

**MONITOR AND CONTROL SOFTWARE FOR GROUND SYSTEMS IN
THE DEEP SPACE NETWORK**

Paul Pechkam

**Network Control Task
NASA/Jet Propulsion Laboratory**

July 22, 2002

ABSTRACT

NASA's Deep Space Network (DSN) is an international network of antenna tracking complexes used for two-way communication with deep space and Earth-orbiting spacecraft. A DSN complex will utilize several antennas concurrently and share ground systems, which include transmitters, telemetry processors and receivers, for examples, from a pool of resources. These complexes are manned operations centers, but they are also networked to a central operations center for remote monitoring. This dynamic allocation of resources in addition to providing worldwide visibility requires a distributed and dynamic monitor and control system. Building a reliable and open software infrastructure is the key to scalable and distributed monitor & control and ultimately, automation at multiple levels.

This has been accomplished by Network Monitor and Control (NMC) software developed at Jet Propulsion Laboratory (JPL). Developed initially to replace legacy monitor and control hardware and software, the NMC was designed to provide a platform for supporting legacy ground equipment, future equipment, and automation of DSN operations. Typical of monitor and control software, NMC includes graphical user interfaces (GUI's) for human operators to input commands to ground equipment and displays for viewing equipment health or tracking status. NMC also introduces automation concepts for operations, but the basis of any automation is the retrieval of monitor data to build the "world" view. This paper will discuss how NMC achieves operational goals and performance required of monitor and control of numerous ground equipment. The focus is on infrastructure and detailed software concepts for providing scalable, reliable and performing software and functionality.

A distributed client-server architecture for infrastructure allows many clients to access or produce hundreds of monitor data items or event messages each. Optimization methods and design patterns are used in server and client architecture to obtain work distribution. In addition, software techniques for building servers include effective thread and network programming in addition to using existing protocols as TCP/IP. Another important factor is a client API layer, which is key to supplying transparent connectivity between ground equipment and monitor & control interfaces, which any form of automation relies on. System architecture for NMC is based on distributed servers that perform different functions such as processing user commands, translation of legacy data, and data management. How these functions are distributed is key to providing scalability and maintainability.

Currently, the NMC is being utilized for 24/7 operations at the three main DSN complexes supporting all of the major spacecraft supported and managed by JPL. A portable version of NMC is being exercised in a Compatibility Test Trailer for off-site monitor and control. Remote monitoring of the three complexes at JPL is accomplished as part of the NMC infrastructure.

INTRODUCTION

The Network Monitor and Control group was tasked to develop a software/hardware architecture to provide monitor and control in the Deep Space Network (DSN). One of the major tasks was to build an infrastructure to replace a legacy system. An infrastructure that used Distributed Computing Environment (DCE) as its core was first implemented. For various reasons, this component was determined too risky for the DSN's future needs. This paper documents the software infrastructure that was put in place as part of what is known as Delivery 1.3 of the Network Monitor and Control (NMC) software. Though the NMC software covers a vast array of functionality, the focus will be on software infrastructure.

The Deep Space Network

NASA's Deep Space Network (DSN) is an international network of antenna tracking complexes used for two-way communication with deep space and Earth-orbiting spacecraft. A DSN complex will utilize several antennas concurrently to track different spacecraft or the same spacecraft (via arraying two or more antennas) simultaneously. To provide round-the-clock tracking of a particular spacecraft, three complexes have been positioned around the world: the Goldstone Deep Space Communication Complex (GDSCC) in Barstow, California; Madrid Deep Space Communication Complex (MDSCC) in Madrid, Spain; and Canberra Deep Space Communication Complex (CDSCC) in Canberra, Australia. These three complexes collect data from the spacecraft, package it, and transport the data back to a control center at Jet Propulsion Laboratory in Pasadena, California, where it can be distributed to the spacecraft's project liaisons or scientists. Tracking a spacecraft is a global affair, and thus, a network of spacecraft tracking complexes provides the backbone for these operations. Each tracking complex comprises of several antennas of different sizes (from 26 meter to 70 meter) and types (including Beam-wave guide and High-efficiency Frequency). Typical services supplied by ground equipment include telemetry, tracking, ranging, and commanding. Each equipment type will report configuration, status or measurement data via monitor data or event messages that can be viewed via displays, or automation software can use the data to determine the next logical step in operations. A particular piece of equipment can produce an unlimited number (in the thousands) of monitor data, thus a system is required that can handle the data flow efficiently. In addition, new types of equipment are being developed such as arraying software or upgrades to current up-linking or down-linking systems that require an open system for easy integration.

