

A Technology Roadmap for Enterprise Connectivity to Control Networks

D. Byron, D. Gaw, E. Koch, A. Marsh, A. Schechter
Coactive Aesthetics, Inc.
4000 Bridgeway, Suite 303
Sausalito, CA 94965

ABSTRACT

In several of the industries to which LonWorks technology applies, system designers and users have identified a need for connectivity between the control network and other higher-level enterprise networks and systems. This connectivity can serve any of several purposes including:

- linking control network subsystems in large networks
- information flow from low-level systems to the corporate information network
- supervisory control originating from other enterprise networks
- connection to Wide Area Networks (WANs)

To gain maximum leverage from existing tools and standards, users are requiring that the enterprise connectivity problem be solved using open solutions. This paper describes some of the requirements for enterprise connectivity to control networks, and explores the key standards and technology options for implementing this capability.

INTRODUCTION

Increasingly in both information and control networks it is critical to leverage existing standards and technologies to implement a system. This is being driven by both the increased complexity of systems being developed and the increased importance of sharing information between different processes in the plant, business, or building. Solutions to the particular problem of connectivity between control networks and information networks can benefit greatly by leveraging the many standards and technologies extant in the information processing world.

In this paper we present examples of the value of connectivity between control and information systems, explore the requirements of such connectivity, and present some of the standards and technologies that can be used to create connectivity solutions. There are of course many ways to solve the connectivity problems we discuss here; we have limited our discussion to only those solutions comprised of open standards and technologies. Using open solutions ensures maximum value and flexibility for the user over the lifetime of the system.

WHY BOTHER ?

One may ask why connectivity between control and information systems is really required. The high-level answer to this question is the value of benefits such as:

- increased quality control
- energy savings
- reduced downtime through preventative maintenance
- centralized, high-level supervisory control
- improved production efficiency (reduced production costs, increased reconfigurability)
- improved management decision-making via more timely information

Achieving these benefits requires global, often detailed information to be transferred to or from the system being controlled or managed. It also requires that this information be gathered in an efficient manner so that the benefits are not outweighed by the costs of acquiring the data. Connecting the existing "islands of automation" to each other and with the rest of the enterprise is the technical challenge that will yield the above benefits.

