N95-14162

27236
P. 20

1995107778

MISSION SAFETY CRITICAL SYSTEMS
RESEARCH & APPLICATIONS
ricis
SYMPOSIUM '92

EVOLUTIONARY TELEMETRY & COMMAND PROCESSOR (TCP) ARCHITECTURE

Mr. John R. Schneider

ABSTRACT

Current development is underway to build a low cost, modular, high performance, and compact Telemetry And Command Processor (TCP) as the foundation of command and data handling subsystems for the next generation satellites. The TCP product line will support command and telemetry requirements for small to large size spacecraft and from low to high rate data. It is compatible with the latest TDRSS, STDN, and SGLS transponders and provides CCSDS protocol communications in addition to standard TDM formats. Its high performance computer provides computing resources for hosted flight software. Layered and modular software provides common services using standardized interfaces to applications thereby enhancing software re-use, transportability, and interoperability. The TCP architecture is based on existing standards, distributed networking, distributed and open system computing, and packet technology. The first TCP application is planned for the 94 SDIO SPAS III mission. The architecture enhances rapid tailoring of functionality thereby reducing costs and schedules during development of individual spacecraft missions.

BIOGRAPHY

Mr. John R. Schneider joined Fairchild Space in December 1991 as a Staff Engineer with the Communications, Data Handling, and Power Systems Department. He currently is working as a system engineer on the TCP hardware and software architecture. Prior to Fairchild Space, Mr. Schneider concentrated as a system engineer on satellite ground systems while employed with NASA/GSFC, NOAA, Mitre, SPACECOM, and Ford Aerospace. His ground systems experience includes RF front end stations, control centers, data processing centers, and communication networks. Notable past projects include Space Station Freedom, EOS-DIS, ERTS/Landsat, TIROS-N, and TDRSS. He holds a BSEE earned in 1968 from the University of Cincinnati.

N

MISSION AND SAFETY CRITICAL SYSTEMS RESEARCH & APPLICATIONS

EVOLUTIONARY TELEMETRY & COMMAND PROCESSOR (TCP) ARCHITECTURE

RICIS Symposium '92
University Of Houston - Clear Lake
Houston, Texas
October 28-30, 1992

John R. Schneider
Communications, Data Handling, & Power Systems
Fairchild Space
Department A-33
Mailstop A-14
20301 Century Boulevard
Germantown, Maryland 20874
(301) 428-6227

THE TELEMETRY & COMMAND PROCESSOR (TCP) IS A VITAL MISSION & SAFETY SYSTEM

The RICIS Symposium '92 focuses on Mission Safety Critical Systems. These systems are characterized as having high criticality and whose correct execution is vital to the successful operation of the mission. In this symposium, these systems include computer controlled real-time applications. This Session II, Generic Architectures For Future Flight Systems, focuses upon architectures of both spacecraft and avionic control systems. This presentation describes a new product, the Telemetry And Command Processor (TCP), currently under development by Fairchild Space. The TCP will serve as the foundation of a control system for future spacecraft.

Mission success is highly dependent upon the real-time control system to reliably perform housekeeping functions, data handling, and information exchange with mission personnel. The need for higher data rates, more on-board processing power, larger storage capacities, and improved communication protocols require the development of new architectures. In addition, industry pressure to rapidly produce new spacecraft with a competitive cost require that modularity and optimum re-use concepts be used in the architecture. In recognition of these needs, Fairchild Space has started development efforts for future real-time control systems - the TCP. The TCP is based on Fairchild's heritage with spacecraft flight data systems especially in providing standardized control systems for multimission spacecraft. Also, the TCP will benefit from hardware and software Independent Research And Development (IR&D) programs. The best features of past systems are engineered into the TCP along with using state-of-the-art technologies and design concepts.

The TCP provides real-time computer based spacecraft control, data handling, and communications with other spacecraft subsystems and with mission personnel. It is compatible with many space-to-ground communication links. The communication link uses the CCSDS protocols in addition to currently used formats (e.g., TDM telemetry and NASA 48 bit command formats). Hardware/software re-use, transportability, and rapid configuration for mission-to-mission adaptability are key design drivers to the TCP. The TCP architecture uses modularity, standardized interfaces, and "information hiding" to satisfy these drivers. The TCP is configured from a toolkit of modular cards using layered software. The cards and software support open system, networking, distributed processing, and packet concepts. Mission unique functions and/or technology insertion is easily achieved through the addition of a new card(s). Reliability from mission-to-mission is also increased through the re-use of proven cards and software. In addition, re-use provides the TCP with the capability to adapt to specific missions at reduced cost and development schedule.

THE TELEMETRY & COMMAND PROCESSOR (TCP) IS A VITAL MISSION & SAFETY SYSTEM

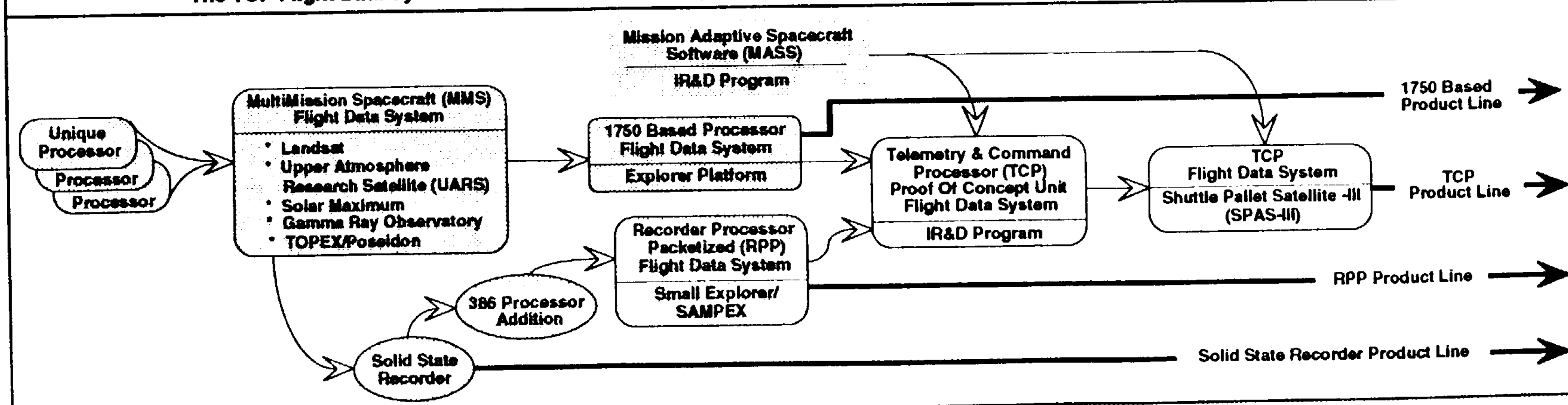
OBJECTIVE

Successful Operation Of The Command And Data Handling (C&DH) Subsystems Is Vital To Spacecraft Mission Execution. Next Generation Spacecraft Range From Small-To-Large Size And From Low-To-High Data Rates. They Will Use State-Of-The-Art Concepts Including CCSDS Communications, On-Board Networking, Distributed Processing, And Open System Architectures. In Addition, Budget Constraints Have Mandated Reduced Costs And Shorter Development Schedules.

