

Distributed computing--an application for a small business
by David Charles Marshall

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science
Montana State University
© Copyright by David Charles Marshall (1982)

Abstract:
Distributed computing systems are growing rapidly in their use in business applications. Geographically dispersed businesses have a need for distributed computing ability with interprocessor communication. Yellowstone Park Service Stations is a remote, geographically dispersed business with the special problem of operating with a six month business season. This thesis discusses the possible application of distributed computer systems to Yellowstone Park Service Stations.

A feasibility and background study of the company and its current situation was done. This was followed by a discussion of current design criteria for distributed systems. It includes interconnect schemes, routing and software protocol level tradeoffs. Special attention was paid to the tradeoffs involved with each of the design choices. The tradeoffs were then analyzed to find the best fit for the application. It was recommended that a network of microcomputers, interconnected through the telecommunications system, is the best choice for the company. Finally, particular systems are recommended for the application.

DISTRIBUTED COMPUTING—AN APPLICATION FOR A SMALL BUSINESS

by

David Charles Marshall

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

December 1982

MAIN LIB.
N378
M355
cop. 2

APPROVAL

of a thesis submitted by

David Charles Marshall

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

Nov 30 1982
Date

Michael A Lassner
Chairperson, Graduate Committee

Approved for the Major Department

Nov. 30, 1982
Date

Wesley Faulkner for D.A. Pierre
Head, Major Department

Approved for the College of Graduate Studies

12-6-82
Date

Michael P. Malone
Graduate Dean

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his/her absence, by the Director of Libraries when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature David Charles Marshall

Date November 30, 1982

TABLE OF CONTENTS

		Page
APPROVAL PAGE		ii
STATEMENT OF PERMISSION TO USE		iii
TABLE OF CONTENTS		iv
LIST OF TABLES		vi
LIST OF FIGURES		vii
ABSTRACT		viii
Chapter		
ONE	INTRODUCTION	1
TWO	FEASIBILITY STUDY	2
	Cost-Benefit Analysis	12
	Benefit Analysis	13
	Cost Analysis	15
THREE	DISTRIBUTED SYSTEM DESIGN FEATURES	18
FOUR	ANALYSIS FOR THE APPLICATION	35
FIVE	CONCLUSION	49
FOOTNOTES		51
SELECTED BIBLIOGRAPHY		55
APPENDICES		61
APPENDIX A – DATA FLOW AND REPORT REQUIREMENTS		62
Data Flow Chart		63
Summary of Report Requirements		64

TABLE OF CONTENTS—Continued

	Page
APPENDIX B – SAMPLE FORMS	66
Service Station Daily Report Form	67
Sales Record	69
Daily Repair Service Report Form	70
Weekly Repair Service Report Form	71
Wrecker Dispatch Log	72

LIST OF TABLES

		Page
1.	Local Office System Benefits	13
2.	Point-of-Sale System Benefits	14
3.	Local Office System Costs	15
4.	Point-of-Sale System Costs	16
5.	Summary of the Cumulative Totals for the Cost-Benefit Analysis	17
6.	Radio Shack System Costs (in dollars)	43
7.	Altos System Costs (in dollars)	44
8.	Burroughs System Costs (in dollars)	45
9.	IBM System Costs (in dollars)	47

LIST OF FIGURES

Page

1. Diagram of YPSS facility locations 4
2. YPSS management structure 4
3. Interconnect configurations 26

Appendix Figure

4. Data flow chart 63

ABSTRACT

Distributed computing systems are growing rapidly in their use in business applications. Geographically dispersed businesses have a need for distributed computing ability with interprocessor communication. Yellowstone Park Service Stations is a remote, geographically dispersed business with the special problem of operating with a six month business season. This thesis discusses the possible application of distributed computer systems to Yellowstone Park Service Stations.

A feasibility and background study of the company and its current situation was done. This was followed by a discussion of current design criteria for distributed systems. It includes interconnect schemes, routing and software protocol level tradeoffs. Special attention was paid to the tradeoffs involved with each of the design choices. The tradeoffs were then analyzed to find the best fit for the application. It was recommended that a network of microcomputers, interconnected through the telecommunications system, is the best choice for the company. Finally, particular systems are recommended for the application.

CHAPTER ONE

INTRODUCTION

The purpose of this thesis is to explore potential computer opportunities for a unique business situation. The Yellowstone Park Service Stations (hereafter known as YPSS), is a small business located in a national park. It is highly diverse in location and somewhat remote. This situation gives it unique problems in effectively computerizing its accounting, inventory, payroll, and budget functions. The primary emphasis of the thesis, therefore, will be to find a system that can solve the communication problems cost effectively.

Poor communication lines, caused by the remoteness, focused the research on computer networks, since they can handle retransmission and prolonged broken communication while remote terminals cannot. The paper is divided into three major chapters. Chapter Two is a description of the current practices of the company. This includes their inherent problems as well as a list of priorities for improvement.

The third chapter is a survey of various types of networks. It contains a discussion of network performance and evaluation criteria.

The fourth chapter deals with a search of potential business systems for the most effective system for YPSS's application. This includes some practical and theoretical design criteria as well as an evaluation of currently available products.

CHAPTER TWO

FEASIBILITY STUDY

Yellowstone Park Service Stations' primary business is operating nine full service gas stations and four auto repair services in Yellowstone National Park.

The business is highly seasonal. Almost all of the business occurs in June, July, and August. A small amount of business begins in May and remains in September and October. Business is virtually nonexistent in November, December, and April, while during January through March a small amount of snowmobile gas is sold. Consequently, the summer finds 150 to 175 employees at 15 locations. In the spring and fall seasons there are approximately 30 to 60 employees at 4 to 8 locations. In the winter, 3 to 8 employees are at 1 to 3 locations.

The business is also geographically dispersed over 90 highway miles in two states. A round trip, from the office to all the stations and back, is about 170 miles and takes about six hours due to traffic. From November 1 to April 30 no road travel is allowed, with the exception of snowmobiles. This leads to communication problems, especially since the telephone service is notoriously poor.

The facilities themselves vary greatly. Built between 1910 and 1970, they range from corrugated metal and wood sheds to modern, large scale (heated) service facilities. Inside temperatures in many locations vary from 0 F to 100 F in the summer and from -60 F to 40 F in the winter. They are located at altitudes of between 5000 and 8000 feet above sea level.

The employees, due to the seasonal nature of the jobs, are primarily college students. Most of them work for only one or two summers (a total of three to six months). Even

seasonal management and office help seldom work more than five or six seasons (two to three years). There are only three permanent employees (see Figure 2), so there is a massive training problem each spring.

At the present time all accounting and business functions are done manually in ledgers. The exceptions are the general ledger, the financial reports based on it, and the payroll. These are sent (individually) for outside data processing in Helena, Montana. The cost is approximately $6,000 per year.

The accounting, inventory, and payroll processes all begin at the various locations each night. The station personnel collect all the raw data needed to summarize the day's business. This includes delivery and transfer tickets, payroll information, pump readings, cash, credit and charge tickets, labor tickets, a list of all accessories (non-fuel items) sold that day, an inventory of all oil and several other products, and the cash register tape. Depending on the location, this may take from one half hour to one and one-half hours of management's time. The next day the reports are picked up by a vehicle circling the Park and delivered to the main office in Gardiner.

During the two to four days following the data collection, a clerk in the office rechecks figures, and calculates total sales in all categories, totals receipts, and an over/short figure. (See report form in Appendix B). Sometime during the fourth through seventh days the report is returned to the station. This means that it can be up to a week before the management at a station knows the day's results. On a twelve to fourteen week "season," this is a long wait. The delay also eliminates any opportunity to catch errors, so they are usually allowed to "bounce" from day to day resulting in erratic and unreliable daily reports.

In the office, this one week delay introduces a delay in generating financial reports. The "current data" is four days old before it is usable. An even further delay, however, is introduced by having the payroll and general ledger done by mail in Helena. The general ledger is thus only updated once a month at a very significant expense. Also, the payroll

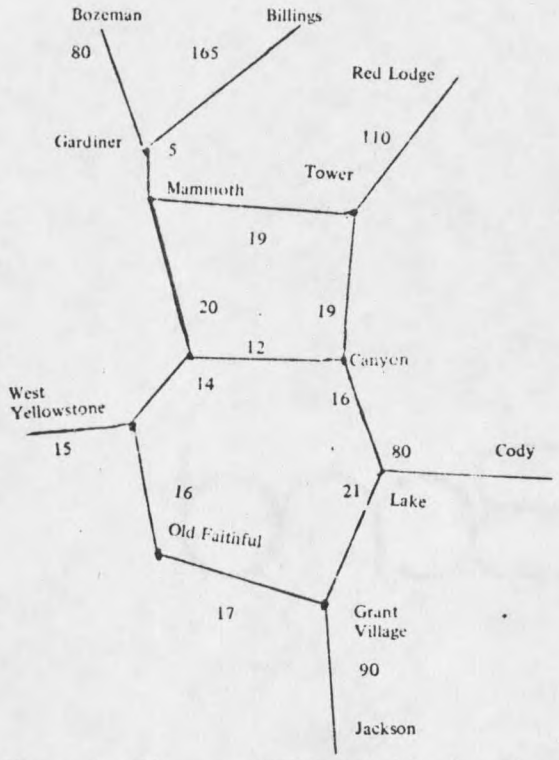


Figure 1. Diagram of YPSS facility locations.

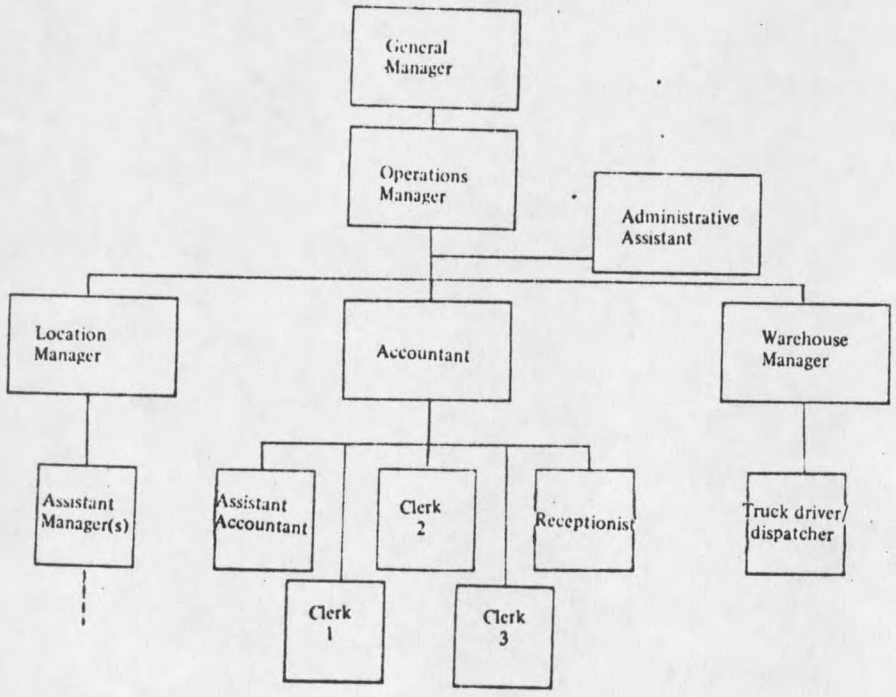


Figure 2. YPSS management structure.

must be complete for the month before the general ledger is sent in to be updated. A delay of approximately one week on the payroll, plus another delay of almost a week on the ledger, means all financial reports are two weeks behind. In a business with a three month operating season, the year is one half over by the time the first financial report is seen. This practice eliminates any flexibility that management has in making decisions based on financial reports during the summer. The reports only can summarize the results after the summer is over.

The primary cost of the outside report and payroll processing, the key punch personnel time,[1] is to redo what has already been done at YPSS. They simply type the data that YPSS sends to them. Thus, not only is there the risk of introducing new errors, but the company pays for the duplication of effort.

The payroll is also sent for outside processing. Verification of time cards and the quantification of hours, meals, etc., is one of the most time consuming processes for the staff. Although this preparation time cannot be reasonably reduced, virtually all delay incurred after that could be eliminated. Currently, it takes about two days to prepare the payroll information, then five to six days for processing (including mailing time), and, finally, an additional two to three days for distribution. This is a total of ten to eleven days. Without the mail time and duplicated effort, certainly four to five days would be an attainable goal for the entire process. In addition to saving duplication of effort, the reduced delay would increase employee good will. Cash advance requests are available to employees who need money before their paycheck arrives. These requests should be reduced substantially by a more timely payroll process, which would save the office staff time. The reduction from eight or nine days down to three days for producing the payroll would make all of the general ledger reports approximately one week more current.

