4690 Store Systems Availability, Networks and Performance

February 16, 2001

Version 1.0

4690_Availability_WP.lwp

Systems Availability, Networks and 4690 Store Performance

Purpose

This document is for the "store systems architect", the individual that is responsible for all of the components that will be integrated together to make one store system. The primary objectives are to address the ways to optimize the availability, reliability, performance and robustness of the 4690 POS Store System and to minimize the probability of system "downtime".

This first section deals with the high availability features of 4690 POS controllers, terminals, the operating system and applications. The second section describes considerations and guidelines for designing the store LAN, since this can relate directly to store stability. The third section discusses the performance aspects of the 4690 Operating System, hardware and applications.

Both the 4690 hardware and 4690 software individually and together provide elements that give the 4690 POS Store System the capability of having the highest availability possible. Some of these elements are automatically provided through the hardware and operating system. Other elements are made available to the application or as system options to provide an increased level of high availability.

The purpose here is to make the store system architect aware of all of these elements so that he knows the choices that exist to enable the least complex but, at the same time, the best store system for meeting the store's requirements of availability and reliability.

This document serves as a reminder to the store architect of the high availability design points of the 4690 POS Store System. It can be used as a checklist. Most all of the components available to make the 4690 POS a robust system are presented together so that they can be reviewed and so that knowledgeable choices can be made as to how they may benefit an individual store system.

The store architect will have many choices to make in terms of controller (server) and terminal model selection, configurations, operating system and application options, networking and probably cost among hundreds of other decisions.

The following information should help the store systems architect to see all of these high availability components in one place so that a conscious decision can be made about each one's place in the store's overall system integration as well as about how the store will be supported.

Enterprise Level

A high availability system can only exist within the context of a high availability management system. Managing a series of stores across an enterprise is a huge and complex task. Whatever decisions can be made to decrease the complexity and the size of the task should be considered high priority and part of the total overall "store" architecture.

It is one thing to design the POS system for one store, and this will be a prime focus of this document. However, it is just as important to consider the management of all the stores from a central site (headquarters). There are decisions that can be made that can make this task easier. There are also trade-offs that may discourage choosing some of these decisions.

The life cycle process of care and feeding of the store system is:

- Installation
- Change Management
- Fault Management

These tasks to enable control and management of all the stores from a home office IT shop might include:

- Monitoring the wellness of stores
- Help Desk support to the stores
- Maintenance updates of software to the systems in the store
- Problem determination and data collection

Operational tasks involved in managing the store include:

- Pull-back of store sales and accounting data
- Price Maintenance - down load of price additions and changes to the stores

In order to manage these types of tasks, there are choices to consider that may help in avoiding potential nightmares later.

The ideal goal would be to have the same hardware configuration, the same levels of software, the same options and the same networking setup in every store. This, of course, is not possible. But the goal remains and it is easy to see the benefits of, at least, planning towards this goal.

At headquarters, you are maintaining only one level of code in the stores and possibly working on the next level of software enhancements as opposed to trying to maintain ten different varieties of systems software across hundreds of stores. Support by the Help Desk is simplified.

Multiple sets of hardware configurations and different software combinations can equal to a set of systems to support that will inevitably produce a proportional (if not exponential) set of unique problems with which to deal.

Consistency across stores should be a goal. Even if it is three configuration types and three levels of software that must be supported, realize that there must be a limit. Control these factors by design.

Monitor store problems and try to be proactive to possible faults in the store. There are a certain levels of system and application event logging from the store's system which should be monitored by headquarters. These may be indications of system errors or warnings of possible future faults. The system is more likely to be able to detect system errors than a user is of recognizing them. Don't wait until store personnel report the problem when it is probable that there is already recorded data by the system of the problem.

There are 4690 tools and information that are available to help with central management, and 4690 interfaces to other processes that aid in these services.

4690 System Structure

4690 hardware and software together provide the basic principles of a Server and backup Server. The concept is that two servers (4690 controllers) can provide file backup with an operating system feature called Multiple Controller Feature (MCF). The Master/File Server controller is the "owner" of designated files. The Alternate Master/Alternate File Server controller contains copies of these files and is ready to assume control in the case of a failure or loss of the Master/File Server.

Multiple controllers can also provide automatic backup to support attached terminals that rely on a controller for disk accesses and application resources. If a terminal loses access to its primary controller, a second controller is able to automatically and immediately provide critical services that were lost.

"File Backup" and "Store Controller Backup (sometimes called Terminal Backup)" are discussed later in this section.

Another 4690 controller type is the Subordinate controller. The Subordinate provides a performance advantage in that it allows the sharing of the processing workload among as many as eight controllers for stores that may have hundreds of terminals generating high rates of traffic.

4690 Store Systems Availability

High Availability Theory

The concept of providing high availability is through redundancy and excess capacity. Redundancy insures that a critical resource can be replaced quickly and most likely automatically. Excess capacity refers to the ability of the system to continue processing the critical workload during the loss of a primary resource and to prepare for and recover after an outage. If one resource has an outage and the workload is switched to a second resource, then the system in fault control mode is capable of handling this workload with minimal or no disruption to the user.

The states of system operation can be seen as:

- Normal Operations
- Standby
- Fault Detection
- Fault Management and High Availability
- High Availability Recovery after Fault

Normal Operations and Standby

The system is running without faults. All functions and resources are available and running as expected. Hardware and software features are monitoring the system and are ready to go into recovery mode.

Fault Detection

The system is aware of the most likely failure points in the hardware and software, and these are being monitored. The system is constantly checking for proper communications between nodes in the system, for missing or incorrect data transmitted or written and for device failures.

The key is to detect an error and recognize the type of fault. The great majority of faults are temporary and, if detected by the system, the system can probably recover itself or at least minimize the recovery process.

Fault Management

Upon detection, the system goes into a "fault control and management" mode. The least that is done is to report the failure through some type of error message or event logging. This informs the user (or optionally a central monitoring site) of the problem, thereby preventing much of the 'guess' work as to what the failure may be.

If faults were ignored and undetected, bad data would likely be processed or the system may not work at all. Recognition of the problem, why it occurred and how to recover would be left to the user - probably a lengthy task. System detection of faults is imperative for high availability systems when recovery has to be in seconds instead of days or weeks.

High Availability Recovery after Fault

The majority of system faults are software induced. The operating system is designed to detect these possibilities and to provide recovery actions.

If control of the problem is possible, the system will attempt to recover from the fault, possibly even restoring the system to a normal state. Examples of recoveries may be with retries, restarting a device, switching a workload to a redundant resource or continuing with processing without the resource that failed.

The High Availability Components

The 4690 POS OS has recognized the requirement for high availability in the retail environment since its inception and has incorporated specific design elements to achieve this. In order to put this process into perspective let's start with a definition of "high availability" in the store environment and then describe those elements that 4690 has developed to support that definition.

The general definition for system availability can be defined as:

$$\text{AVAILABILITY} = \frac{\text{UP-TIME}}{\text{UP-TIME} + \text{DOWN-TIME}}$$

where DOWN TIME can be thought of as the time to repair. In other words, the up-time of a system divided by the up-time plus the down-time (time-to-repair) is defined as the system availability.

Down-time, at first glance, may seem easy to define. But in reality, when one considers partial availability where only part of the system is available, the issue can get very complicated. One has to decide the degree of lost availability depending on the number of terminals that may be unavailable. (In a store with one terminal, the degree is high.) Does one controller being down mean the system is unavailable even though another controller is still supporting the front end?

Availability Factors

There are many factors to consider when specifying system availability requirements. For example:

Different system functions can require different levels of availability. For example, everyone would probably agree that keeping the 4690 front end checkout running has the highest priority.

On the other hand, there are 4690 resources that clearly have a much lower priority, and the user can adapt to their complete loss for an extended period of time. Some sales reports might fit this description.

The availability requirements for a given resource can vary depending upon when the resource is being used, what it is being used for, and what kind of retailer is using it. For example, one user may find losing a certain resource for two hours four times a year acceptable; another user may not. This can be true for users of the same system. In this case, one has to design for the most critical requirement in order to satisfy all the users of the system.

Availability is not only affected by how reliable a system resource is but also how long it takes to recover/repair the system after a resource is lost. This means that system reliability can be markedly affected by the quality of the problem determination aids, help desk services, store personnel training, etc. Being able to respond to, and fixing, a problem can be very powerful in reducing downtime.

Methods Used to Achieve High Availability

There are three general approaches to achieving high availability.

1. A system can keep operating in spite of faults (called Fault Tolerance),
2. The occurrence of faults in the first place can be minimized (called Fault Avoidance), and
3. The process for fixing faults can be enhanced so as to minimize down time (called Maintainability).

High Availability Requirements for POS (4690 POS design goals)

One of the first tasks that must be completed when designing a highly available system is to decide what the specific availability requirements are. The requirements can be presented in a table which specifies availability levels for each major resource of the system. Once these are defined, then the system design can proceed.

