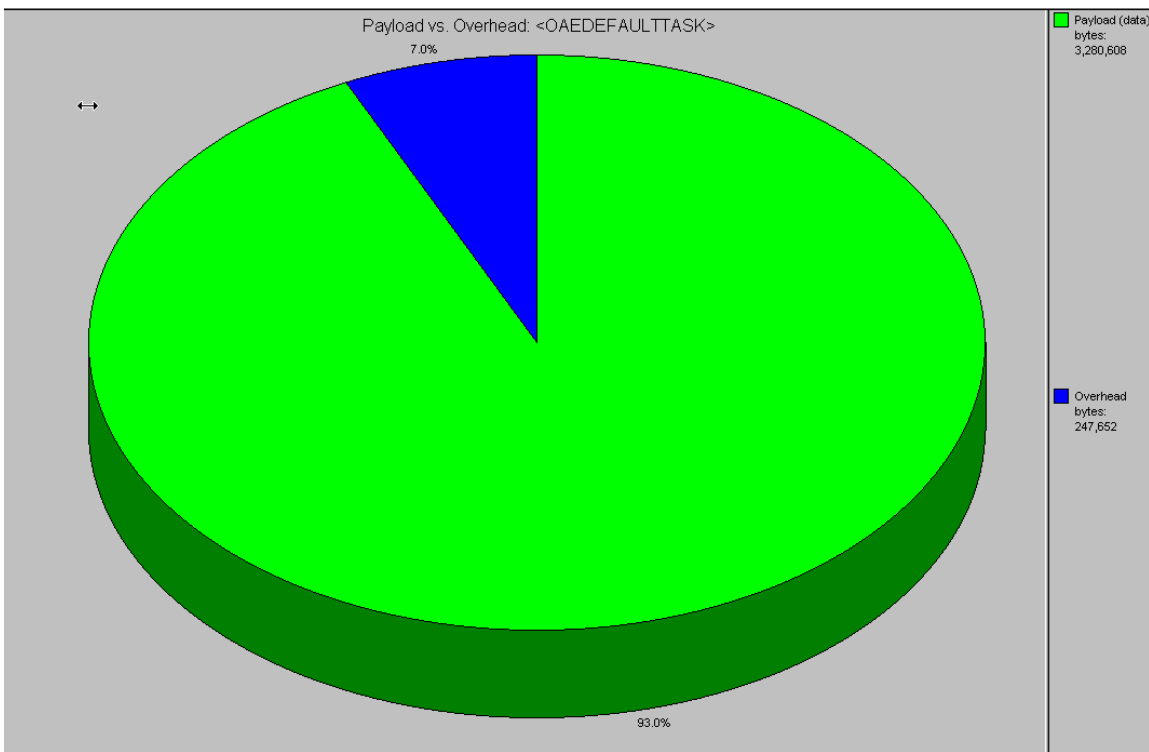
The Network aspect of this analysis included reviewing the major components that link the other systems components together. This includes all of the routers and switches in the path between a

Thin Client, Terminal and SQL server. Please refer to the diagram in the Professional Summary section for a basic review of the network

The network as a whole is the most efficient of the components analyzed. The current network infrastructure provides ample bandwidth for the home office and server segments and well for the remote offices. The Virtual LANs(Vlan) used to support the core of the network and the Terminal/SQL servers are handling and processing the traffic adequately.

The Vlan 14 segment (SQL/Terminal Server segment) average around 50% utilization with peaks up to 100%. However this is only looking at a 100mbps at a time on a segment that technically has 3.2 gigabytes allocated to it. The utilization on VLAN 10 (core segment) segment is much lower around 20%.

For the most part the payload to overhead distribution for SQL TDS packets is very efficient. The following pie chart depicts the percentage distribution from a large query.



The majority of the traffic analyzed was on Vlan 14 so our focus will start here

Excessive amounts of Window Zero were observed during the analysis. All of the Terminal Servers were advertising that their receive window is zero and sending it to the SQL servers. This activity is the result of the Terminal Servers becoming over burdened with data traffic. It was noted that the network interfaces on the Terminal Servers and SQL servers average from 1-2Mbps per second and have peaked as high as 8Mbps per second. Since these interfaces are running at 100mbps and operating in full duplex mode this is not an issue. The issue is further up the packet processing level particularly the IP stack.

The Terminal Server had a TCP receive window size of 8k. The receive window is a flow control mechanism for TCP to use to prevent it from getting overrun with data streams. Since TCP works with streams of data instead of fixed blocks like UDP, a flow control mechanism is needed. When a workstation or server is receiving a stream of data from another peer it will start to fill its receive

buffer. When that station sends an acknowledgment it will advertise how many bytes are left in the buffer, thus prompting the peer to continue to transmit. Usually the buffer will reset after a few packets and will not decrement too much. If the peer however, receives an acknowledgment packet with a window size of zero the peer will stop transmitting to that station until it receives an acknowledgment with a higher window size value. The result of this happening too often may cause unnecessary CPU utilization for the processing in the IP stack that may degrade the overall CPU performance of the server.

Another problem is that upper layer protocols for that TCP session may time out before the peer receives an acknowledgment packet with a new window size. The time-out may cause an entire packet to be re-transmitted thus increasing the number of packets on the network

Following is an excerpt of a packet trace during this activity between Terminal Servers 8 and 2 and SQL servers SBDBMS 2 and 3.

```
46:27.652 129.1.14.17 129.1.14.32 TCP S=4253 D=1433 ACK=396104054 WIN=0
46:27.941 129.1.14.10 129.1.14.31 TCP S=1841 D=1433 ACK=396473269 WIN=0
46:27.941 129.1.14.17 129.1.14.32 TCP S=4253 D=1433 ACK=396104054 WIN=0333
```

That is exactly what was happening on the Vlan 14 segment. All of the Terminal Servers were sending Zero Window packets to the SQL servers because the sheer volume of SQL and NetBIOS transactions were over running the IP stack. The SQL servers were also resending packets because of time-outs. A significant amount of retransmissions were also noted on this segment. The Zero Windows were noted in the other direction as well, that is the SQL servers were advertising them, but it was not nearly prevalent.

The Window size on the Terminal and SQL servers were set to the Windows NT default of 8k.

Recommendation: Increase the Window size to 32k on all servers

A change was made to all of the servers during the analysis period and the Window size was changed to 32k. Further analysis showed that this activity has significantly decreased. Whether this change showed any indicated performance improvement to the system as a whole is pending.

The presence of encapsulated NetBIOS on this segment was noted. NetBIOS is an older protocol and is considered in legacy status. This protocol was used for simple peer-to-peer file sharing between several computers on a non-routed network. Windows NT still supports this protocol in all of its versions. NetBIOS has no logical network layer addressing so it cannot be routed only bridged. For Microsoft to get around this it encapsulated the NetBIOS packet in a TCP/IP packet. The encapsulated NetBIOS packet can now traverse networks and is no longer limited to one broadcast domain. The encapsulation does add packet-processing overhead for the workstation or server's protocol stack and can result in additional CPU cycles to perform this operation. In addition the payload to overhead distribution in each packet is inefficient. In HEALTH CARE CO.'s case the NetBIOS traffic is only used on Vlan 14 and thus may not need to be encapsulated and with the volume of activity may mitigate some CPU overhead on the servers.

The Pathways applications use this protocol for some of its functions and these details will be explained in the Pathways Analysis section of this report.

The ratio of NetBIOS packets to SQL Server TDS packets on Vlan 14 is 75% to 25%.

Recommendation: Employ native Net BIOS protocol on all of the servers on Vlan 14 to further increase protocol efficiency and reduce protocol stack CPU processing.

Keep in mind that if the IT staff uses the Encapsulated NetBIOS to administer and transfer files between servers outside of Vlan 14 then this option may not be a viable one.

As stated in the executive summary the efficient use of sliding window operation was observed and is mostly used for SQL Server packets data resulting from queries originated from the Terminal Servers.

The following trace is an excerpt of what was observed

```

Frame# LEN  Time
 23 126 17:29:38.506374 129.1.14.245          sbdbms3.healthcarei  ETHER  00-80-5F-BD-B7-CA -> sbdbms3.healthcarei
129.1.14.245 -> sbdbms3.healthcarei Id=1120 TCP    t1090  -> t1433  Flags=.AP... Seq=0012881451 Ack=0150593358 TDS    RmtProcCall
 24 1518 17:29:38.511776 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=3d71 TCP    t1433  -> t1090  Flags=.A... Seq=0150593358 Ack=0012881519
 25 1518 17:29:38.511899 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=3e71 TCP    t1433  -> t1090  Flags=.A... Seq=0150594818 Ack=0012881519
 26 1234 17:29:38.512017 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=3f71 TCP    t1433  -> t1090  Flags=.AP... Seq=0150596278 Ack=0012881519
 27 1518 17:29:38.513459 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=4171 TCP    t1433  -> t1090  Flags=.A... Seq=0150597454 Ack=0012881519
 28 1518 17:29:38.513582 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=4271 TCP    t1433  -> t1090  Flags=.A... Seq=0150598914 Ack=0012881519
 29 1234 17:29:38.513700 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=4371 TCP    t1433  -> t1090  Flags=.AP... Seq=0150600374 Ack=0012881519
 30 64 17:29:38.518848 129.1.14.245          sbdbms3.healthcarei  ETHER  00-80-5F-BD-B7-CA -> sbdbms3.healthcarei
129.1.14.245 -> sbdbms3.healthcarei Id=1220 TCP    t1090  -> t1433  Flags=.A... Ack=0150596278
 31 64 17:29:38.518946 129.1.14.245          sbdbms3.healthcarei  ETHER  00-80-5F-BD-B7-CA -> sbdbms3.healthcarei
129.1.14.245 -> sbdbms3.healthcarei Id=1320 TCP    t1090  -> t1433  Flags=.A... Ack=0150598914
 32 64 17:29:38.519045 129.1.14.245          sbdbms3.healthcarei  ETHER  00-80-5F-BD-B7-CA -> sbdbms3.healthcarei
129.1.14.245 -> sbdbms3.healthcarei Id=1420 TCP    t1090  -> t1433  Flags=.A... Ack=0150601550
 33 1518 17:29:38.519894 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=4971 TCP    t1433  -> t1090  Flags=.A... Seq=0150601550 Ack=0012881519
 34 175 17:29:38.520012 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=4a71 TCP    t1433  -> t1090  Flags=.AP... Seq=0150603010 Ack=0012881519
 35 64 17:29:38.521317 129.1.14.245          sbdbms3.healthcarei  ETHER  00-80-5F-BD-B7-CA -> sbdbms3.healthcarei
129.1.14.245 -> sbdbms3.healthcarei Id=1520 TCP    t1090  -> t1433  Flags=.A... Ack=0150603127
 36 126 17:29:38.600033 129.1.14.245          sbdbms3.healthcarei  ETHER  00-80-5F-BD-B7-CA -> sbdbms3.healthcarei
129.1.14.245 -> sbdbms3.healthcarei Id=1620 TCP    t1090  -> t1433  Flags=.AP... Seq=0012881519 Ack=0150603127 TDS    RmtProcCall
 37 1518 17:29:38.604769 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=7071 TCP    t1433  -> t1090  Flags=.A... Seq=0150603127 Ack=0012881587
 38 1518 17:29:38.604892 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=7171 TCP    t1433  -> t1090  Flags=.A... Seq=0150604587 Ack=0012881587
 39 1234 17:29:38.605010 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=7271 TCP    t1433  -> t1090  Flags=.AP... Seq=0150606047 Ack=0012881587
 40 1518 17:29:38.606350 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=7371 TCP    t1433  -> t1090  Flags=.A... Seq=0150607223 Ack=0012881587
 41 1518 17:29:38.606474 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=7471 TCP    t1433  -> t1090  Flags=.A... Seq=0150608683 Ack=0012881587
 42 1234 17:29:38.606591 sbdbms3.healthcarei 129.1.14.245          ETHER  sbdbms3.healthcare00-80-5F-BD-B7-CA -> 0
Proto=IP  IP    sbdbms3.healthcarei -> 129.1.14.245 Id=7571 TCP    t1433  -> t1090  Flags=.AP... Seq=0150610143 Ack=0012881587
 43 64 17:29:38.611917 129.1.14.245          sbdbms3.healthcarei  ETHER  00-80-5F-BD-B7-CA -> sbdbms3.healthcarei
129.1.14.245 -> sbdbms3.healthcarei Id=1720 TCP    t1090  -> t1433  Flags=.A... Ack=0150606047

```


As you can see there are 6 packets sent from the SQL server all sized at 1518 bytes carrying a full payload of data before an acknowledgment is sent. This is an extremely efficient operation. This behavior is exceptional and shows that the network and the protocol stack on the servers are handling the data accordingly. It is ironic to note that in lieu of this preferred behavior the volume of such packets caused the Terminal Servers to choke resulting in the Zero Window issue.