Several activities in addition to those mentioned above also deserve to be considered for computerization. Accounts receivable and accounts payable, for example, would bene-

fit from computerization. The volume of accounts payable is very small, perhaps twenty to thirty accounts over the course of a year. The company's bills are nearly always paid as received so there is little activity. However, they still must be posted to the general ledger. Accounts receivable is different. About 250 accounts are current and perhaps 100 charges per day are posted during the summer. Bills are then produced manually. Automatic bill generation would speed this process substantially.

The annual budget is another function which is currently sent out for processing. This takes virtually no "computing power" to accomplish. In that sense it is wasted money on an outside service. Again, the primary limitation of this process is timing. It takes at least one week, and up to two weeks, to see a budget comparison with the year-to-date situation. Some decisions could benefit by more current knowledge of the situation (expenses or sales) versus the budget estimates.[2]

Other areas that could benefit from computerization are the warehouse and station inventories. Currently, the stations are inventoried completely at the end of the year and daily on about ten items. Tires and batteries are inventoried early in the summer and, occasionally, again later in the summer. One office employee makes an effort to keep the tire and battery inventories up-to-date for all locations. They are usually close to correct, but rarely exact enough for ordering or locating critical items. During the months from June to September, a quick, accurate source of the location of items (especially tires and batteries) is important to station personnel trying to make a sale. Inventory is kept low (very low in September). Locating the required product quickly and accurately may mean making the sale. Often the customer will not wait for nine to ten locations to be called, have their inventory checked, and return an answer.

Sales could be increased with one accurate and easily available inventory by allowing timely updating of inventories in the spring. Each fall (from mid-August on) inventory quantities are allowed to drop on all items. Little or no reordering is done. Business is low

from mid-September until the end of the year so missed sales are minimal. May is not a critical month either, since business is still relatively slow. It would be beneficial to moderately rebuild inventories at this time, however.

In June, sales increase dramatically, often before management rebuilds the inventory levels. Sales reach peak summer levels by July 1. Any low inventories during this period can cause enough lost sales to significantly affect a station's profit for the summer. After managers arrive (about June 1-10), their first priorities are cleaning and fixing the station for the summer and training employees. Ordering inventory is based primarily on the manager's wishes and is often not done until late in the month. This is especially true for new management personnel, who often make up a substantial percentage of the total number of managers. They frequently are unfamiliar with the items that particular station can sell or has sold in the past. By late June, they have begun to realize what sales are being missed, but by early July when the items arrive, many sales have been lost. This could amount to as much as one-third of the summer's sales of a product at that location.

An easily available list of all current inventories, with the last one or two years' sales, could eliminate much of the problem. It would allow the office staff to establish an approximate inventory level for each location in May, based on previous sales and current (exact) inventory levels. Yet it would not require taking a physical inventory at that time. It also would allow new management to order quickly and accurately when they arrive. They would not need to guess at previous sales levels or spend time taking inventory to find out what present levels are.

Warehouse inventory is currently maintained on index cards. While accurate, this is a tedious job and often falls behind by a few days. Automatic updates based on invoices and shipments would be desirable. Also, retrieval by office staff of the quantity of an item currently on hand would be improved. No automatic mechanism is currently present to alert management to low inventory levels. To check all current levels each file must be syste-

matically pulled. A short, quick summary would save time and allow more frequent evaluation of inventory levels.

Another central office function of interest is the dispatching of wreckers and service trucks to disabled vehicles. All service calls in the Park must be authorized and dispatched by a central communications center. During the day, this center is the YPSS office. At night, it is the National Park Service. Each request is logged (see the wrecker dispatch form in Appendix B). While this process would not be readily improved by automation, it would be of significant interest and value to be able to compile statistics on these calls. Questions such as: How many calls? Where to? At what time?, etc., would provide valuable information. The road service business is not very profitable largely because 10 to 20% of the calls are "dry runs" (no vehicle there). The ability to easily compile statistics on these calls should make them easier to reduce, thus limiting wasted money and time. For example, wreckers might be more optimally located than they are. Certain types or locations of calls might be prone to being "dry runs." Perhaps a better way to verify these calls might then be able to be used.

The final area of possible improvement is in the office paperwork. Currently, memory typewriters and copy machines are used. While these perform adequately, a memory typewriter does not have the speed or editing capabilities of a text editor. This is not a major problem, but a computer certainly could decrease the time spent on outgoing communications while increasing their quality. Copies with very minor changes can be produced more easily. It also would allow display of the text before a printed copy is generated.

A list of the reports required for the various functions mentioned above is found in Appendix A. In general, these reports follow current accounting practices and standards. Some reports are completely unique to YPSS, however. These include the location reports, the year-to-year comparison reports, the gas delivery ledger, and the wrecker dispatch log. Several of these forms are found in Appendix B.

YPSS management has discussed several priorities for improvement that would be helped by automation. There are other improvements which could be made that as yet have not been defined as problems. The management's priorities are to:

1. Reduce office staff to save money.
2. Eliminate outside processing.
3. Increase error checking abilities.
4. Reduce computational errors.
5. Reduce time spent on reports by station personnel.
6. Provide more up-to-date inventory (especially on tires).

Other areas where the computer could improve operations are to:

7. Obtain more current data on financial reports for decision making.
8. Obtain a complete, current inventory with sales information, resulting in more timely ordering capabilities.
9. Compensate (both at the station and in the office) for untrained personnel by using faster and simpler procedures. (This could include things like being able to correct report "bounces" from bad data immediately and having less office paper-work forms).
10. Obtain ability to query for answers from the wrecker dispatch log, reducing dry runs.
11. Obtain faster and neater text editing capability.
12. Allow the generation of new reports and summaries in the future from present information with little or no office overhead.

The present system appears to work quite adequately for the company at this time. However, there are problems that could restrict growth. This system offers almost no chance of being flexible enough to reduce the time, effort, or expense involved in process-

ing the accounting functions at a future date. In fact, an additional report requires a significant addition to the office staff's workload, whether it requires new data or not.

The primary areas of a computer's expertise are in speed, extra computing ability (i.e., summarizing and calculating ability), and in reduction of errors. All of these could be achieved to a greater degree than they are with the present manual system. These improvements would help to reduce or eliminate the problems mentioned above.

YPSS's need for computer service can be divided into four basic levels of development. Although they may be done all at once, or separately, any system which implements less than the total should be expandable to include the entire system.

The first step is to get a small office system with editor, ledger, payroll, inventory, and budget functions (inside dashed line on data flow chart in Appendix A). This would be a microcomputer, hard disk, CRT, printer, and possibly a tape backup. All software would be purchased.

The second step would be to expand this to a total main office system. This would not include any extra hardware, but would include all of the software on the data flow chart. The additional software would probably need to be custom written.

Step three would be to locate microcomputers at the remote locations. Communications hardware and software would be needed to relay the data collected at the remote locations to the main office system. This would enhance station access to data. It would also eliminate the data duplication from the location report to the computer because station personnel would enter data directly. Feedback on errors could increase greatly because station personnel would receive instantaneous results from their data.

Step four would be to incorporate point-of-sale cash registers and automatically read pumps into the location microcomputers. This would reduce both data gathering time and the errors associated with it. This step would also greatly reduce the time that station personnel spend on paperwork.

Several factors suggest it would be prudent to implement this plan slowly. First, YPSS management traditionally has been fiscally quite conservative. They prefer to stay with a trusted procedure rather than to commit to a capital expenditure that might produce future headaches. A huge investment would be a large commitment to something very unfamiliar. This is an important consideration since without active management support any system is likely to be unsuccessful.[3] YPSS would tend to agree with Robert Townsend that, "I've never known a company seriously hurt by automating too slowly, but there are some classical cases of companies bankrupted by computerizing prematurely".[4]

A second reason for slow implementation is that few employees remain for the winter. All training must be done during the busy summer season. This would be more easily accomplished if only one new section of the system had to be implemented and taught to personnel each summer.

A third reason will be seen from the cost benefit analysis. The local office system has the shortest payback period. Adding remote data collection has the largest capital investment. This is especially true if the cost of point-of-sale registers, electronic pumps, and heated offices is included. Eventually all locations will probably have new buildings with electronic pumps. However, this is still in the future for many of them. The size of the investment for the remote locations makes pausing after complete office automation logical. The office would be essentially computerized but nothing would be committed elsewhere. To have the payback period be reasonable, the computerizing of the remote locations needs to wait until point-of-sale registers are bought and the locations are upgraded. This could be five to ten years away. Although step three could be accomplished separately (just remote microcomputers), its payback period would be relatively long if point-of-sale data collection did not occur within the next several years. These two divisions—office automation and remote data gathering—correspond to the two major data gathering options described by Birkle and Yearsley in *Computer Applications in Management* (p. 63).

Cost-Benefit Analysis

The primary needs of the company with regard to a system to purchase are listed below in approximate priority order:

1. Low cost. This is vital to purchase approval.
2. Short term payback period. This is needed since present accounting costs are not high.
3. Ease of use. Since it will be used by primarily short term, untrained personnel, anything requiring a knowledge of computers or learning extensive command sets will be prohibitive.
4. Serviceability. This is most vital in the summer. It would be most desirable to have a contract for the summer with fast (one day) service on both hardware and software. In the winter it could be cheaper, slower service since needs are not as pressing.
5. Audit ability. It must meet accepted accounting standards for document traceability.
6. Ability to be networked with other computers and cash registers.
7. Ease of programming. This is important since all program maintenance and modifications done at YPSS must be done by non-programming personnel and since custom programming costs must be kept to a minimum.
8. Ability to expand memory and peripherals. This would be needed if the business expanded substantially in the next ten years or if more processing requirements were introduced.

As in any business venture, price and justifiability (payback period) are preeminent. This means justifying purchase price, and training and operating costs, in terms of reduced costs in the problem areas listed above.

Both major system divisions, office computerization and remote data collection, were analyzed for costs and benefits for a five year period. This is the generally accepted life of an information system.[5] The life at YPSS should be much longer (at least ten years), with only minor changes, since the business is not in a fast growth situation and has limited needs for most of the year.

Benefit Analysis

The Benefit Analysis was made for two overall system types. First, it was done for buying a standard pre-packaged accounting system. In the third year, modules would be added to do location reports and automatic ledger updating, as well as "peripheral" reports such as gas inventory, wrecker dispatching, and year-to-year sales comparisons.

Table 1. Local Office System Benefits

	Year				
	1	2	3	4	5
Sales increase	0	600	1,500	2,000	2,500
Outside process saving	0	6000	6,000	6,000	6,000
Office staff reduction	0	2400	4,800	4,800	4,800
Totals	0	9000	12,300	12,800	13,800
Cumulative totals	0	9000	21,300	35,100	48,900

The sales increase is an estimate based on two factors. One is the speed and accuracy of looking up a request for an item for a customer. The other factor is more timely ordering by both the warehouse and the stations based on their known inventory and past sales. By year five, this is estimated to be 100 tires over the course of the year, or about one every other day someplace in the Park. I believe this is a reasonable estimate of the number of missed sales on batteries and tires due to problems caused by distributing these items. The figure would run even higher if the improvement in inventory causes personnel to be more confident and aggressive in pursuing sales.

The benefit resulting from outside processing reduction is the elimination of the need for that service. Primarily this savings is in eliminating the extra person involved in translating that data. No benefits were estimated for year 1 assuming that a year might be spent testing results against current practice. If not, benefits would begin immediately.

Office staff reduction is based on the elimination of some location report work (specifically report calculations, elimination of the sales journal, and some hand calculations on gas reports and the year to year comparison report). Also, inventory cards for the warehouse would be eliminated. The reduction was calculated from an estimated cost for a report clerk at $7.00 per hour for 16 weeks, or $4800 per year.

Not quantified for either this or the following benefit analysis are those benefits received by more current financial information, reduction of errors and station "bounces," or benefits gained from improved knowledge about wrecker dispatches.

The second benefit analysis is for a complete system. This would include point-of-sale cash registers, pumps tied to the computer, and all data entry at the time of sale by the cash register (propane, charges, credit, etc.).

Table 2. Point-of-Sale System Benefits

	Year				
	1	2	3	4	5
Sales increase	0	1,000	2,000	2,500	3,000
Outside process saving	0	6,000	6,000	6,000	6,000
Station staff reduction	0	1,000	1,500	1,500	1,500
Office staff reduction	0	4,800	9,600	12,000	12,000
Totals	0	12,800	19,100	22,000	22,500
Cumulative totals	0	12,800	31,900	53,900	76,100

Sales increase is as before, with a minor addition due to a more current knowledge of all products stocked at the stations, not just tires and batteries. Current information on how all products are selling around the Park would be available.