HIGH LEVEL REQUIREMENTS

The Deep Space Network requires an infrastructure that:

- Supports operations of up to eight concurrent antennas
- Supports legacy equipment (requiring some form of translation)
- Supports publication of thousands of monitor data
- Supports event notification
- Supports reliable communication to equipment in the form of commands or directives
- Supports transportation of monitor data between sites or to central location over a Wide Area Network (WAN)
- Optimizes for new equipment upgrades
- Supports automation plug-ins.

HIGH LEVEL OPERATIONAL CONCEPTS FOR MONITOR AND CONTROL

- Complexes and their equipment are a dynamic environment, which makes it more difficult to "monitor" and "control". A key operational concept starts with time before the beginning of tracking, in which a set of required ground equipment, which could include a set of transmitters, receivers, or telemetry processors, will be allocated from a common pool of resources to a *connection* associated with a particular antenna and spacecraft. A human operator will configure the equipment and then monitor the equipment for the duration of the track (i.e. while the spacecraft signal is locked and flowing spacecraft data). After the track is complete, the allocated equipment will be released into

the common pool where it will be allocated for the next support on the schedule, which may not be on the same antenna.

- 24/7/365 performance – NASA’s Deep Space Network (DSN) is responsible for round-the-clock tracking of many deep-space and orbital spacecraft. The infrastructure must be reliable and perform in its best capacity at all times. That is, the system should be active with the latest data and be able to communicate with equipment (via operator commands or directives) at any time. This requires a system that achieves 24 hours a day, 7 days a week, 365 days a year level of operation. Any clean-up, routine maintenance, and initialization of new clients must occur without effect on a system that is currently running.
- Automation – As more spacecraft are being supported by the DSN, cost-effective solutions such as automation are being pursued. Thus, a system must be built to support automation plug-ins. By default, automation requires the same information that a human operator would require to perform a specific action. However, good automation needs access to more information so that it can build a database and a sense of history for trend analysis and to predict faults. Some access point must be available for automation to plug in and issue directives and receive monitor data. This also implies a level of security in operating equipment.

SERVICE ORIENTED APPROACH FOR THE INFRASTRUCTURE

In order to fulfill the functional requirements, the approach is to define services that can be contained and distributed. A client will be a user of one or more of these services.

Name services – clients will need to know how to talk to servers on a network level. Physical host information for a client is mapped to a user-defined *physical name*. This is the basis for all network communications.

Functional name services – data needs to be represented by proxies or functions for organizing data. A client may register several functional names to distinguish sets of monitor data. These functional names will then be mapped to a physical name provided by the Name Services.

Monitor Data services – provide a means of publishing and/or subscribing to data items represented by the Functional name services

Messaging services – allows for clients to send data to other clients. These are regarded as peer to peer transactions defined and arranged by the respective clients.

Event notification services – allows for clients to send events that can be logged in a central location. Events are different from monitor data in that combined with previous events they can provide history.

Monitor and Control services – provide protocols and transactional processes for a client to direct another client. For example, these services would be used to send a command from a display to a particular equipment controller. Timeouts and exceptions are provided as part of these services.

Connection services – allocate and manage equipment for a tracking session.

Software is then developed to make these services available to applications.

SOFTWARE DESCRIPTION

The NMC software can be described as a set of *network* applications; they fit perfectly within various network configurations so that the customer can take advantage of the latest server hardware or network hardware and technologies. An Internet-like strategy is adopted; typical of many Internet applications, NMC applications fit on top of the transport layer in the *open system interconnection model* (OSI). TCP/IP, a standard communications protocol and known for its reliability, is used, replacing legacy proprietary protocols.

A *client-server* software model is used. Services are distributed between servers and a client layer (see Table 1). A client will utilize one or more of the services provided by a server and make transactions. Containing some of these services in a client-side library provides *client-client* communication.

Component	Service provided
Connection Engine	Connection Services
Monitor Data Server	Functional Name, Monitor Data, Event Notification Services
Name Server	Name Services
Client library (API)	Messaging, Control Services

Table 1 - Component mapping to service

A client-server and client-client interaction is shown in Figure 1. The Name Services, Functional Name Services and Control Services are utilized to send a command (a request for action) from one client to another client. A client will address the command to a functional name. Through transactions, the functional name will eventually map to host information for direct communications between clients.