After careful inspection of the routers in the home office and branch locations a slight discrepancy was noted as to the configuration of the Frame-Relay encapsulation. On the Home office routers the Frame-Relay encapsulation is set to IETF RFC 1490 and the remote office router is set to Cisco. The use of mismatched encapsulations is not a major issue for the CSU/DSU is processing both types. IETF RFC 1490 is a standard for multi-protocol over Frame-Relay. Since HEALTH CARE CO. is only using IP then IETF is not necessary. Cisco is the default encapsulation. Use the Cisco mode when connecting two Cisco devices and use the IETF mode when connecting your Cisco router to a non-Cisco router using Frame-Relay.

Recommendation: HEALTH CARE CO. should consult with their Frame-Relay provider to verify what encapsulation they are using and set both the Home office and remote routers accordingly.

IPX Type 20 propagation capability was noted on the LAN interfaces only for routers LAKESNY1 and 5.

NetBIOS over IPX uses Type 20 propagation broadcast packets that reach all networks to get information about the named nodes on the network. NetBIOS uses a broadcast mechanism to get this information, because it does not implement a network layer. Routers normally block all broadcast requests. By enabling Type 20 packet propagation, IPX interfaces on the router may accept and forward Type 20 packets.

When an interface configured for Type 20 propagation receives a Type 20 packet, Cisco IOS software processes the packet according to Novell specifications. Cisco IOS software propagates the packet to the next interface. The Type 20 packet can be propagated for up to eight hop counts.

Before forwarding (flooding) the packets, the router performs loop detection as described by the IPX router specification.

You can configure the Cisco IOS software to apply extra checks to Type 20 propagation packets above and beyond the loop detection described in the IPX specification. These checks are the same ones that are applied to help all-nets broadcast packets. They can limit unnecessary duplication of Type 20 broadcast packets. The extra helper checks are as follows:

- Accept Type 20 propagation packets only on the primary network, which is the network that is the primary path back to the source network.
- Forward Type 20 propagation packets only via networks that do not lead back to the source network.

Although this extra checking increases the robustness of Type 20 propagation packet handling by decreasing the amount of unnecessary packet replication, it has the following two side effects:

- If Type 20 packet propagation is not configured on all interfaces, these packets might be blocked when the primary interface changes.
- It might be impossible to configure an arbitrary, manual spanning tree for Type 20 packet propagation.

Recommendation: Remove the IPX type 20 if there is no longer a dependency for this feature across router segments.

The Frame-Relay links to the branch offices operational performance is exceptional. The packet flow is mostly between Thin Client and Terminal Servers. The behavior of this flow is different from that noted on the VLAN 10 and 14 segments. The packet flow observed on the Frame-Relay links is based on a chatty ping-pong type of flow using small packets sizes averaging 64 bytes and maximums up to 250 bytes at times. This ping-pong behavior is acceptable for the traffic type is derived from keyboard and mouse movement plus screens of information from the Terminal Servers being sent to the Thin Client. This type of traffic is considered chatty but since there is no volume of data sent at any one time ever sent to a Thin Client but small streams of data this is acceptable and works well on low speed WAN links. Below is an excerpt from a trace that shows the typical type of Thin Client packet flow:

| <i>Frm#</i> | <i>Len</i> | <i>Source</i> | <i>Destination</i> | | | | | | | |
|-------------|------------|---------------|--------------------|-------|----|-------|--------------|----------------|----------------|--|
| 3 | 67 | 172.27.239.10 | 129.1.14.19 | t1201 | -> | t1494 | Flags=.AP... | Seq=0082670452 | Ack=0161442956 | |
| 4 | 103 | 129.1.14.19 | 172.27.239.10 | t1494 | -> | t1201 | Flags=.AP... | Seq=0161442956 | Ack=0082670461 | |
| 5 | 67 | 172.27.239.10 | 129.1.14.19 | t1201 | -> | t1494 | Flags=.AP... | Seq=0082670461 | Ack=0161443001 | |
| 6 | 498 | 129.1.14.19 | 172.27.239.10 | t1494 | -> | t1201 | Flags=.AP... | Seq=0161443001 | Ack=0082670470 | |
| 7 | 67 | 172.27.239.10 | 129.1.14.19 | t1201 | -> | t1494 | Flags=.AP... | Seq=0082670470 | Ack=0161443441 | |

For specific charts outlining the utilization of the network segments, WAN circuits, protocol distribution and device statistics please see Appendix III Network Statistics and Graphs.

Thin Client Analysis

The Thin Clients operate in a consistent manner and there were no immediate issues uncovered in their operation. The performance and network analysis shows currently this component is operating efficiently.

The Thin Clients are manufactured from View Point Technologies. The model mostly deployed in the field is the TC200 Boundless. This unit provides a 486 processor running at 66Mhz and has 8mb of RAM. There is also no hard disk installed. This is acceptable for no Windows operating system is loaded locally. The Thin Client can use dial up PPP or TCP/IP to connect to a server. The Thin Client connects to the Terminal Server over the WAN using TCP/IP and sending data on the TCP ports of 1207 for ICA mouse and keyboard input and 1494 as the Terminal Server receiving port.

There were no significant trends noticed. However, during the isolated Terminal Server testing the Thin Client did outperform a Fat Client on some occasions. This suggests that the Thin Clients are not to be considered a bottleneck point at all in the transaction path due to their light processing overhead since only screen and keyboard/mouse data are sent and received from the units.

Recommendation: If performance begins to suffer even more and by protocol analysis HEALTH CARE CO. observes the 1K windows zeroing out, adjust the window to 8k.

Recommendation: If the Frame-Relay link becomes congested try utilizing the TCP Slow Start option to enhance performance. Use this option only if the Thin Client is timing out due to heavy traffic volumes and congestion on its Frame-Relay link

To avoid congestion, the TCP standard now recommends using two techniques: **slow-start** and **multiplicative decrease**. They are related and can be implemented easily. Each TCP connection must remember the size of the receiver's window. To control congestion TCP maintains a second limit called the congestion window limit or congestion window. In the steady state on a non-congested connection, the congestion window is the same size as the receiver's window.

Reducing the congestion window reduces the traffic TCP will inject into the connection. To estimate congestion window size, TCP assumes that most datagram loss comes from congestion and uses the following strategy:

Slow-Start (additive) recovery: Whenever starting traffic on a new connection or increasing traffic after a period of congestion, start the congestion window at the size of a single segment and increase the congestion window by one segment each time an acknowledgement arrives.

Multiplicative Decrease Congestion Avoidance: Upon loss of a segment, reduce the congestion window by half (down from a minimum of at least one segment). For those segments that remain in the allowed window, back off the retransmission timer exponentially.

The Thin Client does support slow start but the default is to not have it enabled.

Pathways Analysis

The analysis of Pathways was conducted by observing traffic patterns on Vlan 14 to understand the types of packet flow and transactions used between the Terminal Servers and SQL servers. A functional analysis of some of the widely used applications functions was performed against a single Fat Client on Vlan 14. This Fat Client in essence was basically a client acting as a single user Terminal Server. Our concentration was focused on what happens on the wire when Pathway program modules are loaded and actions are performed in the Pathways menu system. This analysis helps AMI to determine what and were any type performance issues may be present.

HEALTH CARE CO. IT support staff noted that the most widely used Pathways application modules are s follows:

Intake FastPad
Billing
Payroll
Scheduling
Customer Maintenance
Gentrans

AMI only tested some of the application functions in a controlled manner with HEALTH CARE CO. personnel. Testing of every function would require additional time and was beyond he scope of this document.

The application modules reacted in an inconsistent manner based on their function. For example some function will exchange data in a very efficient manner across the network, other functions will also utilize the network efficiently but delays were noted at the SQL servers. Still other functions used a combination of inefficient protocols such as (NetBIOS) to gather data from a different server and efficient protocol (TDS) to gather data from a SQL server.

Let's delve into the mechanics of some of these functions, starting with the loading of the Pathways application.

When the application loads there is some communication to the SQL servers occurring. This transmission activity results in 35 frames and completes in less than one second. This transmission consists of the application and SQL server exchanging version, language and ODBC information. Cursors are open for date and time information. This communication is a dependency so if there is a delay at the SQL server it will be felt by the user.

Loading of Intake FastPad from the menu also results in traffic. An hourglass was observed and this was not only due to the loading of executable modules which was discussed in the Terminal Server section of this report but that 306 packets sending queries and performing cursor operations for eight seconds is occurring in the background. Again, a dependency on the SQL server plus the overhead to loading the modules and the Terminal Server to Thin Client overhead could result in perceived delays to the user for a seamlessly normal operation. The network is handling the packets efficiently and responses are in the milliseconds.

An example of a cursor operation observed is below:

```
00000010: 00 f1 ec 10 40 00 80 06 ee de 81 01 0e f5 81 01 |.ñì.ë.....ö..
00000020: 0e 20 04 3d 05 99 00 a2 66 18 08 d6 cb ba 50 18 |. .=.™.çf..ÖË°P.
00000030: 20 7f 88 1e 00 00 03 01 00 c9 00 00 01 00 0d 00 | .....É.....
00000040: 73 00 70 00 5f 00 63 00 75 00 72 00 73 00 6f 00 |s.p._.c.u.r.s.o.
00000050: 72 00 6f 00 70 00 65 00 6e 00 00 00 00 01 38 00 |r.o.p.e.n....8.
00000060: 00 00 00 00 00 e7 80 00 80 00 53 00 45 00 4c 00 |.....ç....S.E.L.
00000070: 45 00 43 00 54 00 20 00 4e 00 61 00 6d 00 2c 00 |E.C.T. .N.a.m.,.
00000080: 20 00 50 00 77 00 64 00 2c 00 20 00 53 00 65 00 | .P.w.d.,. .S.e.
00000090: 63 00 4c 00 76 00 6c 00 2c 00 20 00 4f 00 72 00 |c.l.v.l.,. .O.r.
000000A0: 67 00 46 00 6c 00 74 00 20 00 46 00 52 00 4f 00 |g.F.l.t. .F.R.O.
000000B0: 4d 00 20 00 55 00 73 00 72 00 20 00 20 00 57 00 |M. .U.s.r. . .W.
000000C0: 48 00 45 00 52 00 45 00 20 00 55 00 73 00 72 00 |H.E.R.E. .U.s.r.
000000D0: 49 00 44 00 20 00 3d 00 20 00 27 00 46 00 41 00 |I.D. .=. .'.F.A.
000000E0: 52 00 42 00 49 00 32 00 27 00 00 01 38 08 00 00 |R.B.I.2.'...8...
000000F0: 00 00 01 38 01 20 00 00 00 01 38 00 00 00 00 |...8. ....8....
```

Again note the use of the period delimiter. This was discussed in the SQL server section but is prevalent in Pathways transactions.