Outside process savings are identical to the local office system calculation.

Office staff reduction is based on the same expense as above. But with all reports essentially done at the station, two or three report clerks could be eliminated eventually. Only the time for checking credit and charge tickets remains for a clerk.

Station staff cost reduction is based on the saving of one hour per day per station. Admittedly this is an estimate. However, time should be reduced on paperwork by the elimination of pump reading, the miscellaneous item sales sheet (see Appendix B), propane tickets, daily inventory (although that would probably still need to be checked weekly), and time spent on the location report form. Time cost was calculated to be approximately $10.00 per hour.

Cost Analysis

Costs have been calculated for both the minimum office system option and for a complete point-of-sale cash register, real-time data gathering system.

Table 3. Local Office System Costs

	Year				
	1	2	3	4	5
Hardware	15,000	0	0	0	0
Software (purchased)	5,000	0	0	0	0
Software (custom)	0	3,000	2,000	500	500
Training	1,500	0	1,500	0	0
Maintenance	0	1,600	800	800	800
Totals	21,500	4,600	4,300	1,300	1,300
Cumulative totals	21,500	26,100	30,400	31,700	33,000

For the minimum cost system the hardware cost is an estimate based on preliminary talks with several vendors. It assumes a microprocessor with some hard disks, a good quality printer and a CRT. It is intended that a large enough system should be purchased so that no expansion is necessary for the first five years.

The purchased software cost is also based on the preliminary vendor interviews. It would include payroll, budget, basic inventory, and the general ledger units, as well as the text editor and operating system.

The custom written software is for the report programs, the automatic ledger and inventory updates, the wrecker dispatch log, and the gas inventory (see data flow chart). It is based on charges of about $12.00 per hour for writing and testing by an outside firm. This is the current student rate in Bozeman. It could be higher if professional help was required. However, it could also be lower if report writing packages allowed most of it to be done in-house.

Maintenance is an estimate based on one service call per month in the summer and two for the rest of the year. Each call is about $200. Year 1 is covered in the purchase price. This should be an ample allowance.

Cost of new personnel training is not included because this should be similar to, or less than, present initial training costs. It is assumed electrical and air conditioning costs would not change significantly. Form costs are also assumed to stay constant.

Table 4. Point-of-Sale System Costs

	Year				
	1	2	3	4	5
Hardware	75,000	0	0	0	0
Software (canned)	10,000	0	0	0	0
Software (custom)	5,000	2,500	500	500	500
Training	3,000	2,000	0	0	0
Maintenance	0	3,000	1,000	1,000	1,000
Totals	93,000	7,500	1,500	1,500	1,500
Cumulative totals	93,000	100,500	102,000	103,500	105,000

The point-of-sale cost estimate is based on the same preliminary discussions. In addition to the office system, it includes microcomputers at seven locations with communication connections. It also includes the purchase of cash registers. It is assumed that this would be done when stations are upgraded and consolidated according to the present com-

pany plan. Until that time it is expected that point of sale data gathering would not be viable.

Software costs are estimated to be double the costs for a local only system. This covers network communication software as well as that for point-of-sale communications. Since all units are identical at all locations, it is assumed that programs written for one location will serve all locations.

Training expense should increase sharply because station personnel are involved. This should be an initial expense only though, since year to year training of new personnel is always required.

Maintenance also will increase, but units probably can be replaced with spare units and returned to the vendor for repairs. Service calls to the stations should then be minimal or nonexistent.

It is obvious from the above payback period comparisons that the remote data gathering system is not feasible at this time. The local office system, however, shows a payback period of less than four years and deserves further study and consideration.

Although the remote system is currently not feasible, any system bought at this time should still be compatible with that goal. Its evaluation may well change in the next several years and remote data gathering is still the ultimate goal.

Table 5. Summary of the Cumulative Totals for the Cost-Benefit Analysis

	Year				
	1	2	3	4	5
Office only (bene.)	0	9,000	21,300	35,100	48,900
Office only (costs)	21,500	26,100	30,400	31,700	33,000
Remote system (bene.)	0	12,800	32,900	56,400	80,900
Remote system (costs)	93,000	100,500	102,000	103,500	105,000

CHAPTER THREE

DISTRIBUTED SYSTEM DESIGN FEATURES

YPSS's computing needs are at disperse locations. Telecommunication lines were found not to be reliable enough to maintain the constant communication channels required to centralize the computing machinery. Therefore, it was decided that instead of remote terminals and one central computer, each location would need some local "intelligence." Distributed computer systems were studied in order to form the basis for design of such a system.

Multicomputer system use is growing extremely fast, due to the demand for related, interconnected processing at dispersed locations in many applications. This distribution increases local speed and control while reducing CPU contention and retaining communication with others.[6] Another primary factor in the growth is the relatively small incremental expansion cost associated with these systems. This growth is just beginning for data processing systems. "Distributed data processing and small business computer systems have exploded from virtually non-existent markets in the mid-1970's. . . . these markets are expected to surge ahead to $5.5 billion by 1982."[7]

The major difficulties with networks are communication and task partitioning. Communication means deciding who can talk to whom, defining their "language," and scheduling the times that they will converse. Task partitioning involves deciding who should know what information and do what jobs. This chapter deals with general guidelines to help solve these problems for particular systems.

One of the basic distinctions between types of distributed networks is coupling.[8] Tightly coupled networks have a common shared memory and shared peripherals. Multi-

processor computers fall into this category. The advantages are less total memory and high data integrity. The problem with tight coupling is the communication and synchronization of processors using the memory. In loosely coupled systems, each processor has its own private memory. Communication is accomplished by message exchanges over a common channel or bus. Memory access is quicker and easier, but more inter-processor communication is required to exchange the information, and, therefore, more hardware is required. Loosely coupled systems are much easier to modify than tightly coupled systems. They can continue with degraded processing service when one or more local failures occur because they consist of independent systems. For example, if a directory type system is used to access data in other computers, only that information which is exclusively in the failed network section will be unavailable for use. Tightly coupled systems may fail entirely with only one bus, memory, or processor failure.

Early systems were generally loosely coupled with one master or lead computer controlling all others. This design is still best for some applications although more "independent" or equal systems are also being designed. Independent systems are designed so no component failure impacts system availability (or impacts it as little as economically feasible).

Computers can communicate over logical channels (physically: buses, cables, radio waves, etc.) by one of three methods. These are circuit switching, message switching, or packet switching. Each has its particular strengths and weaknesses.[9,10]

In circuit switching, a call is established between the two parties involved. It is analogous to a telephone call in that a two-way, full-duplex line is established that is dedicated to those parties' communication. The line is maintained for the duration of the call. Its biggest advantage is that it is compatible with the telephone network. (In fact, it almost always uses that network.)

However, circuit switching has several disadvantages. First, calls are relatively slow to establish (compared to a computer's speed). This means a lot of time is spent waiting for

the call to be set up. Second, the call ties up the line for its duration. This is true even though much of the time may be idle time spent waiting for an answer. This causes the familiar "busy signal" to any other party trying to establish a call. Finally, it requires that the two devices talking to each other be compatible since they talk directly and not through a translator.

With message switching the entire message is given to the network. The network then takes the responsibility of delivering it. If communication lines are open, the message would be delivered quickly. Otherwise, the network stores it and forwards it when the path and the receiver are available. This method utilizes channels more fully than does circuit switching and allows for the translation of the message by the processor that stores it in the network. However, it requires processors with large memories to store the messages at each switch. This is because each switch must retain a copy of messages it has forwarded until it is certain that they have been delivered. For real time or interactive traffic this type of switching can give long delays. Long messages, especially, can tie up the transmission lines, thus blocking many short messages that arrive while the longer ones are transmitting.

The most natural type of transmission between computers is packet switching.[11] Packet switching divides the time intervals on the line into packets or slots. These slots may be fixed or variable in lengths. They may be dedicated to a single sending device or open to contention between many such devices. Network "nodes" may hold the message temporarily to avoid network congestion, but then forward it, destroying their copy when it is successfully received at the next node.

Packet switching does require the use of a small processor at each node. However, it has many advantages over both message and circuit switching. It has high line utilization because it allows one or several devices to talk to one or several other devices concurrently. It can have adaptable routing. This means that since messages are short they can be re-

routed to avoid line congestion or failures. Because messages are broken into short packets they compete, and, therefore, are more consistent in their end-to-end delays. It can also use the processors to translate messages between incompatible devices. Most of the following interconnect schemes, such as time divided buses and loops, use some form of packet switching.

The fundamental design attribute of distributed systems is their interconnect scheme. It may be a single cable dedicated to the two computers, or a shared memory, or it may be a path shared by either time or frequency division between many computers.

Which scheme is chosen depends on the application and, to a large extent, on the physical dispersion of the computers. Hierarchical and memory interconnected systems are usually used for systems in a localized area. Future expansion capabilities also must be considered in the choice of a systems interconnect scheme.

The first class of interconnect schemes is that of shared memory. Memory either can be main or virtual, but one memory serves all processors. Communication only is done through accessing the same data on this memory. These systems are characterized by high data availability and integrity, and high CPU availability to the user.

Three types of physical schemes presently are common for connecting the CPU's and the shared memory.[12] The first is a time shared, common bus with the memory on it. Two serious problems arise when all transfers use the same bus. The first is reliability, since bus failure destroys all communication. The second is contention for the bus. High transfer rates easily can overload the bus. These systems usually require each processor to have some private memory and I/O, in addition to the shared data base, to help control these problems. Resolution of bus contention problems then can be based on fixed priority queueing, FIFO queueing, or daisy chaining. Shared memory system interconnection by shared bus is, however, relatively cheap, simple, and expandable merely by adding devices to the bus (within the bus contention limits mentioned above).

The second physical connection is a crossbar switch. This switch can set up a different "line" from any unit on the system to any other, allowing high transfer rates. However, the limitation is that for N talking devices and M listening devices, there are N and M connections or "lines."[13,14] The switch matrix logic to handle these connections grows rapidly and is difficult or impossible to incrementally expand. The switch must be designed to be large enough to handle all future system expansion. Failure of the matrix switch is catastrophic to the system, so again some private memory is useful to allow soft crash and local processing capabilities.

The third physical scheme is the multibus/multiport shared memory. This also has a line from all talking devices to all listening devices except there is a wire for each line. All logic is controlled at the input-output ports of the device instead of at the crossbar matrix. Contention for passive elements still exists, but not for communication lines. Reliability is improved since failure of one line or unit does not affect others. Incremental expansion and flexibility are limited to the number of ports on the memory, however. The memory units also can be very costly because of the high number of separate ports and the built-in logic to regulate their contention.

Shared memory communication can be logically viewed in two different ways.[15] The first way can be described as virtual memory. In this view all shared memory is "addressable" from any location and regardless of location it is looked upon as if it were in a processor's own "virtual" environment. Addresses not local to the processor simply are sent to the bus, where bus arbitrators handle the actual finding of the data in another module. The difficulty is that too many processors addressing the same memory will develop contention problems. This can be solved by using short term cache memories to save the data. However, cache memories lead to multiple copies of data. This introduces data integrity problems, when one copy is updated while another is being read.

The second logical view, called mailbox, solves this contention problem. Here the shared memory acts only as a message center to "relay" messages by holding them until the receiving processor requests them. Each processor has its own local memory, but communicates through the shared section. This generally works best if the application is well defined and static over time so that message quantities and types are known in advance.

In any shared memory system, access cycles to the memory must be assured integrity before any subsequent accesses. No processor can be allowed to read data while it is being updated. This usually is done by the use of dedicated interrupt lines or by software monitors or semaphores which "lock" the section of memory while it is being written to. In choosing a system, the performance (the extent that the processors interfere with each other) can be calculated ahead of time. Pages 38-43 of Weitzman have a detailed description of this calculation for a specific shared memory system.

In addition to shared memory, a second class of interconnect schemes is the shared bus.[16] Here connection is on a common channel or line, with frequent addressing directing the messages. Many messages coexist, but with unique destinations. The bus can be either parallel or serial. Either central or distributed control of either variable or uniform length slots may be used.

In a centrally controlled polled bus a central controller asks each sender in turn for a message, and then transmits it, if necessary. If high transfer rates are needed, this can be slow since one must wait until asked. While ways of speeding this up exist, they also complicate the system by adding poll lines or frequency modulators.

A second central control system is interrupt driven. This is like the poll except interrupt requests are queued, resulting in no waste of bus time for polling. This works best for loads where first one and then another source monopolizes the line for a long period of time. Separate interrupt lines are needed for each processor, however.