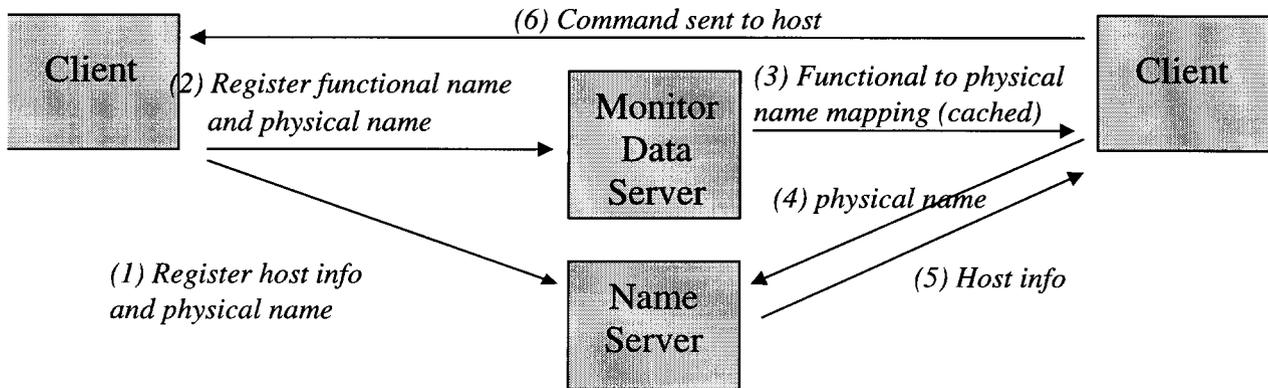


Figure 1 - Client-Server processing of a command (Control)

Another concept used is *publisher-subscriber*, in which a client will *subscribe* to data items that can be *published* by another client. This ensures that only interested data is sent over the network to only the requesting destinations. The client-server interaction for subscribing to data and receiving data is shown in Figure 2. This activity utilizes the Functional name services and Monitor Data services.

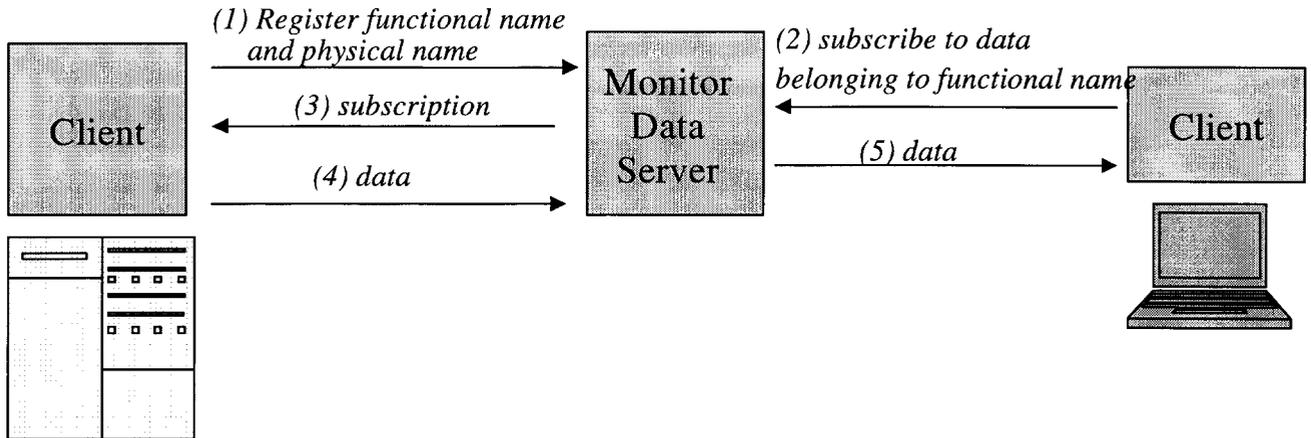


Figure 2 - Client-Server processing of monitor data (Monitor)

A key software component is the Application Programming Interface (API) contained in the form of a shared library. This library contains client-server and client-client protocols for communication and access to the various infrastructure services. These are event-driven asynchronously using threads to provide callback technology or triggers. This allows subscribing clients to react to monitor data updates when they arrive. This reaction could lead to an update to a display or it can cause automation software to react with a response.

An interpreted scripting language is also available for creating simple forms of automation. Graphical displays can be developed using a provided GUI API, and web services can also be provided to produce HTML pages of monitor data and events, for example.

Currently supported platforms include Solaris 2.5.1 and Solaris 8. Infrastructure services are written in C++ allowing for future portability onto various platforms. Connection oriented services are written in C, C++, and a template language called SNAP. Displays are written in X/Motif. For increased portability, standards are used including POSIX where applicable.