AN APPROACH

The Telemetry & Command Processor (TCP) Is The Foundation Of The C&DH Subsystem. It Provides Real-Time Control/Monitor Of Spacecraft Subsystems And The Data Exchange With Mission Personnel. The TCP Supports TDRSS, STDN, And SGLS Transponders. It Provides CCSDS Protocol Communications In Addition To TDM Formats. The TCP Architecture Uses Modular Cards With Layered Software. The Architecture Supports Open Systems, Networking, Packets, And Distributed Computing Concepts. This Approach Enhances Rapid Tailoring Of Functionality And Optimum Re-Use Resulting In Reduced Costs And Schedules For Spacecraft Missions.

The TCP Flight Data System Is Based On Fairchild Space's History With Spacecraft Command And Telemetry Requirements



THE TCP PROVIDES REAL-TIME CONTROL AND DATA HANDLING FOR ALL SPACECRAFT SUBSYSTEMS

Spacecraft C&DH Subsystems generally contain of one or more central processing units surrounded by peripherals. Peripherals include interface units and storage devices. C&DH Subsystems serve the overall purposes of spacecraft subsystem management, data collection, spacecraft health maintenance, and information exchange between the spacecraft and mission personnel. The central processing unit provides the computational power to perform housekeeping functions, data handling, and data communications. Interface units provide the signal conditioning, handshaking, and data transfer with the subsystems. Data storage devices are primarily used for recording on-board data for later playback due to various space-to-ground communication link outages.

In this context, the TCP contains the central processing unit, direct subsystem interfaces, remote interface unit connection via networks, and disk/tape recorder storage device interfaces. Interfaces also exist that allow multiple TCPs to communicate among themselves where more than one TCP are used for reliability purposes and/or for distributed processing. These interfaces provide health and well-being information to the TCPs. The information flow can be across dedicated interfaces or across a networked configuration. For multiple TCPs, the TCP also contains "cross-strapping" of critical input/output interfaces so that only one may serve as a master at a time. In addition, the TCP contains interfaces for use by ground support equipment. These interfaces are used during development for "box" level testing and for the loading/verifying of flight software. The attached viewgraph provides a context view of the TCP's relationship with the spacecraft subsystems.

The TCP supports the overall purposes by reliably providing for command reception, validation, and distribution to the subsystems; the collection, formatting, and distribution of data; the storage and later retrieval of data; the maintenance of on-board time; and the general purpose computational environment to operate flight application software including attitude control, power management, and thermal management.

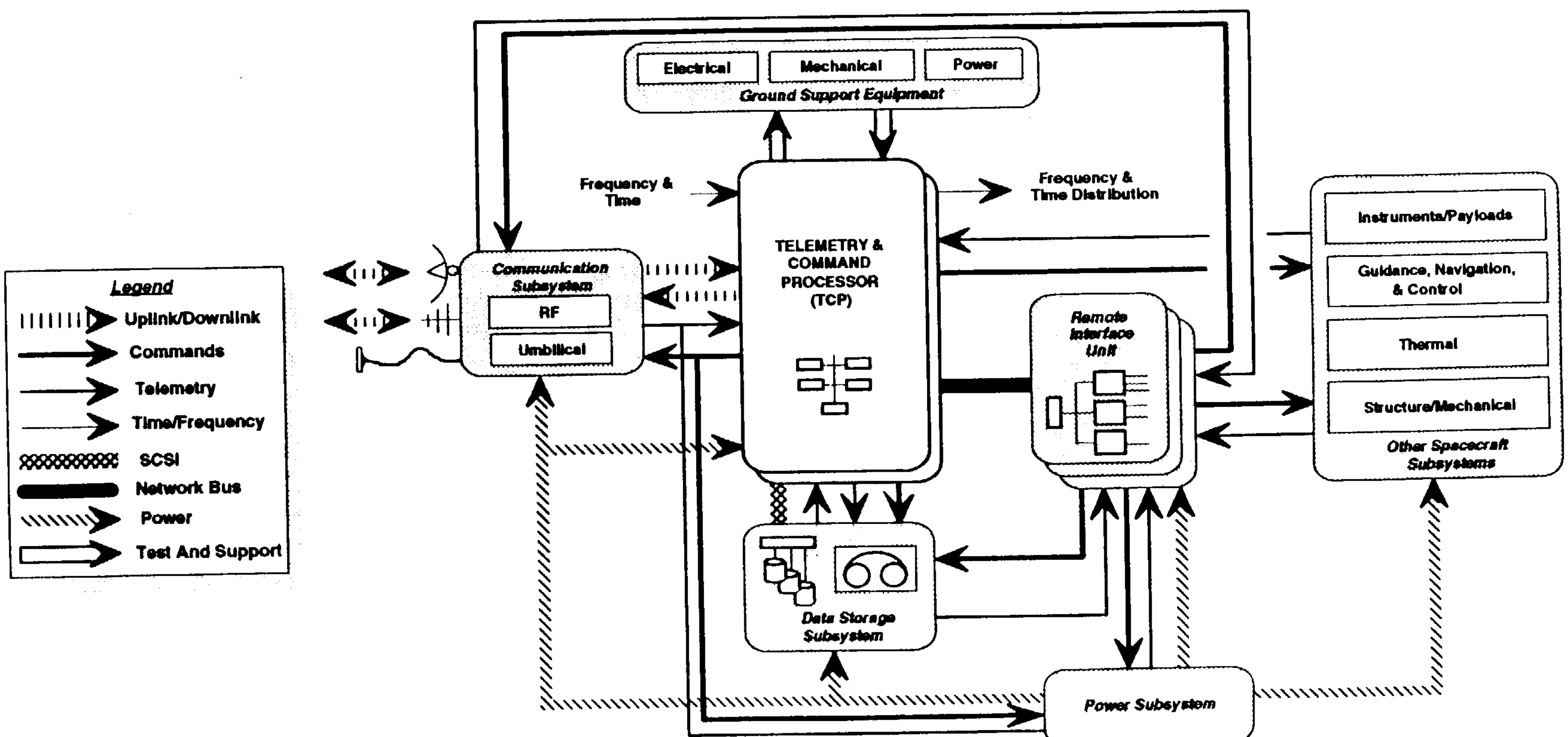
FAIRCHILD SPACE

THE TCP PROVIDES REAL-TIME CONTROL AND DATA HANDLING FOR ALL SPACECRAFT SUBSYSTEMS

The TCP's Overall Purposes Are To 1) Control/Monitor Spacecraft Subsystems, 2) Collect Data, 3) Maintain Spacecraft Health, And 4) Exchange Data And Commands With Mission Personnel.