Both of these two systems essentially are message switched systems although they could be made to appear packet switched if the message length was limited. Methods exist for calculating system capacities for either of these schemes.[17]

A third central control system is the slotted bus. This is a parallel network that collects all messages and then sends them to their destinations. Each sender is designated a unique time interval to use and may fill that interval and then wait for its next slot. A repeater retransmits this message onto an outbound cable with an address for its destination. This waiting can be slow for short, bursty type messages. Allowing slot contention severely complicates message collision software and can force up the number of retransmissions if there are many messages.

Distributed control is more reliable than centralized control since failure of the controller is no longer catastrophic. But the lack of central control also makes it more inefficient, since knowledge of the state of the network is not available at any one place to coordinate routing decisions.[18]

Two major types of distributed control systems exist. The first is frequency division multiplexed. This uses a single bus, but multiple frequencies, allowing only one-to-one communication on a frequency not one-to-many. Users must keep track of who is using each frequency with either a static table or constant updates.

The second distributed control system is time division multiplexed. Centrally synchronized time slots are available for packets. All participants keep track of who is in each slot with constant updates. Complexity is high, but so is reliability and modifiability. Slot or packet lengths can be either fixed or variable length.

A variation of the time division multiplexed system is the global multiple access bus. This type of system uses collision control software to control bus slot contention and retransmits messages which "collide." This network appears best for systems whose mes-

sages are predominantly randomly generated, since retransmission increases system load sharply.

The third interconnect scheme is a loop. In these systems, a message simply travels around the loop until it either reaches its destination or returns to its sender. Bit patterns are used as addresses. The major advantages are no message routing problems, connections that are quick and simple, high throughput, and a relatively low purchase cost. Reliability is the biggest problem since a single failure in the loop or one interface will break the path around the loop.

All types of loops can have reliability increased, but at the cost of hardware redundancy. A standby loop parallel to the main loop can be used, but this requires special interfaces between the two loops to control message routing when the main loop develops a problem. (This also can be a two-way "self heal" loop, which is slightly better.) Double loops and braided loops further increase reliability, but also increase cost and complexity substantially.[19]

Most loops use some type of packet switching (although the packet length may be long enough or variable enough to make it nearly message switching). It may use permanently assigned slots, slot contention with collision control, or one of the following slot control protocols.

The Newhall loop passes a token around the loop. Whoever has the token may send a packet. This is simple, but only allows one transmission at a time. At low transmission rates this is adequate.

The Pierce loop improves this by using fixed slots where packets may be inserted into an empty slot at any time. It is more complex, but allows many messages to be transmitted at a time. It may be best for medium data rates.

The best performance at any data rate is achieved by delay insertion.[20] Here buffers are used to "open" the line for a packet. The buffer delays the loop contents while the

packet is put onto the line. Efficiency is maximized but cost and complexity also are high. It can achieve essentially 100% loop utilization with no collisions and no empty slots.

The fourth type of interconnect scheme is a star. It is relatively simple and line or point breakdowns are minimal problems. It is best used when one sender talks to many receivers.[21] However, a breakdown or congestion at the switch, or central control, is usually catastrophic.

Hierarchical is another interconnect structure that can be used. Each processor has a direct connection with only those directly above or below it in the hierarchy. All messages must be passed through these. This is a tree structure with control always centered at the higher level of any connection. Reliability is a problem, especially for the highest level processors and links. Hardware redundancy will solve this problem but at a higher cost, since these are the most expensive components in the system.

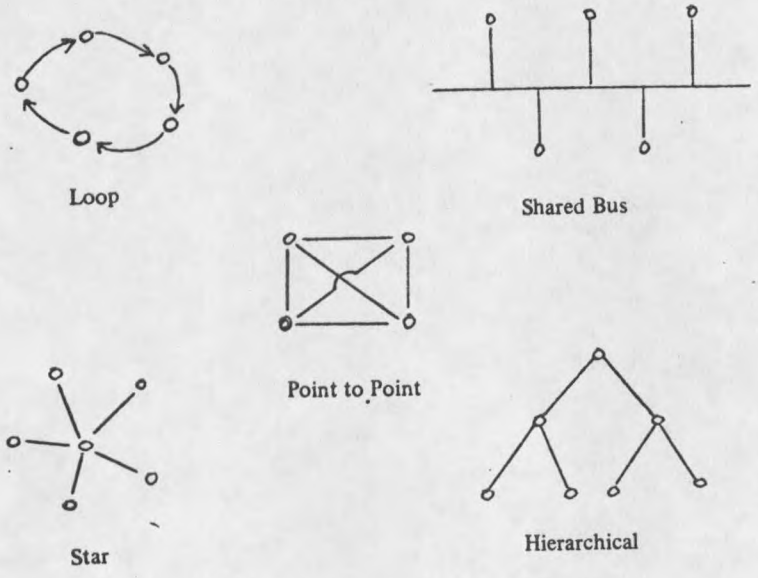


Figure 3. Interconnect configurations.

The final interconnect structure is point to point connected systems—either partially or fully connected. Reliability is very high since every computer is connected to all (or most) others. Cost and complexity grow extremely quickly as the system enlarges. Expan-

sion may be hampered seriously beyond the maximum number of I/O port connections each computer can have. This type of system is most advantageous when there are widespread users with high message transfer rates. (Usually it is a partially connected structure with several processors to a "node" like ARPANET.[22])

Another important attribute of a distributed network is its routing plan. Four major types of plans exist: flooding, random, directory, or adaptive directory.[23] Routing is most significant for multi-node networks since routing between a two or three node network is trivial.

Flooding is the simple case where each node with a message or packet to send makes multiple copies of it, and sends one to each neighbor (except any it received the packet from). The major advantage to this type of routing is that if a route to the destination exists, the packet will get there. Of course, the cost is high in network time and overhead because the message will travel over all links in the network.

In randomly routed networks, each node which receives a packet picks a random neighbor and sends it to him. This is effective at equalizing traffic over all the links in a network, but packet delay in traversing the network can be long if the network is complex. The packet is still, however, guaranteed eventual delivery if a route exists to the receiver.

In a directory method, a static table is maintained at each node telling it what route to use to get to any other node. While the path can be chosen to be short, the table must contain secondary choices since the primary route could be blocked. Unless the table contains all possible choices, it is possible for a route to exist and the table not to contain enough information to be able to select it.

An adaptive directory is probably the most efficient technique. It also requires the most sophisticated software. Adaptive directory solves the problems with directory routing by constantly updating the table for the most efficient route to a destination. This allows

it to avoid failed lines or temporarily congested lines without multiple choices constantly in the routing tables. It keeps the routing flexible, but at the cost of constant communication about the state of the network between its nodes.

Like any other computer, distributed systems are made of hardware and software. Individual processor software is similar to that of a stand alone computer, however extra software must be present to deal with system communications. Several tasks must be accomplished by this software. The first is software synchronization, usually using semaphores or monitors. The second is communication coordination. In tightly coupled systems, this means managing joint memory. In loosely coupled, it means transmitting messages between processors without errors and at the appropriate time. A third job may be task partitioning between system elements. This is the decision about load and file distribution and peripheral use. Finally, diagnostics and recovery must be extended to the entire system.

In tightly coupled (shared memory) systems the common memory must be carefully controlled. Data integrity must be maintained. One way to preserve data integrity is with the use of semaphores, originally introduced by Dkjkstra.[24] This is a simple way of incrementing and decrementing a variable to indicate a memory section's availability. Thus, while one process is using a global variable no other process may access it (the semaphore is closed). Instead the process is queued to wait for that variable to become available (the semaphore to open).

Monitors also can do this synchronization (using languages like concurrent Pascal).[25] Monitors are programs that contain all information on a variable and assist in structuring concurrent processing. Each program or monitor can be used by only one procedure at a time. The variable is controlled exclusively by its monitor and all modifications or accesses to the variable must be done by that monitor. The great advantage is the clarity of programs compared to semaphores because all control is modular and well defined.

Interprocess software communication generally is done by message protocols. There are several different, but related, divisions of these layers in the literature.[26,27,28] Each logical layer of communication has protocols which separate it from all other layers except those directly above and below it. This means that a level uses all protocols in the level below it with no knowledge of how they work. The lowest protocol level (physical) is hardware implemented. The highest level (or levels, depending on the divisions) is the user level protocol and is independent of hardware. It can make a heterogenous system transparent to the users. The advantage of this design is modifiability. It is achieved by allowing each layer to work autonomously to execute the higher level command at the lower level.

After the physical protocol level comes the communication or data link protocol level. This control level uses the hardware to achieve a communication link with a system "grammar". It does error checking, coding and formatting, and creates data transparency automatically, unseen by any higher level of communication. It moves the packet from one node in the network to the next and verifies that the packet was not corrupted. Although many error checking schemes exist, perhaps the best is the cyclic redundance check. It divides the packet by a single common divisor and appends the remainder to the packet for verification.[29]

The end-to-end communication layer uses the data link control layer to accomplish the task of moving messages through a network from source to destination. Here formatting for packaging as well as source and destination codes are defined. It controls use of the physical data links to achieve the logical link that the user requires for his communication. It accomplishes the transfer of packets from one end of the network to the other.

The top layer is the user control level. This level interacts with the user to determine his needs and uses the end-to-end level to reassemble files from packets, control their

security, and talk to other application programs to solve those needs. It also must be able to determine unachievable tasks.

In ISO (International Standards Organization) protocol divisions this top layer is broken into levels five through seven.[30] Level four is the transport level. It accomplishes the full transfer of a message or file. If the message is in many packets, it must reassemble them at the receiving end.

Level five is the session level. It is responsible for achieving the desired communication between the users, even if this requires many messages to be sent and received. It is also responsible for network security by checking for access authorizations.

Level six is the presentation level. It controls display information, coding, and general transportation of information between the application and all peripherals, including the network.

The final level is the application or user process level. It actually executes the application program.

Some systems trade some of this clarity and transparency for speed of execution by imbedding hardware protocols in the higher levels. This tradeoff should be used only if efficiency is paramount.

In general, communications software is independently designed for each system type and can be a large initial cost. Structured levels of protocol with tasks like load partitioning, diagnostics and recovery, and interprocessor communication placed in separate levels simplify and reduce this cost. Monitors and semaphores assure data integrity in shared memory communications.

Distributed multicomputer systems can compete with large computers in some applications. They usually increase CPU availability to the user, are often cheaper, easier to reconfigure, provide more flexible processing capabilities, increased reliability, and the design can be tailored to unique system requirements. However, their hardware and soft-

ware may be more complex, thus increasing the initial investment of both time and money.[31,32]

Cost analysis varies with the specific system. In general, a very large number of micros or minis can be used before reaching the hardware cost of a large computer. Typical mainteinance costs are lower over the system life. However, software costs usually start higher for distributed systems. All these tradeoffs must be analyzed for each system to determine individual cost effectiveness not only of a single computer versus a multicomputer system, but also between the various multicomputer systems.

The choice of hardware and whether to buy or build a distributed system is based on two factors. The first factor is availability and the second is cost. Depending on the interconnect scheme chosen, hardware may or may not be available off-the-shelf. If it is not, in-house development is necessary. Cost may prohibit an interconnect scheme's implementation for either type of development.

Both micros and minis can be connected either master-slave or master-master with off-the-shelf buses and arbitrators. Point to point interconnection is common and relatively inexpensive for small networks.

Minis are connected into more complex systems. For shared memory configurations, only crossbar switches are not generally available. Therefore, they probably would be the most costly system for an application. Shared memories, buffers, cache memories, etc., are all commonly available.

Shared bus networks for both homogenous and heterogenous systems are also available off-the-shelf using co-axial cable and utilizing time division or multiple access (collision control) protocols.

Star systems with communication switches at the center can be found connecting up to 250 computers. Built-in redundancy makes these the most reliable, but expensive off-the-shelf networks.

Bus windows or bus-to-bus connections are available to connect processors or networks to other processor's or network's busses. These devices automatically translate and relay addresses intended for other networks. This allows complete transparency to the requestor. The major risk with these connections is system deadlock. (Requests from both systems waiting for the other to proceed.) It can be avoided, but with additional expense and complexity.

The most important aspect in choosing a successful interconnect system is process characterization. What are the system requirements? These include normal system design considerations plus geographical dispersion.

The process of analysis can be broken into six steps[33] :

1. Process identification.
2. Problem decomposition.
3. Process interaction.
4. Performance requirement specification.
5. Architecture selection.
6. System tradeoff analysis.

Process identification is the first stage of process analysis. It identifies a user's needs with as little implementation suggestion as possible. This is not a complete problem description, but only a system definition, breaking the problem into logical subunits.

