TO BE PUBLISHED IN THE "INTERNATIONAL JOURNAL OF ROBOTICS AND AUTONOMOUS SYSTEMS" AT A FUTURE DATE.

VEHICLE-COMMAND CENTER COMMUNICATIONS in a ROBOTIC VEHICLE SYSTEM

October 1990

Harry A. Scott

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Center For Manufacturing Engineering
Robot Systems Division
Gaithersburg, Maryland 20899

TABLE OF CONTENTS

INTRODUCTION

Background

The U.S. Army Laboratory Command, through its Human Engineering Laboratory (HEL), has undertaken a program designed to demonstrate cooperative, real-time control of multiple land vehicles. The program, called TEAM [1], employs vehicles that will be equipped with remote driving controls, a mission package which includes automatic target detection and tracking, communications equipment and a high-level control system for coordinating all subsystems. A separate remote command center will contain operator driving controls and displays, communications equipment and a high-level control system for coordinating all of this equipment.

NIST's role in this program is to apply its hierarchically-structured real-time sensor-based control system architecture, originally developed for control of robots [2-8], to the design of the control systems in the vehicles and the remote command center. HEL chose this control system approach because of its potential to become a standard control architecture for all Army robotics. This decision parallels the one made by NASA to adopt this architecture for the Flight Telerobotic Servicer control system.

This paper begins with a brief review of aspects of the NIST Real-Time Control System (RCS) architecture which are relevant to design considerations associated with a vehicle-command center communications link. This paper then turns to its primary focus, the communications link itself, examining issues of data content, performance, communications architecture, and standards.

The Control Architecture

An intelligent machine utilizes computers to control a collection of mechanical devices which allow it to physically interact with the environment. The control system uses sensory information to guide the machine in the execution of complex tasks. In the RCS architecture, complex tasks are viewed hierarchically with motor skill functions performed at the lowest levels and coordinated interaction between machines performed at the highest levels. Such a system architecture provides the organization and structure required to effectively integrate the machine's components.

Figure 1 illustrates the RCS system architecture. An important attribute of the architecture, which is evident throughout, is the drive to standardize the controller components. For example, each of the horizontal levels in an RCS consists of Task Decomposition (TD), World Modeling (WM) and Sensor Processing (SP) classes of functions.

This work was funded by the U.S. Army Human Engineering Laboratory. This paper was prepared by U.S. Government employees and is not subject to copyright. Commercial equipment is identified in this paper in order to adequately describe the test configurations. In no case does such identification imply recommendation by the National Institute of Standards and Technology, nor does it imply that this equipment was necessarily the best for the purpose.

Each of the TD modules plan and execute the decomposition of high level goals into low level actions. Goals are decomposed both temporally (into a sequence of actions) and spatially (into concurrent actions performed by multiple subsystems). The world model is the system's best estimate and evaluation of the history, current state and possible future states of the world. It includes both the WM modules and a knowledge base stored in global memory where state variables, maps, lists of objects and events, and attributes of objects and events are maintained. The WM modules provide functions to model and evaluate the state of the world. The SP modules detect events, recognize patterns and filter and integrate sensory information over space and time.

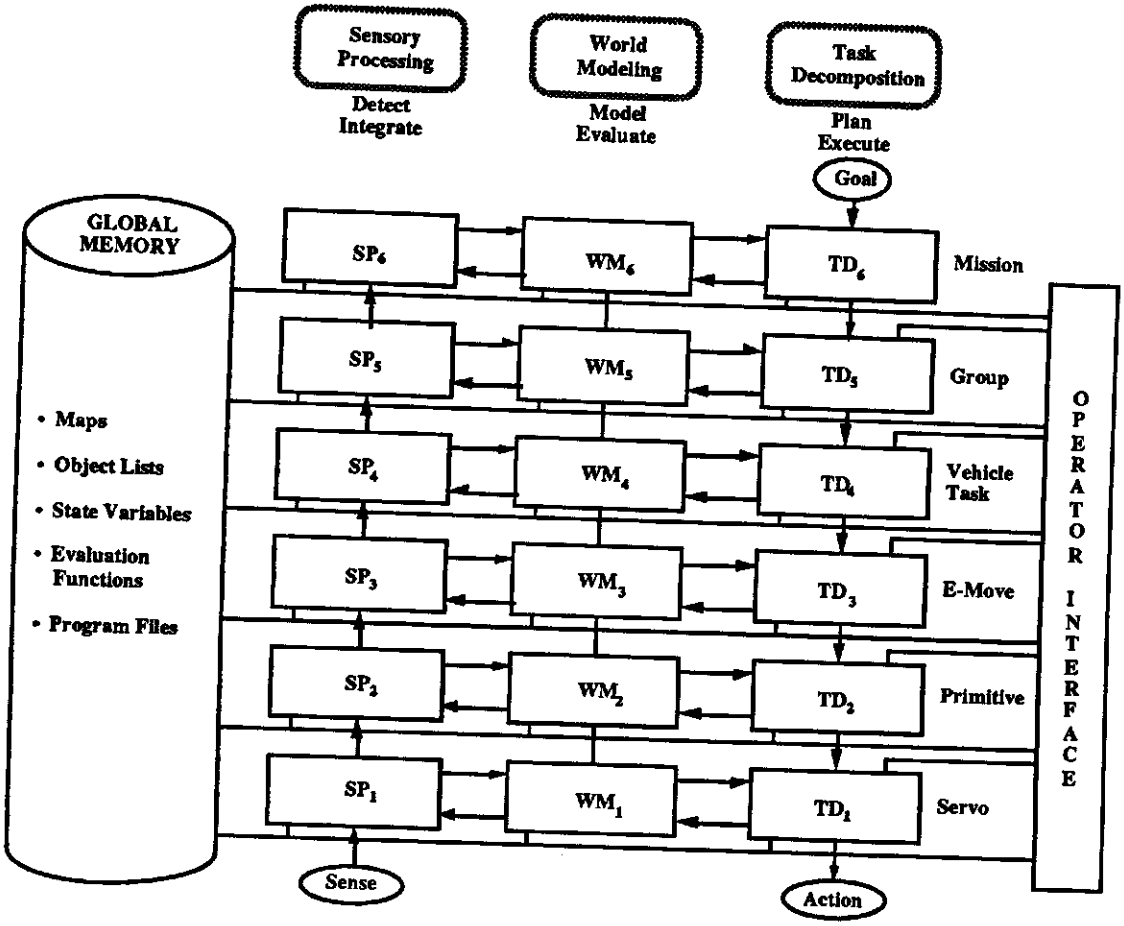

Figure 1. The RCS Control Architecture.

Another example of RCS standardization is seen in the structuring of the control loop at each level. The organization of each loop is always the same. Each loop functions as a state machine where first all inputs to the level are examined. These inputs (the current commanded goal, the current state, and the status feedback of sensors and the next lower level) are analyzed and the appropriate transformation function is applied. Finally, all outputs are transmitted. Each of the modules (TD, WM, and SP) contribute to this Input-Process-Output (IPO) control cycle for each horizontal level in the system. While the organization of each level is the same throughout the hierarchy, the execution speeds required differ from top to bottom. The lowest levels require command updates and responses to sensor information in periods of several milliseconds in order to maintain smooth and stable control of equipment. The higher levels deal with larger, longer term goals and with higher level representations of the world and, therefore, operate over minutes, hours, and even days.

Another significant attribute of the system architecture is that it enforces a clear partitioning of the classes of control functions. The partitioning is handled through interface definition between system components and formal specification of internal functionality. The benefits of such modularism include improved understandability, reusability of code, eased system integration, the support of upgrades and modifications, and efficient distribution of programming efforts.

An important consideration in an RCS design is the integration of the role of humans in control. While the technology of machine intelligence is in its infancy, human dexterity, knowledge and expertise can greatly increase the usefulness of most machines. The smooth integration of human control, and the transparency with which control is exchanged between human and computer is critical to the success of systems using such shared control. The interface for an operator is shown in figure 1 along the right hand side.