Thread models and queue processing provide asynchronous processing to efficiently process both server transactions and client receipt of data. The NMC software is optimized for data transmissions and local and wide area networking architectures.

A typical NMC configuration at a complex will include a Local Area Network (LAN) of at least one workstation for a human operator, which he/she will use as an interface to operate one or more antenna, a shared file-system, and the servers. This configuration is networked with the various ground equipment controllers that it has to communicate with.

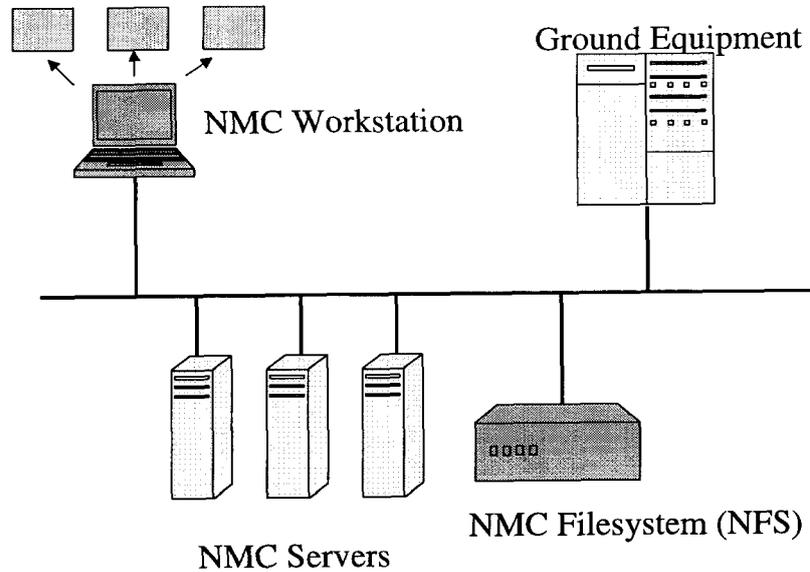


Figure 3- An NMC at a complex

The NMC system at each complex is connected to a central location:

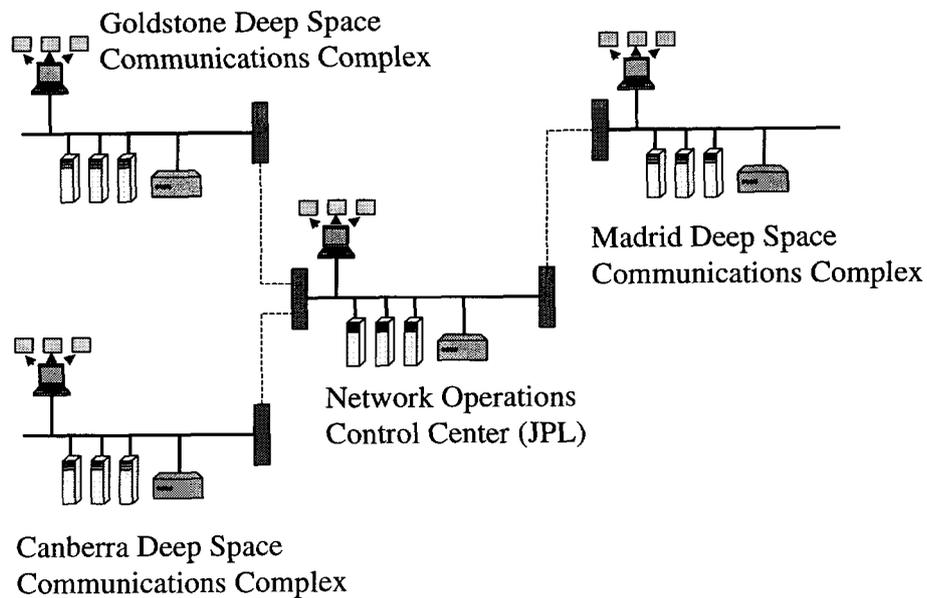


Figure 4 - NMC's connected for remote monitoring

This configuration allows services to be exchanged between NMC systems such that an operator at the Network Operations Control Center at JPL can view monitor data from the Madrid complex.

FEATURES

Reliability – operations is maintained if a server process were to fail. Server pools can be defined such that clients can connect to an available server without interruption. This is accomplished via client caching of all subscriptions, mappings and activities.