In Support Of These Purposes, The TCP Provides The Following High Level Requirements:

* Command Reception, Validation, And Distribution
* Data Collection, Formatting, And Distribution
* Data Storage And Retrieval Management
* Spacecraft Time Maintenance
* On-Board General Processing To Host Flight Software



THE TCP PERFORMS 11 MAJOR FUNCTIONS TO SUPPORT SPACECRAFT REQUIREMENTS

The architectural development of the TCP requires understanding of the major functions and their inter-relationships to satisfy requirements.

The TCP performs 11 major functions. ***Uplink Processing*** interfaces with the communication subsystem for command reception. It provides communication handshaking and synchronization, protocol processing, and command validation. ***Command Handling*** provides storage of procedures and time tagged commands and distributes all commands to the subsystems. ***Downlink Processing*** provides the communication subsystem interfaces for the transmission of data. It performs handshaking with the communication subsystem, modulation processing where appropriate, and protocol processing. ***Data Acquisition*** performs data collection from the spacecraft subsystems and from TCP internal functions. It also performs the routing of data to the proper destination. ***Data Storage Management*** performs the data transfer to/from the storage devices and storage control. For disk storage devices, it provides file manipulations (e.g., open, close, delete, copy, and move) and file management (e.g., file directory maintenance, naming, and dating). ***On-Board General Processing*** provides a general purpose computing environment. It hosts the various mission dependent flight software. Examples of resident software include attitude control algorithms and power resource management. ***TCP Control Management*** performs TCP configuration management and control of operational modes and capabilities. It also provides the TCP health and well-being information to other TCPs in a multiple TCP configuration. ***Time And Frequency Generation*** provides spacecraft time management and clock/frequency generation. It also performs the synchronization of time/frequency with external sources and the distribution of time/frequency to the subsystems. ***Built-In Test*** evaluates internal circuits for proper operation and performance. It interfaces with the ground support equipment for box/card level testing during development and pre-launch activities. It also "loads" the flight software into the TCP. ***Power Conversion And Grounding*** receives spacecraft primary power and generates the secondary power for the internal functions. ***Internal Communications*** provides the routing of all signals, power, and grounds among the functions/cards within the TCP.

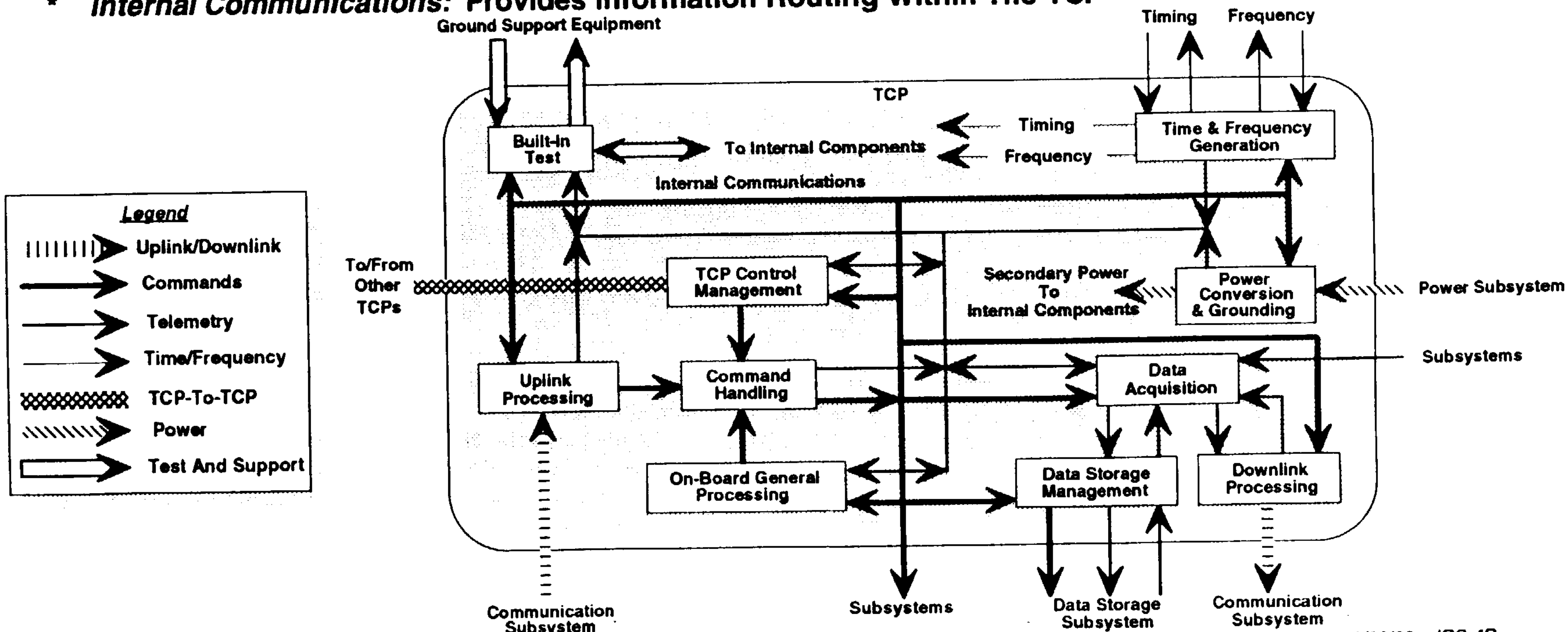
An overview of the inter-relationships of these functions is illustrated in the attached viewgraph. The viewgraph illustrates two key data paths: commands and telemetry. For commands, the communication subsystem provides the uplink signal to the ***Uplink Processing*** function which retrieves the command information for transfer to ***Command Handling***. ***Command Handling*** also receives command information from ***TCP Control Management*** and ***On-Board General Processing***. ***Command Handling*** processes the commands for distribution to the subsystems. It also stores, where appropriate, procedure and time tagged commands. For telemetry, ***Data Acquisition*** collects data from the subsystems and internal functions. It routes the data to four destinations: 1) ***TCP Control Management*** receives internal telemetry for monitoring TCP operations, 2) ***On-Board General Processing*** receives data for input to the resident flight application software, 3) ***Data Storage Management*** receives data that is to be stored, and 4) ***Downlink Processing*** receives data to be formatted for downlink transmission.