There is a know problem that HEALTH CARE CO. has been grappling with regarding a runaway query in Intake FastPad. When a user in Intake FastPad uses the find tab to locate a patient named Smith for example. If the user inadvertently or purposely at the find prompt does not enter in any character but just leaves it blank and the enter key is pressed, the user will wait over 5 minutes while the application performs a full table scan against the SQL server for all patient names.

This results in an error message as follows:

“Runtime 6 overflow” with an okay button. The user presses the return key and the application will respond with **“runtime error 440 automation error”**

The user will then be exited from just the FastPad application module only but the Pathways menu is still available. AMI also observed that even though the user is no longer in FastPad the query is still being executed and the traffic is still present on the wire long after the error message was presented.

A detailed look into the packet traces provides the queries that are executed here is a piece of that initial query:

```
00000000: 00 90 27 11 51 1b 00 80 5f bd b7 ca 08 00 45 00 |...'Q..._...E.
00000010: 05 dc bb 18 40 00 80 06 1a ec 81 01 0e f5 81 01 |.Û..@....ì...ö..
00000020: 0e 20 04 40 05 99 00 ae 4e f8 08 e3 cd a3 50 10 |. .@.™..N..ãÍfP.
00000030: 1e 81 08 d2 00 00 03 00 10 00 00 00 01 00 0d 00 |...Ò.....
00000040: 73 00 70 00 5f 00 63 00 75 00 72 00 73 00 6f 00 |s.p._.c.u.r.s.o.
00000050: 72 00 6f 00 70 00 65 00 6e 00 00 00 00 01 38 00 |r.o.p.e.n....8.
00000060: 00 00 00 00 00 e7 8c 16 8c 16 53 00 65 00 6c 00 |.....çE.E.S.e.l.
00000070: 65 00 63 00 74 00 20 00 43 00 6f 00 75 00 6e 00 |e.c.t. .C.o.u.n.
```

```

00000080: 74 00 28 00 2a 00 29 00 20 00 46 00 52 00 4f 00 |t.(.*) .F.R.O.
00000090: 4d 00 20 00 45 00 6e 00 74 00 49 00 44 00 20 00 |M. .E.n.t.I.D. .
000000A0: 4c 00 45 00 46 00 54 00 20 00 4a 00 4f 00 49 00 |L.E.F.T. .J.O.I.
000000B0: 4e 00 20 00 45 00 6e 00 74 00 44 00 65 00 6d 00 |N. .E.n.t.D.e.m.
000000C0: 20 00 4f 00 4e 00 20 00 45 00 6e 00 74 00 49 00 | .O.N. .E.n.t.I.
000000D0: 44 00 2e 00 45 00 6e 00 74 00 53 00 79 00 73 00 |D...E.n.t.S.y.s.
000000E0: 20 00 3d 00 20 00 45 00 6e 00 74 00 44 00 65 00 | .=. .E.n.t.D.e.
000000F0: 6d 00 2e 00 45 00 6e 00 74 00 53 00 79 00 73 00 |m...E.n.t.S.y.s.
00000100: 20 00 57 00 48 00 45 00 52 00 45 00 20 00 45 00 | .W.H.E.R.E. .E.
00000110: 6e 00 74 00 49 00 44 00 2e 00 54 00 79 00 70 00 |n.t.I.D...T.y.p.
00000120: 20 00 3d 00 20 00 27 00 50 00 27 00 20 00 41 00 | .=. .'P.'. .A.
00000130: 4e 00 44 00 20 00 45 00 6e 00 74 00 49 00 44 00 |N.D. .E.n.t.I.D.
00000140: 2e 00 4f 00 72 00 67 00 53 00 79 00 73 00 20 00 |.O.r.g.S.y.s. .
00000150: 49 00 4e 00 20 00 28 00 31 00 2c 00 33 00 2c 00 |I.N. .(1.,.3.,.
00000160: 34 00 2c 00 35 00 2c 00 36 00 2c 00 32 00 30 00 |4.,.5.,.6.,.2.0.
00000170: 2c 00 32 00 31 00 2c 00 32 00 32 00 2c 00 32 00 |,.2.1.,.2.2.,.2.
00000180: 33 00 2c 00 32 00 34 00 2c 00 32 00 35 00 2c 00 |3.,.3.4.,.2.5.,.
00000190: 32 00 34 00 2c 00 32 00 37 00 2c 00 32 00 38 00 |2.6.,.2.7.,.1.8.
000001A0: 2c 00 32 00 39 00 2c 00 33 00 30 00 2c 00 33 00 |,.2.9.,.3.0.,.3.
000001B0: 31 00 2c 00 33 00 32 00 2c 00 33 00 33 00 2c 00 |1.,.3.2.,.3.3.,.

```

This query is in six packets long and a second similar query is executed before the stream of data is to be returned:

```

|.1..@....\...ö..
00000020: 0e 20 04 40 05 99 00 ae 65 d5 08 e3 ce 3e 50 18 |. .@.™...e...ã.>P.
00000030: 1d e6 21 c5 00 00 03 01 00 44 00 00 01 00 0e 00 |...!Å.....D.....
00000040: 73 00 70 00 5f 00 63 00 75 00 72 00 73 00 6f 00 |s.p._c.u.r.s.o.
00000050: 72 00 66 00 65 00 74 00 63 00 68 00 02 00 00 00 |r.f.e.t.c.h.....
00000060: 38 5c 00 2f 34 00 00 38 02 00 00 00 00 00 38 01 |8\./4..8.....8.

```

```

00000010: 00 57 c1 18 40 00 80 06 1a 71 81 01 0e f5 81 01 |.W..@....q...ö..
00000020: 0e 20 04 40 05 99 00 ae 66 19 08 e3 ce 67 50 18 |. .@.™...f...ã.gP.
00000030: 1d bd b6 1c 00 00 03 01 00 2f 00 00 01 00 0e 00 |...Ŧ...../.....
00000040: 73 00 70 00 5f 00 63 00 75 00 72 00 73 00 6f 00 |s.p._c.u.r.s.o.
00000050: 72 00 63 00 6c 00 6f 00 73 00 65 00 00 00 00 00 |r.c.l.o.s.e.....
00000060: 38 5c 00 2f 34 |8\./4

```

```

|...'Q..._...E.
00000010: 05 dc c2 18 40 00 80 06 13 ec 81 01 0e f5 81 01 |.Ü..@....ì...ö..
00000020: 0e 20 04 40 05 99 00 ae 66 48 08 e3 ce 7d 50 10 |. .@.™...fH.ã.}P.
00000030: 1d a7 15 70 00 00 03 00 10 00 00 00 01 00 0d 00 |.Ş.p.....
00000040: 73 00 70 00 5f 00 63 00 75 00 72 00 73 00 6f 00 |s.p._c.u.r.s.o.
00000050: 72 00 6f 00 70 00 65 00 6e 00 00 00 00 01 38 00 |r.o.p.e.n.....8.
00000060: 00 00 00 00 00 e7 d0 17 d0 17 53 00 45 00 4c 00 |.....çĐ.Đ.S.E.L.
00000070: 45 00 43 00 54 00 20 00 45 00 6e 00 74 00 49 00 |E.C.T. .E.n.t.I.
00000080: 44 00 2e 00 4e 00 61 00 6d 00 53 00 74 00 72 00 |D...N.a.m.S.t.r.
00000090: 2c 00 20 00 45 00 6e 00 74 00 49 00 44 00 2e 00 |,. .E.n.t.I.D...
000000A0: 4c 00 73 00 74 00 4e 00 61 00 6d 00 2c 00 20 00 |L.s.t.N.a.m.,. .
000000B0: 45 00 6e 00 74 00 49 00 44 00 2e 00 46 00 73 00 |E.n.t.I.D...F.s.
000000C0: 74 00 4e 00 61 00 6d 00 2c 00 20 00 45 00 6e 00 |t.N.a.m.,. .E.n.
000000D0: 74 00 49 00 44 00 2e 00 49 00 44 00 2c 00 20 00 |t.I.D...I.D.,. .
000000E0: 45 00 6e 00 74 00 49 00 44 00 2e 00 43 00 61 00 |E.n.t.I.D...C.a.
000000F0: 74 00 2c 00 20 00 45 00 6e 00 74 00 49 00 44 00 |t.,. .E.n.t.I.D.
00000100: 2e 00 45 00 78 00 74 00 49 00 44 00 2c 00 20 00 |.E.x.t.I.D.,. .
00000110: 45 00 6e 00 74 00 49 00 44 00 2e 00 4f 00 72 00 |E.n.t.I.D...O.r.
00000120: 67 00 53 00 79 00 73 00 2c 00 20 00 45 00 6e 00 |g.S.y.s.,. .E.n.
00000130: 74 00 49 00 44 00 2e 00 45 00 6e 00 74 00 53 00 |t.I.D...E.n.t.S.
00000140: 79 00 73 00 2c 00 20 00 45 00 6e 00 74 00 44 00 |y.s.,. .E.n.t.D.
00000150: 65 00 6d 00 2e 00 53 00 73 00 6e 00 2c 00 20 00 |e.m...S.s.n.,. .

```

```

00000160: 45 00 6e 00 74 00 44 00 65 00 6d 00 2e 00 44 00 |E.n.t.D.e.m...D.
00000170: 6f 00 62 00 2c 00 20 00 45 00 6e 00 74 00 44 00 |o.b.,. .E.n.t.D.
00000180: 65 00 6d 00 2e 00 53 00 65 00 78 00 20 00 46 00 |e.m...S.e.x. .F.
00000190: 52 00 4f 00 4d 00 20 00 45 00 6e 00 74 00 49 00 |R.O.M. .E.n.t.I.
000001A0: 44 00 20 00 4c 00 45 00 46 00 54 00 20 00 4a 00 |D. .L.E.F.T. .J.
000001B0: 4f 00 49 00 4e 00 20 00 45 00 6e 00 74 00 44 00 |O.I.N. .E.n.t.D.
000001C0: 65 00 6d 00 20 00 4f 00 4e 00 20 00 45 00 6e 00 |e.m. .O.N. .E.n.
000001D0: 74 00 49 00 44 00 2e 00 45 00 6e 00 74 00 53 00 |t.I.D...E.n.t.S.
000001E0: 79 00 73 00 20 00 3d 00 20 00 45 00 6e 00 74 00 |y.s. .=. .E.n.t.
000001F0: 44 00 65 00 6d 00 2e 00 45 00 6e 00 74 00 53 00 |D.e.m...E.n.t.S.
00000200: 79 00 73 00 20 00 57 00 48 00 45 00 52 00 45 00 |y.s. .W.H.E.R.E.
00000210: 20 00 45 00 6e 00 74 00 49 00 44 00 2e 00 54 00 |.E.n.t.I.D...T.
00000220: 79 00 70 00 20 00 3d 00 20 00 27 00 50 00 27 00 |y.p. .=. '.P.'.
00000230: 20 00 41 00 4e 00 44 00 20 00 45 00 6e 00 74 00 |.A.N.D. .E.n.t.
00000240: 49 00 44 00 2e 00 4f 00 72 00 67 00 53 00 79 00 |I.D...O.r.g.S.y.
00000250: 73 00 20 00 49 00 4e 00 20 00 28 00 31 00 2c 00 |s. .I.N. .(.1.,.
00000260: 33 00 2c 00 34 00 2c 00 35 00 2c 00 36 00 2c 00 |3.,.4.,.5.,.6.,.
00000270: 32 00 30 00 2c 00 32 00 31 00 2c 00 32 00 32 00 |2.0.,.2.1.,.2.2.
00000280: 2c 00 32 00 33 00 2c 00 32 00 34 00 2c 00 32 00 |,.2.3.,.2.4.,.2.
    
```

Recommendation: Perform research into correcting these queries or put in an application control or message box prompting the user not to do this. This action results in thousand of frames needlessly processed by the already overburdened Terminal and SQL servers.