The final component of the RCS architecture is Global Memory (left side of figure 1). Global memory serves as a blackboard where information is posted and readily available to any process in the system. The data structures mentioned earlier (maps, plans, object models) reside in this global memory. This form of memory is essential whenever information is shared between processes.

Implications on Communications

The clean modularity of the figure 1 architecture not only illustrates the functional partitioning of the control system, but also makes clear the major paths of communication needed in such a control system. Further, it suggests something about communications performance since the modules at the lowest levels typically exhibit response times of milliseconds, while the highest levels work in the range of several minutes. These response time differences are reflected in the level of communications performance needed to support the different logical communications paths. In a teleoperation environment, this demands careful consideration of how a single physical link can be shared effectively to provide the required support.

Toward Standard Mechanisms

To date, many of the communications needs of real-time systems have been met through the use of special purpose, custom software and hardware. With the increase in performance of off-the-shelf processor boards and related peripheral boards, it is reasonable to re-examine the potential for use of more standard, and more portable techniques. A review of many available interprocess communications schemes is presented in Reference [9]. Performance measurements for various aspects of communications as it relates to the RCS model are ongoing [10]. The hope of course is that not only will a standard control architecture and standard information interfaces exist, but that standard communications mechanisms may be used in its implementation. The modularity of the RCS architecture makes this a possibility.

The Vehicle-Command Center Link

In the TEAM project example of a teleoperated system, the modules of figure 1 are physically distributed. The Mission and Group control levels and the Operator Interface devices and controllers reside in the command center and the other levels reside on the vehicles. In addition, data is shared by the vehicle and command center. Figure 2 indicates the physical distribution of the control modules and the data. The vehicle modules are shown shaded. Notice that there are a number of communications paths interconnecting the physically separate systems.

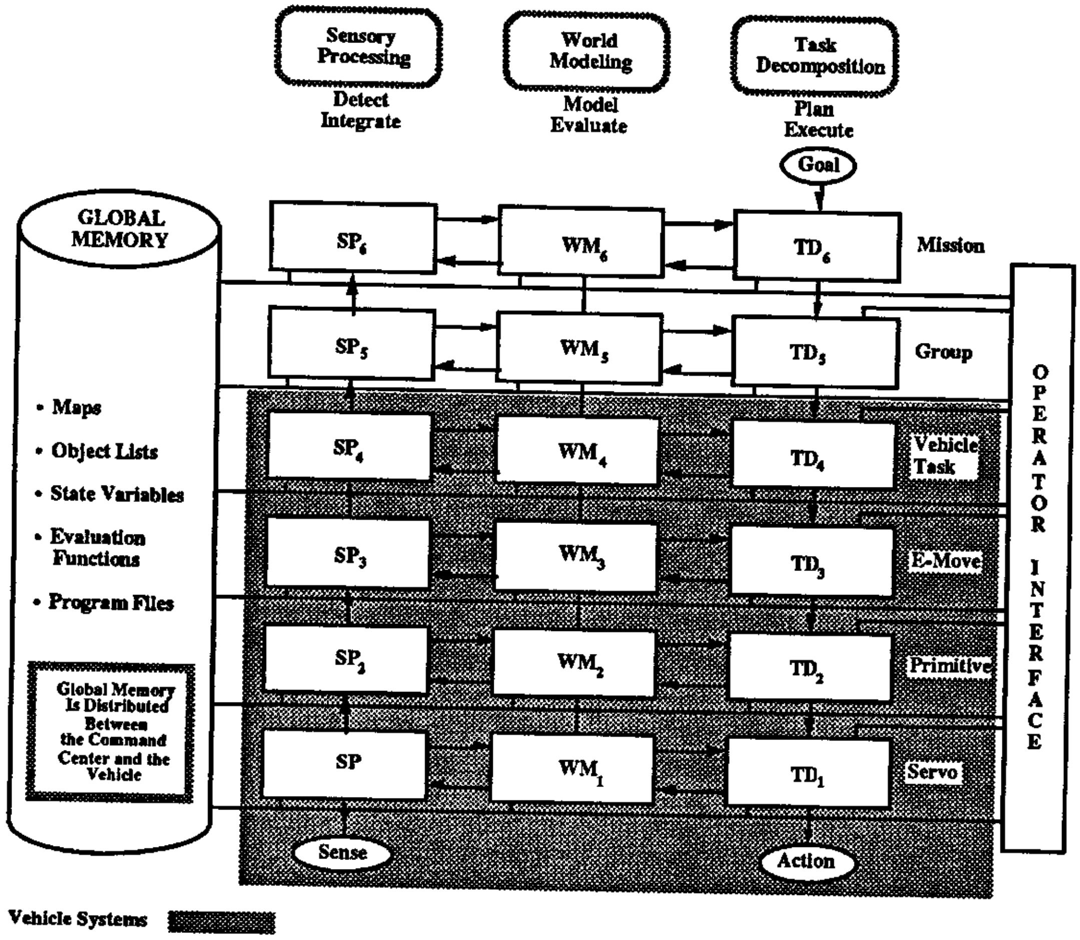

Figure 2. The physical distribution of control modules.

The ultimate implementation goal for the connections between the command center and a vehicle is a single, physical communications link. This physical link must support the several logical paths or links shown in figure 2. Examination of vehicle-command center communications must address the characteristics of the physical connection between the command center and vehicle, the characteristics of communications mechanisms used to move information between processes and the physical connection, the characteristics of the information being exchanged, and how the complete communications path might may be configured and might perform. In addition, it is a goal of the TEAM project that the single communications link operate at a low data rate to facilitate the use of common military radios. This requires special attention to communication timing and performance criteria for the link. These issues are the focus of the rest of this paper.

INFORMATION CONTENT

There are several broad classes of information that are exchanged between the vehicle and command center. They are video information, the data required to support the teleoperated and autonomous functions, and the information needed to support system and software development.

Video

Teleoperation of a vehicle requires that video information from the vehicle be supplied to the operator at the remote command center. This is required for remote driving and for the support of various mission functions (e.g., reconnaissance). The bandwidth required for communicating full-motion video information far exceeds that anticipated for the communication of control information. Common television requires 4 to 6 megahertz of radio frequency bandwidth when transmitted in analog form [11]. Transmission of video, in analog form, also requires different radio equipment than that used for transmission of digital control data. Video information can also be transmitted in digital form, but this requires still greater bandwidth than that for analog transmission, as the following example shows. Maintaining a smooth image of full-motion video requires an image rate of 30 frames per second. If each image requires 8 bits per pixel and the images are 512 pixels x 512 pixels, the required data rate is 30 x 512 x 512 x 8 or about 60 megabits per second [12]. The required bandwidth increases even further if several cameras are used to provide the operator with multiple, concurrent views. Wide-bandwidth radio communication presents several problems. It is highly directional, requires an unobstructed, line-of-sight transmission path, and requires expensive radio equipment.

Narrow-bandwidth radio communication is less hindered by these problems. However, a narrow-bandwidth link designed for the transmission of digital data at 100 kilobits per second, for example, clearly cannot support the full motion video described above. For this reason, video compression techniques with compression ratios in the range of 500:1 to 1000:1 are required. These techniques, which are still under development, will likely include a combination of compression and enhancement of the video data sent from the vehicle at a rate of only a few images per second rather than at 30 images per second.

Potentially, when such compression is achieved, the video information may be handled by the vehicle-command center communication system in the same way as any other control data. Initially though, in the absence of such advanced compression techniques, the video information will be handled separately. In the near term for the TEAM project, this means that the video is handled by separate radio hardware dedicated to the video information and that it is transmitted in analog form. In the long term though, TEAM goals do require driving the vehicle from video images transmitted at low data rates.