FAIRCHILD SPACE

THE TCP PERFORMS 11 MAJOR FUNCTIONS TO SUPPORT SPACECRAFT REQUIREMENTS

The TCP Performs 11 Major Functions Supporting The High Level Requirements. These Functions Are:

* *Uplink Processing:* Provides Command Reception And Validation
* *Command Handling:* Provides Command Decoding, Storage, And Distribution
* *Downlink Processing:* Provides Data Formatting And Distribution To Communication Subsystem
* *Data Acquisition:* Provides Data Collection From Subsystems And Data Routing
* *Data Storage Management:* Provides Tape Recorder/Disk Interfaces For Data Recording And Playback
* *On-Board General Processing:* Provides General Computational Resource For Flight Applications Software (e.g., Attitude Control And Power Management)
* *TCP Control Management:* Provides TCP Configuration And Operational Control
* *Time And Frequency Generation:* Provides Spacecraft Time Maintenance
* *Built-In Test:* Provides Self Checking Tests And Diagnostics And Ground Support Equipment Interfaces
* *Power Conversion And Grounding:* Provides Primary Power Conversion
* *Internal Communications:* Provides Information Routing Within The TCP



THE TCP IS COMPOSED OF MODULAR CARDS AND LAYERED SOFTWARE

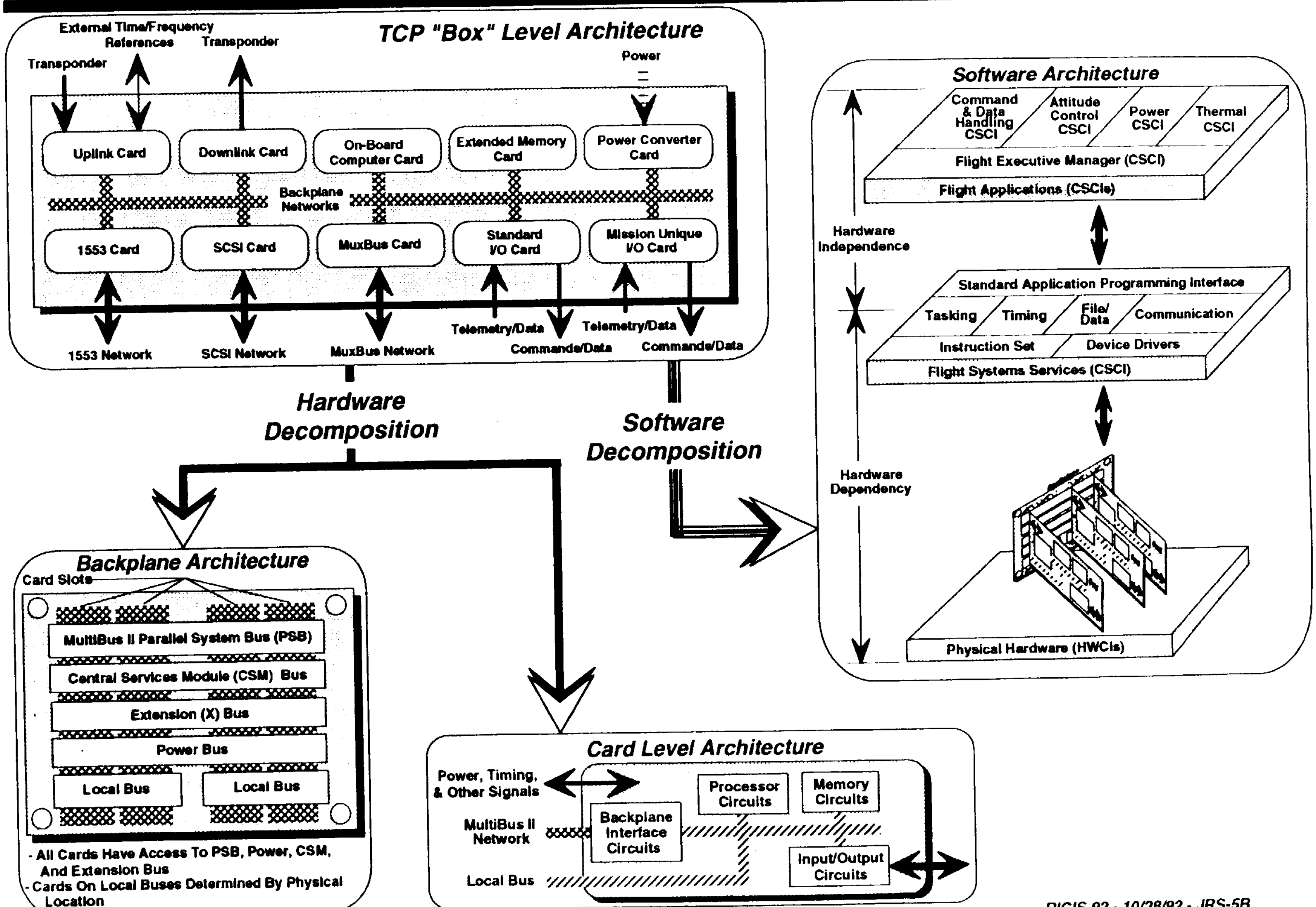
Key design drivers considered during the development of the TCP architecture include hardware/software re-use, transportability, rapid adaptability from mission to mission, modularity, and standardized interfaces. Analysis of the eleven major functions and the design drivers has resulted in an architecture consisting of modular cards connected together via a backplane network within the TCP enclosure. The currently defined card set is illustrated in the attached viewgraph and consists of 10 cards: 1) Uplink, 2) Downlink, 3) On-Board Computer, 4) Extended Memory, 5) Power Converter, 6) 1553 Network I/O, 7) SCSI, 8) MuxBus Network, 9) Standard I/O for direct command and telemetry, and 10) Mission Unique I/O.

The backplane architecture, illustrated in the viewgraph, provides the wiring to interconnect the cards. It is divided into five signal categories. The MultiBus II Parallel System Bus provides the primary communications among the cards and is a network based upon message/packet transmission. The Central Services Module Bus provides the network management signals for the MultiBus II Parallel System Bus. The Extension Bus provides unique signals (i.e., non-network type data) for the cards. Examples of these signals include timing clocks and frequencies. The Power Bus provides the secondary power signals, grounds, and appropriate reset signals. The Local Bus provides a simple, low overhead, network. The Local Bus is primarily for processors to operate with remote memory devices.

Each card, illustrated in the viewgraph, uses the same general architecture. The architecture is based on four components interconnected with a Local Bus. The Backplane Interface Circuits are for communications across the backplane using the MultiBus II network and for network management using the Central Services Module Bus. Processor Circuits and Memory Circuits provide, where required, the computing environment to host the functions allocated to the card. The Input/Output Circuits provide the signal conditioning and handshaking between the card and external devices. A Local Bus interconnects the on-board components together and may be extended into the backplane.

The box level architecture is decomposed into a software architecture in addition to the backplane and set of cards. From an abstract view, the software is allocated to two layers that are allocated to the various TCP cards. The Flight Systems Services CSCI software layer provides the transition from hardware devices to the user environment. This layer provides software to operate the hardware and to perform basic computing services (e.g., communications, tasking, timing, and file/data manipulations). In addition, modules reside in this layer to provide common programming interfaces for application software. The Flight Applications CSCI software layer hosts the application software and a flight executive manager that manages the application software.

THE TCP IS COMPOSED OF MODULAR CARDS AND LAYERED SOFTWARE



COMMUNICATION MODELS AND ARCHITECTURES ARE IMPORTANT INPUTS TO THE DEVELOPMENT OF THE TCP

Government, industry, and international standards and reference models are important considerations during development of control systems. Communication standards and models are important for spacecraft real-time systems like the TCP. In one sense, the TCP provides a "gateway" function between the spacecraft subsystems and the ground system. The TCP "gateway" performs protocol processing on the uplink/downlink, processes information, and performs protocol processing for the information exchange with the subsystems. In addition, the TCP provides peer-to-peer communications between the on-board subsystems and their corresponding ground subsystems.