Scalability – whether it’s to support eight concurrent tracking activities at a site or just one, NMC can scale to the appropriate needs with a full server complement or “NMC in a box” configurations, where all NMC software components are run on a single machine. Software is designed so that the constraints are reduced to hardware traits including CPU and network performance (I/O cards, configuration, etc.). More importantly, an upgrade path is available by allowing the option for faster, more powerful machines.

Interoperability – the client library (currently supporting Solaris 2.5.1 and 8 but easily portable¹) allows applications running on different operating systems or hardware platforms to communicate with the NMC allowing new equipment to easily integrate itself. This also allows automation software a means of interfacing and providing monitor and control. Translators can be built using the API to provide communications between new equipment to talk to legacy equipment.

Performance – to maintain performance, server applications are threaded per client and database locking is managed so that only the required data sets within the servers are locked for access. This provides multi-client access to services. Queue processing algorithms that filter on stale monitor data can effectively keep processes from backlogging and ensuring timely delivery of the “latest” data.

Manageability – tools are provided for viewing the stability and configuration of the various servers. An API is also provided for further accessibility and control.

LESSONS LEARNED

Some lessons were learned during the development and testing of the infrastructure software.

- Develop with performance measuring and analysis in mind – in designing infrastructure, plan for performance measuring and analysis, which should be used in evaluating performance to load conditions and the robustness of the infrastructure. Develop software with hooks into viewing available resources. Many available CPU and/or network measuring tools can be used to indicate inefficient resource usage or bottlenecks in data flow.
- Protect servers from misbehaving clients – it is difficult to determine how a client will behave. An unchecked client can conceivably produce high activity and hoard resources from other clients. Incorporate defense mechanisms like timeouts and disconnect algorithms to protect servers - maintaining accessibility to all its services.
- The network is a limited resource – stale data should be filtered before sending over a network. If using TCP/IP, beware sending small packets that collect on overhead. In addition, many small packets have ways of provoking other ill behaviors like interpacket gap problems between networking hosts or causing packet collisions on half-duplex networks. Coalesce data into larger chunks and enable the Nagle algorithm.
- Beware of thread broadcasting – threads are meant to efficiently use CPU to handle many different tasks simultaneously, but avoid unnecessary broadcasting calls that can accumulate CPU cycles needlessly.
- Wide Area Networks (WAN) behave differently from Local Area Networks (LAN) - Wide Area Network (WAN) behavior is never as reliable as Local Area Networks (LAN). That is, there are multiple points of failure in a WAN that can cause disruption in service. Thus, WAN communication

¹ The client library is only dependent on standard threads and sockets, which are available on many platforms

needs to be developed differently than LAN communication. Heart-beats and timeouts with reconnect algorithms should be employed to handle lengthy outages or routing errors.

- Be wary of 3rd party library components – libraries of constructs that are intended to make development life easier can ultimately be a maintenance and portability problem if they become unsupported. Choose products wisely and that adhere to standards so that they can be replaced easily if necessary.

STATUS

Currently, the NMC is being utilized for 24/7 operations at the three main DSN complexes supporting most if not all the major spacecraft managed by JPL. The Goldstone Deep Space Communications Complex, the most populated with antennas of the three complexes, will use NMC to monitor and control six of its antennas simultaneously on a daily basis. An NMC system has recently been installed at the Network Operations Control Center (NOCC) at JPL in Pasadena, California to provide remote monitoring of the three complexes. A scaled-down version of NMC is being exercised in a Compatibility Test Trailer for monitor and control on the move. The infrastructure is supporting current automation capabilities at all complexes, in addition to supporting new automation technologies currently in development. Current NMC plans include providing support for new operation concepts as tracking multiple spacecraft per antenna.

ACKNOWLEDGMENTS

The author wishes to acknowledge and thank the contributors and implementers of the software: John Angione, Erik Barkley, Robert Balkenhol, Eva Bokor, Geoffrey Coward, Shirley A. Cizmar, Emily Law, Daniel Magallon, Ronald E. Norman, Mike Pester, Steve Stedry, Doug Stones, D'Arcy Tyrrell, Joanne Wisniewski, operations and engineering at Canberra DSCC, Goldstone DSCC, Madrid DSCC, and NOCC. This paper represents their incredible dedication and expertise.