The discussion in the rest of this paper focuses on the control data and other information that is already suited to narrow-bandwidth communication. The expression "vehicle-command center link," as used in the remainder of this paper, is intended to mean this low-data-rate link. The design of a vehicle-command center link should not preclude support for video data when a sufficient low-data-rate mechanism is developed. In addition, functions for supporting the use of video, such as camera positioning and focusing, are functions that are considered appropriate for the link discussed here.

Teleoperation and Autonomy

Control of the robotic vehicle is maintained via command and status information flowing across the communications link. While there exist an unlimited number of ground vehicle configurations, a great deal of similarity exists in the types of functions that need to be controlled. The U.S. Army's Tank Automotive Command (TACOM) performed a study of four military vehicles to develop a basis for communications standardization in vehicle teleoperation [13]. The study identified all of the controls currently available for manned operation and suggested some additional controls needed for teleoperation of the vehicles at the same level of capability attainable by manned operation. Over one hundred vehicle functions and over one hundred vehicle status conditions were identified, covering the spectrum from steering, braking and throttle control, to heater and headlight activation.

In addition to providing remote operation of the "manned operation" type of functions, analysis of the functions required for the TEAM program provides examples of higher level functions that reflect greater vehicle autonomy in completing some tasks. These include *Retro-traverse* (travel a path previously traversed), *Acquire Targets, Cooperative Search* (search an area in a cooperative fashion with another vehicle), and *Track Targets*. These functions, which may make use of various mission package subsystems, must also be supported by the link.

Development

It is highly desirable to support a development connection between a vehicle and remote command center that can remain available for use at any time, including during vehicle operation. This supports downloading, process monitoring and debugging. Robotic vehicles contain complex computing environments that employ multiple backplanes, with each backplane often supporting several high performance processors. Development environments are available that are able to provide communications to all processors in such a target system over a network. Making this sort of connection available over the vehicle-command center link would be useful.

PERFORMANCE CONSIDERATIONS

Of all the non-video information described that is to pass between the vehicle and command center, the vast majority is not time-critical, and for the most part deals with such functions as fuel tank selection, lighting and parking brake activation, or, represent high level commands to which the vehicle responds with a somewhat lengthy (in a real-time sense) autonomous behavior. This type of information forms a large class of commands and status, that are transmitted relatively infrequently and not with real-time urgency.

Although much smaller, there is a class of commands and status that is transmitted frequently and whose communications timing is more critical. This class includes information used in teleoperation driving (steer, brake, and throttle data) and in mission subsystem packages that require similar low level teleoperation control, such as a multi-axis positioning platform for a reconnaissance sensor suite. The frequency of this type of communication is dependent upon the particular system application. For example, the current RCS systems maintain smooth motion of robotic systems by updating position information on a 20 to 30 millisecond (ms) clock period. The vehicle actuator control interface on the TEAM vehicle is limited to a maximum update rate of one new command every 100 ms. This period will likely shorten with actuator controller improvements. But even at 100 ms, this activity represents the most demanding communication occurring on the vehicle-command center link, and cannot be compromised by transmission of other types of information. This requirement for updates at a specific rate calls for a deterministic communications mechanism.

It is also important to include in this class an emergency stop mechanism and a means of communicating a status reflecting a vehicle emergency. The emergency information is not communicated frequently but is, by definition, time-critical.

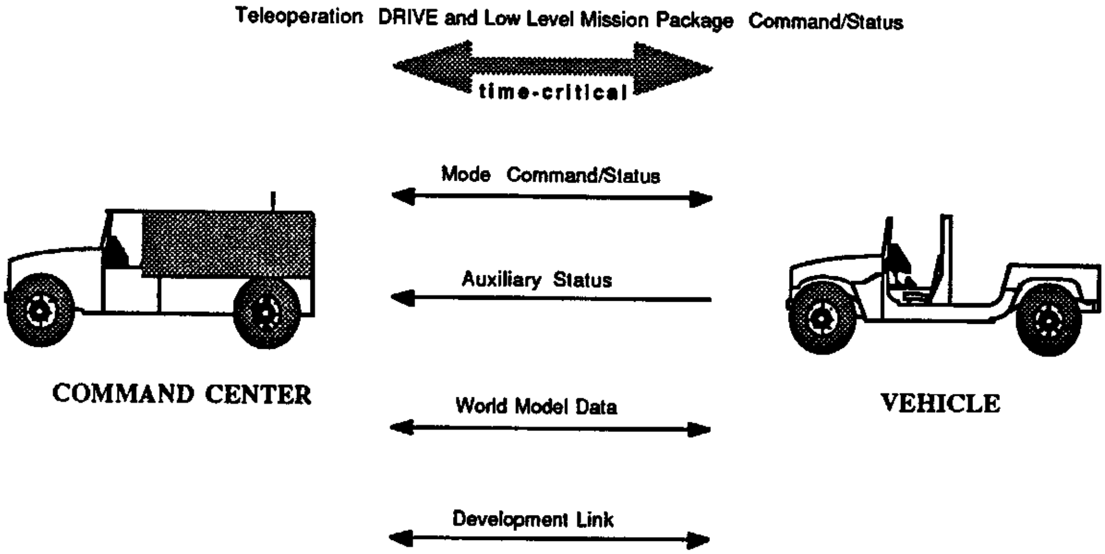

Figure 3. Information types for communication between the command center and vehicle.

Figure 3 further categorizes the types of information exchanged between the vehicle and command center. The Drive and Low Level Mission Package command/status information represent the only class of time-critical information. It includes steer, brake, throttle and certain mission package and emergency condition information.

Mode Command/Status information consists of previously identified high level function commands (*Retro-traverse*, etc.) and the wide variety of vehicle function selections such as lights on/off.

Auxiliary Status information is status reported by the vehicle spontaneously and not necessarily related to the execution of a particular command. Exceeding threshold values for on-board consumables or environmental conditions causes this type of status report.

As shown in figure 1, a world model is a key element of the RCS architecture. Support for the exchange of world model information between the vehicle and command center is required. Vehicle location data, for instance, originates from the vehicle navigation system but must be placed in the command center world model to support such functions as mapping the vehicle location on an operator's display map.

Finally, the inclusion of the previously described development connection is shown.

MANY LOGICAL LINKS, ONE PHYSICAL LINK

The described vehicle-command center link must carry all non-video information to support operation of the vehicle. This includes all types of time-critical and time-noncritical information. Therefore, two types of control must be exerted over the use of this link. Delivery of information to the appropriate destinations must be insured and delivery of time-critical information must be performed without interference by time-noncritical information transmissions. These types of control can be achieved by appropriately multiplexing the various information types on the single communications channel.