For example, the following table is typical for many retail environments. There are 4 system levels of availability requirements defined. The resource and the individuals requiring that resource are described by those levels of availability.

RESOURCE CRITICALITY LEVEL

RQMT	1	2	3	4
	Most Critical: (Must work) Down <10 min, 2/year	One day down is OK, 1/year	One day down is OK, 6/year	Least Critical: Can do without
RESCE	Checkout operatn Capture EFT paymnts log Hard totals	Sales support Store close Host commun. Loans, pickups In store proc.	Price changes Back room rpts Shelf labeling Operator perf.	(none)
	USED BY sales person	USED BY store manager	UESD BY chain manager	

The goal in this 4690 system example is to maximize the availability of the front end using terminal backup, keep sales support running when possible but save the transaction data if the continuation of background processing is delayed, and put off indefinitely price changes, back room reports, etc. in favor of the first two categories if necessary. Training and fast response time to problem alerts will be used to minimize time-to-repair.

Sources of System Faults

The primary sources of system faults can be classified in three categories; hardware, software and the user environment.

Hardware Considerations

All hardware components eventually fail, and in most cases, values for mean time to failure can be determined. These numbers are not freely available however, since they are subject to change and are always subject to interpretation.

The mean times to failure of the hardware components are a major input factor in calculating system reliability. System reliability can be improved by increasing hardware reliability (often at increased cost) or increasing the parallelism (redundancy) of the hardware paths.

Hardware reliability introduces the topic of preventive maintenance. If hardware components can be replaced just before they are due to fail, availability can be increased. Put another way, the unavailability can be planned to occur when it will be least disruptive. Since the system must be down while components are being replaced, whether planned or unplanned, planned unavailability is the better choice.
Keeping good records of component failures may be a way of determining when it would be advantageous to replace hardware components before failure. Some users regularly pull the error log to their host for analysis. This procedure can be used to highlight the onset of a hard failure allowing replacement before a problem becomes severe.

Software Considerations

As is well known, availability is profoundly affected by software quality. Software bugs come from problems not found during test and from code that is not capable of handling new situations. Problems can exist in base code that have been undiscovered from day one or
they can be introduced by changes, both new function and fixes, to otherwise working code.

Software is different from hardware in that once it is stable, it causes few problems thereafter. However, also unlike hardware, it is generally accepted that it is never possible to remove all the problems from software. Fortunately, after a maturing period, code does not present major problems.

Software can be classified as robust if it is able to perform correctly in complex, under-defined environments. Robust code has the capability of handling incorrect inputs. It doesn't wait unconditionally for events that may never complete, and it keeps status in easily accessible form to enhance quick problem determination.

Using the principles of good code design such as simplicity, testability, robustness, consideration of error handling, and visibility, the application programmer has considerable flexibility in creating a highly available system.

Application code, whether IBM's POS application or with user modifications or user developed POS application running on 4690, should be designed with awareness for various system faults, both hardware and software. The code should know the features that 4690 uses and which 4690 provides for applications to use.

The application should consider basic error recovery scenarios such as how to respond when the Master/File Server is not available and therefore certain files are not accessible. If the application is critical such as the Checkout application, how does it continue to run until a Master/File Server is available?

In background applications, such as the IBM POS applications Sales Support which processes the Transaction Log by reading it and updating many accounting files accordingly, the same error recovery procedures should be considered in order to prevent lost or duplicate data from being created. Whether files are mirrored-on-update or mirrored-on-closed will have an impact on the type of recovery procedures the application will have to take in a restart scenario where a new Master/File Server has taken control.

User Environment Considerations

Several customers may have exactly the same hardware and software setup in a store and use the same functions , but this does not imply that the system availability for them will be the same.

High Availability can be affected as well by how much care is given to the installation environment, how well a help desk is able to respond to problems and how well store operators are trained to use the system and the help desk.

Cable connections should be protected so that they can't accidentally be pulled out. Some geographical areas are much more susceptible to store power outages. A UPS (uninterruptable power supply) may be a wise alternative for preserving store-up time for the POS system in such areas.

How closely the user monitors and reacts to system event logs will play an important role in response to system faults and therefore in reducing down-time.

In summary, system availability depends upon a combination of hardware, software, and support considerations. These include such diverse components as good documentation and training, installation of up-to-date code fixes, and possibly an Uninterrupted Power Supply (UPS). Other considerations include a well trained help desk, strict attention to installation instructions, and simplicity of design.

The 4690 High Availability Design Points

Hardware

The 4690 hardware (controllers and terminals) has unique components that allow the 4690 Operating System and applications to achieve higher availability and data integrity than most PC-based systems.

The controller has NVRAM (NonVolatile RAM) which is used by the OS to protect critical data from being lost during disk write functions. The 4690 terminals also have NVRAM that is used by the applications to maintain store totals that are critical to balancing the store's accounting. It prevents the loss of these sales values in the event of PLD (Power Line Disturbance or extended time power loss) to the terminal or in the case of the terminal's disconnection from the store controller(s).

The terminal may also have battery backup which can maintain memory and terminal operation for a period of time. It prevents having to reload all terminals and having to restart transactions in the case of short interruptions of power loss to the terminal.

Software

The 4690 OS provides some automatic functions that add to the robustness of the POS System. The primary function is to insure that data written to the disk is not lost during possible critical failures either to the system itself or to external failures such as PLD (loss of power). Data loss in many PC operating systems is possible in several ways. Data written to the disk can be lost if it is maintained in memory or cache and the system loses power causing the loss of this data in memory. 4690 OS prevents this loss by not caching disk writes, but rather flushing the data to disk on each write.

It is possible that when data is written to disk that some of the "bits" are not recorded accurately on the write. 4690 OS does a "Read-Back Check" (re-READs the data written) to verify that the data is correct.

It is possible that a power loss may cause no data or, even worse, partial data to be written to the disk. 4690 OS puts the data to be written into NVRAM before the write to disk. In the case of power loss followed by a re-IPL of the controller, the OS will check NVRAM for any outstanding disk write and ensure that this data is completed successfully to the disk. Depending on the recovery methods of the application, the worst case is that there may exist duplicate data, but not missing or partial data in the file.

Directories and File Allocation Tables can be corrupted during the loss of power during a write to them or in the case where they were only kept updated in memory. 4690 OS updates the directory to disk whenever there is a change such as when a file is extended in length due to a write to a sequential file and a new EOD (end of data) pointer is created.

Some operating systems can not control the fact that one application can write over the memory data of another or that one application abending can "crash" the entire system. 4690 OS supports "protect mode" which prevents one application from causing either of these situations.

The 4690 Operating System gives controller processor and disk priority to all requests coming from the terminal. This insures that other application tasks that may be active on the controller do not impact the response time to the checkout processing.

Availability Considerations and Choices for the User

The following components are capabilities that are provided through programming, configuration and options of the operating system and/or personalizations selected through the 4690 POS applications.

In most cases, the IBM 4690 applications are written to take advantage of all of these high availability capabilities.

File Sharing with "Locked Records" for Update

Multiple concurrent updates to the same record of a file can cause lost data. The 4690 OS file system provides "shared file" capability with the ability to "lock" records being updated. If this capability is used by the multiple applications with update capability to the same file, they have the tools to prevent any over-write which could result in lost data.

File Backup

The 4690 OS Multiple Controller feature (MCF) with Data Distribution Application (DDA) is the concept of distributed files having various levels of file backup to a disk on a second controller. The Master and File Server definition of a controller gives that controller "ownership" of distributed files. It always contains the "golden" copy of any distributed file. The Alternate Master and/or Alternate File Server will contain the "backup" copy of those files.

Distributed files are defined by the application developer and the specific distributed file mode should be determined by various access requirements of that file. Some of those considerations may include: file size, frequency of access to the file, frequency of updates, degree of criticality of the data and access type of read, write or update to the file.

In the case of some "catastrophic" loss of the File Server such as a hard disk crash, the Alternate controller can be designated as the new Master/File Server and the backup files will now become the "golden" files. Procedures on how and when such designation should take place are detailed in the *4690 OS User's Guide*, and these procedures should be part of the tools of an enterprise help desk.

Store Controller Backup (sometimes called Terminal Backup)

Each terminal is assigned as being "primary" to a specific controller. You may assign groups of terminals to different controllers in order to balance the terminal workload in a large store (50 plus terminals) environment. Each terminal may also be assigned to another controller as its backup. Each terminal communicates with its primary controller for file and pipe requests even if the file or pipe resides on another controller. The controller logically finds the pipe/file and routes the request to the appropriate controller.

In the case where the primary controller for a terminal is not available and the terminal cannot communicate with that controller; the 4690 OS can automatically switch the terminal's communication to the backup controller which can now service the terminal's requests. The typical POS example of this

feature allows another controller to now do price lookups to the Item file and to write transaction log data for this terminal.

It is important in design (and later for store support personnel) to understand how to "allow store controller backup", how to "prevent store controller backup" and how to "recover from store controller backup". These system functions are done from the Store Control Functions and are described in the *4690 POS User's Guide*.

