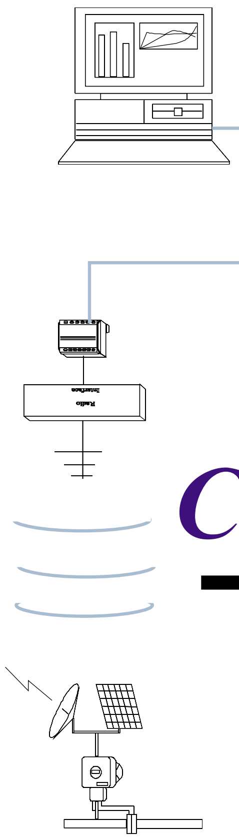
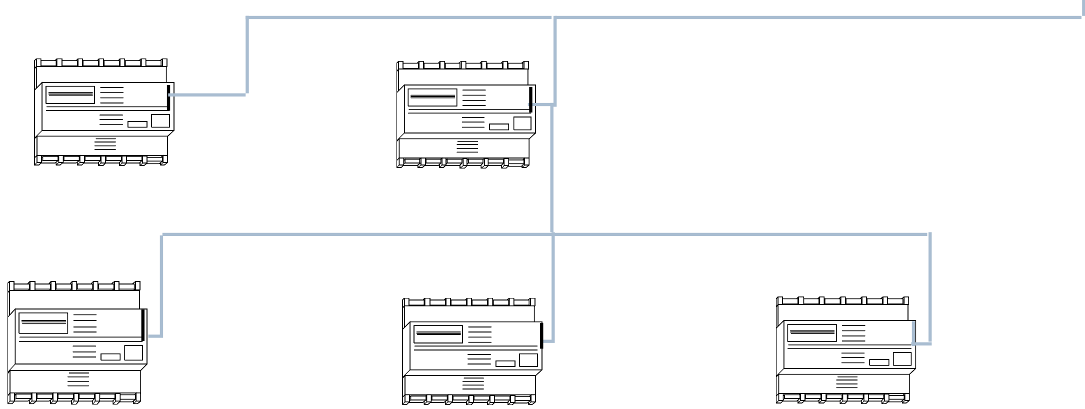


Network 3000 Communications Configuration Guide

(A user's guide to software configuration and troubleshooting for the Network 3000 Communications system.)



Bristol Babcock

D5080 Issue: April, 2001

Notice

Copyright Notice

Request for Additional Instructions

Additional copies of instruction manuals may be ordered from the address below per attention of the Sales Order Processing Department. List the instruction book numbers or give the complete model, serial or software version number. Furnish a return address that includes the name of the person who will receive the material. Billing for extra copies will be according to current pricing schedules.

A Few Words about Bristol Babcock

For over 100 years, Bristol® has been providing innovative solutions for the measurement and control industry. Our product lines range from simple analog chart recorders, to sophisticated digital remote process controllers and flow computers, all the way to turnkey SCADA systems. Over the years, we have become a leading supplier to the electronic gas measurement, water purification, and wastewater treatment industries.

On off-shore oil platforms, on natural gas pipelines, and maybe even at your local water company, there are Bristol Babcock instruments, controllers, and systems running year-in and year-out to provide accurate and timely data to our customers.

Getting Additional Information

In addition to the information contained in this manual, you may receive additional assistance in using this product from the following sources:

Contacting Bristol Babcock Directly

Bristol Babcock's world headquarters are located at 1100 Buckingham Street, Watertown, Connecticut 06795, U.S.A. Our main phone numbers are:

(860) 945-2200
(860) 945-2213 (FAX)

Regular office hours are Monday through Friday, 8:00AM to 4:30PM Eastern Time, excluding holidays and scheduled factory shutdowns. During other hours, callers may leave messages using Bristol's voice mail system.

Telephone Support - Technical Questions

During regular business hours, Bristol Babcock's Application Support Group can provide telephone support for your technical questions.

For technical questions regarding **ACCOL, Open BSI, or UOI call (860) 945-2286.**

For technical questions regarding Bristol's **OpenEnterprise** product, **call (860) 945-2501** or e-mail **openenterprise@bristolbabcock.com**

For technical questions regarding Bristol's **Enterprise Server®** / **Enterprise Workstation®** products, **call (860) 945-2286.**

For technical questions regarding **TeleFlow products call (860) 945-8604**.