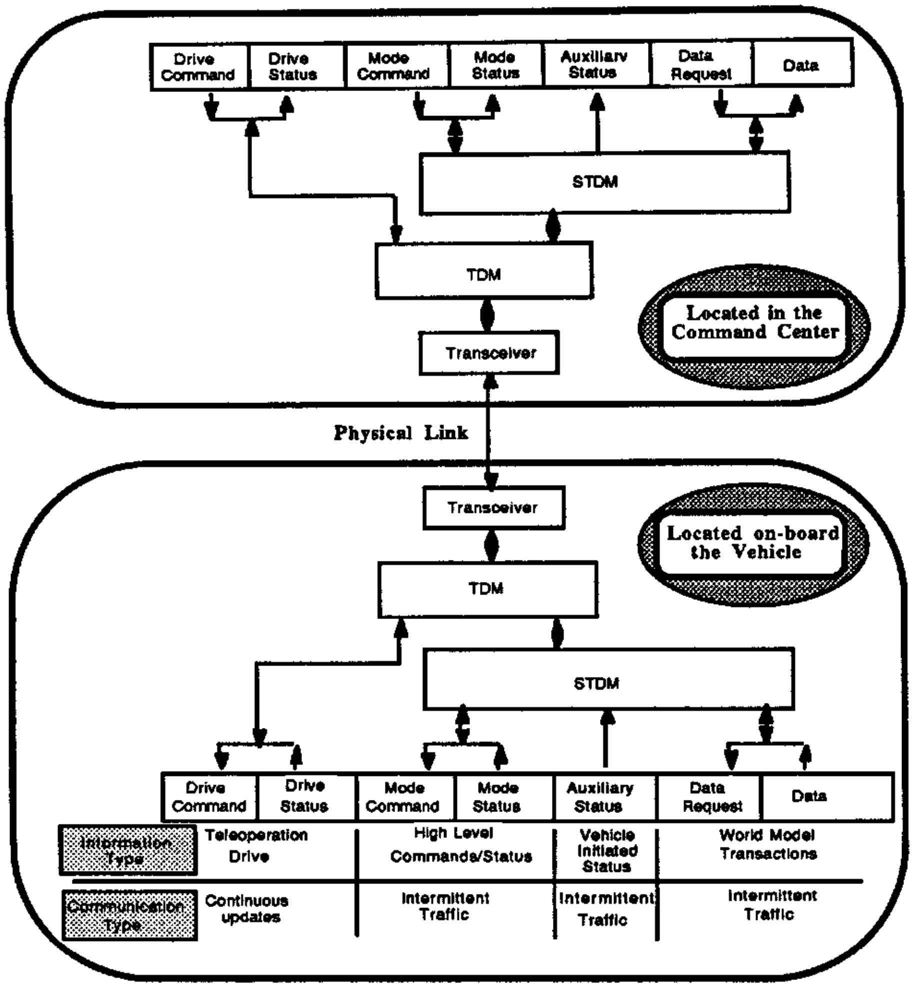

Figure 4. Multiplexing onto a single physical link.

Multiplexing

Figure 4 illustrates a means to multiplex all of the information through a single communications channel. While this could represent an actual hardware implementation, the intention here is just to illustrate the multiplexing concept. Possible hardware configurations are described later.

Application processes transfer information through the communications system via the interfaces shown at the top and bottom of the figure. Each application process is unaware of the use of the communication channel by other processes. Time division multiplexing (TDM) is used to allocate a fixed portion of the bandwidth to the time-critical driving function. The remaining bandwidth is allocated by a more efficient, but non-deterministic statistical time division multiplexing (STDM) technique.

Several advantages are associated with this approach. The time-critical information is not subject to interference from the other traffic. In addition, time-noncritical transactions can use as much of the remaining bandwidth as is available at the time of communication, subject only to other time-noncritical traffic, since the portion of the bandwidth used by each time-noncritical line is not fixed in advance. This works well for intermittent transmission activity. Particularly attractive is the fact that other time-noncritical lines can be added with no changes required in the software supporting the other lines. Also, with only intermittent traffic occurring on these lines, little if any performance drop will be seen. Other time-critical lines could also be added without affecting application code, however, the link performance penalty would be more pronounced since an additional fixed portion of the already limited bandwidth would no longer be available for use by the time-noncritical lines.

The following example shows how the available bandwidth might be used. Figure 5 suggests a message format for a command and status message that would convey the types of time-critical information described. The overall message length is the important consideration here. The example message format includes only that which is deemed time-critical, plus a representative amount of communications protocol overhead. The amount of protocol overhead used in this example is that required by the Synchronous Data Link Control (SDLC) protocol [14]. This is a common link level protocol. The message contents are believed to be representative of information needed for teleoperated driving and are consistent with vehicle actuator controller interfaces that have been used previously. The exact content or format is not important for this examination.

SDLC message overhead:

#bits:	8	8	8	n	16	8
	flag	addr	control	data	crc	flag
				Command or Status Message		

Command Message:

#bits:	8	16	16	16	16
	C#	throttle	brake	steer	emergency

Where:

C#......................................	Is the incremental command number of this command. Indicates a new command.
throttle, brake, steer...........	Are 2 byte specifiers of control parameters. (2 bytes each is consistent with known actuator controller interfaces.)
emergency.............................	Provides a way of including other unusual, but time critical directions. For example, STOP or SHUTDOWN immediately.

Status Message:

#bits:	8	8	16	16	16	16	16	16	32	192
	status#	echoC#	emergency	throttle	brake	steer	engine RPM	vehicle speed	Nav time stamp	Navigation Data 6x32
				Drive Status					Mission Navigation Support	

Where:

status#.............	Incremental number of status message assigned by responding control system.
echoC#..............	Command number of command to which this status refers.
emergency.........	Emergency field for time critical reports of status other than drive information, e.g., critical error conditions, vehicle system failure, etc., that require immediate response.
Drive Status......	Current report of vehicles actuator positions.
Nav time stamp..	Indicates the actual point in time when the vehicle was in the position and orientation given by the Navigation Data.
Navigation Data..	Position and orientation of the vehicle from on-board inertial navigation system (INS), used for various mission functions.

Figure 5. Teleoperation drive message example.

A full duplex communications link is assumed over which the command and status message must be exchanged once per update period. For a communications link of a given bandwidth, the bandwidth remaining for the exchange of time-noncritical information is given as:

$$BW(remaining) = BW(total) - BW(critical)$$

where,

$$BW(critical) = \#bits(critical) / update\ period$$

This yields a bandwidth value in bits/sec, given the number of bits that must be exchanged in each update period (critical) and the duration of the update period expressed in seconds.

The command and status messages are updated at the same rate. Since the status message is longer, it will drive the bandwidth requirement for the critical data and therefore, its length is used in this example.

A bandwidth of 19.2 Kbaud is used in this example. This represents the expected capacity of the target military radios. Assuming a total bandwidth of 19.2 Kbaud, a status message as shown of 384 bits (including 48 bits of SDLC header) and an update period or control cycle of 30 ms:

$$BW(remaining) = 19200 - 384/.03 = 6400 \text{ bits/s}$$

For the current 100 ms update rate of the TEAM vehicle actuator controller:

$$BW(remaining) = 19200 - 384/.1 = 15360 \text{ bits/s}$$

While the remaining bandwidth shown in these examples is certainly not overwhelming, there is sufficient capacity for communication of significant amounts of information in a timely fashion.

The Role of Local Area Networks

Local area network support is available for multi-processor target systems. A typical target system configuration, for example, might include a number of separate backplanes, each containing several processor boards and a connection to the local area network. The network connects the various backplanes as well as one or more development workstations. Network software supports various standard protocols and supports communication across the physical net or backplanes, allowing any process or processor to communicate with any other. The attractiveness of such a configuration lies in the use of standard communications protocols, the commercial availability of hardware and software, the ability to perform transfers of information without concern about the underlying physical media, and the source-destination flexibility.

For the TEAM project, this network configuration will be used both in the vehicle and in the remote command center. A network bridge is a device used to pass information from one network to another, physically separate, network. Bridging together the vehicle and command center networks would contribute greatly to the flexibility of the communications system. The connection between the two network bridges, one located at each network, would be through the vehicle-command center link.

Recall, as discussed earlier, that the complete communications path of interest is that which connects a process in the command center to one in the vehicle. This can be described as a set of the following connected segments: some command center process to the command center communications system, the command center communications system to the vehicle communications system, and the vehicle communications system to some process on the vehicle. The command center process and vehicle process connections to their communications systems are local to the command center and the vehicle, respectively. The local networks on each, supported by standard protocols, are attractive mechanisms for handling these parts of the link.

The key question concerning use of the network as described here is its performance. Using standard protocols and common hardware can bring many advantages. However, it is feasible only if they can be used in a way that meets the system performance requirements.

While further analysis is required, initial work to determine levels of performance has been conducted and is described here to indicate the reasonableness of using a network and standard protocols. A series of measurements were made that determined the time for a message to be sent by one process, echoed back by a receiving process, and received by the initial sender. Very short messages were used to provide an indication of latency time and longer messages were used to show typical performance.