In the event the system knows that the File Server is no longer available but that terminals are still connected to a controller, the OS recognizes that there are special files that must be maintained to record sales (transaction log data). This data resides on the File Server (it is a distributed file). Since the File Server is not available, the OS provides a "local" spool file to capture this data. When the File Server becomes available, this data is "de-spooled" to its original application file - automatically.

Application Timeout

Application Timeout is an operating system function to guard against lost communications between the terminal and a controller that is not otherwise detected. The application must be programmed for special return codes to work in conjunction with this function.

If a terminal pipe or disk I/O request "never" returns due to a fault otherwise not detected, the terminal application can get control back with this time-out option. The operating system meanwhile records the number of time-outs being registered due to local file and pipe requests and, based upon a voting scheme, the OS can determine that the primary controller for this(these) terminal(s) is having a problem. A controller dump and IPL is forced in order to attempt to automatically clear the problem.

LAN Timeout

The LAN Timeout function is a system configuration option to guard against failures between two controllers. With this option and timeout value, the system will wait for the timeout value for most all system calls (SVCs) that are executed remotely. If the system call is waiting because of some failure in the called controller, the requesting controller can record the error and, based on further occurrences (a voting scheme), a decision can be made to dump and IPL the failing controller.

Terminal Off-line

The Terminal Off-line Feature (TOF) is a feature of the 4690 POS Terminal Sales applications. It provides an alternative to "Terminal Backup".

If a terminal (or all terminals) for some reason cannot communicate with any controller for file access, then TOF allows the terminal to continue with the basic checkout function in "stand-alone" mode. TOF has a few flavors depending on whether prices are on the items (and can be keyed by the sales

associate) or not. One option is to have a “copy” of the Item File to reside in terminal RAM disk or on a terminal’s hard disk. Transaction Log data for sales is also kept either in terminal RAM disk or on the terminal’s hard disk.

Checkout can continue as long as there is power to the terminal. When a controller becomes available again, the Transaction Log data is “de-spooled” to file on the controller and price look-up returns to accessing the Item File on the controller. The Terminal Status file can be updated with the terminals accounting totals which were updated and kept in the terminals nonvolatile RAM (NVRAM).

Where a large item file does not fit into available terminal RAM disk, TOF allows an option to create a subset of the Item File or a reduced size item record.

Terminal Off-line can add another level of redundancy to terminal availability. It also comes at another level of complexity. Maintaining the Item file for a set of terminals becomes a task in addition to maintaining it for the controllers. The concept provides added availability but the additional complexity adds more work to system management while adding to things that can go wrong. The point is that this is another tradeoff to be considered. Don’t just look at these functions for their benefits only.

The recommendation would be NOT to use TOF in MCF configurations with greater than 30 terminals unless it is a “must” requirement. This feature for larger stores may be useful where LAN wiring problems cause frequent outages between the controllers and terminals. Typically this is not a problem if a the LAN installation is a good one.

Checklist

This is a summary of items that were discussed in the previous section.

- Plan for managing all of the stores, not just one
- Understand 4690 High Availability Concepts and Components
- Provide Redundant Control Points for Critical Resources
- Store Controller Backup (Terminal Backup)
- File Backup (Mirroring)
- Planning the Help Desk
  - Procedures
  - Education
- Auto-dump and re-IPL
- UPS
- Networking Choices (see next section, *In-Store Networks*)
  - Switches & Backup
  - Segment Sharing

References

The following references to 4690 Store System publications can be found on the Retail Store Solutions Support web page. The way to reach Retail Store Solutions Support is to go to www.ibm.com/solutions/retail/store/support. Click on Publications under "Popular Links".

The following related topics can be found in the corresponding publications.

- *4690 Store System: Planning, Installation and Configuration Guide*
  - Planning Overview
  - 4690 Installation and Migration
- *4690 Store System: User's Guide*
  - Multiple Controller Feature Network
  - MCF Network Problem Analysis and recovery
  - Collecting and Reporting Problem Analysis Data
  - Using the Disk Rebuild Utility
- *4690 Store System: Programming Guide*
  - Designing Applications
  - Using Error Recovery Procedures and Facilities
  - Managing Files
  - User Application Considerations with a LAN (MCF Network)

In-Store Networks

The purpose of this section is to look at in-store networks from a retailer perspective. Current and future trends are considered giving guidelines for in-store networks that will include the POS System as well as other store devices which could include kiosks or multimedia devices, voice, the transfer of supply chain information, network printers and FAXes.

The physical size of the store, which can vary dramatically depending on the type of retailer, the volume of business, and the location, can pose interesting networking challenges. Proprietary traditional networking environments such as IBM store loop, NCR Starlan, etc., allow for large distances between the wiring closet and individual registers (up to 1000 ft. with IBM store loop). They also allow, in certain cases, for a wide variety of wiring to be used in the store. LAN technologies such as 10BaseT ethernet (10 Mbps running over category 5 UTP (unshielded twisted pair) wire) and 100BaseTx ethernet (100 Mbps running over category 5 UTP wire) only allow for up to 100 meter (approximately 330 feet) wire runs as part of the standard. In a large department store, supermarket, or hypermarket, this can raise interesting networking issues due to the distances that have to be covered. It can even affect medium size (in terms of square foot/location) retailers. Many retailers are unaware that some of their current store network configurations are not supported by the ethernet standards since they have wiring runs that are too long.

The cost of the physical facilities in the store today needs to be taken into account. Many stores today are not wired properly to take advantage of advances in networking technology. If 100 Mbps ethernet will be utilized for the LAN some time in the future, then considerations must be made for high quality cabling in the stores, and cable runs must be within distances allowed by the manufacturers and the standards. In stores with multiple floors, wiring closet space should be allocated immediately above and/or below the other wiring closets so that runs are reduced to minimums to interconnect the wiring closets.

Point of Sale (POS) networks are moving from older proprietary technologies to standards based LANs today. Ethernet 10BaseT is the predominant market leader in terms of percentage of adapters installed (as of 1996, about 90% of all adapters were 10BaseT ethernet). Currently, most vendors in a non-retail environment are shifting to producing adapters that are 10/100 Mbps auto-sensing ethernet adapters, but many of these adapters are still being utilized in a 10 Mbps shared or switched environment.

Many major retail vendors (including IBM with the 4694 and SurePOS 700) are now including ethernet adapters as part of the standard POS terminal. This follows the lead of various PC manufacturers of including ethernet adapters as a standard component of the PC configuration.

Java and other graphical software interface standards are providing retailers with the ability to introduce graphical user interfaces (GUIs) at both the store controller (server) and the terminal. The web browser is also emerging as one standard interface for allowing users access to both graphical and textual

information in a manner that is common across many platforms. Additionally, with the emergence of the internet and the increasing number of people utilizing the internet, web browser based interfaces require less training for certain applications, since the basic interface is more likely to be familiar to the user.

An important point to note is that frequently a graphical user interface may require higher networking bandwidths (or increased network utilization) due to an increased amount of data transferred between the controller (server) and the terminal (register or client). Also GUI-based programs are typically larger than text-based programs. This increases WAN (Wide Area Network) utilization for software distribution or applied software maintenance.

In-Store Network Components

In the future, the in-store network will be a LAN based network. The vast majority of these networks will be ethernet networks, although there will still be some retailers that will utilize token-ring or a mixture of token-ring and ethernet based LANs. Proprietary LAN technologies (such as IBM's store loop and NCR's Starlan) will be gradually replaced in stores.

Some of the issues associated with moving to standards based LAN technologies are not immediately obvious to all customers. Ethernet and token ring have limited wiring run distances when utilizing copper wiring. These distances are typically much shorter than some of the proprietary LAN technologies. For example, IBM's store loop allowed for a 2000 foot distance from the patch panel to the register and back (round-trip - or basically a 1000 foot run). The international standards for ethernet are 100 meters (330 feet). Also, the type of wiring that can be used is different. Typically ethernet runs on category 5 wiring - either unshielded twisted pair (UTP) or shielded twisted pair (STP) - where the environment may contain a lot of interference sources. IBM Cabling System type 1 or 2 wiring can also be used in an ethernet environment, although impedance matching devices will be needed at both ends of the wire (the register and the hub). This can mean that the customer may need to rewire the store, either due to an inadequate wiring infrastructure (some proprietary LAN technologies could run over very low quality wire) or due to the distances involved.

The in-store LAN is composed of the network adapters (also called network interface cards or NICs in the networking industry), hubs and/or switches, and possibly routers to provide connectivity from the LAN across the WAN (wide area network) back to a central site.

General Store Network Design Guidelines

The following design guidelines will discuss the uses of LAN adapters, LAN hubs, switches and routers. For an introduction and more detail on these devices, please see *Introduction to LAN Adapters, Hubs, Switches and Routers* the Performance web-site.