Problem decomposition is the next step. The block or flow diagrams for the system are created. Data flow graphs, sequential, if-then- else, do-while graphs, and so on, are useful at this point in creating a process architecture. This is at a total system level.

In process interaction, the process architecture is extended to show interaction between the various processes. Use of exchange functions to define data communication points between the various processes is helpful. Several diagram types exist for this function including N-2 charts.[34]

During performance requirement specification, precise system needs (response times, I/O rates, data speed requirements, minimum error rates, etc.) are described. Also physical constraints like temperature and location are listed. Tradeoffs for systems are not made, but only the goals described. Of course, all of these will affect cost and ease of development, and therefore, system choice. Only after all ideal performance requirements are made, can you proceed to look for the best type of system to satisfy them.

Architecture selection is the next stage. Now it is decided which computer and components best solve the definition in step four. It is determined whether an off-the-shelf solution is satisfactory, or whether either hardware or software must be modified or custom produced. Partitioning of the system will have a tremendous impact on cost, performance and reliability in the future by limiting processing side effects and quantities of data transfer. Failures must also be designed to have minimum impact. Depending on each process (program) size, partitioning within the system will involve tradeoffs between speed, logic complexity, concurrent processing, and reconfiguration ease. Within constraints, logical units should be kept together in a section (processor). Complexity also depends on whether a heterogenous system is necessary. Sizing and timing requirements are similar to those for single computer systems, but now communication must be maintained at proper speeds between processors. Most of these estimates depend on the designer's experience.

System tradeoff analysis is the final step. Performance requirements and size and timing analysis are completed so that various interconnect systems can be compared for relative cost to performance ratios. Direct or indirect message routing, centralized or decentralized control and dedicated or shared pathways can be chosen. The interconnect explanations in this chapter included a brief summary of the various advantages and disadvantages for each scheme. Based on needs for reliability, geography, expandability, performance, and outside service requirements one can compare costs for the application. Also at this point inter-

process communications are set up. Care must be taken to avoid deadlock possibilities as well as to design system error recovery techniques.

While today there exist a wide variety of interconnect schemes, many are relatively untried and in many cases design choices and rules are still unclear. Often tradeoffs are known but still a lack of pre-written systems is hampering development. Networks certainly will increase in popularity though, as small computer costs decrease and point to point communication needs increase. Task partitioning guidelines should develop as more and more distributed systems are designed, just as programming techniques developed in the 1960s and 1970s. Also the basic tradeoffs between concepts such as loops, hierarchical, tightly coupled, shared memory, etc., will be further established for specific applications. As more experience is gained, software and system cost also should decrease, thus further accelerating distributed system development. Home computers and interbuilding communication will probably be the big spur to bring of age interprocessor communication by the 1990s. When this occurs, it will become necessary to deal with the overload which will be created on existing telecommunications facilities. Perhaps technical advances in microwave, satellite, and fiber optics communication techniques will be able to handle the congestion.

CHAPTER FOUR

ANALYSIS FOR THE APPLICATION

Many of the important aspects of distributed systems were discussed in Chapter Three. Their application to YPSS, however, was not considered. As determined in Chapter Two, a local office system is feasible for the company. This system also should be capable of expanding to include remote data gathering. In this chapter, I will first discuss the reasons a service center is not capable of satisfying YPSS's future needs. Next, the theoretical distributed system design considerations, as they apply to the system, will be discussed. Finally, practical (and specific) systems for the application will be analyzed. I will conclude with a recommendation of a possible system.

In the past, many small businesses believed that computers were expensive and best suited for large, repetitive tasks. They believed that their company was too small and had too many unique, non-standard transactions. The cost and power of today's computers can provide special help to a small business. A computer can give flexible answers to "query" type questions, such as, "what if I did this?" Computers also can help reduce the routine workload on a small staff, freeing them for new tasks. Routine tasks can be standard to any business, even a unique one. These might include payroll, accounts receivable, and the general ledger.

Three criteria for computer usefulness are large volume, integrated data, and complexity.[35] Small businesses, especially, need integrated data with minimum cost. The computer can provide integrated data since the complexity, and thus cost, of the system is limited by the very smallness of the business. Large volume is not required since computer costs now are relatively low. It was estimated as long ago as 1967 that the threshold of size

for reasonable payback was 80 to 150 employees.[36] This certainly has fallen even further since then.

YPSS did choose to automate by using a remote processing center. Why cannot this, or on-line remote processing, solve YPSS's ultimate needs? There are several problems, in addition to continually escalating costs, that are common to service centers.[37] The lack of control over data entry errors was discussed in Chapter Two.

One problem is the lack of scope of possible applications. Generally, service centers create and sell only the easy to automate functions, like payroll. These are volume applications and have a high return on the work invested in them.

Another limitation is the need for greater access to the computer. Greater access not only makes modifications easier, but also makes possible the use of a computer for applications like text editing and the wrecker dispatch log, which otherwise could not be automated.

The security of information also is difficult for management to guarantee when unknown data entry personnel handle sensitive documents. Even though a client pays for trust in the service center's integrity, the problem is still out of his control.

A final service center limitation is the potential for failure to produce documents by a given date. Although YPSS seldom has pressing needs, even their regular reports could be delayed if a larger customer's account was critical to the service center.

After the decision is made that a company needs an accessible computer, there are two major options in addition to buying a computer.

One option would be to buy a terminal and/or printer, and lease time via long distance telephone connection to a mainframe system like General Electric's MARK IV or Montana State University's Honeywell. This option was considered briefly but discarded for several reasons.

Although initial investment cost is lower, continuing service costs are much higher than purchasing. Training, software, and conversion costs would be similar to buying a system. A leased line probably would be required since contact time would be eight to twelve hours a day (especially if a wrecker dispatch log was used). It is estimated that any time over five to six hours per day pays for a leased line.[38] Also, expansion to a second terminal would double the line and connect time charges (the primary costs) for this system. It would incur no extra charges with a captive system.

A major factor in the decision to pursue the locally owned system is the availability and level of software and support. Software is much more difficult to find for small business applications on mainframes. It also appears that it would take significantly more training to familiarize clerks with the procedures and commands on a mainframe than it would on a microcomputer.

An alternative to buying a computer is to lease it. There are two reasons to do this. Many computers can be acquired on a lease to buy option. Financially, this is similar to buying. It allows the buyer to discontinue and return the machine up to a certain point. This would be an option for any system YPSS might choose to acquire.

Another reason for leasing a computer is to allow the buyer to expand to an upward compatible model. YPSS would definitely not fall into this category. Most lease rates appear to be established to pay for the machine in two to four years. YPSS should not outgrow the processor in five years and probably would not outgrow it in even ten or fifteen years. It could be conceivable that they would upgrade in the ten to twenty year range when they purchase remote data gathering equipment. This would be likely only if significant communication and/or point-of-sale cash register technical advances were available. In any case, this system almost certainly would last well beyond the two to four years of the lease.

After discarding these options, the remaining decision was to investigate a system for purchase. The first step in this investigation was to define the significant theoretical considerations for the system. While a local office system would be the logical purchase based on the cost and benefit analysis, this evaluation includes the entire distributed network to assure that the system purchased can be expanded at a later date.

The first issue to consider in the design is what the network topology should be. While YPSS facilities are at fixed locations, a few decisions must still be made. For example, there are seventeen cost centers in eleven buildings in seven areas of the Park. There appears to be no need to network each cost center separately. Plans indicate that new facilities eventually will be built to consolidate each area's cost centers into one building. Therefore, it appears logical to network one computer for all cost centers in an area to the main office. The topology then would consist of six remote computers and the main office site.

The next issue is how communication will be accomplished between network nodes. It was decided that batch processing would be acceptable. All data could be collected at the station during the day and then delivered to the office as a summary. Two ways of accomplishing this were explored. First, data could be stored on a diskette and sent to the office by mail or messenger. This is cheap, but relatively slow.

The second way to collect the data is to transfer the information by a direct connection. Packet or message switching did not appear to be feasible because the cost of dedicated lines and "store and forward" equipment would be prohibitive. The telephone network appeared to be the logical choice for interconnection since it is established and relatively inexpensive. It uses circuit switching and, therefore, is susceptible to line failures. A line failure should not be a major problem for YPSS since data is stored locally, and thus it could wait and be updated when communication is restored. Channel capacity is relatively low with this type of interconnection, but with only summary information to transfer, this is also not a major problem. Establishing the call is slow with circuit switch-

ing, but only one or two connections each day are established so that this is not a problem. Messages would be sent in some type of "packet" to facilitate error checking and retransmission. However, the connection would be a circuit dedicated to the call with full-duplex communication. Using the telecommunications network has the advantage that any location can call any other location without routing or connectivity concerns because the telephone network handles them. There will be no line costs since all calls between facilities currently are on a WATS type network.

As just mentioned, routing is done automatically by the telephone switching network and, thus, is not a design problem.

With the communications determined to be done best on the telephone network, the next problem is the type of relationship the computers will have in their communication. They would be independent, loosely coupled computers. Commercially available microcomputers appear to be available in either master-master (point-to-point) connected or master-slave (hierarchical) connected configurations. Either configuration would work adequately for this application. The application relationship is basically master-slave since all data is transferred in one direction, "up" the hierarchy from the data gathering slave to the office master computer. For data flow this is a hierarchical system with the office machine in charge of collecting and processing from the remote slave computers. Any communication between either separate locations or the office and remote computers would be a master to master dialogue. The slave retains equal status in communication because it decides when and how to inform the office, and is not accessed or polled by the central computer.

Shared bus and shared memory interconnects would be completely impossible without some type of permanent line connection. Their cost would be prohibitive since they are not available commercially and would need to be developed specifically for YPSS.

The decision to allow batch processing, rather than real time processing, solves the problem of updating multiple copies of data in the system. Each location will have a current copy of its inventory updated during the day as each transaction occurs. The main office will receive a summary update at night, so its record for the stations will always be one day obsolete. This is considered acceptable since presently the record is usually two to four days obsolete. This is also a practical way of updating the data base since currently no distributed data base management system is being marketed.[39]

The protocol level details for communication software do not concern YPSS. To avoid any possible problems, all equipment should be purchased from the same manufacturer. Even though it is claimed that the multi-vendor approach can work,[40] it is possible that a communication problem could appear. This likely would be catastrophic for the entire project at YPSS. Therefore, one responsible vendor is a better choice than trying to mix several systems.

Practical system constraints also exist for both hardware and software. Price is one constraint for both. It was determined, by comparison with management cost ideas and the benefits analysis, that a microcomputer would be the largest processor that could be justified.

If the full office system is to be implemented smoothly, a fixed disk was found to be highly desirable to avoid shuffling many diskettes. Although this might not need to be part of the initial purchase, it would definitely need to be added before any significant expansion occurred. Therefore, it is included in the system specifications. The hard disk is only required for the office microcomputer system because there is little demand for memory at the remote locations. In fact, due to the altitude, hard disks could not be used at any of these locations. The hard disks are pressure sealed and the seal is not specified to withstand this atmospheric pressure differential.

Temperature presents no problem at the office because air conditioning and heating keep the office between 50 F and 80 F at all times. In most remote facilities cooling would not be a problem, but offices would need to be better heated in several locations. The temperature must be between 40 F and 80 F during working hours. No microcomputer should be run at a temperature below 32 F, thus eliminating winter operations from consideration for computerization.

A hardware maintenance contract would be necessary in the summer to keep the system operating. However, cost could be minimized by having either a spare processor or several spare boards at about $300 to $500 each.[41] These simply could be plugged in to the chassis and the old board returned to the vendor.

The first software constraint centers around its relatively high cost to produce. YPSS could not afford to write their own programs for the entire package. The second constraint is the inability to see and evaluate pre-written software and its documentation before purchase. It must be bought relatively blind to its internal quality. Questions like: What is automatically updated on each program? What can be categorized and how? and What flexibility is there to update? are difficult, if not impossible, to answer before a vendor is chosen.[42]

One answer is to hire a consultant. As one consultant stated, "There is a proliferation of bad stuff. . . . We try to be a guide in the jungle. There is good software, but finding it is difficult."[43] Some hardware companies are willing to be a guide to good software. Most companies, however, are more interested in selling their own software. There appear to be few qualified software consultants in the area.