REFERENCES

- Bokor, Eva, Bryan Camilli, Manuel Gomez, Patrick Olguin, Paul Pechkam, Patricia Santos, Marla Thornton, "Software for Monitoring and Controlling DSN Operations," NASA Tech Brief, Vol. 25, No. 6, 2001.
- Buschmann, Frank, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, "Pattern-Oriented Software Architecture, Volume 1: A System of Patterns", John Wiley & Sons, 1996.
- Graham, Ian, "Object-Oriented Methods: Principles and Practice", Addison Wesley Pub. Co., December 2000.
- Rosenberry, Ward, David Kenney, and Gerry Fisher, "Understanding DCE", O-Reilly & Associates, Inc. May 1993.

Reference herein to any specific commercial product, process or service by trade name, trademark manufacturer or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

The work described here was carried out at the Jet Propulsion Laboratory, California Institute of Technology under contract with the National Aeronautics and Space Administration.



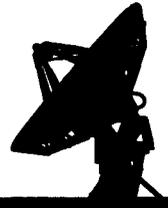
Monitor and Control Software for Ground Systems in the Deep Space Network

Paul Pechkam

July 25, 2002



Introduction



■ Objectives

- Describe the requirements for monitor and control in NASA's Deep Space Network
- Introduce Network Monitor and Control (NMC) software and its components
- Focus on infrastructure
 - design concepts used
 - infrastructure components
 - some lessons learned

■ What is infrastructure?

- Underlying software/hardware foundation of a monitor and control system



Deep Space Network



- The Deep Space Network (DSN)
 - international network of spacecraft tracking complexes
- Each complex is a self contained site of operations
 - pool of antennas; simultaneous tracking (limited to number of antennas) of multiple spacecraft
 - shared ground equipment including transmitters, receivers, telemetry processors, etc.
 - complex-to-complex hand-over for continuous tracking of a single spacecraft
- Remote monitoring
 - monitoring of complex activity available at the Network Operations Control Center at Jet Propulsion Laboratory



Deep Space Network (cont.)



- Operations - typical duties (some are automated) include:
 - allocating equipment from a pool of resources for a track
 - configuring equipment
 - monitoring equipment during track via graphical displays or access to raw monitor data
 - post-track cleanup (stowing the antenna, packaging spacecraft data)
 - releasing equipment for next support
- Operations require an interface to equipment with the ability to control
- Operations require access to monitor data and event messages produced by equipment for monitoring status and health



Infrastructure Requirements



■ Why infrastructure is important

- Increasing number of spacecraft supports and limited resources require reliable operations 24/7/365
- New spacecraft and/or new technologies promote new types of ground equipment that need to be integrated
- Additional load equals more data processed and flowing
- Accessibility and visibility to all ground equipment is required for monitor and control in any form of automation. Data examples include:
 - equipment health
 - signal health as reported by telemetry or signal receivers
- Different tracking configurations such as antenna arraying and/or multiple spacecraft per antenna require increased communication between equipment and feedback back to operations and automation



Building Infrastructure



- Network Monitor and Control (NMC) software uses a network architecture of servers and user workstations
- Good infrastructure will provide
 - manageability
 - reliability
 - performance
 - scalability
 - interoperability
 - security (if needed)
- Hardware
 - off-the-shelf network hardware including switches and routers to interconnect Sun servers and workstations and establish communication with ground equipment



Infrastructure (cont.)



■ Service oriented

- Name services - provide a means for clients to reach other clients
- Functional name services - allow for data representation via formatted names
- Monitor data services - provide for monitor data accessibility
- Messaging services - provide a means for clients to send/receive messages to/from other clients
- Event Notification services - provide event notification and the ability to trigger responses according to specific events
- Control services - provide a means for sending and receiving commands with proper format
- Connection services – allocating and managing resources for an antenna



Infrastructure (cont.)



- Client-server methodology
 - Ground equipment are represented as clients; data recipients (displays, automation software) are also represented as clients
 - Transaction based processing
 - reliable processing of request
- Data flow control
 - publish/subscribe methodology for monitor data and events
 - data is only sent on a network if subscribed to
 - allows for procedures to be triggered on the client when subscribed data is received
 - advanced queue processing
 - prevents backlogging of data