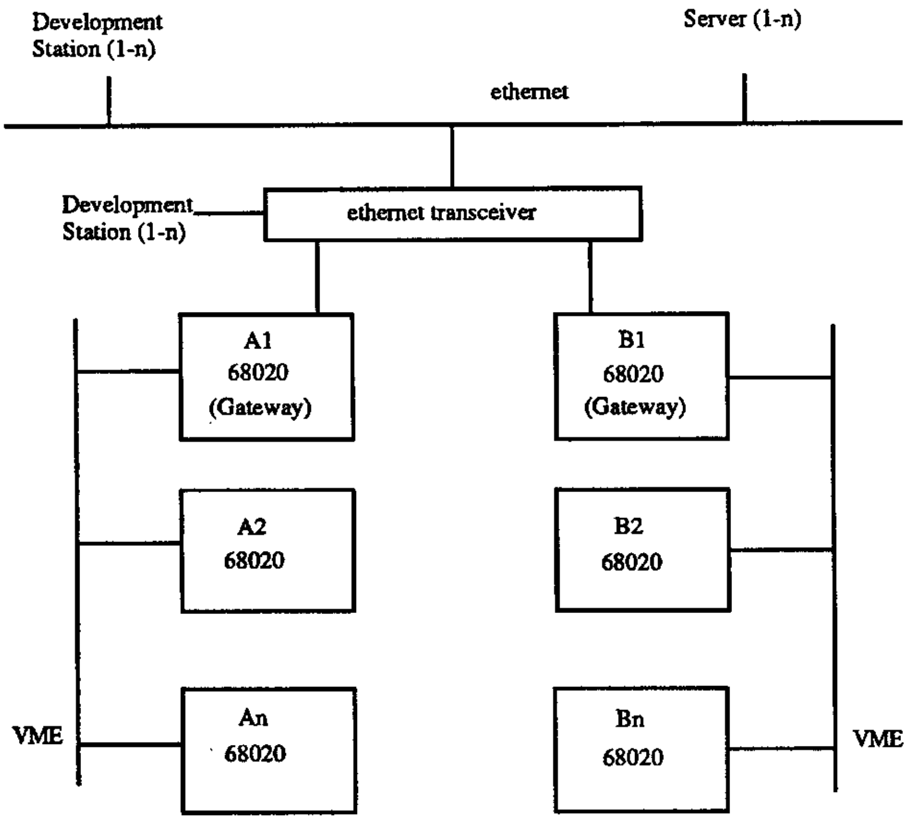

Figure 6. Measurement configuration.

The measurements were made while the processors executed repeated passes through this tight, send-receive loop. Coding was in the C language. Tests were performed for processes executing on different processor boards in the same backplane, and for processes executing on processor boards in different backplanes. All processors were Motorola 68020s [15] and the backplanes were VME [16]. Connection between the backplanes was via an ethernet. Figure 6 shows the configuration used. This is a common target and development configuration for current multiprocessor environments, and was supported by the operating system without custom hardware or software [17]. The same communications protocols were used for communication across the backplane (A-A and B-B) and across the ethernet (A-B), with no change in the test code being required. The protocols used were TCP/IP (Transmission Control Protocol/Internet Protocol) and UDP (User Datagram Protocol) datagrams [18]. Both were supported through a UNIX-like socket interface to the test code. No optimization was attempted and the boards each had a few tasks of higher priorities to handle things like a C shell, network functions, and remote logins.

The measurements were performed with a software analysis tool that combines the functions of a high performance logic analyzer with the ability to access the program symbol table to allow measurements to be made using symbolic program references. This tool monitors all pins of the executing processor unobtrusively and triggers on specified conditions of address, data, or other accessible signals. Measurements reflect the time from the start of a SEND routine, through the RECEIVE routine, and to the start of the next SEND iteration. Table 1 shows the roundtrip times and effective rates for the various message lengths, paths, and protocols. No significant performance differences resulted from conducting the same measurements on both gateway and non-gateway boards.

Table 1. Measured performance.

Protocol	Path	Data length (bytes)	Round trip time (ms)	Effective rate (Kbytes/s)
TCP/IP	ethernet (A-B)	10	14	1.4
		100	15	13
		2000	54	74
	backplane (A-A, B-B)	10	4.9	4.1
		100	5.5	36
		2000	21	190
UDP	ethernet	10	13	1.5
		100	14	14
		2000	51	78
	backplane	10	4.9	4.1
		100	4.9	41
		2000	18	222

An example from the results shows that a message 100 bytes in length can be sent once in each direction across the ethernet using the TCP/IP protocol (the slowest path and protocol tested) in a total time period of 15 ms. This period is small compared to the example 100 ms update rate of the TEAM vehicle actuators. Backplane communications is significantly faster, accomplishing the same length transfer in about 5 ms, and therefore will be useful for communicating within a subsystem. This preliminary data shows that it is reasonable to use these standard mechanisms and protocols for communications in the type of control systems described here. Further analysis is required to determine which of the control and development communications needs can be met by these levels of performance.

PHYSICAL COMMUNICATIONS MECHANISMS

The physical implementation of a communications link between a vehicle and command center can take several forms. Characteristics that must be considered in choosing the type of link implementation include the affect on vehicle mobility, the expected range, the available data rate, and the performance over realistic terrain. The following describes characteristics of the types most frequently employed.

Microwave

Frequency bands in the microwave area are commonly used for communication in robotic vehicle work. Operation in the 1.7-1.85 GHz and 2.2-2.45 GHz military bands is most common [19]. The advantages of microwave operation include easily available bandwidth to support full real-time, noncompressed video, and configurations that allow data to be transmitted on video subcarrier channels.

Unfortunately, a characteristic of microwave communication is that it is very directional, leading to essentially line-of-sight (LOS) operation. This means that a clear path is required between the sending and receiving antennas, unobstructed by any trees, hills, buildings or other objects. In addition, accurate aiming of the sending and receiving antennas is required. Mechanisms required to aim the antennas add to the complexity of the communications system. These problems present significant obstacles to the use of microwave communication for vehicle control in typical outdoor environments.

In addition, even though bandwidth is available, there is an emphasis on transmitting only the minimum amount of information necessary, since any transmission decreases the stealth of a vehicle. In the long term, this makes the high data rate capacity provided by microwave channels less of an advantage.

A variation on the use of microwave transmissions would be the use of satellites. Line-of-sight obstructions are generally reduced because of the satellite's position above the earth and, as before, sufficient of bandwidth is available. Among the disadvantages, data security procedures become particularly important since a satellite broadcasts over a large area. Further, and more important in a vehicle application, is the fact that the 23,000 mile high geosynchronous orbit of a satellite results in a typical 540 ms round trip transmission delay [20]. This sort of delay introduces significant and well known problems for teleoperation.

Fiber Optic

Vehicles connected to a command center via an optical fiber present several desirable characteristics. With no electromagnetic emissions, stealth is excellent. Also, the bandwidth is easily available for video and data with no line-of-sight or interference problems.

The problems associated with fiber are primarily physical. Special equipment must be onboard to pay out the cable, and it is subject to damage from other vehicles running over it. Most operational scenarios dictate paying out the cable and leaving it, but during development researchers typically need to reel it back in, a somewhat painstaking task for lengths of a kilometer or more. Multiple-vehicle operations compound these problems. Reference [21] provides further insight into the issues involved in the application of a bidirectional, optical fiber communications link for a remote system.

RF

Experiments performed under contract to TACOM [19] were performed to measure the effectiveness of vehicle communications at various frequencies ranging from 430 MHz to 40 GHz. The study concluded that there are a number of advantages in the use of frequencies at the low end of the range tested. These include more reliable penetration of foliage and other obstructions, decreased path loss associated with decreasing frequency, ready availability of high power amplifiers and low noise amplifiers at low cost and the ability to use much simpler antennas than is possible at microwave frequencies. This results in a system with much better performance in non-line-of-sight (NLOS) operation than is provided by microwave.