Investment should be made in the best physical facilities possible, as this will pay off in the long term. These include wiring in the store (and its installation), site surveys and access points for wireless network components, environmentals, power and UPS (uninterruptable power supplies), and space allocation (especially in wiring closets).

When implementing a LAN (especially ethernet), the number of devices sharing a segment (called a "collision domain" in ethernet) should be examined. This is not only based on LAN utilization. In a store ethernet environment, collisions will have to be minimized, especially for price lookup functions or in communications between the terminal and its server (especially if the terminal is implemented as a thin client like 4690 today). When looking at the LAN, the applications running on the devices on the LAN need to be taken into account. Different devices and applications drive different utilizations and traffic patterns.

POS traffic between terminals and controllers and among controllers is very low on 4 Mbps Token Ring or 10 Mbps Ethernet LANs. Except for just a few seconds during terminal load, the LAN utilization is under 2% on 4 Mbps Token Ring for as many as 40 busy terminals in a supermarket environment. It would take about 100 terminals in a department store environment to generate the same load. File copies between two 'fast' controllers may generate about 10% utilization on a 10 Mbps Ethernet LAN.

It is most likely that the "other traffic" in the store's network will impact POS traffic before the POS traffic will impact any other in-store activity. The objective in LAN configurations should be to "protect" the POS traffic.

The flatter the network (the fewer layers of routers, switches, bridges, and hubs), the easier it is to manage and operate the network. In addition, the performance will be better, as data will need to go through fewer devices when going from one point in the network to another and during recovery from an outage..

The POS Store Environment

This discussion addresses a typical supermarket store environment, but the concepts can apply across retail stores of various types and sizes (either large format or small format stores). Large stores are typically LAN based, with a large percentage utilizing token ring networks somewhere in the store (whether it is in the back-office or for the check-out lanes).

When the stores are typically larger (more square feet), the distances that the wiring in the store needs to cover is larger. This can tax the 100 meter length restriction that is typically present in category 5 UTP (Unshielded Twisted Pair) ethernet installations.

Larger store formats are typically 15 to 20 lanes (checkout registers) and include a number of other types of functions in the store such as pharmacy, bakery, deli, video rentals and other customer service functions.

Large stores typically generate a larger amount of traffic across the wide area network (WAN). This traffic is due to more credit authorizations, more inventory and sales information in a large format store, and the greater likelihood that a customer loyalty program is in place.

Smaller format stores typically have 5-12 checkout lanes and do not have as many non-checkout functions as larger format stores (for example they may have a bakery and deli, but they won't have a pharmacy and video rental area).

Also, smaller format stores typically have fewer servers and in-store processors than the larger format stores. The fact that there are fewer non-checkout functions in these stores contributes to this.

Design Guidelines

For new stores or retailers that have not installed token ring already, the recommended solution will be to install an ethernet 10BaseT LAN. This is the industry leading LAN technology at present, and the IBM 4694 and SurePOS 700 POS terminals ship with an ethernet adapter standard on the planar. In the case where the store has a token ring in place for the back-office but does not have a LAN installed for the front-end (checkout), ethernet LAN may be the choice for the checkout with a switch or router to connect the token ring and ethernet LANs. In the case that the customer has already installed token ring and it would be cost prohibitive to switch to ethernet, the recommendation would be to stay with a token ring environment. The argument of ethernet vs. token ring is primarily focused on cost vs performance predictability. Ethernet comes standard (for no additional cost) in the IBM POS terminals and in many PCs. Additionally, the per port price for ethernet is cheaper than for token ring in networking equipment. On the negative side, there is a need to design the ethernet LAN to allow for more unused bandwidth so that collisions do not cause problems in terms of response time for the checkout registers.

In order to minimize collisions, it is recommended that not more than 32 to 48 devices be put onto an ethernet collision domain (segment). While this many devices will not necessarily utilize a high percentage of the bandwidth on the LAN, in a check-out environment, collisions on items such as price lookups will want to be avoided since collisions cause response time to suffer. Utilizing a switched connection is the method to separate these segments (also called collision domains).

For non-checkout applications, the load that those applications put on the network needs to be evaluated in order to determine how many of these devices can be put on a collision domain at the same time. The recommendations are to keep these devices on a different collision domain from the store controllers and registers if they have the potential for generating a large amount of traffic at any period of time where the POS system will be in use. For example, if a large file will be downloaded from a central site, or if a Lotus Notes server in the store will be replicating a database across the WAN and the LAN, or if network printers are on the LAN, these functions (devices) should be separated so that its traffic does not impact the POS functions.

If there are two in-store controllers, put them on separate hubs or switches so that a single point of failure is avoided. Each store should have at least two hubs or switches (unless it is an extremely small store). Also, if a switch is being utilized and only one switch can be in place; then put a hub in the same place as the switch and have enough open ports on the hub so that, if the switch goes down, the controllers and router can be plugged into the hub from the switch or at least connect the backup controller to the hub to help minimize single point of failure problems.

Wiring Suggestions

- Wire check-out lanes so that if a hub or switch is lost, it does not take out a whole block of registers in a row. For example, wire registers 1, 3, 5, ... to hub 1 and registers 2, 4, 6, ... to hub 2. That way if a hub is lost, half of terminals are still active.

- Put a dial backup circuit on the router (or other device used for the WAN connection) so that in the case that the WAN is unavailable, the backup circuit for credit authorizations, electronic funds transfers (EFT), etc. can still be utilized.

- Minimize the number of wiring closets in the store. This will depend on the physical size of the store and where all of the networked devices will reside. Having one (or the fewest possible number of) wiring closet in the store minimizes where a person would need to look in order to perform problem determination. Also, try to utilize a single model of hub in a single store. Managing a mix of hub models in the store adds unnecessary complications.

Sample Networking Solutions

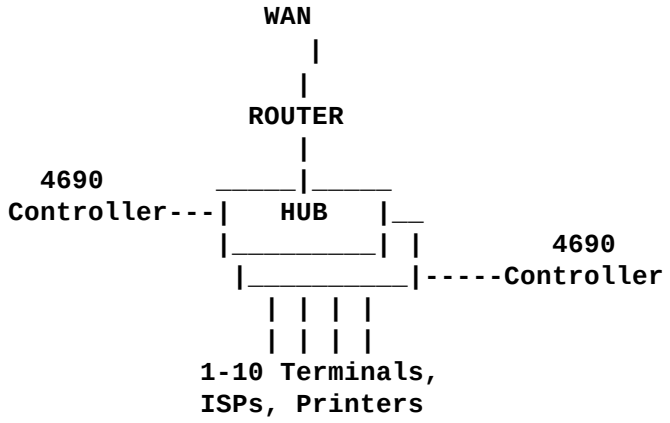
Simple Network

Configuration 1 is a sample solution for a small store which probably has less than 10 terminals and may have some non-POS devices/functions on the LAN which are low impact in terms of LAN traffic.

The "non-POS system" consists of ISPs (In-Store Processors), network printers and other store PCs and devices. The system may or may not be connected to a WAN. The simplest setup is to have enough HUBs stacked so that all the devices, terminals, and controllers can attach to the LAN via these HUBs. The traffic of each device is shared with all others on this configuration.

Although the probability of failing HUBs is low, having a minimum of two HUBs means that someone on site can re-plug terminals from one failed HUB to a good one. If two controllers exist, attach each to a separate HUB. Also, split terminals among at least two HUBs.

Configuration 1



Switched Network

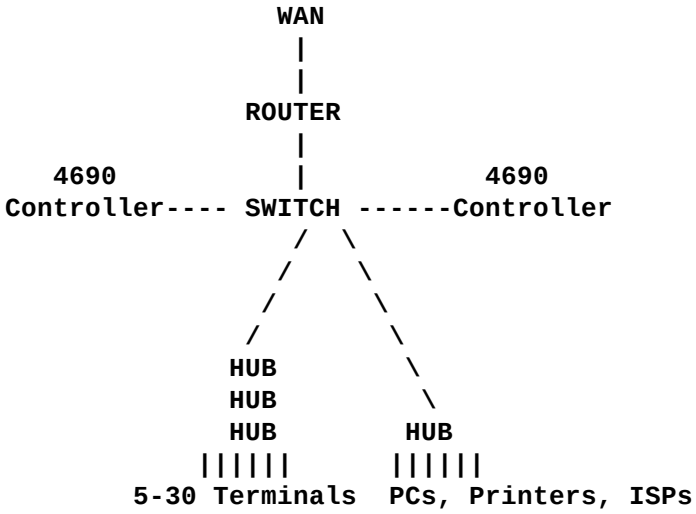
Configuration 2 is a sample solution for a medium size store which probably has 5 to 30 terminals and will have some non-POS devices/functions on the LAN which may have some impact in terms of LAN traffic.

The "non-POS system" consists of ISPs (In-Store Processors), network printers and other store PCs and devices. The system may or may not be connected to a WAN. At this level, it is recommended to 'protect' the POS traffic from other traffic in the network. This is done by implementing a switched connection among different shared segments.