There was no List of Value scrolling traffic observed on a list as patients that is returned to FastPad.

A search for all Patients named Smith resulted in an efficient result set in only 98 frames using sliding windows for a total time of 6 seconds.

However a different search show the variable levels of delays that result in inconsistent service levels and expectations for Thin Client users. An example is shown in the Intake FastPad search for patients named Johnson.

This simple search was longer and was clocked at 12 seconds. There were three delay points over two seconds which shows back to back delays from the database server.

The following delay was 3 seconds due to a cursor operation at frames 7 and 8.

| Relative | Sequence | Bytes | Payload | Source Node | Source Port | Destination Node | Dest. Port | Description |
|----------|----------|-------|---------|--------------|-------------|------------------|------------|--|
| 0.00500 | 7 | 60 | 0 | 129.1.14.32 | 1433 | 129.1.14.245 | 1092 | TCP:Ack Seq# 151133178 Ack# 13415567 Win 32767 |
| 3.94509 | 8 | 209 | 155 | 129.1.14.32 | 1433 | 129.1.14.245 | 1092 | MS_SQL:TDS[1] {CURDELETE ID(00000100), table:} |
| 3.94526 | 9 | 122 | 68 | 129.1.14.245 | 1092 | 129.1.14.32 | 1433 | MS_SQL:RPC (...) |

This delay was 4 seconds for another cursor operation at frame 20.

| Relative | Sequence | Bytes | Payload | Source Node | Source Port | Destination Node | Dest. Port | Description |
|----------|----------|-------|---------|--------------|-------------|------------------|------------|---|
| 4.00479 | 19 | 741 | 687 | 129.1.14.245 | 1092 | 129.1.14.32 | 1433 | MS_SQL:TDS Continuation |
| 4.21972 | 20 | 60 | 0 | 129.1.14.32 | 1433 | 129.1.14.245 | 1092 | TCP:Ack Seq# 151133396 Ack# 13421925 Win 32080 |
| 8.65914 | 21 | 421 | 367 | 129.1.14.32 | 1433 | 129.1.14.245 | 1092 | MS_SQL:TDS[1] {CURDELETE ID(00000900), table:} |
| 8.65948 | 22 | 122 | 68 | 129.1.14.245 | 1092 | 129.1.14.32 | 1433 | MS_SQL:RPC (...) |
| 8.65958 | 23 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1092 | MS_SQL:TDS[2] {ROW"JOHNSON,"}EVENTNOTICE OHNSON |

Another 2 second delay towards the end of the trace at frame 78.

| | | | | | | | | |
|----------|----|-----|-----|--------------|------|--------------|------|---|
| 9.11455 | 77 | 455 | 401 | 129.1.14.245 | 1092 | 129.1.14.32 | 1433 | MS_SQL:TDS Continuation |
| 9.11492 | 78 | 60 | 0 | 129.1.14.32 | 1433 | 129.1.14.245 | 1092 | TCP:Ack Seq# 151167650 Ack# 13428477 Win 32767 |
| 11.21975 | 79 | 342 | 288 | 129.1.14.32 | 1433 | 129.1.14.245 | 1092 | MS_SQL:TDS[1] {CURDELETE ID(00000800), table:A} |
| 11.22002 | 80 | 122 | 68 | 129.1.14.245 | 1092 | 129.1.14.32 | 1433 | MS_SQL:RPC (...) |


```

| . . D . I . . . . . ÔP .
| . $ô . . . . .
| s . p . _ . c . u . r . s . o .
| r . o . p . e . n . . . . . 8 .
| . . . . . ç . . . . . S . E . L .
| E . C . T . . E . n . t . I .
| D . . N . a . m . S . t . r .
| . . . . E . n . t . I . D . . .
| L . s . t . N . a . m . . . .
| E . n . t . I . D . . F . s .
| t . N . a . m . . . . E . n .
| t . I . D . . I . D . . . .
| E . n . t . I . D . . C . a .
| t . . . . E . n . t . I . D .
| . . . . E . x . t . I . D . . .
| E . n . t . I . D . . O . r .
| g . S . y . s . . . . E . n .
| t . I . D . . E . n . t . S .
| y . s . . . . E . n . t . D .
| e . m . . S . s . n . . . .
| E . n . t . D . e . m . . D .
| o . b . . . . E . n . t . D .
| e . m . . S . e . x . . F .
| R . O . M . . E . n . t . I .
| D . . L . E . F . T . . J .
| O . I . N . . E . n . t . D .
| e . m . . O . N . . E . n .
| t . I . D . . E . n . t . S .
| y . s . . = . . E . n . t .
| D . e . m . . E . n . t . S .
| y . s . . W . H . E . R . E .
| . . E . n . t . I . D . . T .
| y . p . . = . . ' P . '
| . . Â . N . D . . E . n . t .
| I . D . . O . r . g . S . y .
| s . . I . N . . ( . 1 . .
| 3 . . . 4 . . . 5 . . . 6 . .
| 2 . 0 . . . 2 . 1 . . . 2 . 2 .
| . . 2 . 3 . . . 2 . 4 . . . 2 .
| 5 . . . 2 . 6 . . . 2 . 7 . .
| 2 . 8 . . . 2 . 9 . . . 3 . 0 .
| . . 3 . 1 . . . 3 . 2 . . . 3 .

```

Otherwise, the data was sent in milliseconds using large packs and employing sliding windows.

This activity suggests that either the cursors and queries must be looked at or the database is very busy and is taking longer than necessary for simple operations.

Recommendation: Investigate into optimizing query/cursor performance or verifying the accuracy of these queries. Look into whether the Multiphase Query Optimizer is used.

Pathway application modules are dependent on open connections to the SQL server. This can enhance performance but at the same time cause a ripple effect to the end users, for performance issues at the SQL server can impact the modules running on the Terminal Servers.

An experiment was conducted to validate this behavior. On our test Fat Client the application was sitting idle in Intake FastPad. The network cable was pulled out to simulate a network failure or hiccup. The application then presented the following error message:

"Run Time error 4002 A8501(ms)(odbc sql server driver)(tcp/ip sockets)(general network error) when the OK button is pressed the user is then exited completely from the application altogether.

This behavior shows that the application is not very resilient to network or SQL server performance issues. It does not recover gracefully from simple network issues. The Winsock routines may not be optimized to return control back to the application efficiently.

Recommendation: Check into the application network library API function calls to determine if coding changes can be applied to the application so simple disruptions do not force a complete shutdown of the application for the user.

Recommendation: Check into Winsock implementation regarding graceful recovery if the application is not handling this activity. The use of Asynchronous based Winsock function should be in use.

Similar delays were noted in the Customer Maintenance application module. When performing a search to select a customer, delays over 2 seconds were noted from the database server.

Below is a five second delay from back to back SQL server packets

| Relative | Sequence | Bytes | Payload | Source Node | Source Port | Destination Node | Dest. Port | Description |
|----------|----------|-------|---------|--------------|-------------|------------------|------------|---|
| 27.89493 | 170 | 436 | 382 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | MS_SQL:TDS Continuation |
| 27.89528 | 171 | 60 | 0 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | TCP:Ack Seq# 152156948 Ack# 14471637 Win 32767 |
| 32.28610 | 172 | 93 | 39 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS[1] (CURDELETE ID(00000900), table:tc) |
| 32.34103 | 173 | 1,514 | 1,460 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | MS_SQL:SQL Language UNICODE(0)select ENTID.NAMSTF |

Here was the last Select statement used before the above delay:

```
MS_SQL:
SQL Language UNICODE(0)select count(*)as cnt FROM (ENTID INNER JOIN PAT ON ENTID.ENTSYS = PAT.PATSYS) LEFT JOIN ENTDEM ON ENTID.ENTSYS = ENTDEM.ENTSYS WHERE ENTID.ORGSYS IN (1,3,4,5,6,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,166,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,
```

This query continues for several packets.

A second 2 SQL server delay was noted again for a cursor operation:

| Relative | Sequence | Bytes | Payload | Source Node | Source Port | Destination Node | Dest. Port | Description |
|----------|----------|-------|---------|--------------|-------------|------------------|------------|--|
| 32.34574 | 179 | 894 | 840 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | MS_SQL:TDS Continuation |
| 32.50581 | 180 | 60 | 0 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | TCP:Ack Seq# 152156987 Ack# 14478033 Win 31927 |
| 34.72552 | 181 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS[1] (CURDELETE ID(00000900), table:) |
| 34.72673 | 182 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |

The query sent last before this delay was noted:

```
MS_SQL:
SQL Language UNICODE(0)select ENTID.NAMSTR AS NAME, ENTID.ID AS ID, ENTID.AC TRSTS AS ACT, ENTDEM.SSN AS SOCSEC, ENTDEM.DOB AS BIRTHDATE, ENTID.ORGSYS AS ORGSYS, ENTID.CAT AS CATEGORY, ENTDEM.SEX AS SEX, ENTID.EXTID AS EXTERNALID, ENTID.ENTSYS FROM (ENTID INNER JOIN PAT ON ENTID.ENTSYS = PAT.PATSYS) LEFT JOIN ENTDEM ON ENTID.ENTSYS = ENTDEM.ENTSYS WHERE ENTID.ORGSYS IN (1,3,4,5,6,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132
```

However once the initial delay has passed the SQL server does send the data in an efficient manner over the network:

| Relative | Sequence | Bytes | Payload | Source Node | Source Port | Destination Node | Dest. Port | Description |
|----------|----------|-------|---------|--------------|-------------|------------------|------------|--|
| 32.34574 | 179 | 894 | 840 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | MS_SQL:TDS Continuation |
| 32.50581 | 180 | 60 | 0 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | TCP:Ack Seq# 152156987 Ack# 14478033 Win 31927 |
| 34.72552 | 181 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS[1] (CURDELETE ID(00000900), table:) |
| 34.72673 | 182 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.72794 | 183 | 1,230 | 1,176 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.72892 | 184 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.73013 | 185 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.73159 | 186 | 60 | 0 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | TCP:Ack Seq# 14478033 Ack# 152159907 Win 8760 |
| 34.73139 | 187 | 60 | 0 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | TCP:Ack Seq# 14478033 Ack# 152162543 Win 8760 |
| 34.73144 | 188 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.73265 | 189 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.73386 | 190 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.73508 | 191 | 946 | 892 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.73583 | 192 | 60 | 0 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | TCP:Ack Seq# 14478033 Ack# 152165463 Win 8760 |
| 34.73588 | 193 | 60 | 0 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | TCP:Ack Seq# 14478033 Ack# 152168383 Win 8760 |
| 34.73593 | 194 | 60 | 0 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | TCP:Ack Seq# 14478033 Ack# 152169275 Win 8760 |
| 34.73598 | 195 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.73719 | 196 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.73840 | 197 | 1,230 | 1,176 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.73938 | 198 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.74059 | 199 | 1,514 | 1,460 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.74181 | 200 | 729 | 675 | 129.1.14.32 | 1433 | 129.1.14.245 | 1098 | MS_SQL:TDS Continuation |
| 34.74239 | 201 | 60 | 0 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | TCP:Ack Seq# 14478033 Ack# 152172195 Win 8760 |
| 34.74244 | 202 | 60 | 0 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | TCP:Ack Seq# 14478033 Ack# 152173371 Win 7584 |
| 34.74248 | 203 | 60 | 0 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | TCP:Ack Seq# 14478033 Ack# 152174831 Win 8760 |
| 34.74253 | 204 | 60 | 0 | 129.1.14.245 | 1098 | 129.1.14.32 | 1433 | TCP:Ack Seq# 14478033 Ack# 152176966 Win 8760 |

The TDS continuation packets containing the data:

```

. . . . . Q . . . E .
E2A@ . . . | O . . . .
. . . . . J . . . . U e N P .
| c { . . . A . . 347121
925 . . . . . $ . .
. . . . . F . . . N . . S
MITH . . . . .
00 . . . 8164 . . A . . 173
075387 . . . . .
O . . . . . M . . . O ( . . N
. . . . . WRIGHT
. . . . . 003381 . . A . . 5
36542127 . V ^ . . . .
. . . . . M . . . O ( . .
N . . . . . ZACHA
RY . . 003381 . . A
. . 00000000 . ^ | . .
. . . . . M . .
( . . N . . . . . ZA
CHARY . . 003381 . .
. . A . . . . 000674 . ^
| . . . . .
M . . ( . . N . . . .
ZEENA . . 00 . . 16
8 . . A . . 000 . . 000 .
    
```

Recommendation: Investigate into optimizing query/cursor performance or verifying the accuracy of these queries. The parsing and processing time at the SQL server regardless of whether it is due to the SQL server's sizing or an inefficient query. The delays do ripple back to the Thin Client user.