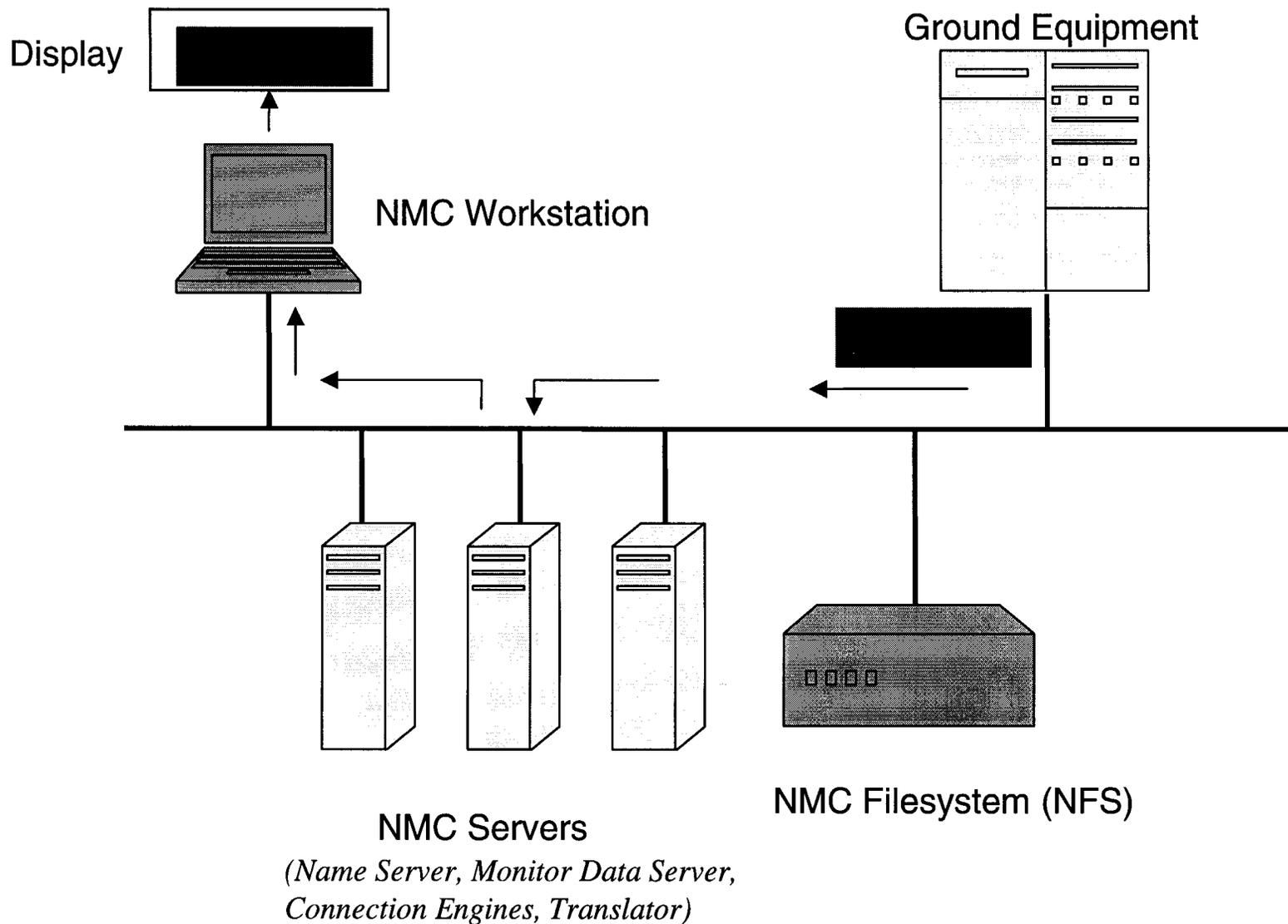
Infrastructure (cont.)



- Base software components
 - Name Server
 - Monitor Data Server
 - Client library and Application Programming Interface (API)
 - contain client-server and client-client protocols
 - Shared file system - based on NFS
- Additional components for managing ground equipment
 - Connection engine/server(s) - a server process representing an antenna and its configured/assigned equipment.
 - Translator - a server process used to translate legacy protocols
 - User Interface - suite of graphical displays for real-time monitoring and control of assigned equipment