After somewhat blindly picking the pre-written software, there are two choices for custom programming. The first is to buy a report writing package. This is inexpensive, but would require some in-house personnel training and time. The second is to have the programs custom written by a software firm. The programs are often of high quality, but are

usually expensive. Either choice would be acceptable for YPSS. A combination of the two might prove best. The volume, frequency, or speed of obtaining reports presents no constraints because YPSS has limited requirements for these things. Due to these limited needs, it is not necessary to do software efficiency modeling or analysis.[44]

Specific systems were evaluated to conform to these requirements. They come from the two major types of computer manufacturers: personal computer and mainframe manufacturers.[45]

An attempt was made to be somewhat exhaustive in a search for a system with sales and service available in Bozeman. This is the primary source of high technology equipment for Gardiner based businesses. Systems supported from Billings are considered acceptable, though. The systems outlined below appear to be a good cross section of the available market within a reasonable support distance of Gardiner.

Other systems were contacted, but for various reasons did not submit enough information to be evaluated. For example, the DEC sales representative is based in Boise and, therefore, was difficult to meet with. The Apple dealership experienced some key personnel turnover and so was unable to propose a system at this time. NCR decided not to mail a design option and meeting with them in Billings was not feasible. Any of these companies might be more closely evaluated before a purchase is made.

The first system investigated was from Radio Shack. It has the virtue of being readily available and inexpensive. All maintenance work, however, is sent out from the local store. A second problem is that it would have to be mixed with a second vendor's point-of-sale cash registers. It will not handle multiple input lines or an automatic computer answering service. This would inconvenience the remote call in, since someone would need to be available on a schedule to answer the telephone for the computer.

Probably the biggest problem with this system, however, was that Radio Shack's own software was not adequate for even the local office needs. Inventory, payroll, and general

ledger files were all too small. This fact, combined with the lack of a report generator, indicated that there would be extensive custom programming. The system, therefore, was dropped from consideration.

Table 6. Radio Shack System Costs (in dollars).

Hardware	
Local	
Microcomputer	5,000
Hard disk 5MB	2,500
Printer (letter quality)	2,000
Second terminal	700
Total (local office hardware)	12,200
Remote system additions	
6 remote computers @ 2300 ea.	13,800
Cash registers	na
Communications hardware	3,000
Total for entire system	29,000
Software	
Business systems*	1,000
Operating system	0
Word processor	200
Report generator	na
Total software	1,200

*Systems are not adequate for the application without significant modification. A large (perhaps up to $5,000) cost could be incurred for custom modification.

The second personal computer system was Altos, marketed by Western Telecomputing in Bozeman. It is a little more expensive than Radio Shack, but has better local hardware support and better software. The software has no local support, though. Once again, the undesirable situation of remote networking with a second vendor's point-of-sale cash registers must be considered. Software is adequate in function, but is very basic and does not integrate the various functions well. A significant amount of custom programming would likely be required for this system as well.

Training and installation of software is not included with the package and could be a moderate extra cost.

Table 7. Altos System Costs (in dollars).

Hardware	
Local system	
Microcomputer	5,000
Hard disk 5MB	0
Printer (quality dot matrix)	750
Second terminal	600
Total office hardware	6,350
Remote system additions	
6 remote computers @ 2800	16,800
Cash registers	na
Communications hardware	7,000
Total hardware cost	30,150
Software	
Business system*	2,500
Operating system	0
Word processor	500
Report generator	400
Communication software	1,000
Total software	4,400

*Relatively few business functions compared to other software packages.

The Burroughs system was designed primarily for the business environment. It is a tremendous improvement over any of the personal computer systems. This is true primarily of the built in communication abilities and the range of peripheral devices available to interconnect with the system. The software appeared to be excellent, although relatively expensive. Many desirable features were standard, such as the report generator and the word processor. Probably the only drawback to the software was that it was designed for a general business environment and not specifically for the combination of product distribution and retailing that YPSS does. It did, however, contain many features that software available from the personal computer manufacturers did not, such as budget and fixed asset packages.

I believe that the Burroughs local office system is equal to any "standard" office package on the market. The price is also reasonable. Their software package is exceptional when compared with the personal computer packages.

The major problem with Burroughs is its cost. It is two to three times as expensive as most of the personal computers. Due to the fact that its remote cluster station cannot operate as a stand alone unit, any remote intelligence at YPSS's locations would require a powerful (and expensive) computer. If they had offered a satellite, stand alone microcomputer with cash register compatibility for $2500 to $4000, they would have been an excellent choice.

Table 8. Burroughs System Costs (in dollars).

Hardware	
Local system	
Microcomputer	16,000
Hard disk 8.4MB	0
Printer (high quality dot matrix)	2,000
Second work station	4,100
Total for office system	22,100
Remote additions	
6 microcomputers @ 9000	54,000
Communications hardware	0
Cash registers	na
Total system	76,100
Software	
Business system	10,000
Operating system	0
Report generator	0
Word processor	0
Total software	10,000

The IBM system had by far the most options and possible configurations available for the application. This is a major advantage, especially if implementation is done slowly. Decisions can be changed at various steps in the process with minimal loss, if alternatives are available.

Two problem areas were discovered with IBM. One is that the remote microcomputers are relatively expensive. They are almost $6000 each, compared to several manufacturers in the $2500-$3000 range. However, they could be optional to the configuration since the cash registers can accomplish the data collection without extra intelligence. The other

problem is that their software, both system and application, has some drawbacks. The operating system was reported to be especially slow. However, it was decided that YPSS could live with this since report frequency and load requirements would be light. The power of the system was excellent, so no problems are expected with that.

The business application software problem was solved in such a way that it was actually turned into a major advantage. IBM only offered standard business software. It was good quality, but no better than other packages. Several companies exist that write excellent business software specifically for IBM machines. Two of these companies are local. One company, ABC Corporation from Helena, has a good general business package which offers better business function integration and a closer custom application fit than IBM's package.

The most promising firm is from Billings, Management Computer Systems, and offers a special petroleum marketing package. This package has been specifically designed and installed for several companies who, like YPSS, distribute and retail gasoline products at a number of locations. The fact that this package was designed very close to the application makes it an excellent choice. Desirable features should be maximized when compared to a standard package. Integration of functions is much more complete than with any of the other packages so that custom programming should be minimal. Management Computer Systems also markets IBM hardware at about ten percent below the IBM prices quoted here. This means that the entire system can reasonably be supplied by one independent vendor, serviced by IBM maintenance and support.

The software problems can be solved very well by this approach. The expensive remote hardware problem can be dealt with by considering the multiple hardware configuration options. The office system can be purchased now and the final remote configuration determined later. At the time that expansion to the remote installations occurs, the final configuration can be decided based on then current conditions. This decision will most

Table 9. IBM System Costs (in dollars).

Hardware cost			
Local system			
Microcomputer			5,870
Hard disk 15MB			7,000
Printer (high quality matrix)			3,000
Second terminal			800
Total office system			16,670
Remote additions			
6 microcomputers @ 5870 (opt.)			35,220
6 cash registers @ 4500			27,000
Communication equipment			6,000
Total system hardware			84,890
Software cost			
Business system (IBM)	6,000		
Business system (ABC Corp.)		5,000	
Business system (Management Computer Systems)			7,500
Operating system	0		
Word processor	0		
Report generator	1,250		
Total software costs	7,250	6,250	8,750

likely be either to pay the full cost of remote microcomputers and cash registers or to eliminate the microcomputers and just use the cash registers for remote data gathering. Data summaries would then be written by the cash register to a diskette which could be transferred physically or by telecommunication lines to the main office. This option would save $35,000 to $40,000, but would eliminate the ability to have local inventories available and to see reports generated on site each day.

It appears that for YPSS the large business computer manufacturers offer a better product than the personal computer stores. They are more expensive initially, but fit YPSS's requirements much more closely. The personal computer manufacturers' costs would escalate very quickly from the purchase price because all training, support and maintenance is extra. A considerably larger amount of expensive custom programming is needed and the construction of a remote network would require hiring a consultant.

In most cases hardware support is similar in price between the various manufacturers, but software is unsupported, or supported at a great distance. The cost of flying consultants from California for software modification would be very high. With Petroleum Marketing software on IBM equipment, support is available in Billings.

The mainframe manufacturers offer more hardware and software choices with a better "fit" for the application, but at a higher price. Software can be obtained that is much closer to YPSS's needs, reducing custom programming costs. Training and installation is generally free. At the time of remote networking, easily compatible equipment is available from the same vendor as the office system. IBM, with Petroleum Marketing software, appears to be the best choice since support is near by (Bozeman and Billings) and they offer the package with the most options, the software closest to the application, and they have a competitive price.

It was decided that it was profitable and reasonable for YPSS to consider computerization. Service centers appeared to have several drawbacks as long term solutions to this problem. Perhaps their biggest problems are their continuing expense and relative inflexibility. Leasing is a possible alternative, but is not significantly different from purchasing.

A number of theoretical and practical system limitations were discussed such as topology, routing, and data-base distribution. Although the choice is tightly constrained in most areas of the distributed system design by the lack of available systems, YPSS's requirements were flexible enough so that some alternative system designs were found to be acceptable. Several companies were considered for cost and ability to fit the application. At this time IBM appears to be the best choice since they support the maximum hardware flexibility and the best software at a reasonable price.

CHAPTER FIVE

CONCLUSION

The thesis has shown that YPSS has a number of problems which could be solved cost effectively by office automation. It would also offer them future accounting flexibility at a relatively low constant cost. Although unique in many ways, solutions exist for most of their special problems. In some cases, such as wrecker dispatch, the computer actually offers significant benefits over current practice because of the uniqueness. Other functions, such as the payroll, have standard solutions.

It was recommended that YPSS accomplish automation in two major steps. First a local office system, and later a remote data gathering system to interface with the office system. This has the advantage of delaying, until some future date, the high cost of the remote equipment. Yet it still offers the advantages and experience of an office system now.

The various types and design features of distributed systems were also discussed. Many of the potential solutions are not cost effective or are only in the research phase of development. Guidelines and tradeoffs between these systems will be further defined as more systems are implemented, but the fundamental components will remain. The questions of routing, coupling, and interconnect schemes will remain important. Software is already somewhat modularized by the various protocol level designs. It is likely that one standard set of protocols will develop, possibly based on ISO's seven protocol levels.

A process characterization is important for any design process. Process identification and decomposition were done for this system in Chapter Two. Process interaction was done for YPSS, but was too detailed and extensive for inclusion in this paper. Performance requirements and physical constraints were specified in Chapter Four. In general, they

were not tight enough to cause major design problems. An architecture selection was then made on the basis of an analysis of several locally available systems. A particular system was recommended to best fit YPSS's requirements. System tradeoff analysis was mentioned with regard to the selected system. Final decisions on this subject are likely to be made only after an office system is installed and remote data collection is determined to be feasible.

FOOTNOTES

FOOTNOTES

[1] Jules A. Cohen, *How to Computerize Your Small Business.* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1980), p. 20.

[2] Interview with Charles Gallagher, Yellowstone Park Service Stations, Gardiner, Montana, 23 August 1982.

[3] Donna Hussain and K. M. Hussain, *Information Processing Systems for Management.* (Homewood, IL: Richard Dirwin, Inc., 1981), pp. 223-228.

[4] Robert Townsend, *Up the Organization.* (New York: Alfred A. Knoff, Inc., 1970), p. 37.

[5] Hussain, *Information Processing Systems,* pp. 238-243.

[6] Barbara Liskov, "Primitives for Distributed Computing," *Proceedings of the Seventh Symposium on Operating Systems Principles.* (10-12 December 1979): 33.

[7] Stephen T. McClellan, "Distributed and Small Business Computing... A Fast Track," *Datamation* 26 (May 1979): 124.

[8] Cay Weitzman, *Distributed Micro/Minicomputer Systems.* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1980), p. 4.

[9] Roy D. Rosner, *Packet Switching.* (Belmont, LA: Lifetime Learning Publications, 1982), pp. 32-38.

[10] E. W. Davies, D. L. A. Barber, W. L. Prize, and C. M. Solomonides, *Computer Networks and Their Protocols.* (New York: John Wiley and Sons, Inc., 1979), pp. 51-53.

[11] Rosner, *Packet Switching,* pp. 38-39.

[12] Weitzman, *Distributed Systems,* pp. 23-30.

[13] Donald W. Davies and Dereck L. A. Barber, *Communication Networks for Computers.* (New York: John Wiley and Sons, Inc., 1973), p. 181.

[14] Weitzman, *Distributed System,* pp. 26-27.

[15] Ibid., pp. 31-35.

[16] Ibid., pp. 43-60.

[17] Ibid., pp. 47-50, 58-59.

[18] Rosner, *Packet Switching,* pp. 144-145.

[19] Weitzman, *Distributed Systems,* pp. 73-79.

[20] Ibid., pp. 68-69.