For technical questions regarding **Network 3000 hardware products call (860) 945-2502.**

For technical questions about ControlWave call (860) 945-2244 or (860) 945-2286.

You can e-mail the Application Support Group at: **bsupport@bristolbabcock.com**

The Application Support Group maintains an area on our web site for software updates and other technical information. Go to: **www.bristolbabcock.com/services/techsupport/**

For assistance in interfacing Bristol Babcock hardware to radios, contact **Communication Technologies** in Orlando, FL at **(407) 629-9463 or (407) 629-9464**.

Telephone Support - Non-Technical Questions, Product Orders, etc.

Questions of a non-technical nature (product orders, literature requests, price and delivery information, etc.) should be directed to the nearest sales office (listed below) or to your Bristol-authorized sales representative.

Major U.S. Sales Offices

Watertown, CT (860) 945-2262
Birmingham, AL (205) 980-2010
Ontario, CA (909) 923-8488
Farmington, NM (505) 327-3271
Houston, TX (713) 685-6200
Richardson, TX (972) 238-8935

Major International Sales Offices:

Bristol Babcock Ltd (UK): (441) 562-820-001
Bristol Babcock, Canada: (416) 675-3820
Bristol Babcock Asia Pacific 61-8-9791-3654
BBI, S.A. de C.V. (Mexico) (525) 254-2131

Please call the main Bristol Babcock number (860-945-2200) if you are unsure which office covers your particular area.

Visit our Site on the World Wide Web

For general information about Bristol Babcock and its products, please visit our site on the World Wide Web at: **www.bristolbabcock.com**

Training Courses

Bristol Babcock's Training Department offers a wide variety of courses in Bristol hardware and software at our Watertown, Connecticut headquarters, and at selected Bristol regional offices, throughout the year. Contact our Training Department at **(860) 945-2269** for course information, enrollment, pricing, and schedules.

Who Should Read This Manual?

This document is intended to be read by a system engineer or technician who is configuring or trouble shooting Network 3000 communications. It assumes that controllers have been installed, ACCOL programming has been completed and the controllers have running ACCOL loads, and that all network cabling has been connected and tested.

NOTE: Although *some* of the information in this book is also applicable to ControlWave-series controllers, they are not part of the Network 3000 product line, and so will not be covered as part of this book.

This document assumes familiarity with the following subjects:

- Users must have some level of knowledge regarding how their Network 3000 hardware is to be used. In particular, they should be familiar with the data update requirements for their particular process. Users should also have access to whatever drawing or document specifies the addresses of each controller.

- Users must be familiar with using certain Open BSI Utilities programs such as the Open BSI Setup Tool (Open BSI 2.3 or *earlier*) or NetView (Open BSI 3.0 or *newer*), and DataView. For information on these subjects, see the *Open BSI Utilities Manual* (Ver 2.*x* document# D5076 or Ver 3.*x*/4.*x* document# D5081).

- Users may need to alter signal values, and certain other parameters in their ACCOL load as part of the Network tuning process. As such, some familiarity with ACCOL signals is helpful. See *An Introduction to ACCOL* (document# D4056) and the *ACCOL II Reference Manual* (document# D4044) for details.

- Users with Open BSI 2.3 (or *earlier*) may need to edit their Network Topology files using the NETTOP program. This subject is discussed in the *NETTOP and NETBC User Manual* (document# D4057). Users with Open BSI 3.0 (or *newer*) may need to edit their Network Definition (NETDEF) files using NetView, and specify RTU configuration parameters using LocalView. NetView and LocalView are discussed in the *Open BSI Utilities Manual* (Ver 3.*x*/4.*x* document# D5081).

Table of Contents

Introduction

Bristol Babcock's Network 3000-series of remote process controllers (DPC 3330s, RTU 3305s, 3530s, etc.) run ACCOL loads to measure or control some part of a process. Besides performing these control duties, each Network 3000 controller is designed to operate as a **node** in a data network.

The data transmitted through the network is sent according to a pre-defined format called a network **communication protocol**. There are two basic types of network protocols supported by Bristol devices.[1] **BSAP** (Bristol Synchronous/Asynchronous Protocol) is a Bristol Babcock proprietary protocol which is supported by all Network 3000 devices[2]. **IP** (Internet Protocol) is a more industry-standard protocol, and is supported by many types of computers world-wide. Currently the 386EX Protected Mode DPC 3330/DPC 3335 controllers with Ethernet hardware installed are the only Network 3000-series controllers which support the IP protocol. Bristol Babcock's ControlWave controller also supports IP, however, ControlWave communications are not covered in this book.