On the Fat Client the following observations were made in regards to the CPU utilization when loading different Pathways modules.

Loading Payroll 50% spike
Scheduling 60% spike

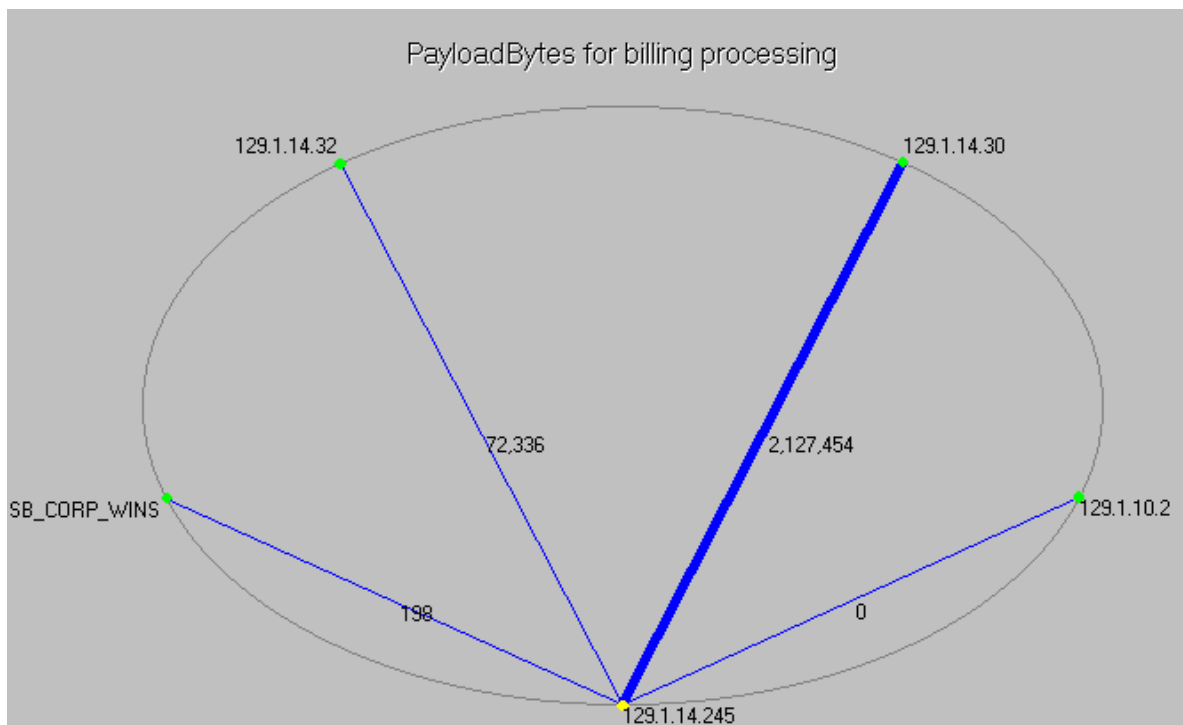
Bill Processing 70% spike
Intake FastPad 40% - slowest to load due to traffic dependencies to SQL server.

Please refer to the Terminal Server Analysis section for additional tests with the Fat Client.

Further analysis was conducted with HEALTH CARE CO. Pathways support personnel to step AMI through several popular user functions and recorded the performance activity. Here is what was discovered:

Bill Processing was interesting for we were able to observe first hand a tremendous delay in the View/Print Existing Claims function. There was no hourglass icon, which could leave the user to believe that application has frozen during the delay, plus there is no other control and the mouse pointer was no longer visible. The delay was calculated to over one minute of SQL TDS processing and another over 2 minutes of NetBIOS processing. There was no one back to back packet delay as in other functions. These delays are the results of just the volume of data being retrieved using inefficient methods. The following conversation map shows the relationships that are formed during this popularly used transaction.

The amount of data sent to and from the SQL server in the form of many cursor commands is represented by the node with the address of 129.1.14.32 and the NetBIOS traffic is represented by the address of 129.1.14.30. The Fat Client is 129.1.14.245.



As you can see the misappropriate use of the legacy protocol over the more efficient one is occurring. Why is NetBIOS being used??

To answer that question we will follow the transaction through its sequence of events. First, several Select commands and cursor operations are executed before some initial data is returned.

One of the commands is as follows:

```

L
|...Q...E.
|...@...ö...ö.
|...(.I...w...(wP.
|!$.
|e.l.e.c.t.*.
|F.R.O.M. U.s.r.
|c.l.s.o.r.g. W.
|H.E.R.E. C.l.s.
|I.D. i.n. (.s.
|e.l.e.c.t. C.l.
|s.I.D. f.r.o.m.
|.U.s.r.C.l.s.L.
|n.k. w.h.e.r.e.
|.U.s.r.I.D. =.
|'.E.M.O.R.A.N.
|').
    
```

After several Select command and cursor operations are performed, some data will be returned as such:

```

|...Q...E.
|Ütk@...a¹...
|ö.I.(.~...P.
|~@B..rket..Mark
|etN%...(.GARD
|EN GROVE.
|Garden
|Grove.....Loca
|tion..LocationN&
|.....INDIANAP
|OLIS..Indianapol
|is Area.....Ma
|rket..MarketN'..
|.....DALLAS ARE
|A..Dallas Area..
|.....Market..Mar
|ketN(.....KAN
|SAS..Kansas.....
|..Market..Market
|N).....SOUTHW
|EST MO..Southwes
|t Missouri.....
|.Market..MarketN
|*...2...SPASADE
|NACT..S Pasadena
|Certified.....
    
```

Then the following NetBIOS sequence starts:

| Relative | Sequence | Bytes | Payload | Source Node | Source Port | Destination Node | Dest. Port | Description |
|----------|----------|-------|---------|--------------|-------------|------------------|------------|--|
| 4.22510 | 143 | 101 | 47 | 129.1.14.245 | 1063 | 129.1.14.32 | 1433 | MS_SQL:RPC (...) |
| 4.23000 | 144 | 76 | 22 | 129.1.14.32 | 1433 | 129.1.14.245 | 1063 | MS_SQL:TDS[2] (RETURNSTATUS 0){DONEPROC (Comp |
| 4.33400 | 145 | 60 | 0 | 129.1.14.245 | 1063 | 129.1.14.32 | 1433 | TCP:Ack Seq# 1233377 Ack# 394142709 Win 8284 |
| 4.33405 | 146 | 60 | 0 | 129.1.14.245 | 1064 | 129.1.14.32 | 1433 | TCP:Ack Seq# 1234583 Ack# 394194900 Win 8470 |
| 17.90400 | 147 | 92 | 50 | 129.1.14.245 | 137 | SB_CORP_WINS | 137 | UDP_NETBIOS:Name Service Query Request for SBDBMS |
| 17.90407 | 148 | 104 | 62 | SB_CORP_WINS | 137 | 129.1.14.245 | 137 | UDP_NETBIOS:Name Service Query Response for SBDBM |
| 17.90900 | 149 | 60 | 0 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | TCP:Connection setup Seq# 1248695 Ack# 0 Win 8192 |
| 17.90905 | 150 | 60 | 0 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | TCP:Connection setup reply Seq# 394251364 Ack# 124869 |
| 17.90910 | 151 | 60 | 0 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | TCP:Ack Seq# 1248696 Ack# 394251365 Win 8760 |
| 17.90914 | 152 | 126 | 68 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | TCP_NETBIOS:Session Request Called=SBDBMS1 Calling= |
| 17.90924 | 153 | 60 | 0 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | TCP_NETBIOS:Positive Session Response |
| 17.90929 | 154 | 228 | 137 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | SMB:Negotiate Pid(CAFE) PC NETWORK PROGRAM 1.0(X |
| 17.90948 | 155 | 163 | 38 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | SMBReply:Negotiated Pid(CAFE) NT LM 0.12 |
| 18.00400 | 156 | 238 | 121 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | SMB:Session Setup (ANDX 75) Pid(CAFE) Client @ using W |
| 18.00419 | 157 | 212 | 115 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | SMBReply:Session Setup (ANDX 75) Pid(CAFE) Action 000C |
| 18.04000 | 158 | 320 | 203 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | SMB:Session Setup (ANDX 75) Pid(CAFE) Client far@STA |
| 18.04500 | 159 | 218 | 121 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | SMBReply:Session Setup (ANDX 75) Pid(CAFE) Action 000C |
| 18.05900 | 160 | 176 | 55 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | SMB:Transaction2(Query Path Info) Pid(CAFE) Info Level 00 |
| 18.05914 | 161 | 158 | 47 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | SMBReply:Transaction2(Query Path Info) Pid(CAFE) |
| 18.06500 | 162 | 138 | 47 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | SMB:Check Directory Pid(CAFE) Tid(0802)PwHC\WORKF |
| 18.06511 | 163 | 93 | 2 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | SMBReply:Check Directory Pid(CAFE) |
| 18.06518 | 164 | 186 | 65 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | SMB:Transaction2(Find First) Pid(6C40) Flags(00AA) Searchi |
| 18.06900 | 165 | 1,514 | 1,403 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | SMBReply:Transaction2(Find First) Pid(6C40) Number of Ent |
| 18.07021 | 166 | 1,514 | 1,456 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | TCP_NETBIOS:Session |
| 18.07142 | 167 | 1,494 | 1,436 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | TCP_NETBIOS:Session |
| 18.07262 | 168 | 60 | 0 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | TCP:Ack Seq# 1249730 Ack# 394254863 Win 8760 |
| 18.25400 | 169 | 60 | 0 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | TCP:Ack Seq# 1249730 Ack# 394256303 Win 7320 |
| 18.25900 | 170 | 1,514 | 1,403 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | SMBReply:Transaction2 Pid(6C40) |
| 18.26021 | 171 | 1,514 | 1,456 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | TCP_NETBIOS:Session |
| 18.26142 | 172 | 1,494 | 1,436 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | TCP_NETBIOS:Session |
| 18.26262 | 173 | 60 | 0 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | TCP:Ack Seq# 1249730 Ack# 394259223 Win 8760 |
| 18.45400 | 174 | 60 | 0 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | TCP:Ack Seq# 1249730 Ack# 394260663 Win 7320 |
| 18.45900 | 175 | 1,514 | 1,403 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | SMBReply:Transaction2 Pid(6C40) |
| 18.46100 | 176 | 1,514 | 1,456 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | TCP_NETBIOS:Session Message |
| 18.46221 | 177 | 1,494 | 1,436 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | TCP_NETBIOS:Session |
| 18.46341 | 178 | 60 | 0 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | TCP:Ack Seq# 1249730 Ack# 394263583 Win 8760 |
| 18.65400 | 179 | 60 | 0 | 129.1.14.245 | 1065 | 129.1.14.30 | 139 | TCP:Ack Seq# 1249730 Ack# 394265023 Win 7320 |
| 18.66000 | 180 | 1,514 | 1,403 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | SMBReply:Transaction2 Pid(6C40) |
| 18.66121 | 181 | 1,514 | 1,456 | 129.1.14.30 | 139 | 129.1.14.245 | 1065 | TCP_NETBIOS:Session |

As you can see there is a dependency to find the other SQL server but the question remains why is this NetBIOS present?