The TCP uses a variety of networks to communicate with the spacecraft subsystems. Selection of each network is viewed with compatibility to the International Standards Organization/Open Systems Interconnect (ISO/OSI) Reference Model. The ISO/OSI model provides seven layers of services for efficient and reliable communications from one entity to another connected by networks. The underlying concepts of the model are to provide consistent and uniform interfaces between the layers and for each layer to provide higher levels of service than the layer underneath. The first layer, ***Physical***, provides the hardware and media interconnections forming the network. The second layer, ***Data Link***, moves information from one network device to another. It also provides flow and error control. The third layer, ***Network***, provides additional services to move information segments and performs routing management. The fourth layer, ***Transport***, provides reliable end-to-end data transfer between communicating users. The fifth layer, ***Session***, establishes and manages connections between communicating users. The sixth layer, ***Presentation***, provides data format translation to ensure that the data representation is understood by the communicating users. The seventh layer, ***Applications***, provides basic data handling services for communicating users. Some of these services are Message Handling; File Transfer, Access, and Management; Virtual Terminal; Directory Services; and Network Management.

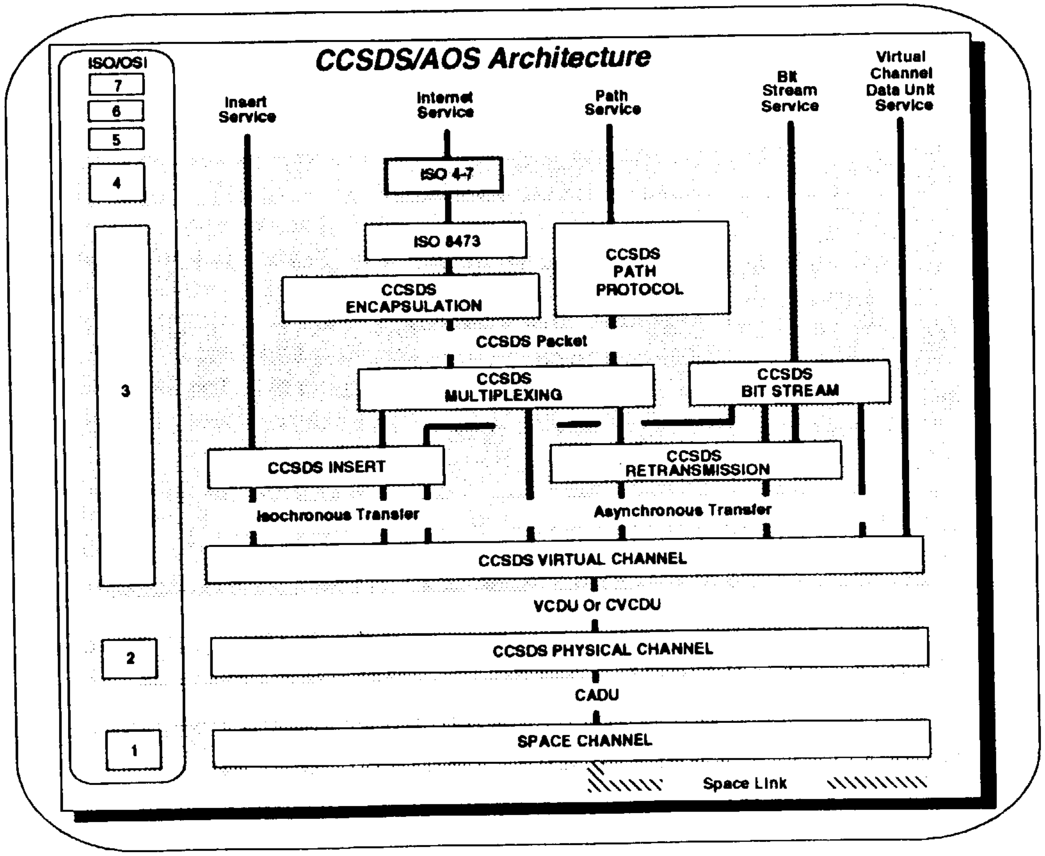
The recent emergence of the CCSDS protocols is of particular importance to the TCP. The primary objective of CCSDS is a new communication architecture, based on the ISO\OSI model, for communicating various data types between a spacecraft and the ground system. The Advanced Orbiting Systems (AOS) and Telecommand architectures of the CCSDS protocol suite are the basis of the TCP's Uplink and Downlink Cards. Telecommand provides reliable and efficient transfer of control information from a ground source to the spacecraft. Standardization of protocol layers and interfaces provides for a common means of ground-to-space communications. The AOS architecture is a full suite of data services between space and ground systems. Data types range from packets with well defined formats to bit streams that are unstructured. The architecture allows Internet and/or path data units to transfer across multiple interconnected subnetworks. AOS has been designed from packet and virtual channel technologies to provide a dynamic means to efficiently assign bandwidth on an as needed basis. Both Telecommand and AOS enhance re-use among multi-missions and cross-support among multiple ground resources/agencies.