In addition to the choice between these two protocols, it is possible to have separate BSAP and IP networks as part of the same overall system, or to have a BSAP network serve as a **sub-network** of an IP node.

BSAP Networks

At the top of a BSAP network is a host computer, called the **network master**. The network master is usually a PC Workstation running **human-machine interface (HMI)** or **supervisory control and data acquisition (SCADA)** software such as Intellution® FIX®, Iconics Genesis, or OpenEnterprise software, in combination with Bristol Babcock's **Open Bristol System Interface (Open BSI)** software. The HMI/SCADA software at the Network Master allows the operator to view what is going on in the network through graphical displays, trends, or printed logs and reports.[3] Below the Network Master are the remote process controllers.

The controllers in the network are organized into a hierarchical structure of one or

---

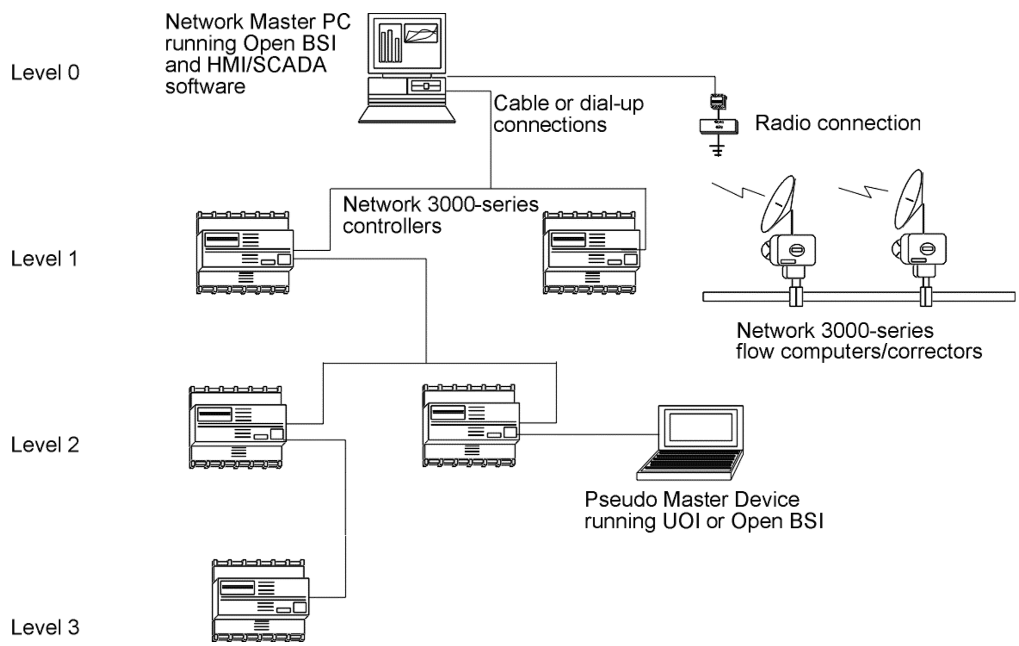
[1] *In addition to these standard types, certain Bristol controllers support 'custom' protocols for communication with various foreign devices, PLC's etc. See the ACCOL II Custom Protocols Manual (document# D4066) for details.*

[2] *Some non-Bristol devices can be made to communicate via BSAP, provided that special communication software programs called **communication drivers** are created for them.*

[3] *Pseudo Master devices can be connected to lower levels of the network to view data. These are similar to Network Masters, however, they are not considered to be 'nodes' in the network, and so do NOT appear in the NETTOP/NETDEF files.*

more **levels**. A BSAP network can support up to six levels (not including the Network Master referred to as level 0.) The number of levels required varies depending upon the size and scope of your project. Each controller is assigned an **address** to allow data to be routed correctly. Addresses and network structure are specified using either the **Network Topology (NETTOP) Program** or the Open BSI **NetView** program.[4] The network information stored internally by a node is called its **Node Routing Table**, and is updated whenever an ACCOL load is downloaded, or a valid time synchronization message is sent.