LOGICAL vs PHYSICAL ARCHITECTURE

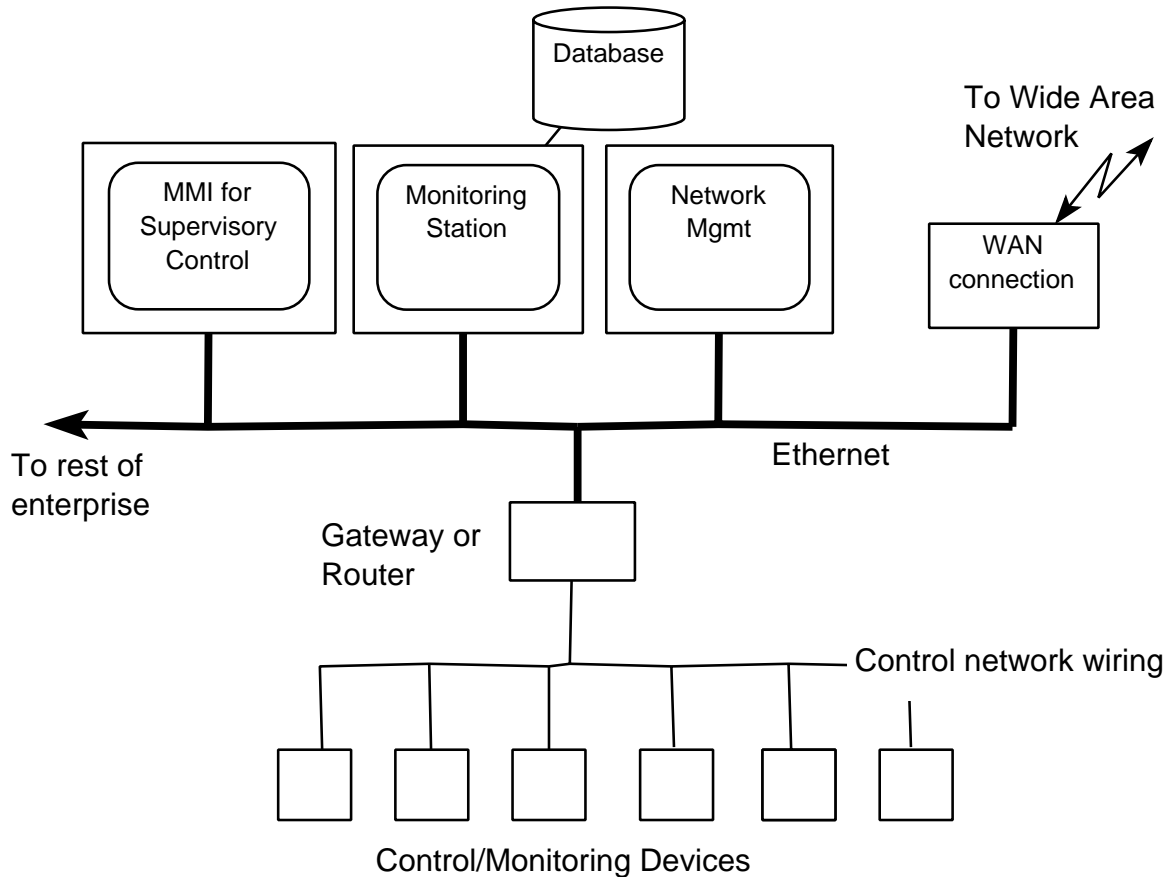


FIGURE 1: Standard Physical Architecture for Integrated Information/Control Network

Figure 1 shows the standard physical architecture for an integrated information and control

network. This sort of diagram is often presented as *the* architecture of the system. However, this is only the physical architecture. What is perhaps more important is the *logical* architecture which shows how the information flows and how the major software components and processes of the system fit together. Having a clear understanding of the logical architecture of the system enables designers to make better choices about standards and technologies which can be brought to bear on the problem.