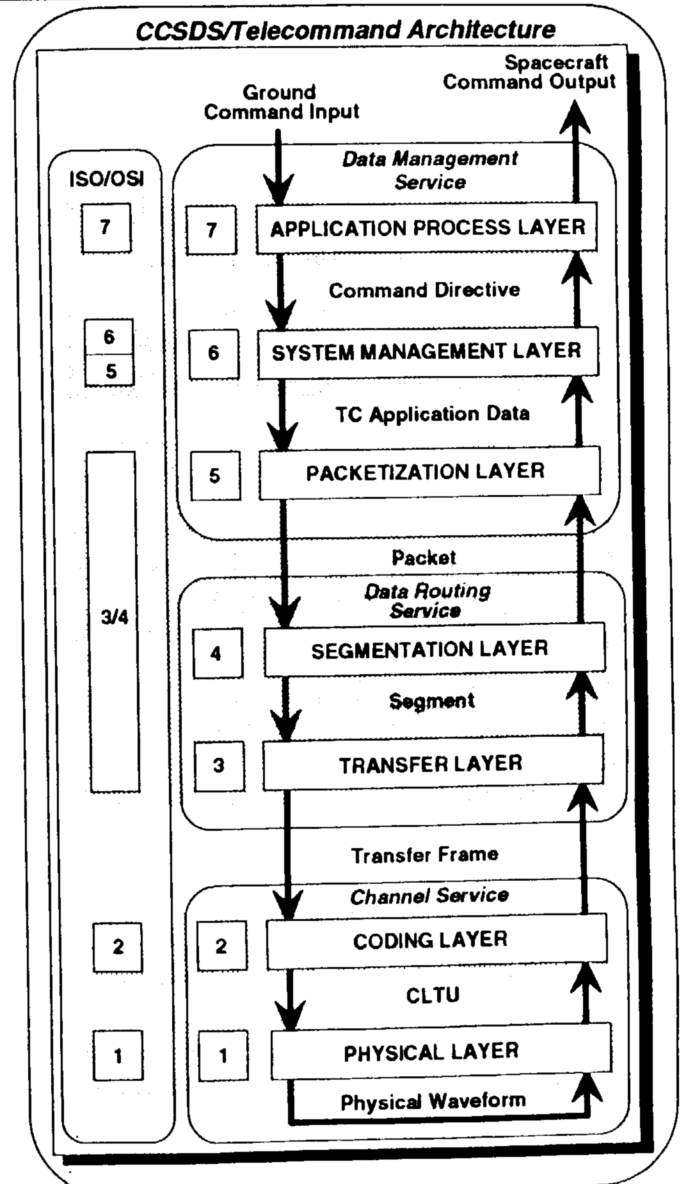
FAIRCHILD SPACE

COMMUNICATION MODELS AND ARCHITECTURES ARE IMPORTANT INPUTS TO THE DEVELOPMENT OF THE TCP

ISO/OSI Reference Model

Layer	Layer Name
7	NETWORK APPLICATIONS
6	PRESENTATION
5	SESSION
4	TRANSPORT
3	NETWORK
2	DATALINK
1	PHYSICAL





OPEN SYSTEMS, LAYERED, AND DISTRIBUTED COMPUTING CONCEPTS ARE ALSO IMPORTANT TO THE TCP

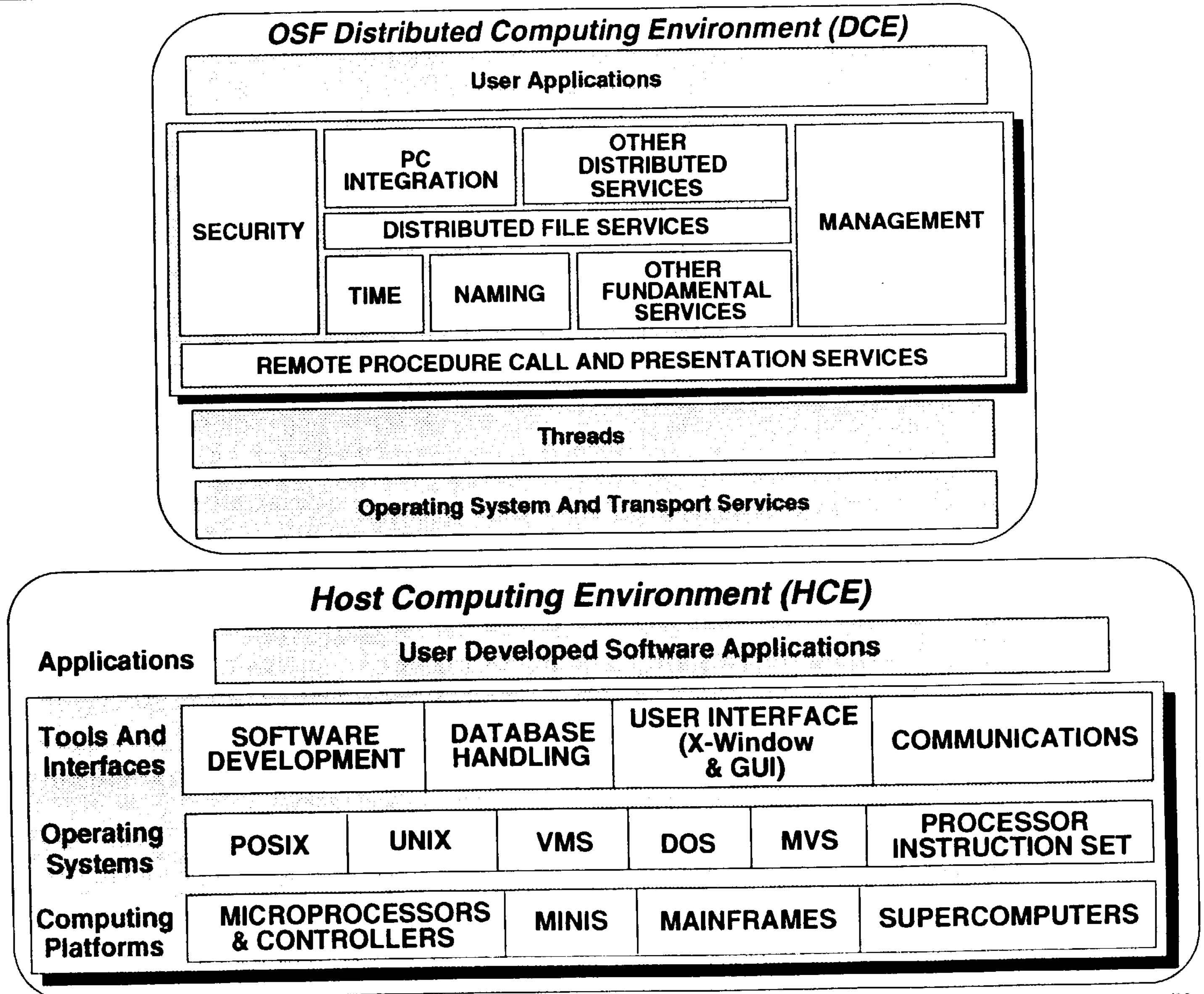
Similar to communication standards and models, computing standards and concepts are important. As a "gateway", the TCP provides a computing environment to process commands for distribution to the subsystems and to process data for transmission to the ground. In addition, the TCP provides a general purpose computer to host flight application software. This software receives data from the subsystems, performs algorithm processing, provides processed data for the downlink, and generates control information for the subsystems. An objective of the TCP is for flight application software to operate in a peer-to-peer interaction with its corresponding ground software. To achieve this objective, open systems, modularity, layering, and distributed computing concepts are design drivers to the development of the TCP architecture.

General computing environments contain four overall layers in its architecture. Layer one, ***Computing Platforms***, contains the computing hardware devices. These devices range from microprocessors/controllers to supercomputers. Layer two, ***Operating Systems***, provides the basic software to manage the underlying hardware devices. This layer ranges from instruction sets for individual processors to full functional operating systems like UNIX, DOS, and VMS. The open system concepts have led to a new full functional operating system that provides common and consistent application programming interfaces independent of the underlying hardware. This system is the Portable Operating System Interface (POSIX). Layer three, ***Tools And Interfaces***, provides basic data handling services for the user applications in addition to the software development and test environment. Some basic tools and interfaces include file manipulations, data base handling, user interfaces (e.g., graphical user interfaces, window managers, and display/keyboard controls), and the linkage to communication services that transfer data among applications and between applications and external hosts. Layer four, ***Applications***, contains the user developed software applications that configure the computing environment to perform the intended mission.

The Open Software Foundation (OSF) has established a Distributed Computing Environment (DCE) reference architecture to promote interoperability within a heterogeneous, networked, computing environment. The primary goal of DCE is to provide a complete, integrated, and uniform set of services to support distributed applications regardless of the underlying hardware. The DCE architecture is layered with the bottom layer providing basic interaction services with the host platform. The highest layer interacts with the user applications. Threads support concurrent programming, multiple executions, and synchronization of global data. It is ideally suited to support client/server interactions. Presentation services provide translations to ensure that the data representation is common to the distributed users. Remote Procedure Call establishes the connection between communicating applications on different hosts. Time ensures a single time reference is used between applications on different hosts. Naming identifies distributed resources on the network. Distributed File Service implements the client/server model and enables global file accesses to appear as a local access. PC Integration allows minicomputers, mainframe, and PC users to share resources in a distributed environment. Security provides the distributed environment with authentication, authorization, and user account management services.