POS terminals are on one segment 'isolated' from the non-POS devices on a separate segment. These segments come together at a switch. Controller to controller traffic can also be isolated from the terminals by attaching them directly to ports on the switch.

The likelihood of a switch going down is small; however, the physical backup for this is to (1) have a spare switch or (2) have a spare HUB or (3) move all the switch connections to spare HUB ports. In the latter two cases, the performance may suffer but the network will still be up.

Configuration 2



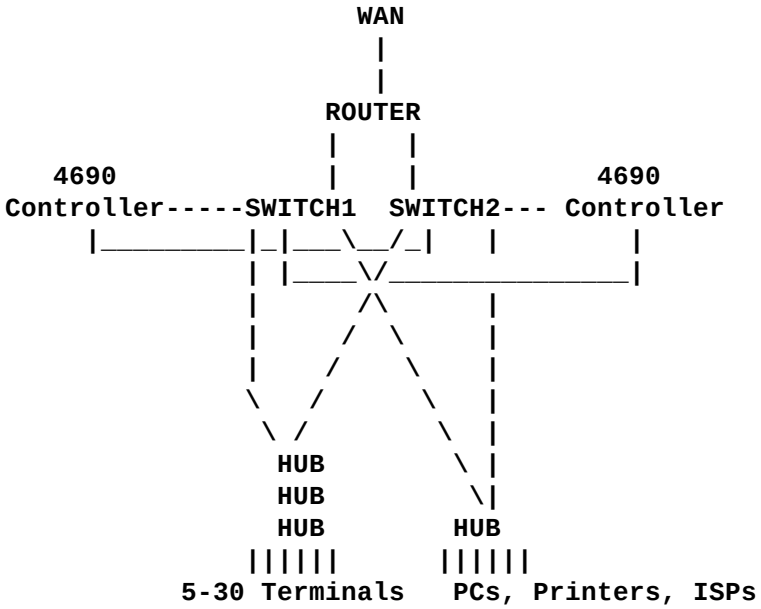

```
Objective:
    Reduce probability of collisions with POS traffic.
    -  "Filter" POS system (Controllers plus Terminals) from
       non-POS traffic (PCs, Printers & ISPs)
    -  Segment "controller to controller" traffic from "terminal to
       controller" traffic.
```

Switched Network with Redundancy

Configuration 3 adds a level of availability and redundancy. If the concern is about a switch going down, the network could be designed with two switches as shown below. "Spanning Tree", which is a configuration feature of all switches, will prevent loops in an Ethernet network.
The benefit of a two switch design is that it would not require someone on the premises with networking skills. If a switch fails, the other one would take over and the network would run with minimal interruption ("spanning tree" takes 25 seconds to reroute).

The router, both controllers and both shared segments of the HUBs are each connected to each switch.

Configuration 3



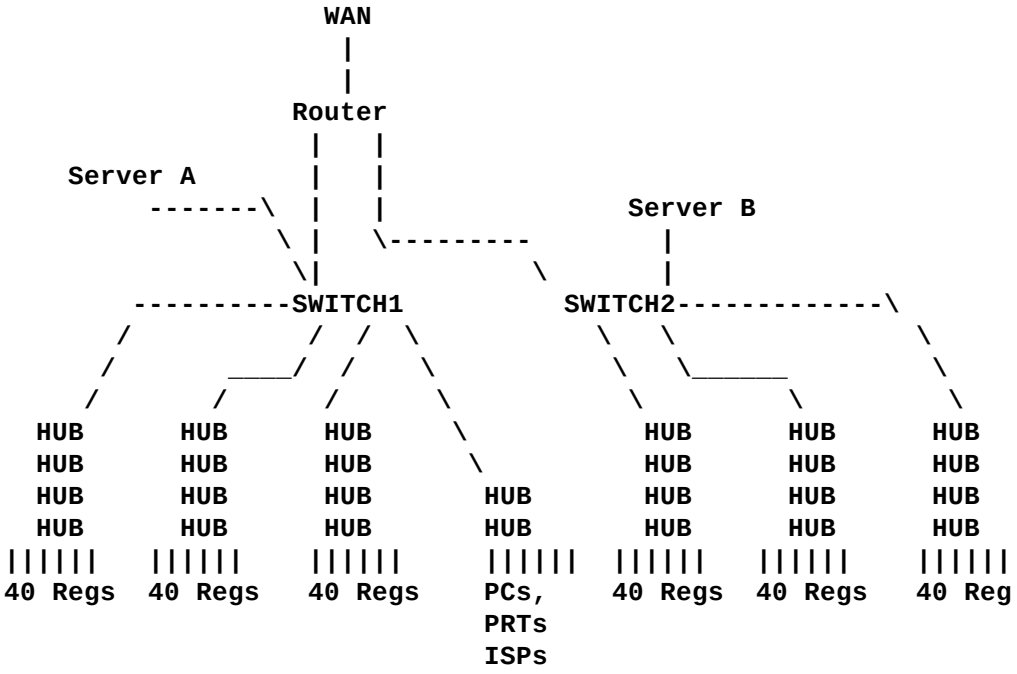
Objectives:
1) Highest availability of the network
2) Reduce probability of collisions with POS traffic.
   - "Filter" POS system (Controllers plus Terminals) from non-POS traffic (PCs, Printers & ISPs)
   - Segment "controller to controller" traffic from "terminal to controller" traffic.

Large Store Network with Redundancy

Configuration 4 shows how 200+ terminals might be networked with HUBs and switches.
In this situation, to go back to configuration 3 which shows how to add redundancy and automatic recovery of a switch failure. Most switches today have a minimum of 12 or 16 ports. From each SWITCH, there is one connection to **each** stack of hubs, to **each** 4690 controller and to the router. Only 10 ports out of the 12 will be used.

In configuration 4 below, the connections from each HUB segment and each Server would **ALSO** go to the other SWITCH to add redundancy. **The connection lines are not shown to simplify the diagram. The number of 4690 Controllers (Servers) may need to be greater than what is shown in order to support 200+ terminals.**

Configuration 4




```
Objectives:
  1) Support "super large" POS network.
  2) Highest availability of the network
  3) Reduce probability of collisions with POS traffic.
     -  "Filter" POS system (Controllers plus Terminals) from
        non-POS traffic (PCs, Printers & ISPs)
     -  Segment "controller to controller" traffic from "terminal to
        controller" traffic.
     -  Segmentation of Terminal traffic.
```

Network Guideline Summary

The following is a checklist of network considerations that have been discussed in this section.

- Wire - Does the type of wire match the requirements of function and devices supported?
- Distance - Manage and design for the physical size of the store.
- Bandwidth -
  - POS traffic is typically low in utilization.
  - 10 Mbps Ethernet and 4 Mbps Token Ring are sufficient for 4690 Terminal segments; 32 to 48 terminals are recommended per shared segment or collision domain.
  - Know the non-POS traffic from other in-store LAN traffic.
  - Beware that the traffic during the recovery process may be several times greater than during "normal" traffic.
- Create shared segments (collision domains) with the use of one or more switches.
- Redundancy - Enable Tree Spanning with switches.
- Future - If rewiring an old store or wiring a new store, consider the LAN requirements of future store functions and devices.
- Network Utilization Guidelines - Token Ring networks can operate efficiently (maintain good response times) at sustained utilizations of 60% to 75%. Ethernet is susceptible to collisions which can impact response times when the LAN utilization is greater than 35%.

Performance Considerations

There are not many choices that are made in the POS system design selection that do not have some impact on the performance of that system. Some of these design selections and parameters do require more attention than others. This section describes those that are worthy of specific attention.

Performance Impact Factors

Key factors for performance can be categorized into the groups; environment, profile and application.

- Environment

The environment includes factors like the retail category. Is it Supermarket, Department Store, Specialty, Drug Store, Discount or Cruise Line. This has implications on the profile below.

Also included is the factor of the configuration. How many controllers and terminals are present and what models are used? Processor power, disk performance and network performance have an obvious impact on system performance.

- Profile

The profile includes parameters like transaction rates, item sales rates, percentage of credit transactions and the number of items in the item record file.

- Application

The primary 4690 applications are General Sales Application (GSA), Supermarket Application (SA), SurePOS Application Client/Server Environment for 4690 OS (SurePOS ACE), and Chain Sales (CS). Each one of these has some degree of modification in its executing environment.

There are many business partner solutions that support application environments like cruise ships, restaurants, casinos and theme parks that require unique functions in addition to retail sales.

Performance Basics

The primary performance goals for the POS system are:

1. Maintain good response times at the front-end (terminal sales).
2. Maintain high checkout throughput at the terminals.
3. “Timeliness of Data” processing by background processing.

These are achieved based upon the environment with appropriate hardware selection and configuration and an application designed to meet these goals.

Good response time for "price look-up" means minimizing the processing per item by the Sales Application. High checkout throughput, while maintaining good response times, may mean multiple controllers supporting a large number of terminals to spread the work load. It certainly involves special consideration in Checkout application design. A system may easily support the many terminal requests for a few terminals, but programming design must consider the complete picture for the system.