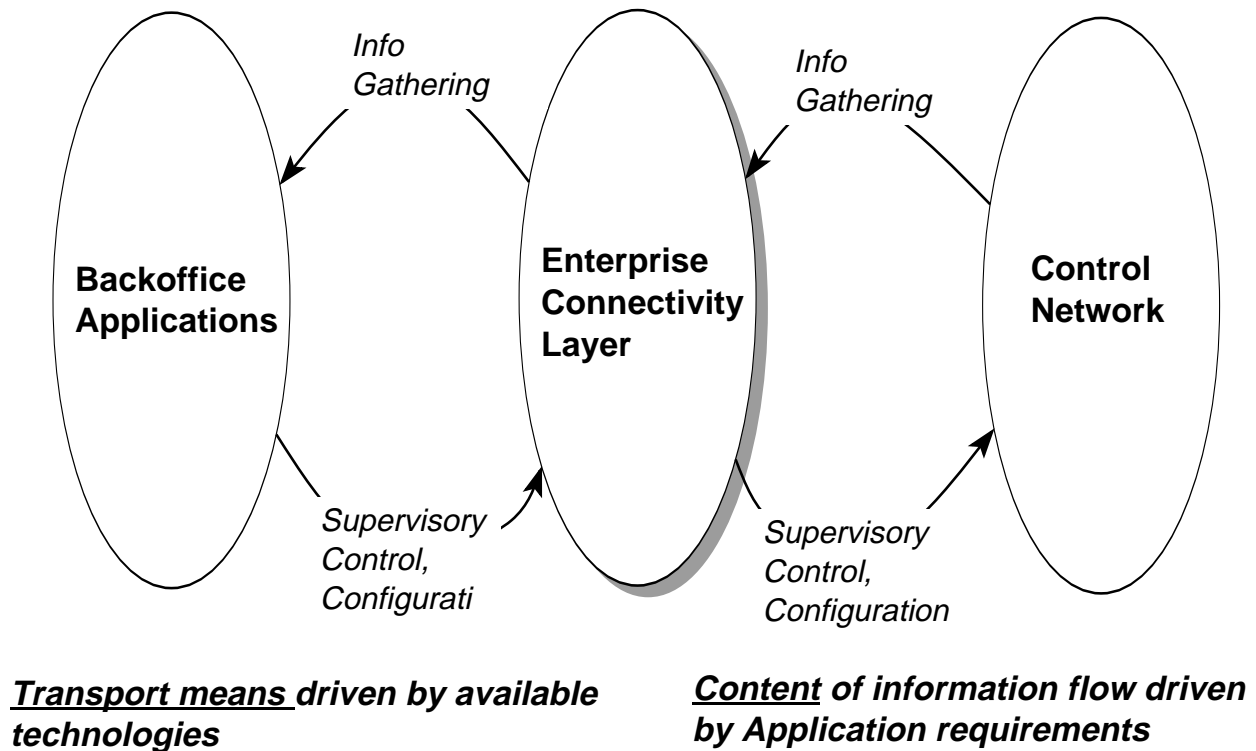


FIGURE 2: Logical Architecture of Information/Control System Connectivity

Figure 2 shows one possible logical architecture for an integrated information/control network. The Control Network component refers to the entire network of intelligent devices performing the core monitoring and/or control functions. This component should be familiar to developers of LonWorks systems.

The Backoffice Applications component refers to the entirety of the enterprise information systems. This may include accounting systems, high-level operator stations, management information systems, production control systems (e.g., for scheduling factory machines), and other components of the traditional SCADA system.

The middle component, the Enterprise Connectivity Layer, is often neglected or considered part of one of the other two systems. This connectivity layer, however, is the key to providing generic connectivity between the newer control networks and the mostly existing backoffice applications.

The connectivity layer is responsible for presenting information from the control network in a form that is understood by the backoffice applications.

The architecture in Figure 2 is centered around the Enterprise Connectivity Layer. In a very simple connectivity solution this layer may be nothing more than a protocol converter to connect the control network protocol to one known by the relevant backoffice applications. A DDE server could also be considered the connectivity layer in solving a range of simple connectivity problems. For more complex situations, the connectivity layer must provide an intelligent interpretation of the control network to the backoffice, as well as establishing sensible entry points for backoffice command and/or control.

APPLICATION REQUIREMENTS

To understand the requirements for a connectivity solution we must look at the reasons for providing connectivity in the first place. Below we review several types of application functionalities that drive various connectivity requirements and discuss how they relate to the high-level benefits of connectivity.

Trending

Trending for some is the most important long-term function available over networks. By measuring real-world information represented by analog signals, one can display and store valuable information. Whether it be temperature, pressure, level, flow, or any other analog signal (4-20 mA, 1-5 VDC, etc.), this information can be represented in a real-time visual or analytical format. By storing snapshots of this data, an operator can view a statistical curve reflecting changes of a particular variable (or variables) over time. Trending helps to attain benefits such as quality control and energy savings.

Status Logging

Status logging can be viewed as the digital counterpart to analog trending. Whereas analog trending is real-time based, status logging is monitoring the state of a particular variable at a particular time of reference. For example, one could be monitoring whether a particular piece of equipment was either 'On' or 'Off', or whether a particular variable was 'High' or 'Low', or what time a device went off-line. This status can be viewed as it is polled, or stored for future reference. The status logging function contributes to quality control, reduced downtime, and supervisory control.

Alarming

This function is consistent with either digital or analog signals. When a particular variable reaches a pre-assigned value, an alarm can be triggered. This can be as simple as setting off a horn, or as complex as triggering an entire new set of commands and functions. For example, if monitoring a temperature point, and the analog measurement reaches a predetermined set-point, a chiller may be turned on. Or, if the state of a digital variable changes, a horn can blast, and a light flash. Alarming information can also be stored for trending purposes with time and date recorded. The alarming function contributes to high-level supervisory control and

quality control.

Supervisory Control

This function is for the control of collected data. Operators track this data with their system, or take specified actions under certain conditions - actions such as modifying process temperatures, for example, or shutting down equipment that may be malfunctioning.

Supervisory control is the overall control of the various nodes on a network from one central location. This function drives the requirement for data to flow from the upper level network into the control network.

Accounting

This function can also be referred to as process management. Whereas Supervisory Control is management of all the subcomponents of a network, accounting would be specifically tailored to functions such as inventory control, WIP (Work In Progress), yields, cycle times, material tracking, costs, and quality control. Accounting contributes to increased production efficiency and improved management decision-making through more accurate production information.