The major problem is the practical unlikelihood of obtaining sufficient bandwidth in the existing frequency bands to fully support a robotic vehicle with both video and control communication. The available bands are already heavily congested with licensed commercial and government users. The recommendation of the TACOM study is to transmit all command and control data at the lower frequencies described and to use microwave frequencies for the video.

This recommendation fits well with TEAM goals since the goal for remote driving video is to eventually reduce the amount of data transmitted to a point were a low frequency channel is sufficient. Further, low frequency transmission of command and control data may permit the use of common military radios. A number of such data radios are available with simple RS-232 [22] or RS-422 [23] communications interfaces to support interconnection to computing hardware.

Using RF frequencies for the command and control information still presents a number of problems. While less severe than with microwave transmissions, there are still limitations in transmitting over hills or other large obstructions. Multipath reflections and interference cause reception problems. The result is that the receiving systems must be able to sort out the good information from the corrupted messages and received noise. Appropriate support for error detection and correction must be developed. In addition, stealth is decreased anytime a transmission is performed since transmitting reveals the vehicle position.

IMPLEMENTATION PROPOSAL

Based on the discussions in the previous sections, figure 8 illustrates a possible configuration for the vehicle communications system and figure 9 illustrates a possible configuration for the command center communications system. The configurations include time-division multiplexing, statistical multiplexing, network bridging, and the use of an RF radio link. All the connections shown, except for the ethernet and VME backplanes, represent simple serial connections (e.g., RS-232 or RS-422).

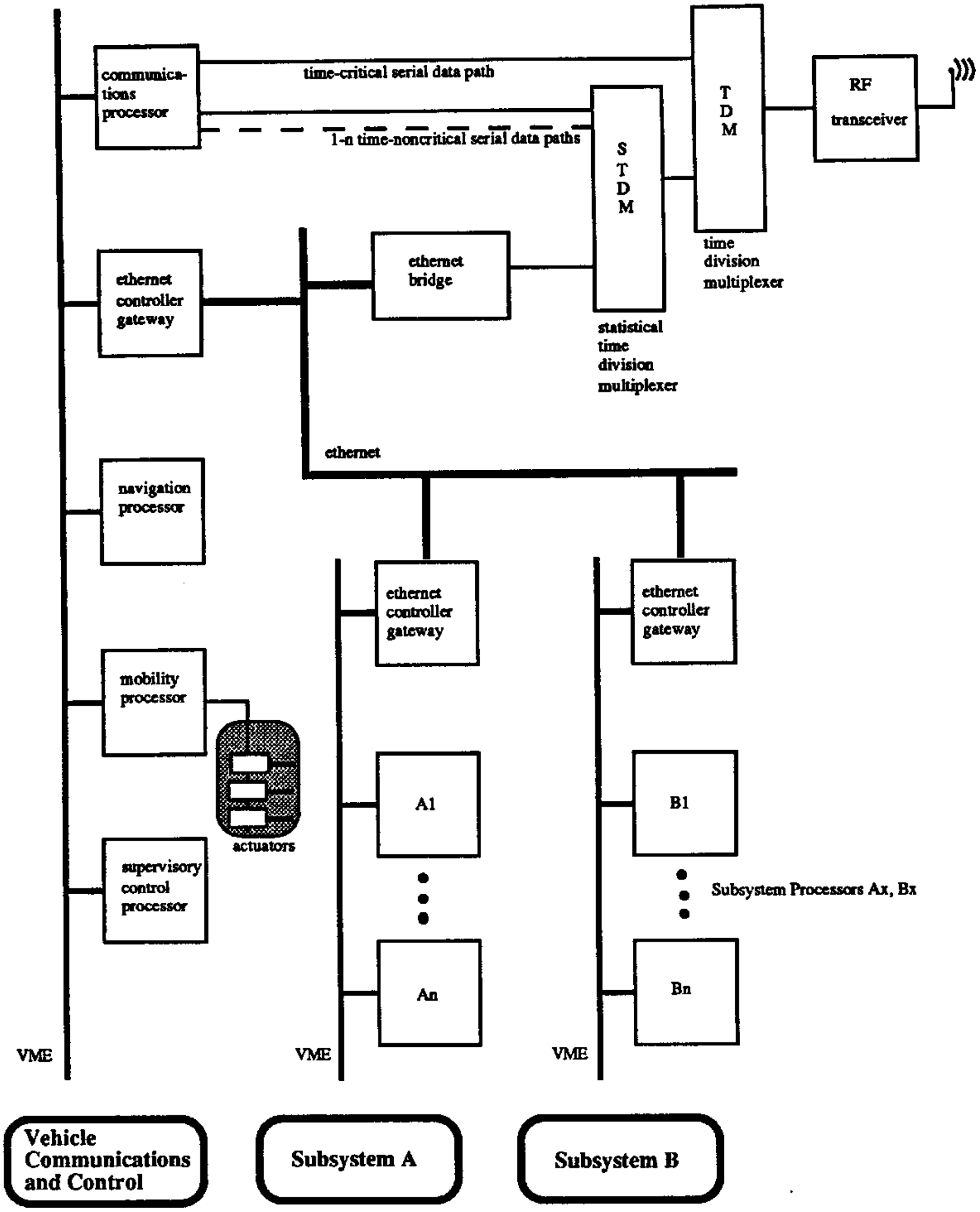

Figure 8. Vehicle communication system configuration.