If an average transaction selling 5 items requires 40 controller requests (like disk I/Os) from the terminal, then a store with 100 terminals selling at 30 customers per hour per terminal will produce 120,000 requests per hour or 33 requests per second for the entire system. Whereas forty disk requests for one transaction may provide good response times for a few terminals, 33 requests per second could create a bottleneck in the system that introduces poor response times during peak sales throughput.

"Timeliness-of-Data" is the ability of the background application (Sales Support) to process the Transaction Log (as input) and update all of the system's "accounting files" at the same rate as terminal sales throughput. This means that reports are timely in that the data they show reflects sales from within the last five minutes, and there is no "backlog" of processing that will delay the "store closing" procedures at the end of the day.

It is impossible to address all the combinations of environments, profiles and applications. Probably no two customers or even no two stores have the exact same total environment. The intent is to address some basic environments that will give an understanding for the most important performance parameters so that their impact may be understood in various situations.

The following table presents some of the typical profile variables for three major environments. The throughput parameters (such as "Transactions per hour per terminal") represent peak periods of sales sustained over an hour. In the Department Store, some terminals selling cards just before Mother's Day may be selling at 3 customers per minute (or 180 customers per hour); however, at the same time, other terminals in appliances may be servicing a customer every 15 minutes (or 4 transactions per hour per terminal). Hence, 25 transactions per hour per terminal is an average over the whole store but during a peak sales period. For department stores, this would typically be a period during the Christmas season.

Environment / Profile Variable	Supermarket	Department Store	Mass Merchandiser
Terminals/Store	20	150	35
Xtns/Hour/Terminal	30	25	30
Items/Xtn	33	2.5	5
Xtn/Hour/Store	600	3,750	1,050
Items/Hour/Store	20,000	9,375	5,250

From a performance impact perspective, supermarket is “item” oriented. Their numbers emphasize items sold. In department stores, the emphasis is on the number of transactions (or customers).

Therefore, supermarket processing is busiest in the area of accessing the ITEM record file or other processing related with the selling of an item. This could be a path in the Checkout Application or in the Sales Support application such as processing Item Movement and Department Totals updates.

In the department store environment, processing representative of the “transaction” will be the most frequent activity. This may include functions like operator sign-on and sign-off, writing to the Transaction Log, requesting credit to a host or writing to electronic journal logs in the Sales application. In the background processing (Sales Support), the impact may be updating files for operator and terminal accounting and for customer information.

4690 System Priority

4690 OS uses operating system and application priorities to aid in providing high availability and high performance for the most important POS functions.

- All I/O From Terminals Have the Highest Priority over other Controller Applications
  - First considered are files with 'PRIORITY' on the OPEN statement
    - Item Record File
    - Terminal Status File
  - Next are all other terminal requests
  - Last from terminal are Load/Dump requests

- Controller Applications Are Next and Can be Assigned Priorities within this Group

4690 Controller Application Priority

Priority can be assigned to controller applications. In general, this is not necessary. Where a system is not stressed to its maximum recommended resource utilizations, applications running at the same application priority (default of ‘5’) will share the resources with very little performance impact of one application upon another. Only in the case where there is system performance degradation due to intense processing of several tasks is it recommended to assign priorities to applications. Priorities may be assigned to 4680/90 controller applications in two ways.

For more information on how to use controller application priority, recommendations and a description of how 4690 application priority works, please see the Controller Application priority document on the Performance Web-site.

System Utilizations

Resource limitations are guidelines that say, "If the system is kept within these boundaries, then the system will maintain performance stability". These resource limitations are given in the "recommended" utilizations of the table below.

When one of these limits is exceeded, it may or may not represent performance degradation in the system. If there is evidence of poor performance, then one of these resources is likely a clue as to where the 'bottleneck' may be.

These system utilizations are guidelines that are good boundaries for performance stability in a multi-tasking environment. They represent utilizations sustained over a period of time, not just spikes representing a few seconds to a minute.

The MAX column represents the point at which a sustained utilization at this rate will probably mean there is no capacity left from which "good" performance can be expected.

	Recommended	MAX
* CPU Utilization	60 %	80 %
* Hard Disk Utilization	50 %	70 %
Host Utilization	50 %	70 %
* Loop Utilization	40 %	70 %
LAN (Ethernet) Utilization	35 %	35 %
LAN (Token Ring) Utilization	70 %	80 %

* These resources are measurable with the 4690 OS internal utility available under "Problem Analysis Data Collection". Host and LAN utilizations must be measured with an external device as the 4690 system does not 'see' all of this traffic.

THE RECOMMENDED UTILIZATIONS DO NOT REPRESENT THE FULL CAPACITY OF THE SYSTEM.

The above recommended factors allow other applications to run simultaneously without impacting the sales throughput and response time. The idea is that there is CPU, Disk, Host and Loop and/or LAN utilization left to support other tasks. These other tasks are not, of course, without their limitations.

CPU and Disk Utilization Considerations

A batch application (such as applying 10,000 file updates to a file that takes "x" amount of time) will theoretically take 100% of the system. For example, it might be 75% on the disk and 25% in the CPU, if this task were measured by itself. The "75%" disk utilization is not "bad". It will, however, have some impact on applications that are running at the same, or lower, application priority.

Terminal requests to the 4690 controller automatically have higher priority than any controller applications, so there would not be any impact to the terminal applications from running these batch applications. Console response times may be impacted, but, unless there are also other tasks running at

the same priority or higher than the batch task and the console application, the impact will probably be small.

Several batch tasks that are intense in processing could have some impact on controller console response time. In this case, the batch tasks could be assigned an application priority lower than "5" which is the system default for all applications as well as for applications started at the consoles. “Lower than 5” would be priorities 6-9. Applications running at this “lower” priority would have very little impact on console tasks which means that operator console response time would not be impacted by this application which uses the disk so heavily.

The point is that high utilizations may not always imply "running out of gas". The guidelines refer to values for the "interactive" activity of the system where good system response times are important to the user.

Before Java and After Java

The 4690 OS and its applications were designed for, written for and, in many stores, are still running in 4683 terminals with 6 MHz 286 processors and 2 MB of RAM. There are still controllers with 16 MHz 386 processors and 8 MB of RAM, and the terminals are attached via 38.4 Kbps Store Loop. The POS applications have grown with added function over time. Sometimes the combination of enhanced function added year by year causes performance to become sluggish on the oldest of hardware and/or requires more RAM in the terminal or controller.

That was “BEFORE JAVA”.

Until Java applications came along, there was really no ‘hard’ line drawn that said a function will not run on a certain processor. Java draws that line. With the planning for Java applications in the POS terminal or controller, the time has come to realize that there is new minimum hardware to consider.

With the announcement of Java in V2 R2 of 4690 OS, the capability for graphics and all else that Java programming has to offer are now possible on the 4690 POS system. Java trades new function and programmer productivity for an increased processor and RAM load. Java requires the speed of more current processors and lots of RAM in order to meet the performance capabilities of past languages.

In the 4690 world, the minimum recommended terminal for Java applications is a 4694-244 with a 6x86 P166 processor. The minimum RAM should be 32 MB for partial Java application implementation such as a GUI interface to a current CBASIC checkout application. A full Java Checkout application will more likely require 64 MB of RAM or more.

A Pentium processor with at least a P166 and 32 MB to 64 MB RAM is also the recommended minimum system for a controller with Java applications, and token ring or ethernet must be the TCC (Terminal to Controller Communications) medium to support Java in the terminal.

Perspective

Today's processors are over 200 times as fast as those used in the initial years of the 4680 POS System. The Ethernet and token ring are over 100 times as fast as the store loop. DASD space has increased over 100 times per drive. However, the disk access times are effectively only four times as fast as they were on the first PC/AT hardfile.

Therefore, the disk is the primary bottleneck in the 4690 POS system. Whereas many operating systems use disk caching (keeping file data in memory) to avoid as many real disk accesses as possible, the 4690 OS file system forces record writes into physical disk I/Os to ensure data integrity.

When it comes to hardware configurations and system design, consideration of file accesses is an important factor. In "ninety five percent" of the cases, performance may not be an issue with the POS system. A small store (10 terminals or less) with today's latest processing technology, 4694 terminals and Pentium processors in the controller, will probably require no tuning for performance. For those that have stores in the "five percent" category, there are likely other choices that should be considered.

Basic Configurations

The following are some very basic configuration concepts. They do not distinguish older (slower, 486 processors) hardware versus newer (faster, vintage 2000 Pentium III) hardware, nor do they address the application modifications and all of the personalization options possible.

In most cases, a faster model will not be necessary for the POS workload unless the customer is considering crossing the "Java line". A faster model will, of course, be more than adequate and will be a requirement only when other tasks are a consideration for execution on the POS system.

Two disk drives will be recommended per controller in instances where considerable increase in throughput is required. Otherwise, DASD should be chosen to insure adequate space for all possible files.