Monitoring

Monitoring is keeping track of, regulating, or controlling the operation of processes in equipment, machinery, and networks. This can be as basic as monitoring a single temperature point via a digital input, or monitoring the overall condition of processes in a networked environment. This information can be displayed in a variety of formats, including real-time, polled, graphical, or others. Monitoring is a general function and contributes to all of the high-level benefits we have mentioned.

System Performance Evaluation

This function is the monitoring and evaluation of overall system performance. With this information, one can determine if the system performance is consistent with expected or desired results. For example, is this process being performed in a timely and profitable manner? Modifications can then be made to the system to improve performance as required. This function contributes to management decision-making, quality control, and improved production efficiency.

Network Management

Network Management is ideally "real-world" management of network functions and parameters. This includes troubleshooting any node over the network, analyzing networking functions and variables, and providing diagnostic tools, all from one central location. Other required network management functions include re-booting equipment, accessing all control ports from one location (e.g. one modem), and re-routing network traffic from the desktop. This contributes to reduced downtime by allowing preventative maintenance and facilitating quick troubleshooting.

APPLICABLE TECHNOLOGIES

The information processing industry has spent large amounts of money and resources developing methodologies that allow distributed applications to exchange information. The general paradigm that has been embraced for accomplishing this is referred to as client/server. This generally means that there is some application running which supplies information (server) to another application that is requesting it (client). These applications are typically running on different platforms which are linked by some type of network. This definition is quite general, and in fact there are a wide array of different methodologies that could be considered client/server. Such methodologies include everything from simple file servers all the way up to complex on-line transaction processors. Each methodology involves a set of standards which allow different vendors operating on different platforms to develop servers and clients that will be capable of exchanging information. In order for information to be exchanged and integrated with control networks, it will be necessary for the enterprise connectivity layer discussed above to adhere to many of the established standards in the client/server arena.

In addition to the client/server standards, which determine how information is exchanged at the application level, there are a large number of technologies referred to as "middleware". Middleware refers to a collection of technologies that provides the glue to enable different computers operating on the same network to exchange information at a relatively low level. It includes a hierarchy of technologies that can be classified in the following manner:

(1) Media - This includes the low-level means used to exchange information packets, e.g. ethernet.

(2) Transport Stack - This is a protocol layer built on top of the media, e.g. TCP/IP.

(3) Network Operating System - This is a basic set of services that are required by any distributed environment, and includes such things as security, distributed file systems, and naming services.

(4) System Management - This is a set of utilities and services that are used to manage the distributed network and environment, e.g. SNMP.

(5) System Services - These are a set of applications that provide basic services to users and their applications, e.g. email.

Figure 3 lists some of the more well-known technologies/methodologies used in client/server architectures. To build up connectivity solutions from off-the-shelf standard components, it is important to know how these various standards and technologies fit together.

CLIENT
APPLICATION

MIDDLEWARE

SERVICES

SQL
RPC
EMAIL
ORB
HTTP

SYSTEM MANAGEMENT

SNMP
CMIP
DME

NETWORK O/S
(DCE)
(Netware)

Directory
Security
Distributed File
Messaging
RPC

TRANSPORT STACK

NetBIOS
TCP/IP
SNA
IPX/SPX

MEDIA

Ethernet
Token Ring
SDLC
ISDN

SERVER

FILE SERVERS

DBMS SQL

ON LINE TRANSACTION PROCESSING (OLTP)

OBJECTS OLE

CORBA

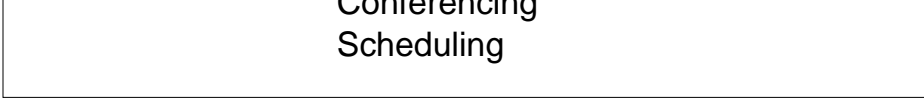
Open Doc

GROUPWARE Email

(Lotus Notes) Multimedia Documents (WWW)

Workflow

Conferencing



Conferencing
Scheduling

FIGURE 3: Technology breakdown in terms of Client/Server

Discussing all of these technologies is beyond the scope of this paper. Below are brief descriptions of those technologies which are most important for connectivity between the information and control systems.