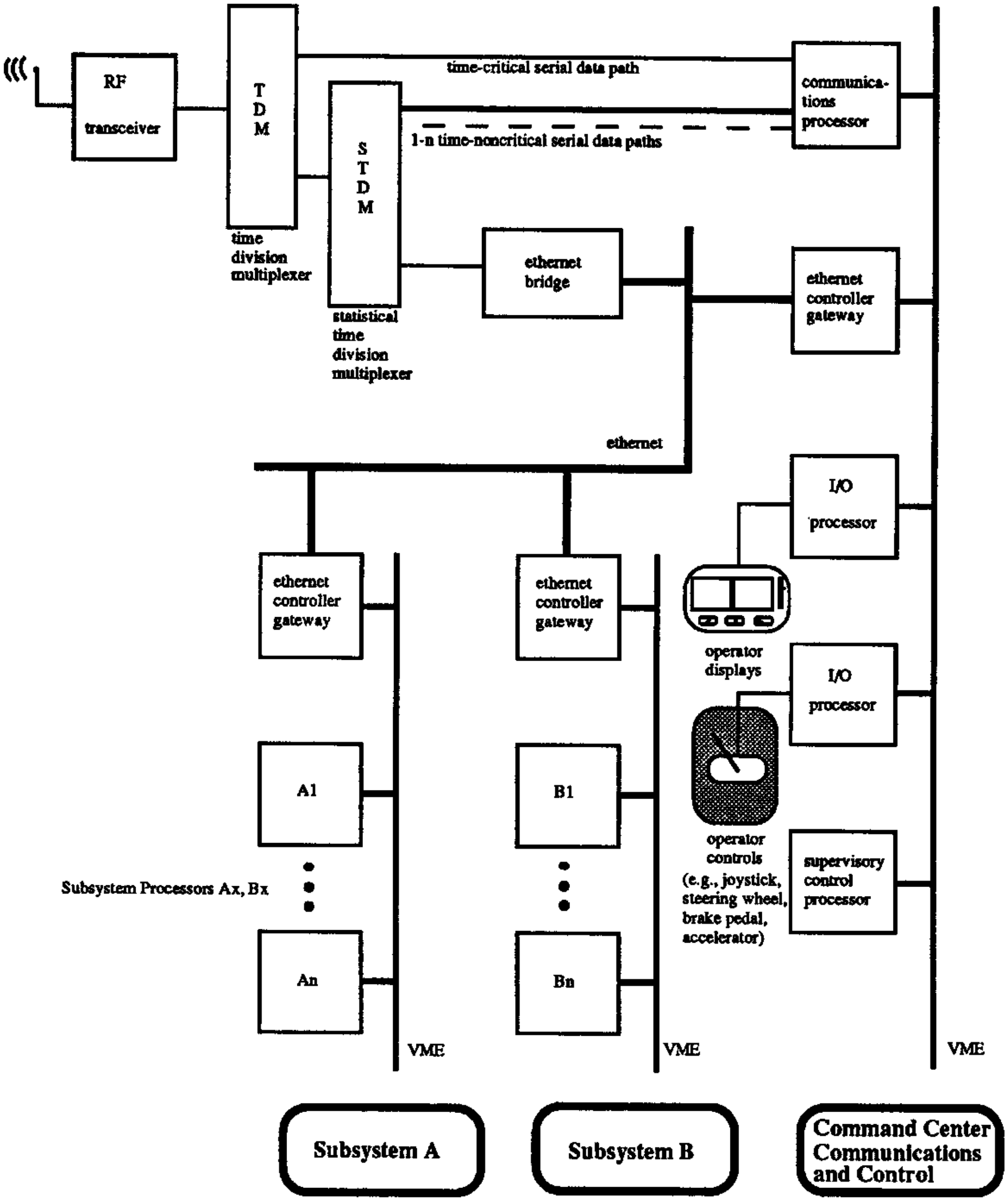

Figure 9. Command center communication configuration.

Commercial hardware exists to perform the multiplexing and bridging functions, however other implementation methods are possible. Any combination of the multiplexing and bridging functions could be performed by the communications processor (Comm Processor) in the Vehicle Communications and Control backplane. In addition, the bridging function may be supplied by the computer operating system if it offers a serial line interface protocol for its networking mechanism. An example of such of mechanism is the SLIP or Serial Line Interface Protocol that works with Internet protocols [24].

It is useful to point out how this configuration provides communication flexibility that supports the control flexibility inherent in the RCS control architecture. One of the strengths of the RCS architecture is that the partitioning of the control system into the modules shown in figure 1 results in information interfaces between these modules that support communications with the control system at any level. The following examples show how a flexible communications mechanism can support the various requirements of the control architecture. Also remember that the lower levels execute faster than the higher levels and thus require communications mechanisms to support more frequent exchanges of information.

An example of the need for higher speed communications is the low-level operator control of a vehicle actuator in real-time. The shaded boxes of figures 8 and 9 show an actuator on the vehicle and an operator input device in the command center, respectively. A fast, predictable link between the operator input device and the actuator is provided. This link consists of several segments as shown in table 2. These segments are listed in terms of the data source, the information path, and the data destination.

Table 2. Use of multiplexing for time-critical information.

	SOURCE	PATH	DESTINATION
COMMAND	I/O Processor	VME backplane	Comm Processor
CENTER	Comm Processor	time-critical serial line	RF transceiver
LINK	RF transceiver	RF transmission	RF transceiver
VEHICLE	RF transceiver	time-critical serial line	Comm Processor
	Comm Processor	VME backplane	Mobility Processor

The command data from the operator input device travels through these segments to reach the input interface of the Primitive level of the vehicle Mobility Processor, which provides the low-level control desired. The input device motion is detected by a Command Center Communications and Control I/O processor and the corresponding data is sent across the VME backplane to the Communications Processor. Then it is passed to the RF radio along a time-critical path provided by multiplexing. The vehicle receives the command and directs it by a time-critical path to the Vehicle Communications and Control system Communications Processor. It is then sent via VME backplane to the Mobility Processor, which treats it as a primitive level command, and generates the appropriate servo level commands for the actuators.

A second scenario illustrates how the actuator might receive command information to support a less frequently updated, higher level command. A decision is made at the Group level to perform a multiple vehicle mission. To complete this mission the vehicle in this example must perform a *Retro-traverse* function, that is, it is to move along a path previously traveled. *Retro-traverse* becomes the Vehicle Task level command.

Retro-traverse will take several minutes to execute, as is typical at the Task level, and thus, can be sent to the vehicle by a time-noncritical means. It is sent from the appropriate Group level control processor in the command center to the Command Center Communications and Control system Communications Processor and on to the RF radio along a time-noncritical path provided by multiplexing. Alternatively, it could be sent over the ethernet and through the bridge, with the bridge output multiplexed as a time-noncritical path. Tables 3 and 4 represent these two alternatives.

Table 3. Use of multiplexing for time-noncritical information.

	SOURCE	PATH	DESTINATION
COMMAND CENTER	Control Processor	VME backplane	Comm Processor
	Comm Processor	time-noncritical serial line	RF transceiver
LINK	RF transceiver	RF transmission	RF transceiver
VEHICLE	RF transceiver	time-noncritical serial line	Comm Processor
	Comm Processor	VME backplane	Superv. Contr. Proc.
	Superv. Contr. Proc.	VME backplane	Mobility Processor

Table 4. Use of bridging and multiplexing for time-noncritical information.

	SOURCE	PATH	DESTINATION
COMMAND CENTER	Control Processor	VME backplane	Ethernet Controller
	Ethernet Controller	Ethernet	Bridge
	Bridge	time-noncritical serial line	RF transceiver
LINK	RF transceiver	RF transmission	RF transceiver
VEHICLE	RF transceiver	time-noncritical serial line	Bridge
	Bridge	Ethernet	Ethernet Controller
	Ethernet Controller	VME backplane	Superv. Contr. Proc.
	Superv. Contr. Proc.	VME backplane	Mobility Processor

The vehicle receives the command and directs it by a time-noncritical path to the Vehicle Communications and Control system Comm Processor, or if ethernet bridging is used, to the ethernet controller via the network. The command is then sent over the backplane to the Supervisory Control Processor, the processor responsible for the Vehicle Task level commands. It is sent via VME backplane to the Mobility Processor and appropriately decomposed into E-move, Primitive and Servo level functions to cause actuator movements.

These simple examples show how such a configuration can support the access to various control levels that is a strength of the RCS architecture. Further, by permitting time-noncritical exchanges to occur via slower paths, the faster paths remain available for time-critical applications, such as low level actuator control. Neither mechanism is better than the other, they each support different levels of communications performance that allow the RCS designer more flexibility.

Note that from the application point of view, the dedicated bandwidth for time-critical data is always available. Thus, continuous, low-level teleoperation (e.g., Primitive level functions) can proceed while higher level interactions (e.g., Task level functions) or non-control functions (e.g., diagnostics, monitoring) proceed concurrently.

STANDARDS

There are no standards in place to address all of the issues relevant to a vehicle-command center communications link. There are however standard models that can help structure the design and implementation of the link, and standard communication protocols that should be considered. The tradeoff to be considered in using standard mechanisms is between the performance penalty imposed by the overhead of the standard approach and the potential benefits of standard approaches. Among the advantages of a standard approach are portability, consistency and available support and documentation. These benefits help simplify the integration of individual components into a complex system.

To the extent possible, the implementation should follow the the International Standards Organization (ISO) Open Systems Interconnection Reference Model [25]. This seven layer model defines a partitioning of the communications functions and provides a framework for standard protocols to be defined for the various layers. Protocols have been and are being developed to support the functions defined for the layers. If new protocols are required for this application, the layering specified by the OSI model should be followed to enhance compatible communications between systems.

The NIST Standard Reference Model for intelligent control systems provides insight into how the control architecture can be partitioned, and hence, helps define what sort of communication is required across the vehicle-command center link. This model has been documented in several forms, each in the terminology of a particular application area. These include NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM) [8] and Mining Automation Real-Time Control System Architecture Standard Reference Model (MASREM) [26]. Further, a more general approach is formulated in an Architecture for Real-Time Intelligent Control Systems (ARTICS) [27]. SARTICS is being explored to evolve a standard set of computing hardware, reusable software modules and a software development environment which can be applied to a wide variety of automation and robotics problems. These models are consistent with the structure for the TEAM RCS discussed in this paper.