After reviewing the traces with HEALTH CARE CO. Pathways support personnel we uncovered that after the initial sequence of the transaction talks to the SQL server. The NetBIOS portion of the transaction, which is the larger of the two protocol operations, is the result of the application pulling/reading static claim numbers from a file on a different server.

This NetBIOS sequence continues for several thousand frames and is far less efficient than the observed sliding window based TDS sequences.

This is what the NetBIOS packets contain from the static file on a different server

```

. . . . . | . . . . . E
. . . . . @ . . . . . ?
. . . . . ) . . . . . ã . . . . . ÓP
. . . . . f . . . . . SMB2
. . . . . |
. . . . . b . . . . . ?
. . . . . 8 . . . . . 8 . . . . . ! . . . . . Í
. . . . . C . L . 0 . B . 8 . 3
. . . . . ~ . 1 . . . . . M . D . B . C . L
. . . . . M . 3 . 5 . 6 . 2 . 3 . 0 . . . .
. . . . . M . D . B . x . . . . . $ |
. . . . . | i u * . . . . . \ . . . . . 4 . . . .
. . . . . Z . 1 . . . . . | ñ . { ^ * . . . . . È
. . . . .
. . . . . C . L .
. . . . . 1 . B . 8 . 7 . ~ . 2 . . . . . M
. . . . . D . B . C . L . M . 3 . 5 . 6 .
. . . . . 2 . 5 . 0 . . . . . M . D . B . x
. . . . . . . . . . . d . ð i u * . . . . . ã
. . . . . $ . . . . . 4 . . . . . 7 . Û \ 1 . . . . . 2 >
. . . . . ð { ^ * . . . . . È . . . . .
. . . . .
. . . . . C . L . 1 . 3 . 9 . 7 .
. . . . . ~ . 2 . . . . . M . D . B . C . L
. . . . . M . 3 . 5 . 6 . 2 . 5 . 2 . . . .
. . . . . M . D . B . x . . . . . ð j
. . . . . . ý u * . . . . . C . 4 . . . . . ã
. . . . . . ] 1 . . . . . O " | ^ * . . . . . È
. . . . .
. . . . . P . . . . .
. . . . . C . L .
. . . . . 1 . 7 . 9 . 7 . ~ . 2 . . . . . M
. . . . . D . B . C . L . M . 3 . 5 . 6 .
. . . . . 2 . 5 . 3 . . . . . M . D . B . x
. . . . . . . . . . . L . s ý u * . . . . . ~
. . . . . ` . . . . . 4 . . . . . d J @ ] 1 . . . . . | b
. . . . . Q | ^ * . . . . . È . . . . .
    
```

Other application function may use the NetBIOS transport, as was verified in general traces and protocol distribution statistics gathered. AMI did not review all application functions for this activity would be beyond the scope of this project. So, it currently is unclear exactly how many different application functions use TDS and NetBIOS in a transaction.

Recommendation: Consider eliminating the NetBIOS portion of the communication and having the static claim number information in another table for more efficient retrieval.

The Scheduling function also had a major delay when performing an office search for every employee with their name beginning with R. Again there was no hourglass to depict to the user a waiting period. This is not a Terminal Server presentation issue for these tests were done on a Fat Client and the application modules are executed locally.

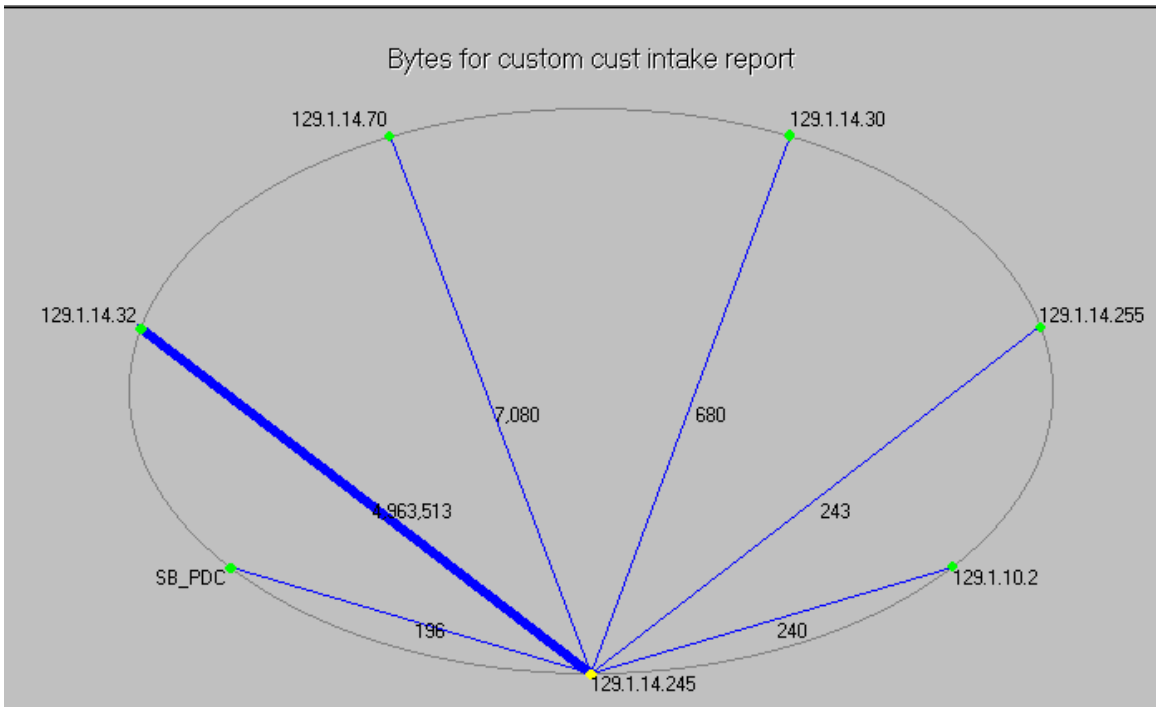
A major delay of 25 seconds was noted from back to back packets during a DML process:

| Relative | Sequence | Bytes | Payload | Source Node | Source Port | Destination Node | Dest. Port | Description |
|----------|----------|-------|---------|--------------|-------------|------------------|------------|---|
| 9.29100 | 21 | 1,514 | 1,460 | 129.1.14.245 | 1082 | 129.1.14.32 | 1433 | MS_SQL:TDS Continuation |
| 9.29221 | 22 | 352 | 298 | 129.1.14.245 | 1082 | 129.1.14.32 | 1433 | MS_SQL:TDS Continuation |
| 9.29249 | 23 | 60 | 0 | 129.1.14.32 | 1433 | 129.1.14.245 | 1082 | TCP:Ack Seq# 395247453 Ack# 2307414 Win 32767 |
| 34.34100 | 24 | 93 | 39 | 129.1.14.32 | 1433 | 129.1.14.245 | 1082 | MS_SQL:TDS[1] {CURDELETE ID(00000900), table:tc} |
| 34.35000 | 25 | 1,514 | 1,460 | 129.1.14.245 | 1082 | 129.1.14.32 | 1433 | MS_SQL:SQL Language UNICODE(0)select DISTINCT ENTII |
| 34.35500 | 26 | 1,514 | 1,460 | 129.1.14.245 | 1082 | 129.1.14.32 | 1433 | MS_SQL:TDS Continuation |
| 34.35621 | 27 | 60 | 0 | 129.1.14.32 | 1433 | 129.1.14.245 | 1082 | TCP:Ack Seq# 395247492 Ack# 2310334 Win 32767 |

Recommendation: Consider making a change to the application code to exhibit some form of message or hourglass during these operations. Have these messages be consistent across all functions. And again look into query optimization options.

The Custom Customer Intake Report Payer Report function also reported tremendous delays during the analysis.

This delay is attributed the overwhelming number of cursor and DML activity repeated plus the dependency to communicate to other servers. The Dependency are depicted in the following conversation map:



The trace the cursor and DML activity is repeated for 14000+ times thus resulting in a 7 minute delay due to the chatty nature of this transaction. An excerpt of the chatty nature of the packet flow is below. Note the frame numbers and relative time in seconds.

| | Relative | Sequence | Bytes | Payload | Source Node | Source Port | Destination Node | Dest. Port | Description |
|-------|-----------|----------|-------|---------|--------------|-------------|------------------|------------|---|
| 14728 | 436.35100 | 14728 | 235 | 181 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14729 | 436.41500 | 14729 | 1,126 | 1,072 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[1] (CURDELETE ID(00000800), table:P) |
| 14730 | 436.42100 | 14730 | 178 | 124 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14731 | 436.43000 | 14731 | 281 | 227 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (RETURNSTATUS 0){DONEPROC (Mc |
| 14732 | 436.44000 | 14732 | 122 | 68 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14733 | 436.44600 | 14733 | 85 | 31 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (DONEINPROC (More results to come,R |
| 14734 | 436.44607 | 14734 | 122 | 68 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14735 | 436.46000 | 14735 | 301 | 247 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[1] (ROW " VIRGINIA SUFFOLK PEI |
| 14736 | 436.46600 | 14736 | 233 | 179 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14737 | 436.51500 | 14737 | 1,126 | 1,072 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[1] (CURDELETE ID(00000800), table:P) |
| 14738 | 436.52500 | 14738 | 178 | 124 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14739 | 436.54000 | 14739 | 257 | 203 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (RETURNSTATUS 0){DONEPROC (Mc |
| 14740 | 436.54600 | 14740 | 122 | 68 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14741 | 436.55500 | 14741 | 85 | 31 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (DONEINPROC (More results to come,R |
| 14742 | 436.56100 | 14742 | 233 | 179 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14743 | 436.60500 | 14743 | 1,126 | 1,072 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[1] (CURDELETE ID(00000800), table:P) |
| 14744 | 436.61000 | 14744 | 178 | 124 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14745 | 436.63100 | 14745 | 222 | 168 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[5] (RETURNSTATUS 0){DONEPROC (Mc |
| 14746 | 436.63600 | 14746 | 122 | 68 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14747 | 436.64500 | 14747 | 85 | 31 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (DONEINPROC (More results to come,R |
| 14748 | 436.65100 | 14748 | 233 | 179 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14749 | 436.69500 | 14749 | 1,126 | 1,072 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[1] (CURDELETE ID(00000800), table:P) |
| 14750 | 436.70100 | 14750 | 178 | 124 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14751 | 436.71600 | 14751 | 282 | 228 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (RETURNSTATUS 0){DONEPROC (Mc |
| 14752 | 436.72100 | 14752 | 122 | 68 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14753 | 436.72600 | 14753 | 85 | 31 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (DONEINPROC (More results to come,R |
| 14754 | 436.73100 | 14754 | 235 | 181 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14755 | 436.77000 | 14755 | 1,126 | 1,072 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[1] (CURDELETE ID(00000800), table:P) |
| 14756 | 436.77600 | 14756 | 178 | 124 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14757 | 436.79000 | 14757 | 516 | 462 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (RETURNSTATUS 0){DONEPROC (Mc |
| 14758 | 436.80000 | 14758 | 122 | 68 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14759 | 436.81000 | 14759 | 85 | 31 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (DONEINPROC (More results to come,R |
| 14760 | 436.81007 | 14760 | 122 | 68 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14761 | 436.82100 | 14761 | 85 | 31 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (DONEINPROC (More results to come,R |
| 14762 | 436.82107 | 14762 | 122 | 68 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-RPC (...) |
| 14763 | 436.82600 | 14763 | 85 | 31 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (DONEINPROC (More results to come,R |
| 14764 | 436.82607 | 14764 | 126 | 72 | 129.1.14.245 | 1084 | 129.1.14.32 | 1433 | MS_SQL-SQL Language UNICODE(0)if @@TRANCOUN |
| 14765 | 436.83500 | 14765 | 89 | 35 | 129.1.14.32 | 1433 | 129.1.14.245 | 1084 | MS_SQL-TDS[3] (DONE (More results to come)){DONE (f |

Recommendation: Research into optimizing the number of cursor functions and mitigate the chatty nature of this function. The network can handle and process these packets but the sheer volume causes the application to wait and digest all of this information.
 be available for backward compatibility

Appendices: Analysis Statistics, Graphs And Supplemental Information

Appendix I - Terminal Server information

No additional information.