DCE

The Distributed Computing Environment (DCE) is an integrated collection of technologies that allow applications to be created, used and maintained in a distributed environment. It is intended to operate in a seamless manner across disparate hardware platforms and networking transport stacks. It is a middleware package that forms a Network Operating System (NOS). DCE was developed by the Open Software Foundation (OSF) and first released in 1992. It provides all the basic services of a NOS including:

- Distributed file service
- Distributed time service
- Directory and naming service
- Remote procedure calls
- Threads
- Network security service

Almost all major computer and operating system vendors are either currently or have announced plans to support DCE.

DCE is probably the top candidate for a long-term solution to enterprise connectivity in a wide range of applications. It may not be appropriate for connectivity to embedded devices.

DBMS

Database Management Systems (DBMS) are collections of information that can be accessed and updated from client applications. The standard method for accessing and updating information in these databases is through the use of the Structured Query Language (SQL), which was developed by IBM Research in the mid-1970s. A client application requests data and services (e.g. filtering) from a database server by submitting an SQL query. The database server responds to the client's request by returning data to the client and/or updating the DB in the requested fashion. The server is responsible for providing secured access to shared data. There is a very wide range of operations that can be performed on databases using SQL. Numerous standards have been adopted for SQL databases including SQL-89, SQL-92, and SQL3. Currently over 200 vendors offer SQL products on almost every conceivable platform.

DBMSs are often a core part of any connectivity solution. Many SCADA packages support

archiving historical and alarm data to standard DBMSs. The standardization of SQL makes a DBMS a good candidate to implement the Enterprise Connectivity Layer described above, since many applications have SQL interfaces.

GROUPWARE

Groupware is a somewhat ill-defined catch-all term for a collection of technologies that are designed to increase the interaction and throughput of individuals working within an organization. It includes the following general categories of applications:

- Multimedia documents such as HTTP
- Workflow, which is used to automatically route events and work from one application to another
- Email
- Conferencing, which includes everything from electronic bulletin boards and chat rooms to real-time video and audio using such packages as IBM's Person to Person or FutureLabs' Talkshow
- Scheduling

Being a collection of technologies, there is no distinct history of groupware, although Lotus Notes could be considered a package that attempts to integrate many of the above technologies into a single operating environment.

Groupware is not likely to be part of the actual connectivity solution, but may drive the choice of standards so that data captured from the control network is transmitted/stored in a form compatible with the Groupware solution being used.

OLTP

On-line transaction processing (OLTP) is the technology that coordinates the activities of disparate applications and their access to shared databases. The core component of OLTP is a TP Monitor that is responsible for managing a transaction from its point of origin (the client) through the one or more servers and back to the client. It is difficult to create an exact definition of a transaction but it can be said to be a collection of database accesses and data manipulation. A transaction can request data from numerous sources (using SQL), and pass the information to other applications to be processed. The TP Monitor will manage the entire transaction process to ensure that it either completes successfully or that any operations already performed as part of a failed transaction are rolled back to restore any DBMSs to their original state before the transaction was submitted. Currently there are standards efforts by X/Open to create a consistent interface for all transactions, but there are numerous TP architectures and models offered by various vendors.

OLTP is not likely to be part of the connectivity solution.

SNMP

Simple Network Management Protocol (SNMP) is a protocol designed to communicate network management information. It lies on top of a transport protocol such as TCP/IP or (usually)

UDP/IP. It is part of the OSI Management Framework established by the International Standards Organization (ISO) in the mid-1980s.

The SNMP protocol supports four operations: get, get/next, set, and trap. These are asynchronous commands that allow a management application to gather or disseminate information across a network. An "agent" (really a specialized server) executes on SNMP-enabled nodes on the network and services these operations. While SNMP was originally designed for network management, its generic design and built-in support for such concepts as alarms make it an attractive option for other applications such as supervisory monitoring of a control network. The SNMP protocol includes a mechanism for extending the capabilities of agents.

SNMP is widely used in the data networking industry to manage devices such as routers and bridges. Commercial implementations of agents for embedded platforms are widely available.

CORBA

Common Object Request Broker Architecture (CORBA) was created as an open object infrastructure by the Object Management Group (OMG), a standards consortium supported and endorsed by the International Standards Organization (ISO). Object interfaces are described in Interface Definition Language (IDL), which is based on and similar to C++.

Information flows between distributed objects or "components" by an object making a request to a CORBA-compliant Object Request Broker (ORB). The ORB addresses the request to the appropriate target object, regardless of whether the target object resides within the same process, within a separate process on the same machine, or on another machine on the network. This is similar to a Remote Procedure Call (RPC), except that the call is made on a specific object which encapsulates its own data.