FAIRCHILD SPACE

OPEN SYSTEMS, LAYERED, AND DISTRIBUTED COMPUTING CONCEPTS ARE ALSO IMPORTANT TO THE TCP



THE REFERENCE MODELS ARE INTER-RELATED WITHIN THE TCP TO FORM PEER-TO-PEER INTERACTIONS WITH THE GROUND

The emergence of many models and reference architectures requires a system architect to understand the interactions and relationships of the models when building a new system. Each of the models describes an important aspect of networking and distributed processing and their role in supporting open systems and peer-to-peer communications. The CCSDS, ISO/OSI, Host Computing Environment (HCE), and DCE models are relevant to the TCP. The attached viewgraph depicts the overall model relationships within the TCP and its peer-to-peer communications with the ground system.

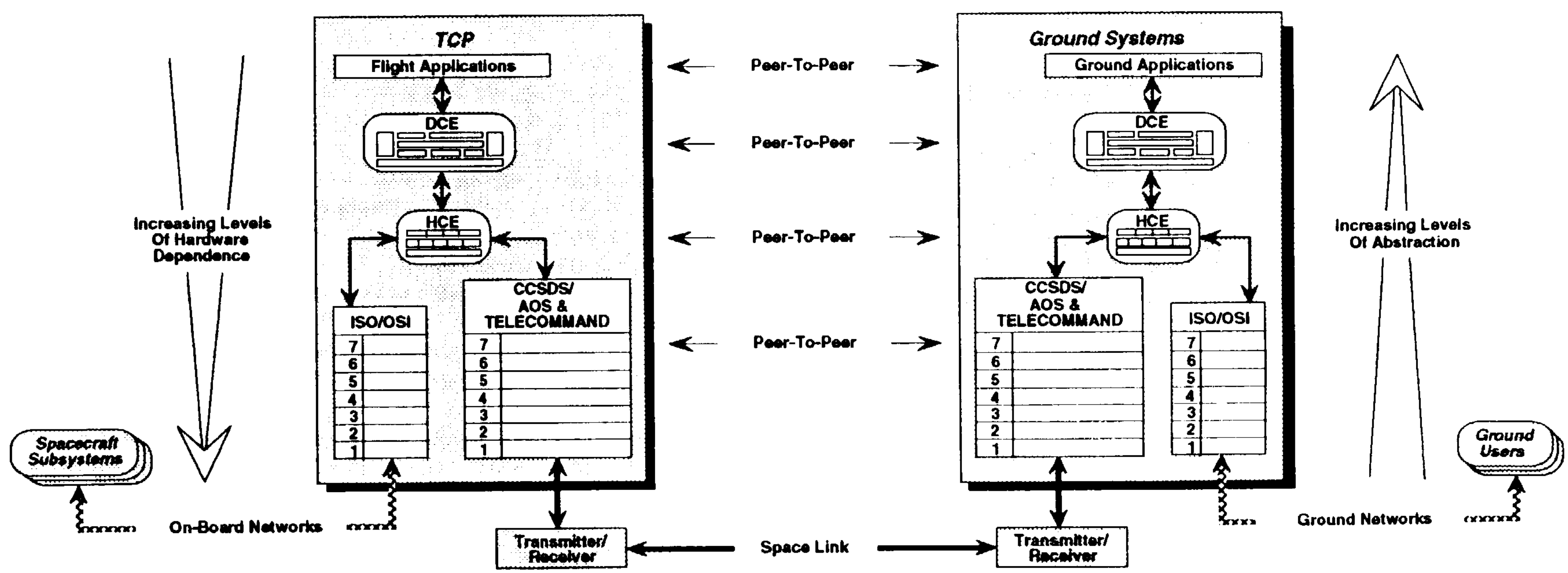
The end-to-end system is viewed as three networks connected together to form the path between the end user(s) and the spacecraft subsystems (e.g., sensors and actuators). The first network connects the end user(s) with the ground system. It may use any available standard network (e.g., Ethernet, Internet, or X.25) or may be custom. The second network, space link, connects the ground system to the spacecraft/TCP. The third network connects the TCP to the end sensors/actuators. It may also use standard networks or may be custom. Both the ground system and TCP provide "gateway" functions in the sense that both form the linkage between two different networks and perform the protocol processing for each connected network. For the space link, the ground system and TCP use the CCSDS AOS and Telecommand architectures. The physical path is established at layer 1 with peer-to-peer communications at the upper layers. For the ground network and on-board networks connecting to their respective users, both employ protocol processing based upon the ISO/OSI model.

In addition to the "gateway" functions, the ground system and TCP provide important processing capabilities to manage the spacecraft mission. This processing consists of the HCE, DCE, and applications software. The HCE provides the general computational platform for hosting the applications in addition to providing the connection with the networks. The applications software would reside directly on the HCE if it were not desirable to support distributed processing between the TCP and other on-board subsystems or between the TCP and the ground system. However, because distributed processing provides significant advantages to systems such as easier global access to data and higher levels of abstraction to the user, the DCE is placed between the HCE and the applications.

The combination of models forms an architecture that promotes hardware independence and abstraction. At the lowest levels, the architecture is highly dependent upon the hardware implementation and its resident operating system. User applications at this level must know them in detail. As user applications are moved higher up in the architecture, the underlying hardware and configuration become hidden from them and the underlying layers provide higher and higher levels of services. At the highest level, the application-to-system interface becomes purely logical where the application need specify only what it needs. The system will perform the implementation of the need.

