Today, the e-Business model is a competitive necessity for most large corporations. While the implementation of e-Business is often dependent on mainframe TCP/IP, surprisingly, the monitoring of workload and performance for this critical component has not been systematically considered. To effectively manage mainframe TCP/IP, more than just the mainframe must be controlled: Visibility through the network to the desktop must be present. With mainframe TCP/IP-based applications becoming mission-critical, monitoring the stack, associated applications, and the network components is just as important. This article examines the important parameters and components to monitor and manage this new and complex environment.

Background information is provided for those who may be new to mainframe TCP/IP or VTAM. Those who are more experienced may wish to lightly scan these sections and concentrate on the meatier ones. This article will focus exclusively on IBM’s implementation of mainframe TCP/IP, since this is the most widely used. Examples shown are from a technology developed by Applied Expert Systems.

MONITORING NEEDS

The monitoring needs of mainframe TCP/IP can be categorized into the following seven areas, as shown in Figure 1.

1. TCP/IP stack (address spaces)
2. shared storage areas (Communications Storage Manager, VTAM buffer pools)
3. applications (socket attached, TCP/IP services: FTP, Telnet, SMTP)
4. routers, channel-attached processors, offload devices
5. servers
6. clients
7. network connections

For each of these categories, information needs to be collected both in real-time and historically. Real-time information is needed for diagnostics and fire-fighting, but the value of historical data is underestimated. Understanding historical data allows us to be proactive. For example, you may be able to create a profile of an average user and then project future growth based on that number.

Knowing the historical trends of workload, that is, growth over time, will let you see how your usage has changed. Many installations have a complex mix of workload: Telnet, FTP, and 20 or 30 socket-attached applications. In such an environment, analyzing current usage and predicting future growth can be a daunting task.

TCP/IP STACK (ADDRESS SPACES)

As far as the TCP/IP stack, some things that need to be monitored are the CPU times (TCP, SRB, I/O, Hiperspace), EXCPs and real
memory usage of the address spaces associated with the stack. These may be the main TCP/IP address space, the SNALink address space, or the FTP and Telnet servers. Figure 2 shows an example of CPU usage monitoring.

In real-time, you may find that numerous FTP jobs have pushed your CPU usage to unacceptable limits. Historically, by knowing both the number of Telnet sessions and CPU time for the Telnet server, you may be able to create a profile of a Telnet user and then project future growth based on that number. For example, if you know that a typical Telnet user uses two minutes of CPU time and does 2,000 EXCPs, and you anticipate having 5,000 such users in the next year, you now have an idea of how much CPU you may need. This same method can be used to extrapolate the needs of any application.

**SHARED STORAGE**

With TCP/IP 3.4, the parameters for TCP/IP buffer pools were no longer set by the user, but a new concept was introduced: Communications Storage Manager (CSM). The following sections will provide an overview of CSM, which will be familiar to those already using CSM, but may serve as an introduction to those new to this area.

CSM is a part of VTAM that allows authorized host applications to share data with VTAM and other CSM users without having to physically copy the data. CSM is used by VTAM, TCP/IP and user-written applications. In fact, a CSM user can be any system-authorized application program or product that resides on an S/390 host running, at a minimum, Version 4 Release 3 of MVS/ESA.

Using CSM is a way to optimize system performance for the transfer of bulk data. By providing a way for authorized applications to share buffers, CSM improves system performance during the transfer of bulk data by reducing the processing required for data movement. So, CPU resources (CPU cycles, memory bus and cache) are saved.

**APPLICATION USE OF CSM**

CSM includes an application programming interface (API) that allows users to obtain and return CSM buffers, change ownership of buffers, copy buffers and perform other functions related to CSM buffer management. Applications must be authorized to use CSM. Users set up and access data that resides in CSM buffers. These buffers are obtained from buffer pools that are identified by their buffer size and storage type, as shown in Figure 3.

Data space storage is a common area data space and is associated with the master scheduler address space. This association results in a data space that exists for the life of the system.

**CSM DEFINITIONS AND DYNAMIC EXPANSION**

CSM includes an application programming interface (API) that allows users to obtain and return CSM buffers, change ownership of buffers, copy buffers and perform other functions related to CSM buffer management. Applications must be authorized to use CSM. Users set up and access data that resides in CSM buffers. These buffers are obtained from buffer pools that are identified by their buffer size and storage type, as shown in Figure 3.

Data space storage is a common area data space and is associated with the master scheduler address space. This association results in a data space that exists for the life of the system.

**WHY MONITOR CSM?**

CSM must be monitored because there may be storage leaks, the entire system may be out of capacity, or too much capacity may have been allocated.

- **Storage leaks**: as with any system that allocates and release buffers, the users may allocate buffers but do not release all of them. A user may continuously allocate nine buffers and release eight. However, after some time, that user is holding on to many buffers which are unavailable to others. Without monitoring and looking at historical data, the only time that storage leaks may be found is when a serious system problem, such as a TCP/IP abend, occurs.

- **Capacity issues**: CSM storage is allocated in various size pools (4KB, 16KB, etc.) with certain maximum numbers. Without monitoring, it is not possible to know if the values allocated are adequate. Ideally, you want to neither run out or allocate more than you need. To approach this balance, you must know the pattern of usage. It is true that with dynamic expansion, more buffers may be allocated if needed, but there will be a response time degradation at that time.