The latest version of CORBA, CORBA 2.0, was released in December 1994; CORBA 1.1 was released in 1991. CORBA is a generic abstract standard for distributed objects, and is not designed for a specific application (e.g. OpenDoc and OLE/COM are specifically designed for compound documents). As a standards definition, CORBA is not specific to any industry, platform, or company. It is specifically NOT supported by Microsoft, which supports the competing OLE/COM. Component technology in general is in its infancy.

To the extent that enterprise solutions are moving to object technology, this is also a strong candidate for the long-term connectivity solution.

OLE

Object Linking and Embedding (OLE) , and Network OLE, also known as the Common Object Model (COM), is Microsoft's standard for distributed objects. The two together are referred to as OLE/COM. An OCX is a component in an OLE compound document. OLE/COM is not CORBA-compliant and is supported only by Microsoft.

In information flow in OLE/COM, COM acts as an ORB, except that it is currently limited to a single machine. With the release of Microsoft's Cairo (late 1996?), COM is supposed to provide local/remote transparency and become "distributed COM". OLE objects are not object-oriented in the traditional way, in that there is no real inheritance, instantiation, etc.; they are more like groups of related functions.

OLE/COM is Microsoft's standard for distributed objects, and is only supported by Microsoft. OLE 1 was released in 1990 as a DDE-based way to open sub-components in compound documents. In OLE 1 the sub-component simply opened in its own window under the appropriate application. OLE 2 was introduced in 1993 with COM, and forms both a distributed object service architecture and a compound document framework. OLE/COM is primarily designed for compound documents.

Within the Windows environment OLE/COM is a powerful connectivity option due to the growing number of applications that support it. However, it does not provide connectivity between different platforms as DCE or CORBA do.

OpenDoc

OpenDoc is a compound document oriented CORBA-compliant framework for the desktop competing directly with OLE/COM. OpenDoc is from Components Integration Lab (CI Labs), which was formed in September 1993 by Apple, IBM, Novell, Oracle, Taligent, SunSoft, WordPerfect, and Xerox (i.e. everybody but Microsoft).

Information flow in OpenDoc is as in CORBA, with compound document oriented features added such as a platform-independent container file format (Bento) and Uniform Data Transfer, which allows document storage, data transfer, drag-and-drop, copy-and-paste, and linking operations to be performed using the same method invocations. OpenDoc uses a CORBA-compliant ORB from IBM called System Object Model (SOM).

IMPLEMENTING THE ENTERPRISE CONNECTIVITY LAYER: AN EXAMPLE

Figure 4 shows a sample implementation of an HTTP Control Network Server using the generic architecture described above. This system provides a flexible, powerful monitoring/control capability that can be run over both LANs and WANs connected to the control system. Standard Web browsers are used for graphical display of information, so that a user may open the appropriate URL from any location on the World Wide Web and view a HTML page that displays real-time information associated with the control system and allows the user to push buttons or make selections that effect control over this system. The only custom component of the system is the "Network Monitoring/Control Application" . All other components are standard off-the-shelf components. For all of the components there are multiple vendors of interchangeable products (e.g., HTML Browser, Network Interface, HTTP Server). The possibilities for this type of architecture are endless and represent many opportunities in next generation SCADA and remote monitoring/control applications.

In this system a custom application is written ("Network Monitoring/Control Application") which (a) monitors the network and writes out periodic status/data files in HTML format, and (b) responds to control requests from the HTTP server. The control requests are typically written to the Network Data/Status database. Alternatively, some servers also support various types of direct communication with an application("Direct Server/App Interface" in the figure).

Users access the information from standard Web (HTML) browsers over a LAN or WAN. The HTTP Server supports multi-user access to the control network information. The HTTP server and HTML documents are platform-independent and so can be accessed from any type of machine.

Typically the Control Network module and Enterprise Connectivity Layer would be running on the same physical computer. The Logical Network Interface is a high-level API to the control network. This component takes care of servicing the network and hides details of the control network protocol. The Physical Network Interface provides the actual connection between the control network and the computer hosting the Enterprise Connectivity Layer.