Appendix II - SQL Server information

No additional information.

Appendix III - Network Statistics and Graphs

Router and switch log information

```
Lake> (enable) sh top 65 inter 300
Start Time: 10/09/2000,12:40:57
End Time: 10/09/2000,12:45:58
PortType: all
Metric: util
```

| Port | Band- width | Uti % | Bytes (Tx + Rx) | Pkts (Tx + Rx) | Bcst (Tx + Rx) | Mcst (Tx + Rx) | Error (Rx) | Over flow |
|------|----------------|----------|--------------------|-------------------|-------------------|-------------------|---------------|--------------|
| 3/1 | 10 | 90 | 322786496 | 903393 | 533 | 2929 | 0 | 0 |
| 4/16 | 10 | 4 | 14814549 | 25731 | 332 | 227 | 0 | 0 |
| 4/10 | 100 | 3 | 227445135 | 328480 | 753 | 220 | 0 | 0 |
| 4/8 | 100 | 2 | 218930761 | 310889 | 753 | 220 | 0 | 0 |
| 3/11 | 100 | 2 | 194171551 | 281623 | 753 | 220 | 0 | 0 |
| 4/7 | 100 | 2 | 161898754 | 289951 | 753 | 220 | 0 | 0 |
| 5/6 | 100 | 2 | 155207144 | 235584 | 752 | 220 | 0 | 0 |
| 5/3 | 100 | 1 | 108247813 | 183944 | 753 | 220 | 0 | 0 |
| 3/7 | 100 | 1 | 97371682 | 165771 | 753 | 220 | 0 | 0 |
| 5/8 | 100 | 1 | 93742023 | 176507 | 752 | 220 | 0 | 0 |
| 1/1 | 100 | 1 | 72613222 | 247331 | 0 | 247331 | 2003 | 0 |
| 4/3 | 10 | 1 | 3683272 | 9456 | 390 | 252 | 0 | 0 |
| 4/5 | 100 | 0 | 70140150 | 203557 | 753 | 220 | 0 | 0 |
| 4/6 | 100 | 0 | 61885976 | 176821 | 753 | 220 | 0 | 0 |
| 4/2 | 10 | 0 | 2685216 | 10251 | 278 | 219 | 0 | 0 |
| 4/24 | 100 | 0 | 50156424 | 334249 | 191 | 292 | 0 | 0 |
| 3/12 | 100 | 0 | 52019763 | 115638 | 753 | 220 | 0 | 0 |
| 3/9 | 100 | 0 | 48393975 | 106095 | 753 | 220 | 0 | 0 |
| 5/24 | 100 | 0 | 42624512 | 78735 | 753 | 219 | 0 | 0 |
| 5/7 | 100 | 0 | 43493266 | 53206 | 752 | 220 | 0 | 0 |
| 3/6 | 100 | 0 | 36568918 | 105049 | 753 | 220 | 0 | 0 |
| 4/11 | 100 | 0 | 34265765 | 113375 | 753 | 220 | 0 | 0 |
| 3/8 | 100 | 0 | 30704528 | 66378 | 753 | 220 | 0 | 0 |
| 4/9 | 100 | 0 | 18659097 | 43410 | 753 | 220 | 0 | 0 |
| 5/20 | 100 | 0 | 17270586 | 32114 | 589 | 292 | 0 | 0 |
| 5/10 | 100 | 0 | 13147834 | 19506 | 752 | 220 | 0 | 0 |
| 4/20 | 100 | 0 | 11886768 | 16479 | 585 | 220 | 0 | 0 |
| 1/2 | 100 | 0 | 8478118 | 42497 | 0 | 42497 | 140 | 0 |
| 5/9 | 100 | 0 | 6463064 | 14946 | 752 | 220 | 0 | 0 |
| 4/1 | 10 | 0 | 316259 | 2481 | 218 | 219 | 0 | 0 |
| 4/17 | 10 | 0 | 264872 | 2288 | 145 | 231 | 0 | 0 |
| 3/15 | 100 | 0 | 5026180 | 26417 | 587 | 292 | 0 | 0 |
| 3/17 | 100 | 0 | 4744308 | 26554 | 587 | 292 | 0 | 0 |
| 3/24 | 100 | 0 | 3991198 | 9539 | 0 | 220 | 0 | 0 |
| 3/23 | 10 | 0 | 197841 | 1533 | 95 | 219 | 0 | 0 |
| 4/19 | 100 | 0 | 3269939 | 7622 | 585 | 220 | 0 | 0 |
| 5/21 | 100 | 0 | 3095540 | 11996 | 589 | 292 | 0 | 0 |
| 3/22 | 100 | 0 | 2805401 | 9728 | 586 | 225 | 0 | 0 |
| 5/1 | 10 | 0 | 132013 | 1633 | 753 | 220 | 0 | 0 |
| 4/15 | 10 | 0 | 121270 | 662 | 407 | 220 | 0 | 0 |
| 3/13 | 10 | 0 | 104814 | 562 | 337 | 221 | 0 | 0 |
| 5/16 | 100 | 0 | 1904448 | 3865 | 588 | 292 | 0 | 0 |
| 4/14 | 10 | 0 | 92441 | 486 | 266 | 220 | 0 | 0 |

| | | | | | | | | |
|------|-----|---|---------|-------|-----|-----|---|---|
| 5/23 | 100 | 0 | 1748353 | 7555 | 753 | 219 | 0 | 0 |
| 5/12 | 100 | 0 | 1480319 | 9500 | 753 | 220 | 0 | 0 |
| 4/22 | 100 | 0 | 1171171 | 5713 | 585 | 220 | 0 | 0 |
| 3/18 | 100 | 0 | 1037547 | 12478 | 587 | 292 | 0 | 0 |
| 5/17 | 100 | 0 | 986271 | 4862 | 588 | 292 | 0 | 0 |
| 4/21 | 100 | 0 | 707357 | 4467 | 585 | 220 | 0 | 0 |
| 5/4 | 100 | 0 | 659224 | 6347 | 752 | 220 | 0 | 0 |
| 3/16 | 100 | 0 | 611657 | 2611 | 587 | 292 | 0 | 0 |
| 3/21 | 100 | 0 | 547330 | 2663 | 587 | 297 | 0 | 0 |
| 3/20 | 100 | 0 | 292058 | 2226 | 753 | 220 | 0 | 0 |
| 4/18 | 100 | 0 | 285065 | 1448 | 585 | 220 | 0 | 0 |
| 4/12 | 100 | 0 | 195714 | 1614 | 753 | 220 | 0 | 0 |
| 3/19 | 100 | 0 | 158621 | 845 | 586 | 220 | 0 | 0 |
| 5/14 | 100 | 0 | 156590 | 821 | 585 | 219 | 0 | 0 |
| 4/13 | 100 | 0 | 155430 | 811 | 585 | 220 | 0 | 0 |
| 4/23 | 100 | 0 | 156704 | 879 | 588 | 291 | 0 | 0 |
| 5/19 | 100 | 0 | 163118 | 936 | 588 | 292 | 0 | 0 |
| 3/14 | 100 | 0 | 177382 | 1195 | 587 | 292 | 0 | 0 |
| 3/5 | 100 | 0 | 85788 | 973 | 753 | 220 | 0 | 0 |
| 5/11 | 100 | 0 | 89037 | 995 | 752 | 220 | 0 | 0 |
| 5/13 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5/15 | 100 | 0 | 25256 | 290 | 8 | 220 | 0 | 0 |

Lake (enable) sh top 65 interval 300
 Start Time: 10/09/2000,12:46:23
 End Time: 10/09/2000,12:51:25
 PortType: all
 Metric: util

| Port | Band- width | Uti % | Bytes (Tx + Rx) | Pkts (Tx + Rx) | Bcst (Tx + Rx) | Mcst (Tx + Rx) | Error (Rx) | Over flow |
|------|----------------|----------|--------------------|-------------------|-------------------|-------------------|---------------|--------------|
| 1/1 | 100 | 1 | 79984427 | 237075 | 0 | 237075 | 21876 | 0 |
| 1/2 | 100 | 0 | 65244413 | 165476 | 0 | 165476 | 1610 | 0 |
| 2/15 | 10 | 0 | 1935320 | 9899 | 276 | 220 | 0 | 0 |
| 2/3 | 10 | 0 | 1360089 | 7254 | 276 | 220 | 0 | 0 |
| 2/27 | 10 | 0 | 826777 | 6036 | 238 | 221 | 60 | 0 |
| 2/13 | 10 | 0 | 855096 | 4497 | 276 | 220 | 0 | 0 |
| 3/32 | 10 | 0 | 618303 | 3887 | 238 | 221 | 0 | 0 |
| 2/2 | 10 | 0 | 580332 | 5079 | 276 | 220 | 79 | 0 |
| 5/18 | 10 | 0 | 506478 | 3257 | 277 | 219 | 97 | 0 |
| 2/45 | 10 | 0 | 503042 | 3661 | 238 | 221 | 0 | 0 |
| 4/6 | 10 | 0 | 544616 | 2402 | 277 | 219 | 0 | 0 |
| 5/2 | 10 | 0 | 403649 | 3246 | 277 | 219 | 0 | 0 |
| 4/7 | 10 | 0 | 451898 | 1901 | 277 | 219 | 0 | 0 |
| 3/43 | 10 | 0 | 440916 | 1897 | 238 | 221 | 0 | 0 |
| 5/47 | 10 | 0 | 382847 | 2319 | 228 | 221 | 0 | 0 |
| 4/41 | 10 | 0 | 363630 | 1877 | 224 | 221 | 10 | 0 |
| 5/27 | 10 | 0 | 357968 | 2181 | 231 | 221 | 0 | 0 |
| 5/42 | 10 | 0 | 358512 | 2202 | 229 | 221 | 0 | 0 |
| 3/11 | 10 | 0 | 358812 | 2718 | 276 | 220 | 0 | 0 |
| 2/24 | 10 | 0 | 348619 | 2636 | 276 | 220 | 0 | 0 |
| 2/46 | 10 | 0 | 344638 | 2488 | 238 | 221 | 0 | 0 |
| 3/10 | 10 | 0 | 332910 | 2039 | 276 | 220 | 8 | 0 |
| 3/48 | 10 | 0 | 320031 | 2329 | 238 | 221 | 7 | 0 |
| 3/30 | 10 | 0 | 278625 | 2113 | 238 | 221 | 37 | 0 |
| 2/11 | 10 | 0 | 277425 | 2338 | 276 | 220 | 0 | 0 |
| 4/21 | 10 | 0 | 220681 | 1629 | 277 | 219 | 0 | 0 |
| 3/3 | 10 | 0 | 214279 | 2004 | 276 | 220 | 0 | 0 |
| 4/9 | 10 | 0 | 210901 | 1187 | 277 | 219 | 0 | 0 |
| 4/3 | 10 | 0 | 192927 | 1943 | 277 | 219 | 4 | 0 |
| 2/16 | 10 | 0 | 189343 | 1214 | 276 | 220 | 1 | 0 |
| 2/10 | 10 | 0 | 189383 | 2008 | 238 | 221 | 0 | 0 |
| 4/24 | 10 | 0 | 170601 | 1113 | 277 | 219 | 15 | 0 |
| 3/19 | 10 | 0 | 164910 | 1735 | 276 | 220 | 0 | 0 |
| 3/24 | 10 | 0 | 160147 | 1834 | 276 | 220 | 0 | 0 |
| 3/4 | 10 | 0 | 156926 | 1717 | 276 | 220 | 0 | 0 |
| 4/20 | 10 | 0 | 156530 | 1095 | 277 | 219 | 0 | 0 |
| 4/8 | 10 | 0 | 155213 | 1400 | 277 | 219 | 1 | 0 |
| 5/32 | 10 | 0 | 150756 | 1458 | 230 | 221 | 0 | 0 |
| 3/44 | 10 | 0 | 152553 | 1400 | 238 | 221 | 0 | 0 |
| 5/29 | 10 | 0 | 149710 | 1217 | 230 | 221 | 0 | 0 |
| 2/6 | 10 | 0 | 148982 | 1138 | 276 | 220 | 0 | 0 |
| 3/18 | 10 | 0 | 135766 | 1379 | 276 | 220 | 0 | 0 |
| 4/43 | 10 | 0 | 134933 | 1281 | 235 | 221 | 0 | 0 |
| 2/47 | 10 | 0 | 134621 | 1188 | 238 | 221 | 3 | 0 |