In our system, we found that some of the pools were never used, so the allocations for them could be reduced dramatically. It is also important to monitor continuously. If a new application or product is installed that requires a new mix of buffers, then you want to ensure they are available as needed. Considering that VTAM, TCP/IP and any...
Authorized application can use CSM, planning for its use becomes critical.

**VTAM BUFFER POOLS**

The following sections will provide an overview of VTAM buffer pools that will be familiar to many, but may serve as an introduction to those new to this area. VTAM uses buffer pools to control the handling of data: VTAM control blocks, I/O buffers, and channel programs that control the transmission of data. Each buffer pool handles storage for a different VTAM service; therefore, each network will have a different mix of buffer pools.

When you set a buffer pool size that is too small, VTAM will reach the buffer pool limit and then dynamically allocate more space. When the current need is satisfied, VTAM will then dynamically deallocate space in the buffer pool. Specifying small buffer pools conserves storage, but can cause frequent CPU use for expansion and contraction.

VTAM buffer pools are also used for TCP/IP traffic. Applications such as Telnet, Web Server, and CICS sockets use VTAM buffer pools. The buffer pool types specifically used by TCP/IP are IOBUF, LFBUF, CRPLBUF, TIBUF, and CRA4.

- **IOBUF**: Used for input/output data. Every PIU that enters or leaves VTAM resides in an I/O buffer. This pool is 31-bit addressable.
- **LFBUF**: One buffer is required for each active application program with an EAS value (on APPL definition statement) less than 30. If the EAS value is greater than 30, this information is contained in SFBUF. One buffer is required for each TSO user who is logged on.
- **CRPLBUF**: RPL-copy pool. One buffer is required for each VTAM application program request until the operation is complete.
- **TIBUF**: Used to perform input/output operations for CSM-capable protocols. This pool is in 31-bit storage.
- **CRA4**: Used for scheduling and error recovery.

**VTAM BUFFER POOL TUNING**

VTAM buffer pools can be tuned basically by controlling the amount of storage used. Since the pools are dynamically expanded, base allocation and expansion increments can be checked. Allocation of the buffer pools is a trade-off between using storage and the CPU time that will be used when expansion has to be done. Ideally, you want to allocate enough storage to do expansion only for times of fairly high usage. This avoids allocating storage that is rarely used, yet provides high throughput.

**APPLICATIONS (SOCKET-ATTACHED, TCP/IP SERVICES: FTP, TELNET, SMTP)**

**Workload**: Workload for an application is the number of bytes transferred over the network and the number of sessions or clients accessing the application. Workload combined with CPU usage, as described in the following section, will allow you to anticipate the CPU needs for your applications.

**CPU Usage**: The same factors monitored for the TCP/IP stack — CPU times (TCP, SRB, I/O, Hiperspace), EXCPs and real memory usage — need to be monitored for any application, whether it is socket-attached or a TCP/IP service such as FTP or Telnet. As stated earlier with respect to Telnet, if you know that a typical user uses two minutes of CPU time and does 2,000 EXCPs, and you anticipate having 5,000 such users in the next year, you can forecast your CPU usage. The same method can be used to extrapolate the needs of any application: CICS sockets, Web server, MQSeries, or ADSM.

**ROUTERS, CHANNEL-ATTACHED PROCESSORS, OFFLOAD DEVICES**

Many installations are using channel-attached devices, such as the Open Systems Adapter (OSA) or routers such as the IBM 2216 or the CISCO CIP card for high-speed...
Failure Analysis (unreachable, flapping, or unreachable routes.

Performance Analysis (response time distribution, route/segment analysis)

Usage Analysis (sharing, usage, dominance)

Failure Analysis
Failure analysis must be categorized by type of failure charted against route, segment and time. Failure types include flapping, looping, destination unreachable, and packet loss. Routing flaps occur when routers cannot agree on how a packet should be routed. This may happen if one of the hops is not usable because either the link or the router is having problems. Routing flaps may be detected by doing a TraceRoute test. Figure 4 shows an example of looping in the TraceRoute output. Note that addresses 137.72.13.112 and 137.72.13.244 are looping.

Performance Analysis
Response time for routes may be analyzed in terms of the best and worst routes, the best and worst segments and also over the time of day. In tuning your network, adding capacity to the worst segments in terms of response time may have the most impact on the network in terms of improving service for the entire network.

Usage Analysis
In addition to the aforementioned, sharing analysis shows the most shared segments and hosts. This analysis can be used together with response time analysis to see which segments would provide the best return on tuning efforts. The most shared segments are the ones that are points of failure in your network. Also, often, the most shared segments are the worst in terms of response time. This makes sense because the most shared segments are the ones that are most used, and therefore may have the worst response time so they are the most congested. Paying attention to these segments will have the most dramatic impact on your network.

SUMMARY
This article presented an overview of the monitoring needs for mainframe-based TCP/IP. Many pages could be written for each of the seven categories presented. What I've attempted to do here is to define a basic set of parameters and categories for TCP/IP monitoring to the end user, including the mainframe, the applications, and the network. The growth in intranets and mainframe TCP/IP has paralleled that of the Internet. For stable, controlled growth of mainframe TCP/IP, including the network, monitoring and reporting of the areas described must be set in place.

Nalini Elkins is Chief Technology Officer of Applied Expert Systems. Previously she was Senior Consulting Engineer for Boole & Babbage, and worked for many years in network design, performance and planning for Chevron Corporation. Nalini can be contacted via email at Nalini_elkins@aesclever.com.

© 1999 Technical Enterprises, Inc. For reprints of this document contact editor@naspa.net.