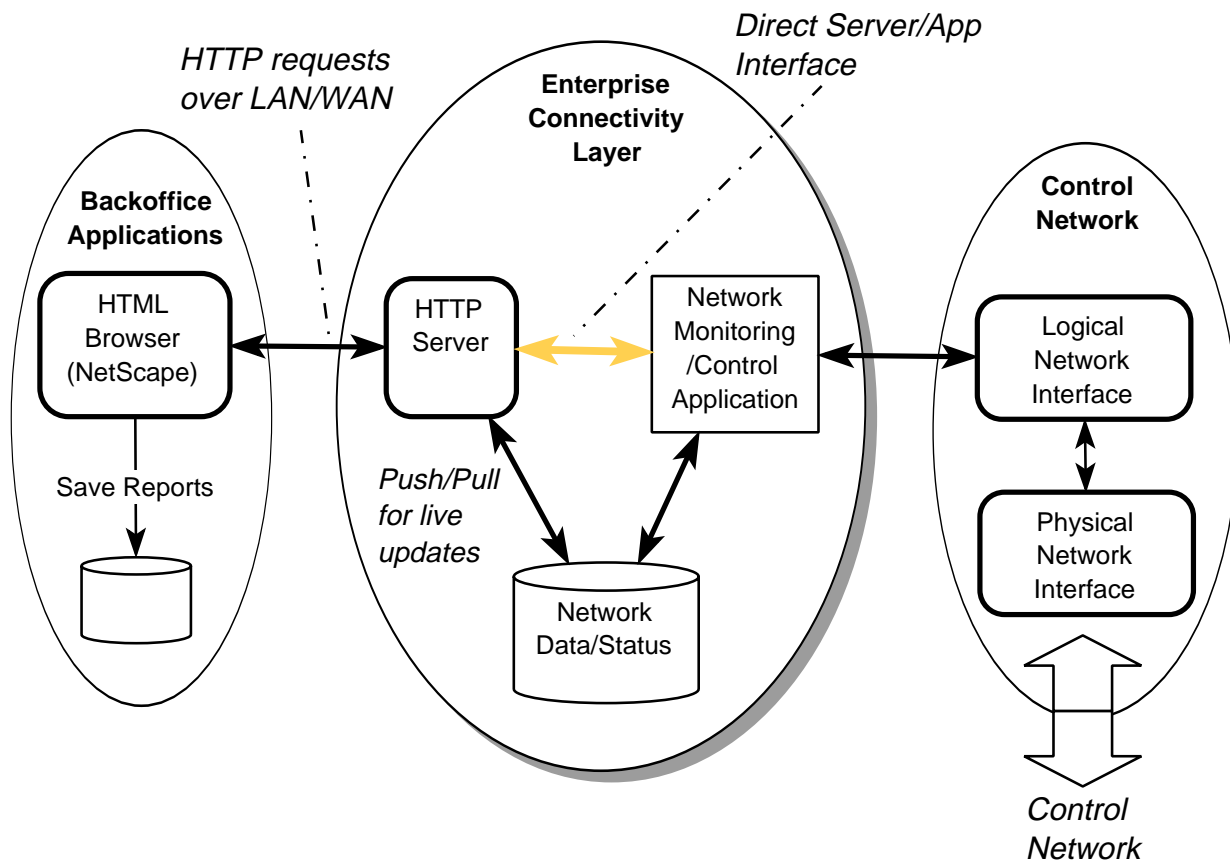


FIGURE 4: Sample Implementation of an HTTP Control Network Server

CONCLUSION

In solving the Information/Control network connectivity problem, system developers can benefit greatly by leveraging existing open standards from the data processing world. We have provided a very brief overview of some of the technology options that address this connectivity problem.

The Enterprise Connectivity Layer was introduced as a useful architectural component of the integrated information/control system. This component can be implemented with a combination of available standard components.

REFERENCES

"The Essential Client/Server Survival Guide", Robert Orfali, Dan Harkey, Jeri Edwards, John Wiley & Sons, Inc. 1994.

"The Essential Distributed Objects Survival Guide", Robert Orfali, Dan Harkey, Jeri Edwards, John Wiley & Sons, Inc. 1996.

"Distributed SCADA: From the plant floor to the executive suite", Pat Toole Jr. I&CS Magazine, Jan. 1996.