[21] E. E. Swartzlander, Jr. and B. K. Gilbert, "Supersystems: Technology and Architecture," *IEEE Transactions on Computers* C-31 (May 1982): 401.

[22] Paul E. Green, Ed., *Computer Network Architecture and Protocols.* (New York: Plenum Press, 1982), pp. 340-342.

[23] Rosner, *Packet Switching,* pp. 193-207.

[24] H. M. Deitel, *An Introduction to Operating Systems.* (Reading, MA: Addison-Wesley Publishing Company, 1983), p. 88.

[25] Weitzman, *Distributed Systems,* pp. 107-113.

[26] Ibid., p. 115.

[27] Rosner, *Packet Switching,* p. 108.

[28] Davies, *Networks and Protocols,* p. 277.

[29] Rosner, *Packet Switching,* pp. 127-129.

[30] Ibid., p. 108.

[31] Weitzman, *Distributed Systems,* p. 206.

[32] Swartzlander, "Supersystems," p. 400.

[33] Weitzman, *Distributed Systems,* p. 206.

[34] Ibid., pp. 214-220.

[35] Alexander M. Blanton and Joseph Traut, *Computers and Small Manufacturers.* (New York: Computer Research and Publication Associates, 1967), p. 36.

[36] Ibid., p. 38.

[37] Ibid., p. 137.

[38] Davies, *Networks and Protocols,* p. 99.

[39] Franklin F. Kero, ed., *Protocols and Techniques for Data Communication.* (New York: Prentice-Hall, Inc., 1981), p. 434.

[40] Alyn Gorman, Edmund J. Armon, and Kathleen Ternback, "Survey of Small Business Computers," *Datamation* 23 (September 1977): 190.

[41] Chris Hankins and Alex d'Agapeyeff, "Developing Software for Micro's," *Datamation* 23 (July 1977): 79.

[42] Barry W. Boehm, "Software and Its Impact: A Quantitative Assessment," *Datamation* 19 (May 1973): 48.

[43] Frederick Muller, "The Business-wise Microcomputer," *Infosystems* 27 (February 1980): 81.

[44] B. Kumar and Timothy A Gonzalves, "Modeling and Analysis of Distributed Software Systems," *Proceeding of the Seventh Symposium on Operating Systems Principles* (10-12 December 1979):2.

[45] Lawerence Tiedelman, "Microcomputers Mean Business," *Infosystems* 26 (June 1979): 104-107.

[46] Edward Yourdon and Larry Constantine, *Structured Design.* (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1979), pp. 43-47.

SELECTED BIBLIOGRAPHY

SELECTED BIBLIOGRAPHY

Business Application Sources

Albrecht, Leon K. *Organization and Management of Information Processing Systems.* New York: The MacMillan Company, 1973.

Barcomb, David. *Office Automation—A Survey of Tools and Technology.* Bedford, MA: Digital Equipment Corporation, 1981.

Biggs, Charles L.; Birks, Evan G.; and Atkins, William. *Managing the Systems Development Process.* Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1980.

Birkle, John R., and Yearsley, Ronald. *Computer Applications in Management.* New York: Halstad Press, 1976.

Blanton, Alexander M., and Traut, Joseph. *Computers and Small Manufacturers.* New York: Computer Research and Publications Association, 1967.

Boehm, Barry W. "Software and Its Impact: A Quantitative Assessment." *Datamation* 19 (May 1973): 48-59.

Boutell, Wayne S. *Computer Oriented Business Systems.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1968.

Brandon, Dick A. *Data Processing Cost Reduction and Control.* New York: Van Rostrand Reinhold Company, 1978.

Brightman, Richard W. *Information Systems for Modern Management.* New York: The MacMillan Company, 1971.

Clark, Frank J.; Gale, Ronald; and Gray, Robert. *Business Systems and Data Processing Procedures.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1972.

Cohen, Jules A. *How to Computerize Your Small Business.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1980.

Data Processing Monograph Series. *Management of Online Systems.* Detroit, MI: American Data Processing, Inc., 1968.

Donaldson, Harnish. *A Guide to Successful Management of Computer Projects.* New York: Halstad Press, 1978.

Edwards, Morris, "Modem and Multiplexer Update: Compact and Cheaper." *Infosystems* 28 (November 1981): 50-58.

Enger, Norman L. *Management Standards for Developing Information Systems.* New York: Amacom, 1976.

Feidelman, Lawerence. "Microcomputers Mean Business." *Infosystems* 26 (June 1979): 104-107.

Gorman, Alyn; Armon, Edmund J.; and Fernbach, Kathleen. "Survey of Small Business Computers." *Datamation* 23 (September 1977): 180-191.

Hawkins, Chris; and d'Agapeyeff, Alex. "Developing Software for Micros." *Datamation* 23 (July 1977): 76-80.

Heany, Donald F. *Development of Information Systems.* New York: The Ronald Press Company, 1968.

Hussain, Donna, and Hussain, K. M. *Information Processing Systems for Management.* Homewood, IL: Richard D. Irwin, Inc., 1981.

McCarthy, E. Jerome, and McCarthy, J. A. *Integrated Data Processing Systems.* New York: John Wiley and Sons, Inc., 1973.

McClellan, Stephen T. "Distributed and Small Business Computing . . . A Fast Track." *Datamation* 25 (May 1979): 124-134.

Muller, Frederick. "The Business-wise Microcomputer." *Infosystems* 27 (February 1980): 80-83.

Nadler, Gerald; Johnson, James; and Bailey, James. *Design Concepts for Information Systems.* Norcross, GA: American Institute of Industrial Engineers, Inc., 1972.

Orlieky, Joseph. *The Successful Computer System,* New York: McGraw-Hill Book Company, 1969.

Perry, William, and Fitzgerald, Jerry. "Designing for Auditability." *Datamation* 23 (August 1971): 46-50.

Rhodes, Wayne L. Jr. "The Disproportionate Cost of Data Entry." *Infosystems* 27 (October, 1980): 70.

Townsend, Robert, *Up the Organization.* New York: Alfred A. Knopf, 1970.

Wilkinson, Bryan. "An Application Audit." *Datamation* 23 (August 1971): 51-54.

Distributed Network Sources

Almes, Guy, and Lazowski, Edward D. "The Behavior of Ethernet-like Computer Communications Networks." *Proceedings of the Seventh Symposium of Operating Systems Principles.* New York: The Association for Computing Machinery, Inc., 1979, 66-81.

Bacon, Jean. "An Approach to Distributed Software Systems." *Operating Systems Review* 15 (October 1981): 62-74.

Berglund, Ralph. "Comparing Network Architectures." *Datamation* 24 (February 1978): 78-85.

Cypser, R. J. *Communications Architecture for Distributed Systems.* Reading, MA: Addison-Wesley Publishing Company, 1978.

Davies, Donald W., and Barber, Dereck L. A. *Communication Networks for Computers.* New York: John Wiley and Sons, Inc., 1973.

Davies, Donald W.; Barber, D. L. A.; Price. W. L; and Solomonides, C. M. *Computer Networks and Their Protocols.* New York: John Wiley and Sons, Inc., 1979.

Deitel, H. M. *An Introduction to Operating Systems.* Reading, MA: Addison-Wesley Publishing Company, 1983.

Fischer, M. J.; Griffoth, N. D.; and Lynch, N. A. "Global States of a Distributed System." *IEEE Transactions on Software Engineering* SE-8 (May 1982): 198-202.

Gifford, David K. "Weighted Voting for Replicated Data." *Proceedings of the Seventh Symposium on Operating Systems Principles.* New York: The Association for Computing Machinery, Inc., 1979, 150-162.

Green, Paul E. Jr., ed. *Computer Network Architectures and Protocols.* New York: Plenum Press, 1982.

Irani, K. B., and Khabbaz, N. G. "A Methodology for the Design of Communication Networks and the Distribution of Data in Distributed Supercomputer Systems." *IEEE Transactions on Computers* C-31 (May 1982): 419-434.

Kim, K. H. "Approaches to Mechanization of the Conversation Scheme Based on Monitors." *IEEE Transactions on Software Engineering* C-31 (May 1982): 189-197.

Kumar, R., and Gonzalves, Timothy A. "Modeling and Analysis of Distributed Software Systems." *Proceedings of the Seventh Symposium on Operating Systems Principles.* New York: The Association for Computing Machines, Inc., 1979, 2-8.

Kuo, Franklin F., ed. *Protocols and Techniques for Data Communications.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1981.

Leventhal, Lanee A. *Introduction to Microprocessors: Software, Hardware, Programming.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1978.

Liskov, Barabara. "Primitives for Distributed Computing." *Proceeding of the Seventh Symposium on Operating Systems Principles.* New York: The Association for Computing Machinery, Inc., 1979, 33-42.

Liskov, B. "On Linguistic Support for Distributed Programs." *IEEE Transactions on Software Engineering* C-31 (May 1982): 203-210.

Lorin, Harold. *Aspects of Distributed Computer Systems.* New York: John Wiley and Sons, Inc., 1980.

Loveland, Richard. "Puting DecNet Into Perspective." *Datamation* 25 (March 1979): 109-114.

Lunn, K., and Bennett, K. H. "An Algorithm for Resource Location in a Loosely Linked Distributed Computer System." *Operators Systems Review* 15 (April 1981): 16-20.

Martin, James. *Computer Data-Base Organization.* Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1977.

Minoura, T., and Wiedehold, G. "Resilient Extended True-Copy Token Scheme for Distributed Database Systems." *IEEE Transactions on Software Engineering* SE-8 (May 1982): 167-172.

Montgomery, Warren. "Polyvalues: A Tool for Implementing Atomic Updates to Distributed Data." *Proceedings of the Seventh Symposium of Operating Systems Principles.* New York: The Association for Computing Machinery, Inc., 1979, 143-149.

Needham, R. M. "Systems Aspects of the Cambridge Ring." *Proceedings of the Seventh Symposium of Operating Systems Principles.* New York: The Association for Computing Machinery, Inc., 1979, 82-85.

Paxton, William. "A Client Based Transaction System to Maintain Data Integrity." *Proceedings of the Seventh Symposium on Operating Systems Principles.* New York: The Association for Computing Machinery, Inc., 1979, 18-23.

Reed, David P. "Implementing Atomic Actions on Decentralized Data." *Proceedings of the Seventh Symposium on Operating Systems Principles.* New York: The Association for Computing Machinery, Inc., 1979, 163.

Ries, Dr. D. R., and Smith, G. C. "Nested Transactions in Distributed Systems." *IEEE Transactions on Software Engineering* SE-8 (May 1982): 167-172.

Rosner, Roy D. *Packet Switching.* Belmont, CA: Lifetime Learning Publications, 1982.

Saxton, Dr. W. A., and Edwards, Morns. " DEC Closes in on Networking Goal." *Infosystems* 27 (April 1980): 83-85.

Soloman, Marvin H., and Finkel, Ralph A. "The Roscoe Distributed Operating System." *Proceedings of the Seventh Symposium on Operating Systems Principles.* New York: The Association for Computing Machinery, Inc., 1979, 108-114.

Swartz, Mischa. *Computer Communication Network Design and Analysis.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1977.

Swinehart, Daniel; McDaniel, Gene; and Boggs, David. "WFS: A Simple Shared File System for a Distributed Environment." *Proceedings of the Seventh Symposium on Operating Systems Principles.* New York: The Association for Computing Machinery, Inc., 1979, 9-17.

Weitzman, Cay, *Distributed Micro/Minicomputer Systems.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1980.

Yourdon, Edward; and Constantine, Larry L. *Structured Design: Fundmentals of a Discipline of Computer Program and Systems Design*. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1979.