FAIRCHILD SPACE

THE REFERENCE MODELS ARE INTER-RELATED WITHIN THE TCP TO FORM PEER-TO-PEER INTERACTIONS WITH THE GROUND



USE OF REFERENCE MODELS AND SOFTWARE RESULT IN MODULAR AND LAYERED TCP ARCHITECTURE

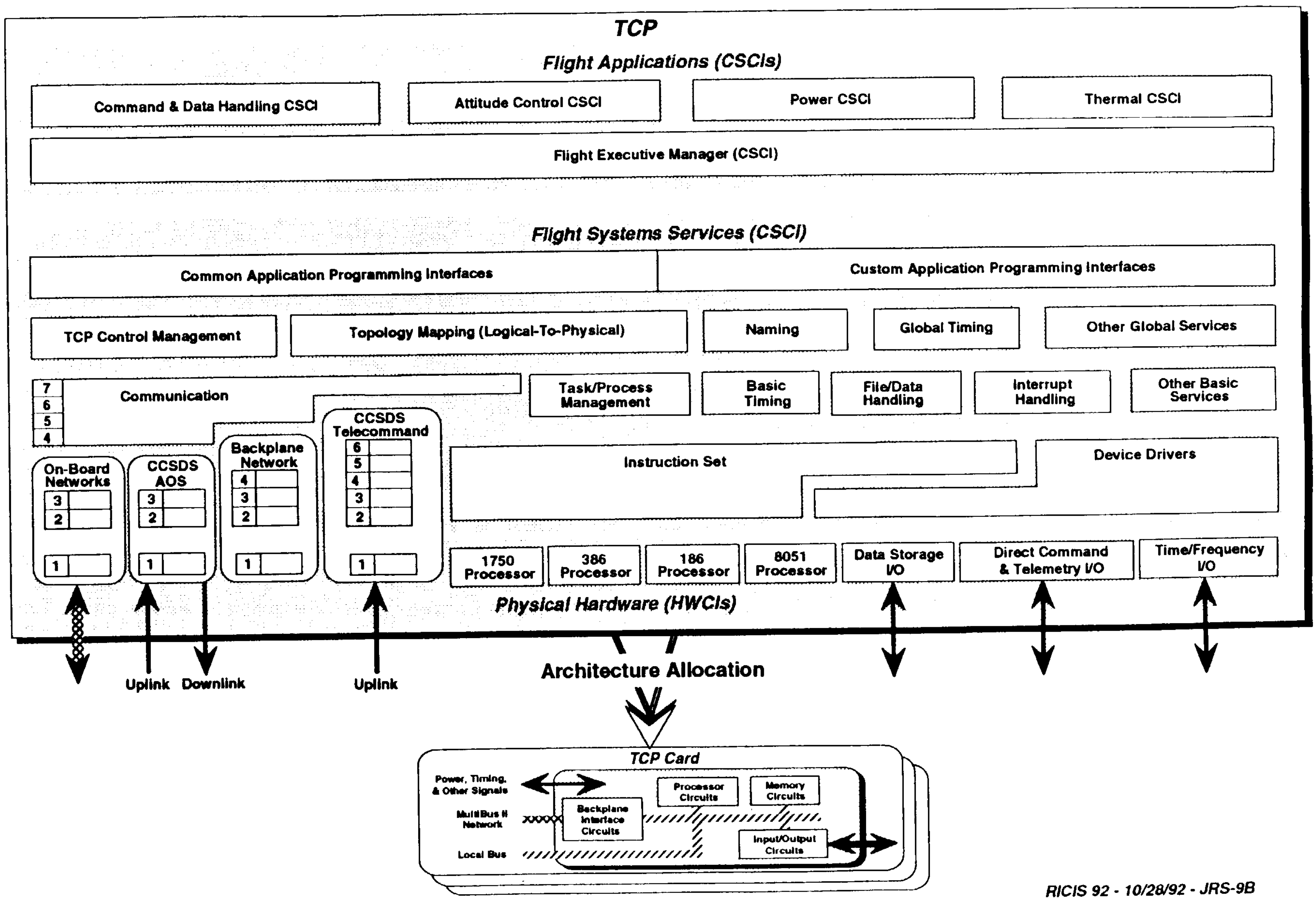
The CCSDS, ISO/OSI, HCE, and DCE models are important inputs to the TCP architecture development. However, two negative aspects of these models are 1) they contain a significant amount of processing overhead and 2) they provide many services not required for a particular implementation. Generally for spacecraft missions, the control systems are constrained by size, power, and weight requirements. These constraints limit the amount of processing capability that can be achieved. Environmental factors (e.g., radiation, shock, vibration, and thermal) also are factors in the amount of available processing capability. In addition, user applications are the highest priority for mission success with the housekeeping functions being the lowest priority. Within the constraints, a prudent "stripping down" of the models can be achieved while still maintaining the overall concepts of open systems, distributed processing, and networking. The general technique used within the TCP is to maintain the model's lower layers intact and replace the upper layers with a single, efficient, software module that preserves the outer interfaces and as much of the services as possible. In addition, the layers and services not required for flight are removed. As the computing performance/size ratio improves through advancements in processor technology, new higher performing processors can be inserted into the TCP allowing for the addition of those layers and services that were initially removed.

The attached viewgraph shows the TCP architecture. It illustrates the use of the models and modular and layered design techniques. The architecture consists of three high level layers (Physical Hardware, Flight Systems Services, and Flight Applications) that contain sub-layers. The Physical Hardware layer contains all the hardware devices including processors, dedicated I/O devices, and communications media contained within layer 1 of the networking models. The Flight Systems Services layer contains sub-layers using elements from the ISO/OSI, CCSDS, HCE, and, DCE models. In the case of communications, the protocol stack is maintained to at least layer 4 and, for Telecommand, to layer 7. The remaining upper layers have been combined into one software module. It serves the various networks and provides the linkage to the HCE. For the HCE, a full operating system is not used. Rather, the processor instruction sets are used coupled with programs to perform task/process management, basic timing, file/data handling, and interrupt handling. The next sub-layers use the DCE model to provide higher levels of abstraction and hardware independence to the flight applications. The provided services include the overall management of the TCP, naming, global timing and synchronization, resource mapping that translates from logical names to physical locations as one of its features, and common programming interfaces. Within the flight applications layer is a flight executive manager that provides management of the upper mission application programs like attitude control, power, and thermal.

The overall TCP architecture is allocated to the individual TCP cards and, subsequently, to the individual major components on the card. For example, the backplane network protocol stack is allocated to the backplane interface circuits for all cards. For the downlink card, the CCSDS AOS network is allocated to the input/output circuits. And for the on-board computer card, almost all of the flight system services and flight applications layers are allocated to the processor circuits.

FAIRCHILD SPACE

USE OF REFERENCE MODELS AND SOFTWARE RESULT IN MODULAR AND LAYERED TCP ARCHITECTURE



TCP - DESIGNED AND BUILT FOR THE NEXT GENERATION SPACECRAFT

In summary, this symposium describes Mission Safety Critical Systems that have high criticality to the successful operation of a mission. The TCP, a computer based real-time control subsystem, is one of them. The TCP is the foundation of Command & Data Handling Subsystems. It provides command handling, spacecraft health control and monitor, time management, data storage, data exchange with the ground, and the hosting of flight applications software. Its architecture is based upon communication and computing reference models and architectures. Modularity and standardized interface concepts have led to a TCP composed of a set of modular cards connected together through a backplane with the cards containing layered software. The architecture enhances optimum re-use and transportability to support rapid mission adaptability while reducing costs and shortening development schedules.

TCP - DESIGNED AND BUILT FOR THE NEXT GENERATION SPACECRAFT

In Summary, The TCP

* Is An Important Mission & Safety System For The Success Of The Spacecraft Mission
* Provides Command Handling, Spacecraft Health Control And Monitor, Time Management, On-Board Processing, Data Storage, And Data Exchange With Mission Personnel
* Uses ISO/OSI Reference Models, CCSDS Protocols, Open System And Distributed Processing Concepts, And Packet Techniques
* Is A Set Of Modular Cards With Layered Software That Enhances Re-Use And Transportability To Support Rapid Adaptation To Individual Missions Thereby Reducing Costs And Shortening Schedules