TACOM has undertaken an effort to promote robotic vehicle communications interoperability by working toward a standard for vehicle-command center communications. This standard addresses both the communications system functionality, in a manner consistent with the OSI model, and the information content and encoding for the link [13].

With its wide use on development workstations and its appearance on high performance target boards, the TCP/IP protocol [18] is being examined for its suitability in real-time environments. One recent encouraging analysis [28] has concluded that the protocol is not the factor which limits the performance of TCP/IP, but rather that the operating system-to-network interfaces are the areas which need improvement. Further, it is felt that these are areas in which performance can be readily improved . The point here isn't that TCP/IP can do the job (especially since a common view is that it will eventually be replaced by fully OSI compatible protocol suites), but that the level of performance required for robotics and the use of standard high level protocols are not mutually exclusive. Additionally, the benefits of using a standard approaches demand that they be evaluated.

Work has also been performed that leads to similar conclusions about the performance of implementations of the bottom four layers of an OSI structured system [29].

SUMMARY

The development of the vehicle-command center communications link for teleoperated, intelligent, robotic vehicles requires careful analysis in many areas. The constraints imposed by the desired low data rates, time-critical data, line-of-sight communication characteristics, and slow implementations of standard protocols all pose significant challenges. However, the proposed configuration, coupled with higher performance available products, appears to be a possible solution. Offering multiple communications paths, with various performance characteristics, is a key element of this approach.

Communications technologies are continually advancing in performance. It appears that with sufficient analysis and attention to the areas described, it is possible to follow consistent communications and control reference models, and to make use of available standards to develop a well structured, extensible, and fully functional vehicle-command center communications link. Future field testing to be conducted as part of the TEAM program will seek to verify this.

REFERENCES

[1] Szabo, S., Scott, H.A., and Kilmer, R.D., "Control System Architecture for the TEAM Program," Proceedings of the 2nd International Symposium on Robotics and Manufacturing Research, Education, and Applications, Albuquerque, NM, November 16-18, 1988 .

[2] Albus, J. S., Barbera, A. J., and Nagel, R. N., "Theory and Practice of Hierarchical Control", Proceedings of the Twenty-third IEEE Computer Society International Conference, Washington, DC, September 15-17, 1981.

[3] Barbera, A. J., Fitzgerald, M. L., and Albus, J. S., "Concepts for a Real-Time Sensory-Interactive Control System Architecture," Proceedings of the Fourteenth Southeastern Symposium on System Theory, Blacksburg, VA, April 15-16, 1982 .

[4] Barbera, A. J., Fitzgerald, M. L., Albus, J. S., and Haynes, L. S., "RCS: The NBS Real-Time Control System", Proceedings of the Robots 8 Conference and Exposition, Volume 2 - Future Considerations, Detroit, MI, June 4-7, 1984.

[5] McCain, H. G., "A Hierarchically Controlled, Sensory Interactive Robot in the Automated Manufacturing Research Facility", Proceedings of the IEEE International Conference on Robotics and Automation, St. Louis, MO, March 25-28, 1985.

[6] McCain, H. G., Kilmer, R. D., Szabo, S., and Abrishamian, A., "A Hierarchically Controlled Autonomous Robot for Heavy Payload Military Field Applications", Proceedings of the International Symposium on Intelligent Autonomous Systems, Amsterdam, The Netherlands, December 8-11, 1986.

[7] Albus, J. S., "System Description and Design Architecture of Multiple Autonomous Undersea Vehicles", NIST Technical Note 1251, National Institute of Standards and Technology, Gaithersburg, MD, September 1988.

[8] Albus, J. S., "McCain, H. G., and Lumia, R., NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)", NIST Technical Note 1235, 1989 Edition, National Institute of Standards and Technology, Gaithersburg, MD, April 1989 (supersedes NBS Technical Note 1235, July 1987).

[9] Gauthier, D., Freedman, P., Carayannis, G., and Malowany, A., "Interprocess Communication for Distributed Robotics," IEEE Journal of Robotics and Automation, Volume RA-3(6), pp. 493-504, December 1987.

[10] Wheatley, T.E., "Mapping Processes to Processors for Space Based Robot Systems," Proceedings of the IEEE International Conference on Systems Engineering, Dayton, OH, August 24-26 1989.

[11] Martin, J., "Data Communications Technology," Prentice Hall, Englewood Cliffs, NJ, pp. 107-108, 1988, ISBN 0-13-196643-X 025.

[12] Herman, M., Chaconas, K., Nashman, M., and Hong, T., "Low Data Rate Remote Vehicle Driving," Proceedings of the Third IEEE International Symposium on Intelligent Control, Arlington, VA, August 1988.

[13] Mariani, D., "Robotic Vehicle Communications Interoperability," TACOM Technical Report No. 13387, U.S. Army Tank-Automotive Command (TACOM) Research, Development and Engineering Center, Warren, MI, August 1988.

[14] Tanenbaum, A., "Computer Networks," Prentice Hall, Englewood Cliffs, NJ, pp. 254-257, 1988, ISBN 0-13-162959-X.

[15] Anon., "MC68020 32-Bit Microprocessor User's Manual," Prentice Hall, Englewood Cliffs, NJ, 1985, ISBN 0-13-566860-3.

[16] Anon., "The VMEbus Specification," Revision C.1, Motorola, Inc., October 1985.

[17] Anon., "VxWorks Volume 1," Wind River Systems, Inc., Emeryville, CA, 1988.

[18] Comer, D., "Internetworking with TCP/IP Principles, Protocols, and Architecture," Prentice Hall, Englewood Cliffs, NJ, 1988, ISBN 0-13-470154-2 025.

[19] Anon., "Radio Communication System for Robotic Vehicles," U.S. Army Contract #DAAE07-86-C-R066 for U.S. Army Tank Automotive Command (TACOM), Final Report, Comdel Electronics, Beverly, MA.

[20] Martin, J., "Data Communications Technology," Prentice Hall, Englewood Cliffs, NJ, pp. 185-192, 1988, ISBN 0-13-196643-X 025.

[21] Vidula, B.S., and Smith, R.H., "Bidirectional Communications Link for Optical Fiber Payout from a Remotely Operated Vehicle," Proceedings of the Eighth International Fiber Optic Communications and Local Area Network Exposition (FOC/LAN 84), Las Vegas, NV, September 17-21, 1984.

[22] Anon., "Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange", EIA Standard RS-232-C, Electronic Industries Association, Washington, D.C., June 1981.

[23] Anon., "Electrical Characteristics of Balanced Voltage Digital Interface Circuits," EIA Standard RS-422-A, Electronic Industries Association, Washington, D.C., December 1978.

[24] Anon., "SLIP.DOC", VxWorks Manual Pages for Beta Distribution of SLIP for VxWorks Release 4.0.2 with Sun OS 4.0 SLIP Implementation, Wind River Systems, Inc., Emeryville, MD, October, 1989.

[25] Anon., International Organization for Standardization, "Open Systems Interconnection - Basic Reference Model, ISO/TC 97/SC21, International Standard ISO/IS 7498 rev., 1984.

[26] Albus, J.S., Quintero, R., Huang, H., and Roche, M., "Mining Automation Real-time Control System Architecture Standard Reference Model (MASREM)," NIST Technical Note 1261 Volume 1, National Institute of Standards and Technology, Gaithersburg, MD, May 1989.

[28] Clark, C.D., Jacobson, V., Romkey, J., and Salwen, H., "An Analysis of TCP Processing Overhead," IEEE Communications Magazine, Volume 27(6), pp. 23-29, June 1989.

[29] Heatley, S., and Stokesberry, D., "Analysis of Transport Measurements Over a Local Area Network," IEEE Communications Magazine, Volume 27(6), pp. 16-22, June 1989.