APPENDICES

APPENDIX A

DATA FLOW AND REPORT REQUIREMENTS

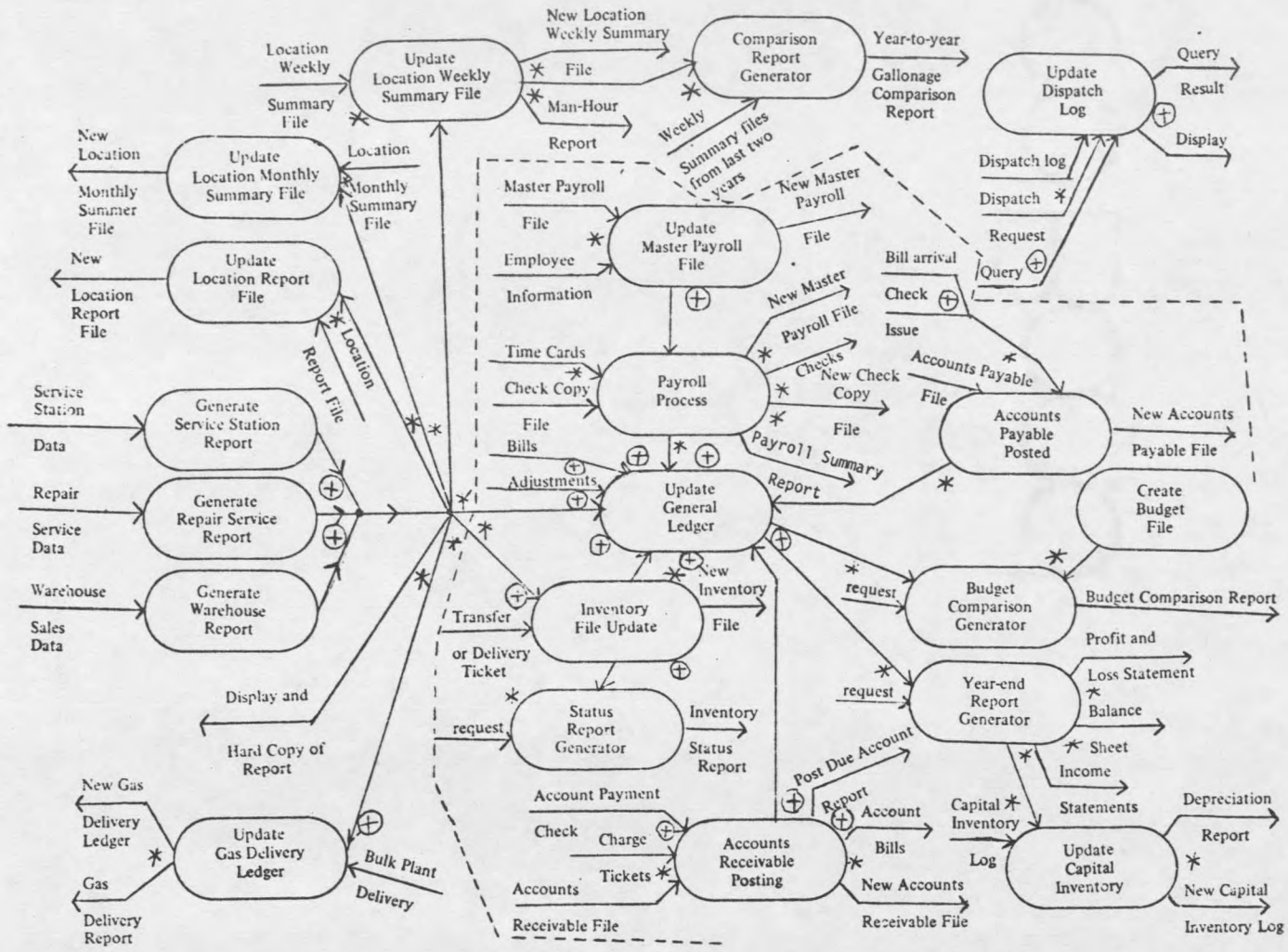


Figure 4. Data flow chart.

SUMMARY OF REPORT REQUIREMENTS

General Ledger Reports
- General ledger display
- Balance sheet
- Income statement
- Profit and loss statement
- Location profit and loss statements
- Capital inventory log
- Budget summary
- Budget-year-to-date comparison

Payroll Reports
- Employee checks
- Employee record file
- Employee year-to-date records
- Employee W-2 forms
- Payroll verification summary

Inventory Reports
- Complete inventory status and sales to date
- Partial inventory status
- Low inventory report

Receivable and Payable Reports
- Checks issued
- Accounts payable summary
- All accounts payable (current)
- All accounts receivable (current)
- Bills generated
- Overdue accounts report

In addition to the reports listed on the previous page, all files would need to be able to be displayed. These reports and displays are all standard. They would be included in the purchased software (inside dotted line on Figure 4).

The following reports are non-standard and would need to be generated by custom software.

Location Related Reports
- Daily station report
- Daily miscellaneous item sales sheet
- Daily propane sales summary
- Daily repair service report
- Weekly repair service summary
- Weekly service station summary
- Monthly service station summary
- Monthly repair service summary
- Location year to year comparison
- Gallons/man-hour

Others
- Gas ledger and summary
- Bulk plant report
- Warehouse sales ledger and summary
- Wrecker dispatch log
- Outside wrecker log and summary

APPENDIX B

SAMPLE FORMS

Service Station Daily Report Form (page 1)

Data is collected and entered at the station each day on this form. Calculations are done in the office on the following days (see next page). It summarizes sales and inventory at the station for that day.

SS 306—SM—1962

Date _____ **YELLOWSTONE PARK SERVICE STATIONS**

Station No. _____ *DAILY SERVICE STATION REPORT* Report No. _____

Block A — *Pump Meter Readings*

	FULL SERVICE					SELF-SERVICE				
	Pump	Regular	Unleaded	Diesel No. 1	Diesel No. 2	Pump	Regular	Unleaded	Diesel No. 1	Diesel No. 2
	1					1				
	2					2				
	3					3				
	4					4				
	5					5				
	6					6				
1	TOTAL									
2	LESS TOTAL Previous Report									
3	SALES This Report									
4	LESS Priming-Testing									
5	SALES-GAL									
6	PRICE per gal									
7	SALES $ (5x6)									

Block B — *Gasoline Invoices**

Invoice No.	Gals. Regular	Gals. Unleaded	Gals. Diesel No. 1	Gals. Diesel No. 2
Totals				

Block C — *Gasoline Inventory Record*

	Gallons Regular	Gallons Unleaded	Gals. Diesel No. 1	Gals. Diesel No. 2
1 Forward				
2 Receipts*				
3 Total Available				
4 Less Sales				
5 Book Inventory				
6 Actual Inventory				
7 Over-Short				

Block D — *INVENTORY RECORD*

Item	Opening Inventory	*Add Receipts	Total Available	Less Ending Inventory	Total Sold Used & Trans.	*Less Station Use & Transfers	Sales - Units	Price Per Gallon	Amount
1 Conoco "New Generation" 10-40 - 5 qt (Gallons)								$	$
2 Conoco "New Generation" 10-40 - Qt (Gallons)									
3 Conoco "Super" 10-30 Gallons									
4 Conoco "Super" / Fleet 20 & 30 (Gallons)									
5 Conoco Par (Gallons)									
6 Pennzoil 10-40 (Gallons)									
7 Pennzoil 30 (Gallons)									
8 Valvoline / Castrol 20 50 (Gallons)									
9									
10								TOTAL OIL SALES	
11 Transmission Fluid (pints)									
12 Anti-Freeze Coolant (Gallons)									
13 Tires (new)									
14 Car Batteries									
15 Air Filters									
16 Oil Filters									
17 Tire Chains									
18 YPSS Mats									
19									
20									

Basic Report Information Compiled at Station by _____ Total Regular Hours _____ Total Overtime Hours _____

*Attach copies of Warehouse Delivery Ticket (SS-205), Material Transfer (SS-206), Valley Motor Invoices or Gasoline Invoice to substantiate.



Service Station Daily Report Form (page 2)

This is the calculation summary of the report form on the preceding page.

Report No. _____

Block E *RECAP OF SALES*

FUEL SALES (from Block A—Line 7)			
1.	Gal. Regular, Full Service @	$	
2.	Gal. Unleaded, Full Service @		
3.	Gal. Regular, Self-Service @		
4.	Gal. Unleaded, Self-Service @		
5.	Gals. **TOTAL GAS SALES****		
6.	Gals.—Diesel No. 1** @		
7.	Gals.—Diesel No. 2** @		
8.	Gals. Propane		
OIL SALES (from Block D—Line 10)			
9.	Gals. TOTAL OIL SALES		
10.	Tires and Tubes		
11.	Fed. Excise Tax on Line 10. Items**		
12.	Labor		
13.	Truck Revenue		
14.	Other Merchandise		
15.	**TOTAL SALES (add 5 through 14)**		
16.	Wyoming Sales Tax (No tax on these items**)		
17.	**TOTAL SALES TO ACCOUNT FOR (15 plus 16)**		
18.	Charge Sales	$	
19.	(a) Master Card/Visa		
	(b) Conoco/Exchange Credit Card Inv.		
	(c) Foreign Credit Card Inv.		
	(d)		
20.	TOTAL CREDIT ITEMS (18 plus 19)		
21.	TOTAL CASH RECEIPTS (17 minus 20)		
22.	CASH DEPOSITED		
23.	Cash Over/(Short) (21 minus 22)		

Cash Register Tape $_____

_____

This Report Completed by:

SS302-10M-1981

YELLOWSTONE PARK SERVICE STATIONS *Station* *Date* *Report No.* **SALES RECORD**

TIRES				LABOR			TRUCK REVENUE		OTHER MERCHANDISE				
Quantity	SIZE	Fed. Excise	AMOUNT	SS-301 No	TYPE	AMOUNT	SS-301 No	AMOUNT	Quantity	DESCRIPTION	AMOUNT	Quantity	DESCRIPTION	AMOUNT
		$	$			$		$			$			$

TUBES

Quantity	SIZE	Fed. Excise	AMOUNT
		$	$
TOTAL TIRES & TUBES			

TOTAL | TOTAL | TOTAL OTHER MERCHANDISE

IMPORTANT! Do NOT charge Sales Tax on Excise Tax. Tax must be charged on all other items.

Sales Record or "Yellow Sheet"

This is a list of all non-petroleum products sold at a station during a day. It includes services and materials. It is sent to Gardiner, Montana each day with the location report.

Daily Repair Service Report Form

This is comparable to the station report form, but repair services only sell parts and labor on a service order (ticket). Again, forms are done daily at the station.

55605-3M-1981

YELLOWSTONE PARK SERVICE STATIONS
DAILY REPAIR SERVICE REPORT

REPORT NUMBER _____
PAGE _____

LOCATION _____ DATE _____ TO _____ (inclusive)

Ticket Number	Memo	Parts	Labor	Towing	Mileage	Winch Time	Accessories	Miscel.	Sales Tax	Total
TOTALS										

Weekly Repair Service Report Form

This is used simply to enter each day's report, and then summarize the week's activity.

55-606-1M-1961

YELLOWSTONE PARK SERVICE STATIONS

WEEKLY REPAIR SERVICE REPORT SUMMARY

REPORT NUMBER _____

LOCATION _____ DATE _____ TO _____ (inclusive)

Page Number	Memo	Parts	Labor	Towing	Mileage	Winch Time	Accessories	Miscel.	Sales Tax	Total Sales

1. TOTAL SALES TO BE ACCOUNTED FOR _____
2. LESS: TOTAL CASH DEPOSITS _____
3. TOTAL CHARGE ITEMS _____
4. TOTAL CREDIT CARD INVOICES _____
5. OTHER (EXPLAIN OVER) _____
6. TOTAL LINES 2, 3, 4 & 5 _____
7. CASH (OVER) or SHORT (Line 1 minus Line 6) _____

FOR OFFICE USE ONLY

Net Sales _____

Sales Tax _____

Total Sales _____

LOCATION MANAGER _____

AUDITED AND COMPLETED BY _____

Wrecker Dispatch Log

This form is completed for any request for wrecker in Yellowstone National Park.

55401—6A6—1982

YELLOWSTONE PARK SERVICE STATIONS

Towing Service Dispatcher's Record of Request for Service

Date:..... 19..... Time Service Request Received:.....

Origin of Service Request: .....

.....

Location Where Service is Required:.....

.....

Nature of the Problem: .....

.....

.....

DESCRIPTION OF THE VEHICLE:

..... (YEAR) ..... GROSS VEHICLE WEIGHT: (If other than passenger vehicle) ..... (MAKE)

..... (BODY STYLE) ..... (COLOR)

..... (STATE LICENSE NO.) ..... (OWNER/OPERATOR'S NAME)

10-20 of Our Equipment When Dispatched:.....

.....

What Equipment Was Dispatched? ..... (VEHICLE NO.) ..... (WRECKER OR SERVICE TRUCK)

Time of Departure: .....

Driver of Equipment .....

ETA at Scene .....

Time of 10-23 at Vehicle: .....

Time of 10-24 .....

Com. Center Contact :.....

On-Scene Ranger: .....

..... (DISPATCHER)

Dispatch No. .....

Work Order No. .....

IF VEHICLE IS TO BE TOWED, WHERE DOES CUSTOMER WISH TO GO (In-Park or Entrance)?

.....

MONTANA STATE UNIVERSITY LIBRARIES
3 1762 10166043 7

N378
M355
cop.2

Marshall, D. C.
Distributed Computing - an application for a small business

DATE | ISSUED TO
JUL 1 [redacted]

MAIN LIB
N378
M355
Cop. 2