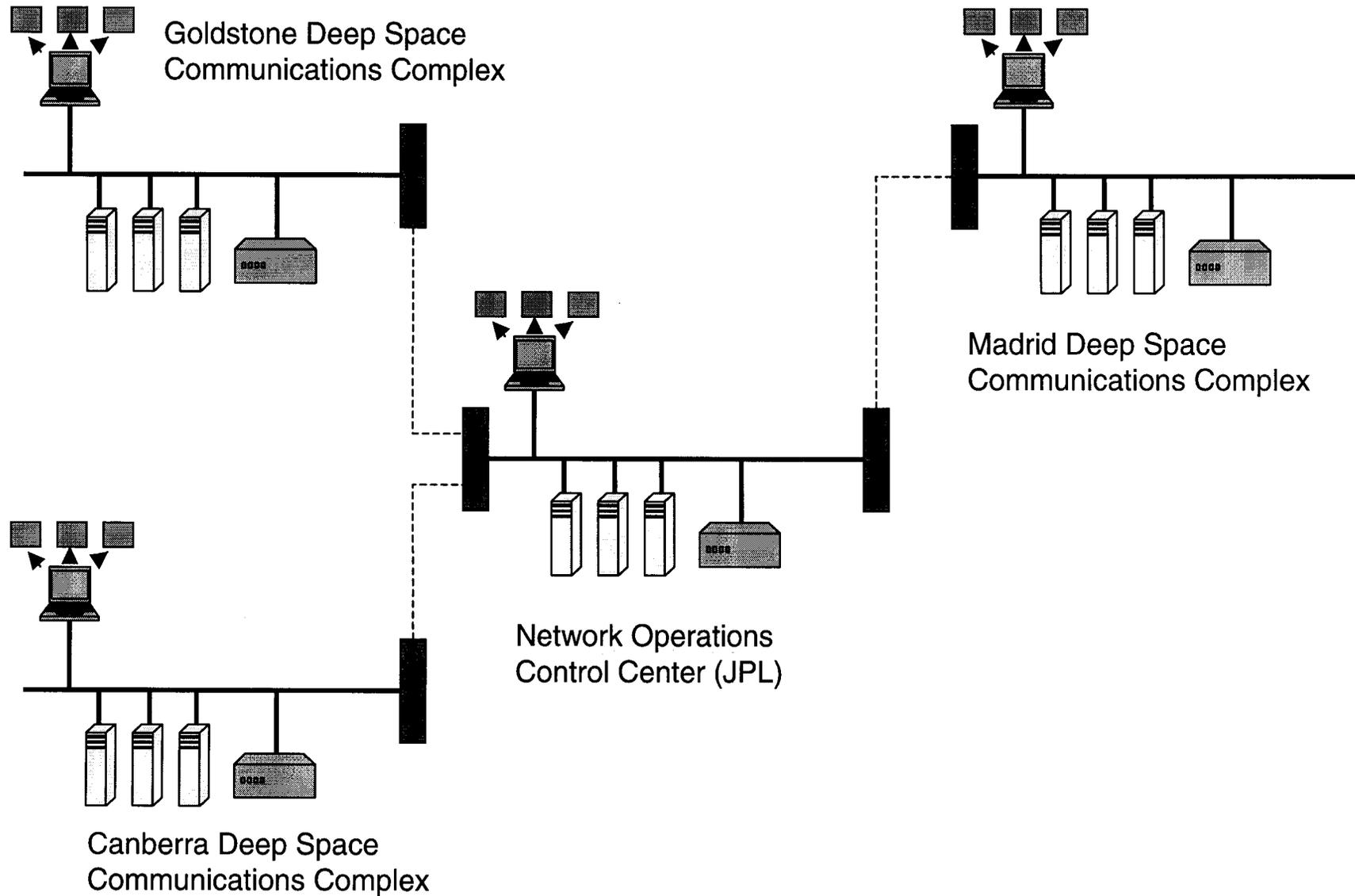


NMC in a Deep Space Communications Complex





Distributed NMC's in the Deep Space Network





Achieving Performance



- **Server optimizations**
 - Thread-based for parallel access to services and data
 - Identify and discard stale data
- **Client optimizations**
 - Thread-based to allow for parallel event-driven applications
- **Use existing protocols and accepted standards**
 - compatibility and portability
 - POSIX
 - TCP/IP as a reliable protocol



Performance (cont.)



■ Obtaining scalability

- server processes are network processes and can run on one host or distributed on many hosts

■ Obtaining reliability

- fail-over capability
 - client-side caching
- server pools



Lessons Learned



-
- Plan for performance measuring and analysis
 - Protect servers from non-nominal client activity
 - The network is limited resource
 - Avoid inefficient thread models
 - A Wide Area Network (WAN) is not as reliable as a Local Area Network (LAN)
 - Avoid server dependencies on shared resources e.g. shared filesystem



Conclusion



- Software and hardware infrastructure is an important part of ground communications (in the DSN)
 - Allow operations to worry about the spacecraft only
 - Ready for new equipment and/or operational scenarios
 - Automation relies on data availability and feedback from equipment
 - Automation can better predict the environment with increased data access and evaluate decisive action
 - Numerous applications beyond spacecraft tracking
- NMC is operational and supporting live spacecraft tracking