There is no specific TCC (Terminal to Controller Communication) method chosen. The Store Loop, Ethernet and Token Ring all perform equally well for most POS functions. For terminal load scenarios, Ethernet and Token Ring will be two to three times as fast as the store loop, but for Terminal Sales functions, little difference will be noticed.

The primary configuration choices that will be considered here are:

- Number of Controllers in System
- Number of Hard Disk Drives per Controller
- Terminal to Controller Ownership/Backup
- Number of Terminals per Controller

- Host Link Connections

Where the configuration description says it supports "30 supermarket terminals or 80 department terminals", this is a "ballpark" estimate to give a *feeling* for the store environment and size. Many variables, not addressed here, can have impacts in either reducing or increasing this number of terminals.

File Server will refer to the Master as well as the File Server controller. No 'split' Master-File Server configuration is recommended due to the complexity it adds to the system. Alternate or Alternate File Server refers to the Alternate File Server and the Alternate Master as the same controller.

Maximum Configuration Considerations

Some of the larger configurations shown below may be limited by these rules regarding the maximum number of terminals that may be physically attached in a 4690 POS configuration.

When using Ethernet or Token Ring Terminal to Controller Communications (TCC), the maximum number of 4693 and/or 4694 terminals supported by one controller is 128 terminals**.** This includes primary terminals and those possibly in backup. I.e.; two 4690 controllers could support 128 terminals if terminal backup was also provided.

In the case of some 4693 terminals where a satellite terminal (model 202s) can be attached, these do not count toward the limit of 128. Therefore, there could be a maximum of 256 terminals supported for the situation where half the terminals are model 202s attached to 4693s.

When using SLOOP TCC , the limit on terminals is the number of model 1 terminals that can be attached to one loop. The maximum number of 4683 or 4693 terminals is 128 (64 model 1 plus 64 model 2) per loop. The maximum number of 4694's per loop is 64 since the 4694 does not support model 2s. One controller can support a maximum of two loops. Each loop may be in either primary mode or in backup mode.

There is also the possibility of a mix of loop and ethernet-or-token ring attached terminals. Then the same limits apply to the loop and to the controllers, which are 64 model 1s per loop and 128 total terminals per controller.

Configuration #1 - One Controller System - Small Store

This is one controller typically supporting up to 10 terminals in any environment. One controller could easily support more terminals from a performance and limitation point. In order to provide higher availability, the application's Terminal Off Line Feature (TOF) can be utilized to ensure sales if access is lost to the controller.

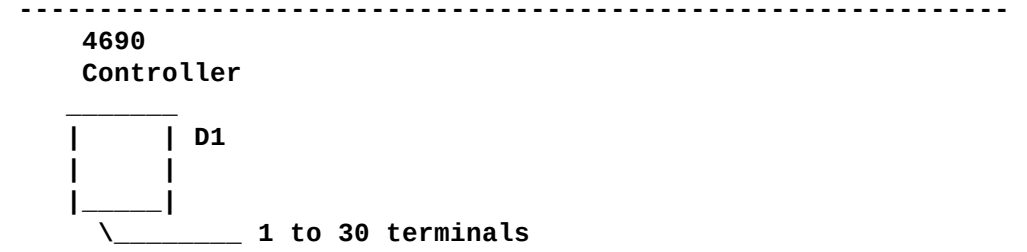


Figure - Configuration #1

All other configurations will assume at least two controllers to provide file backup and controller backup. The concept here is that when one controller is "down", the system will still support the store during periods of peak sales with acceptable performance.

Configuration #2 - The Basic Two Controllers

Probably the most common configuration and the one that fits "90%" of the stores is two controllers with one hard disk each. This configuration supports up to 30 supermarket terminals or up to 80 department store terminals.

There are equally good reasons for any of the following terminal assignments:
1) All terminals on File Server; the Alternate acts as the backup to the File Server for the terminals.
2) All terminals on Alternate File Server; the File Server acts as the backup to the Alternate File Server for the terminals.
3) Terminals split between File Server and Alternate; each controller acts as backup to the terminals on the other controller.

The Host Link is typically through the File Server with the ability to run through the Alternate File Server when it is designated as the File Server. In the latter case, a physical host connection would also have to exist to the Alternate.

If either controller is "down", the other controller will be able to support the full store.

The limitation of this system is likely to be "timeliness of data" (Sales Support starts to get behind) during peak sales of the largest stores. The disk is likely to become the bottleneck.

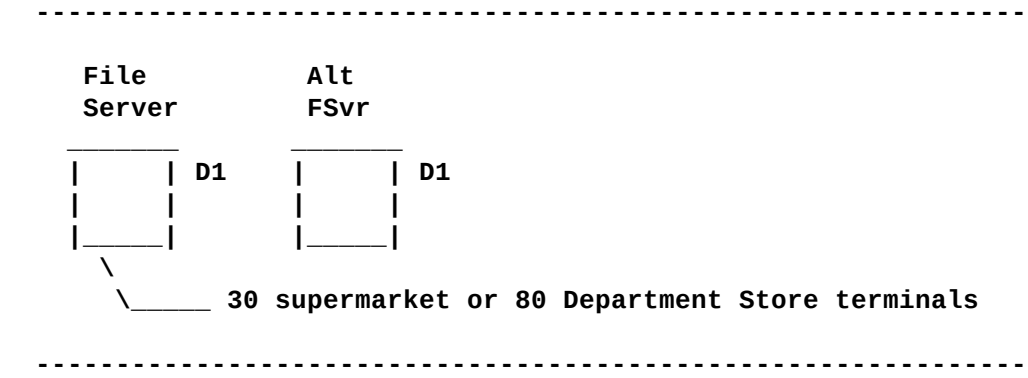


Figure - Configuration #2

Configuration #3 - Two Hard Drives per Controller

The next step in achieving more throughput capability for the store is to add a second hard disk to each controller. This is not for space but in order to add disk utilization cycles (disk accessing capability) to the system.

The bottleneck in configuration #2 is probably the disk on the File Server. It is receiving access requests from terminal sales for transaction log writes, terminal status file updates and PLU activity for terminals primary to ('attached to') the File Server as well as all the updates to accounting files by the background sales support application.

A second hard drive allows strategic placement of files so that accesses to files by sales support can be placed on one drive while files accessed by Checkout (the terminals) can be on the second drive. 4690 allows simultaneous access to two drives by different tasks. This can result in a 20% to 50% increase in throughput capability depending on the applications, file placement and the type of disk drive. The performance of the IDE drives can be maximized by ensuring that each drive has a separate disk controller.

This configuration supports up to 50 Supermarket terminals and up to 120 department store terminals.

There are equally good reasons for any of the following terminal assignments:

1) All terminals on File Server; the Alternate acts as the backup to the File Server for the terminals.
2) All terminals on Alternate File Server; the File Server acts as the backup to the Alternate File Server for the terminals.
3) Terminals split between File Server and Alternate; each controller acts as backup to the terminals on the other controller.

The Host Link is typically through the File Server with the ability to run through the Alternate when it is designated as the File Server. In the latter case, a physical host connection would have to exist to the Alternate.

If either controller is "down", the other controller will be able to support the full store.

The limitation of this system is likely to be the disk utilization on the File Server. The way to off-load more processing from the File Server is to have a configuration that does not support terminals primary to the File Server. A third controller allows this to be accomplished.

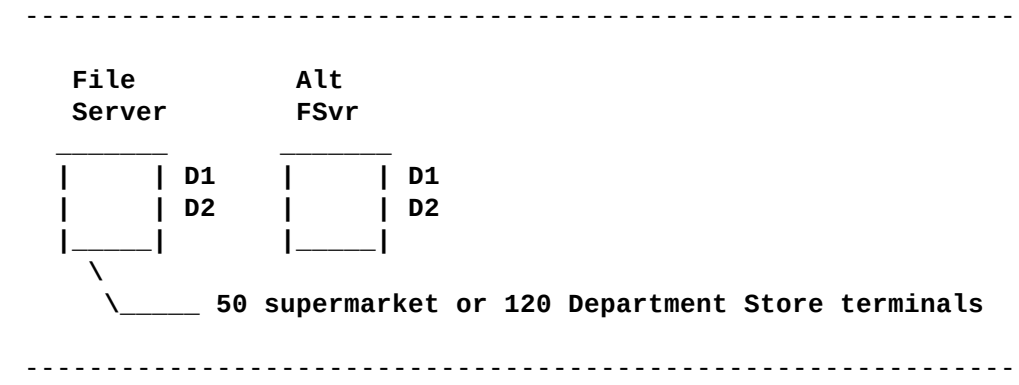


Figure - Configuration #3

Configuration #4 - Three Controllers

The way to off-load more processing from the File Server is to have a configuration that does not support terminals primary to the File Server. A third controller allows this to be accomplished. This controller is called a Subordinate controller.

This configuration supports up to 100 Supermarket terminals and up to 250 department store terminals.