| | | | | | | | | |
|------|----|---|--------|------|-----|-----|---|---|
| 3/13 | 10 | 0 | 132736 | 1372 | 276 | 220 | 0 | 0 |
| 2/32 | 10 | 0 | 131100 | 1273 | 238 | 221 | 0 | 0 |
| 2/36 | 10 | 0 | 131541 | 1279 | 238 | 221 | 0 | 0 |
| 5/8 | 10 | 0 | 125644 | 1333 | 277 | 219 | 0 | 0 |
| 2/39 | 10 | 0 | 124440 | 1323 | 238 | 221 | 0 | 0 |
| 2/42 | 10 | 0 | 125558 | 1276 | 238 | 221 | 0 | 0 |
| 3/2 | 10 | 0 | 125426 | 1265 | 276 | 220 | 0 | 0 |
| 2/8 | 10 | 0 | 122388 | 1317 | 276 | 220 | 0 | 0 |
| 5/41 | 10 | 0 | 117769 | 1240 | 229 | 221 | 0 | 0 |
| 2/17 | 10 | 0 | 117064 | 1153 | 276 | 220 | 0 | 0 |
| 4/1 | 10 | 0 | 116415 | 1129 | 166 | 221 | 0 | 0 |
| 2/44 | 10 | 0 | 107122 | 949 | 238 | 221 | 0 | 0 |
| 3/15 | 10 | 0 | 102899 | 1055 | 276 | 220 | 0 | 0 |
| 5/28 | 10 | 0 | 102655 | 1094 | 230 | 221 | 0 | 0 |
| 2/21 | 10 | 0 | 103367 | 1108 | 276 | 220 | 0 | 0 |
| 5/23 | 10 | 0 | 99696 | 1068 | 277 | 219 | 0 | 0 |
| 2/20 | 10 | 0 | 101951 | 1043 | 276 | 220 | 0 | 0 |
| 2/43 | 10 | 0 | 100372 | 974 | 238 | 221 | 0 | 0 |
| 5/43 | 10 | 0 | 101708 | 1116 | 278 | 220 | 0 | 0 |
| 2/48 | 10 | 0 | 101144 | 1045 | 238 | 221 | 0 | 0 |
| 4/16 | 10 | 0 | 100330 | 1100 | 277 | 219 | 0 | 0 |

```
Lake> (enable) sh top 65 interval 300
Start Time: 10/09/2000,17:43:42
End Time: 10/09/2000,17:48:44
PortType: all
Metric: util
```

| Port | Band- width | Uti % | Bytes (Tx + Rx) | Pkts (Tx + Rx) | Bcst (Tx + Rx) | Mcst (Tx + Rx) | Error (Rx) | Over flow |
|------|----------------|----------|--------------------|-------------------|-------------------|-------------------|---------------|--------------|
| 2/25 | 10 | 0 | 1217600 | 8277 | 270 | 220 | 0 | 0 |
| 2/44 | 10 | 0 | 784124 | 1877 | 270 | 220 | 180 | 0 |
| 5/47 | 10 | 0 | 530899 | 6367 | 256 | 218 | 0 | 0 |
| 3/29 | 10 | 0 | 414180 | 3646 | 266 | 220 | 0 | 0 |
| 5/40 | 10 | 0 | 431421 | 1750 | 256 | 218 | 1 | 0 |
| 5/21 | 10 | 0 | 417756 | 1851 | 249 | 221 | 0 | 0 |
| 5/48 | 10 | 0 | 405509 | 2330 | 256 | 218 | 0 | 0 |
| 1/1 | 100 | 0 | 7758543 | 38829 | 0 | 38829 | 1064 | 0 |
| 3/17 | 10 | 0 | 248041 | 1547 | 248 | 221 | 0 | 0 |
| 2/17 | 10 | 0 | 229537 | 1126 | 248 | 221 | 0 | 0 |
| 4/28 | 10 | 0 | 184908 | 1263 | 260 | 218 | 0 | 0 |
| 3/37 | 10 | 0 | 176516 | 1236 | 265 | 220 | 0 | 0 |
| 4/14 | 10 | 0 | 170840 | 1935 | 300 | 219 | 1 | 0 |
| 4/12 | 10 | 0 | 165302 | 1501 | 248 | 220 | 3 | 0 |
| 2/46 | 10 | 0 | 154326 | 1173 | 270 | 220 | 0 | 0 |
| 2/15 | 10 | 0 | 152090 | 1228 | 248 | 221 | 0 | 0 |
| 4/40 | 10 | 0 | 143050 | 1606 | 259 | 218 | 0 | 0 |
| 3/6 | 10 | 0 | 131509 | 1335 | 248 | 221 | 0 | 0 |
| 5/18 | 10 | 0 | 124843 | 1284 | 249 | 221 | 0 | 0 |
| 5/23 | 10 | 0 | 124587 | 1271 | 249 | 221 | 0 | 0 |
| 2/39 | 10 | 0 | 121420 | 1141 | 270 | 220 | 7 | 0 |
| 2/19 | 10 | 0 | 120203 | 1277 | 248 | 221 | 0 | 0 |
| 2/42 | 10 | 0 | 120722 | 1148 | 270 | 220 | 0 | 0 |
| 3/27 | 10 | 0 | 111830 | 1206 | 267 | 220 | 0 | 0 |
| 4/39 | 10 | 0 | 108122 | 1266 | 259 | 218 | 0 | 0 |
| 4/42 | 10 | 0 | 106441 | 1247 | 258 | 218 | 0 | 0 |
| 4/46 | 10 | 0 | 106441 | 1247 | 258 | 218 | 0 | 0 |
| 2/4 | 10 | 0 | 97362 | 1052 | 300 | 221 | 0 | 0 |
| 2/48 | 10 | 0 | 96313 | 1005 | 270 | 220 | 0 | 0 |
| 2/30 | 10 | 0 | 96090 | 997 | 270 | 220 | 0 | 0 |
| 4/44 | 10 | 0 | 93009 | 1000 | 258 | 218 | 0 | 0 |
| 2/35 | 10 | 0 | 93779 | 982 | 270 | 220 | 0 | 0 |
| 2/36 | 10 | 0 | 93907 | 984 | 270 | 220 | 0 | 0 |
| 2/23 | 10 | 0 | 94175 | 972 | 248 | 221 | 0 | 0 |
| 2/47 | 10 | 0 | 94281 | 988 | 271 | 220 | 0 | 0 |
| 2/24 | 10 | 0 | 95340 | 1048 | 247 | 221 | 0 | 0 |
| 2/27 | 10 | 0 | 94115 | 986 | 270 | 220 | 0 | 0 |
| 2/28 | 10 | 0 | 94163 | 988 | 270 | 220 | 0 | 0 |
| 2/29 | 10 | 0 | 93907 | 984 | 270 | 220 | 0 | 0 |
| 4/32 | 10 | 0 | 90291 | 995 | 260 | 218 | 0 | 0 |
| 3/20 | 10 | 0 | 90031 | 959 | 248 | 221 | 0 | 0 |
| 4/43 | 10 | 0 | 90897 | 997 | 258 | 218 | 0 | 0 |
| 3/25 | 10 | 0 | 92529 | 984 | 267 | 220 | 0 | 0 |
| 2/18 | 10 | 0 | 91825 | 953 | 248 | 221 | 0 | 0 |
| 3/36 | 10 | 0 | 91141 | 974 | 265 | 220 | 0 | 0 |

| | | | | | | | | |
|------|----|---|-------|-----|-----|-----|---|---|
| 3/41 | 10 | 0 | 91013 | 972 | 265 | 220 | 0 | 0 |
| 3/42 | 10 | 0 | 90383 | 968 | 264 | 220 | 0 | 0 |
| 3/44 | 10 | 0 | 90831 | 974 | 264 | 219 | 0 | 0 |
| 3/45 | 10 | 0 | 91217 | 977 | 264 | 219 | 0 | 0 |
| 4/2 | 10 | 0 | 90070 | 962 | 248 | 220 | 0 | 0 |
| 4/20 | 10 | 0 | 90997 | 979 | 248 | 220 | 0 | 0 |
| 3/13 | 10 | 0 | 92345 | 971 | 248 | 221 | 0 | 0 |
| 4/31 | 10 | 0 | 87795 | 956 | 260 | 218 | 0 | 0 |
| 3/16 | 10 | 0 | 88355 | 942 | 248 | 221 | 0 | 0 |
| 3/18 | 10 | 0 | 88611 | 946 | 248 | 221 | 0 | 0 |
| 2/16 | 10 | 0 | 89495 | 956 | 248 | 221 | 0 | 0 |
| 3/21 | 10 | 0 | 89471 | 956 | 248 | 221 | 0 | 0 |
| 3/22 | 10 | 0 | 89855 | 957 | 248 | 221 | 0 | 0 |
| 3/24 | 10 | 0 | 88099 | 938 | 248 | 221 | 0 | 0 |
| 2/8 | 10 | 0 | 88222 | 940 | 248 | 221 | 0 | 0 |
| 5/43 | 10 | 0 | 87423 | 968 | 256 | 218 | 0 | 0 |
| 5/15 | 10 | 0 | 88561 | 946 | 249 | 221 | 0 | 0 |
| 5/16 | 10 | 0 | 88177 | 940 | 249 | 221 | 0 | 0 |
| 2/9 | 10 | 0 | 88094 | 938 | 248 | 221 | 0 | 0 |
| 3/33 | 10 | 0 | 89832 | 852 | 266 | 130 | 0 | 0 |

Appendix IV - Thin Client information

No additional information.

Appendix V - Pathway application information

No additional information.