There are equally good reasons for any of the following terminal assignments:

1) All terminals on Subordinate; the Alternate acts as the backup to the Subordinate for the terminals.
2) All terminals on Alternate; the Subordinate acts as the backup to the Alternate File Server for the terminals.
3) Terminals split between Subordinate and Alternate; each controller acts as backup to the terminals on the other controller.

The Host Link is typically through the File Server with the ability to run through the Alternate when it is designated as the File Server. An alternative is to support the host link and communications through each controller. This would be a solution for a system whose terminal
to host traffic was very busy.

If any one controller is "down", the other two controllers will be able to support the full store.

As more terminals equals to more customer throughput, the Sales Support application's throughput load increases to a point where it cannot process all the updates to accounting data fast enough to keep up with peak sales.

One application option that is available here is the GSA RAM disk function to process some files in memory. Supermarket has a performance enhancement option which processes accounting files in memory. Both schemes allow greater Sales Support throughput capability while also maintaining the data integrity of the data being processed in memory.

There is a system design limitation of 256 terminals based upon the choices recommended here. Since, in this configuration, the File Server will not support terminals, the Alternate and Subordinate can only support a maximum of 128 terminals each with controller backup of terminals.

See the note prior to these configurations describing the “Maximum Configuration Considerations”. The above limit refers to 4683 or 4693 terminals with model 2 terminals attached. If 4694 or SurePOS terminals were attached the maximum limit in configuration #4 would be 128 terminals.

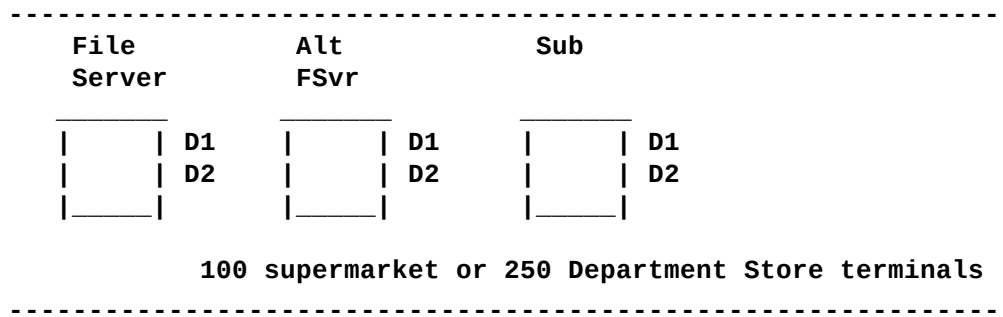


Figure - Configuration #4

Configuration #5 - Four Controllers

To add more physical terminals to the configuration for a large department store, a second Subordinate controller is added to the configuration.

By choice here, the File Server still does not support terminals as primary or in backup.

This configuration supports up to 384 terminals in a department store.

1) One third of terminals on Alternate; backup is first Subordinate.
2) One third of terminals on first Subordinate; backup is second Subordinate.
3) One third of terminals on second Subordinate; backup is Alternate.

If any one controller is "down", the other three controllers will be able to support the full store.

See the note prior to these configurations describing the “Maximum Configuration Considerations”. The above limit of 384 terminals refers to 4683 or 4693 terminals with model 2 terminals attached. If 4694 or SurePOS terminals were attached the maximum limit in configuration #5 would be 192 terminals

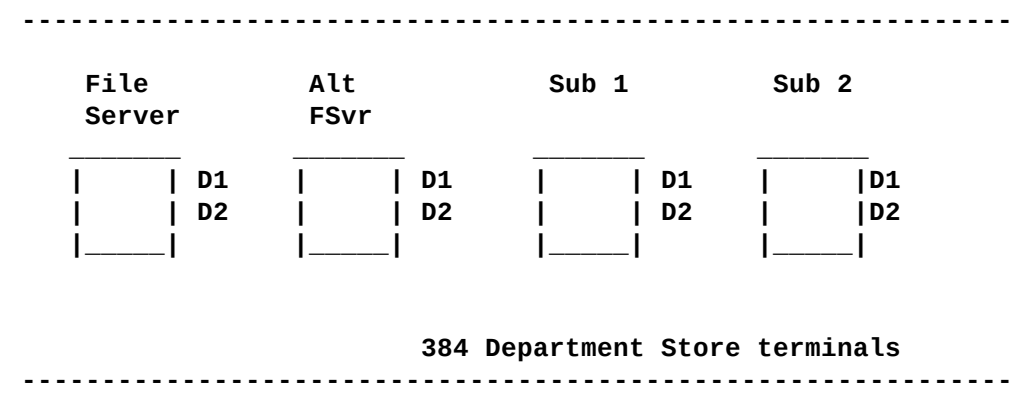


Figure - Configuration #5

Maximum Configuration

See the note prior to these configurations describing the “Maximum Configuration Considerations”.

The physical design limit for a 4690 Store System network is eight controllers. If controller backup was provided for all the terminals, the configuration could support 1024 terminals when using 4683 or 4693 terminals with model 2s attached. This limit for 4694 or SurePOS terminals would be 512 terminals. The actual limitations of such a system would probably be related to the application and the workload based upon the environment.

POS Keyed Files

The following are basic objectives and assumptions for the usage and implementation of keyed files in the 4690 POS Operating System. The primary objective is one of performance: be able to access records from very large files in the shortest time possible. 90% of the time, the desired record should be found on the first read to the file.

- The files may contain 100 records or millions of records.
- The keys are random (as opposed to sequential type keys).
- Reorganization of the file at various intervals is not required.
- Records can be Read, Updated, Added and Deleted.
  This allows for some ability of housekeeping of the file. Records that are deleted free up space for new records.
- Support 'Read with Lock' and 'Write with Unlock' to allow shared read/write access to the same file (database).

Implementation

The following include some of the assumptions of the 4690 Keyed File implementation. They exist largely based upon the objectives as well as the type of data that is being processed in the POS store environment.

- Records are fixed length.
- Key is fixed length (up to 508 bytes)
- Maximum record size for a file is 508 bytes.
- There is only one level of keys.
- Hashing is the most efficient way to locate one record in a database.
  The hashing algorithm has to be able to handle various types of keys and still give fast access:
  - packed numeric keys (Item record keys)
  - Alphabetic keys (Names)

The file size is fixed at origination or build time. It is created based upon a selected number of records that the user feels will meet the requirements for that file for "x" number of years. For example, a store has 100,000 unique items today, but it expects to grow to possibly handling as many as 500,000 unique item codes in five years. The file will be created for 625,000 possible records. (That is room for 500,000 records with a "packing factor" of 80%. 500K records fits in 80% of the available space. The extra space gives room for records to fit efficiently allowing minimal accesses per record.)

This allows the file to grow without reorganization of the file; i.e. no store management of the database is required. If a file eventually becomes "full", the file will need to be rebuilt for a larger size. There are utilities to "create a direct file from a keyed file" and to "create a keyed file from a direct file" which can aid in this process.

The limits of fixed length records and one level of key may be disadvantages to some generic requirements. You cannot have multiple records with the same key. For example, you cannot find all the "Smith's" in the data base using the keyed file services.

Support of 'Read with Lock' and 'Write with Unlock' allows shared read/write access to the same file. Lock occurs only at the "block size" level. "Block" size is a sector , 512 bytes in 4690. It is the size of the "record" that the keyed file system manages. A hashed key would point to a block. The system reads the "block" and the record is likely to be one of several records in that block. 4690 OS locks at the block level in order to reduce the possibility of locking out other concurrent updates to other records in the same database.

For more information on Keyed Files, recommended algorithms, packing factors and general hints, please see the Keyed Files document on the Performance Web-site.

Performance Checklist

The following is a checklist of performance tuning ideas, configurations and features that have been discussed in the previous section.

- Understand POS Environment and Performance Parameters
- Performance Goals
- System Priorities
- System Utilizations
- Configurations
  - Number of Controllers
  - Two Hard Disks per Controller ?
  - Balance POS Workload
  - Which Controller is Primary and Backup to Terminals?
  - What Application Options Improve Store Throughput Capability?

References

The following references, and others, can be found on the Performance Web-site. This is part of the Retail Store Solutions Software Support web page. The way to reach Retail Store Solutions Software Support is to go to www.ibm.com/solutions/retail/store/support and click on "Self-service and SLP Support" under the Point of Sale Software heading. At this page, select "Performance" to get to the references below.

- Keyed Files
- 4690 OS Performance Perspective
- 4690 OS Performance Perspective (4690 V2 & Java)

- 4690 Controller Application Priority
- One Hard Disk vs. Two
- 4610 and 4610HS Printer Performance

- 4690 Disk and Processor Times
- POS Terminal Processor Comparison
- Introduction to LAN Adapters, Hubs, Switches a Routers

- Terminals by Token-Ring
- System Busy
- A CSF/DDS/NT Performance Perspective

- TCC Options for 4690 - Performance
- Large Store Performance Tuning
- Why is Sales Support Processing Slow?