The level of a given controller specifies how many intervening **communication line**s there are between it, and the network master. The first level controllers are called **top-level nodes** because data must travel over only 1 communication line to reach the Network Master. A communication line can consist of a direct cable connection, a radio or satellite link, or a dial-up modem connection. Each communication line is configured independently with its own **baud rate**, **poll period**, **timeout**, etc.



Each controller (node) serves as a **master** to the nodes connected to it on the level immediately below, and as a **slave** to the node connected to it on the level immediately above. A node can have many slaves but only one master. Each master **polls** its slaves for data, which it retains in memory until it is polled by *its master*.

[4] *The choice of whether NETTOP or NetView is used to configure your network depends on the software in your system. If you are using DOS-based ACCOL Tools (AIC, ABC, etc.), or Open BSI Utilities versions lower than 3.0, then NETTOP must be used. If you have Open BSI Utilities 3.0 (or newer) then NetView must be used. If you want to use a mixture of older DOS-based ACCOL Tools with Open BSI 3.1 (or newer) you will need to use NetView, and then use the NETBCX utility to generate network files compatible with the DOS-based tools.*

In this way, data flows from slave to master, slave to master, etc. until it reaches the Network Master, where it is made accessible to the operator via HMI software. From a given node, direct **peer-to-peer communication** (using ACCOL Master/Slave modules) is only possible to its Master node, any connected slave nodes, and any siblings (nodes on the same level which share the same master). If communication is required to any node not in these categories, it must be routed up using Master/Slave modules at each individual level of the network, until it reaches either the Network Master, or a Master which is a sibling to another Master. The message can then be routed down, again, using Master/Slave modules at each level, until it reaches the desired node.

What Type of Data Gets Polled in a BSAP Network?

When a BSAP Master polls its slave nodes for data, it passes message traffic from other nodes in the network, and collects response messages from its slaves. The type of data collected includes alarms, RBE signal changes, audit trail, or array data, etc. The various methods of BSAP network data collection are described below:

Data Collection Method:	What is this used for?	What sort of configuration activities are required?
Alarm Data Collection	Collecting analog alarm and logical alarm signal data, when these signals enter or leave an alarm state.	No user configuration is required, other than declaring signals as alarms in the ACCOL load. Alarms will automatically be sent to the top-level node(s). Configuration may be required, however, within the HMI/SCADA package in order to retrieve the alarms.
Report By Exception (RBE) Data Collection	Collecting RBE logical signals which change status, or RBE analog signals which have a value change exceeding a specified deadband.	Signals must be declared as RBE signals. An RBE Module must be configured in each ACCOL load containing RBE signals, and expanded memory is required. Compatible RBE Manager software must be available for the particular Network Master HMI/SCADA software.

Data Collection Method:	What is this used for?	What sort of configuration activities are required?
Remote Data Base Access (RDB) Data Collection	Collecting signal, array, signal list, archive or audit trail data. RDB access is used by Open BSI, UOI, and other programs specifically written to use RDB commands. Often this type of collection occurs based on user request, i.e. calling up a DataView Window, etc.	Configuration varies depending upon the program being used.
Template Data Collection	Collecting signal data from all global signals, in the network, at specified intervals, and storing it in the Enterprise Server Real Time Data Base.	Users will need to assign signals to Scan Time Classes, configure the Real Time Data Base and Template Data Collection, create a Serial CFE Port in Top-Level Nodes, and reserve expanded memory in the top-level node for template data.
Peer-to-Peer Communication	Sending signal and/or array data from one controller, to another controller, so the other controller can perform some processing on it.	Users must configure Master/EMaster and Slave Modules in their ACCOL loads. NOTE: Master/EMaster Modules and Slave Modules should NOT be confused with Master and Slave communication described previously. They are different.

IP Networks

The 386EX Protected Mode DPC 3330 / DPC 3335 controllers with PES03/PEX03 or *newer* firmware support IP. ControlWave controllers also support IP, however, they are not part of the Network 3000 product line.

IP is a communications protocol which allows computers on *different* networks to exchange information with one another. IP is *also* the standard communication protocol used on the **Internet,** a constantly changing set of computer networks that millions of people around the world use for business, government, and educational purposes. The 'IP', in fact, stands for **Internet Protocol**.[5]

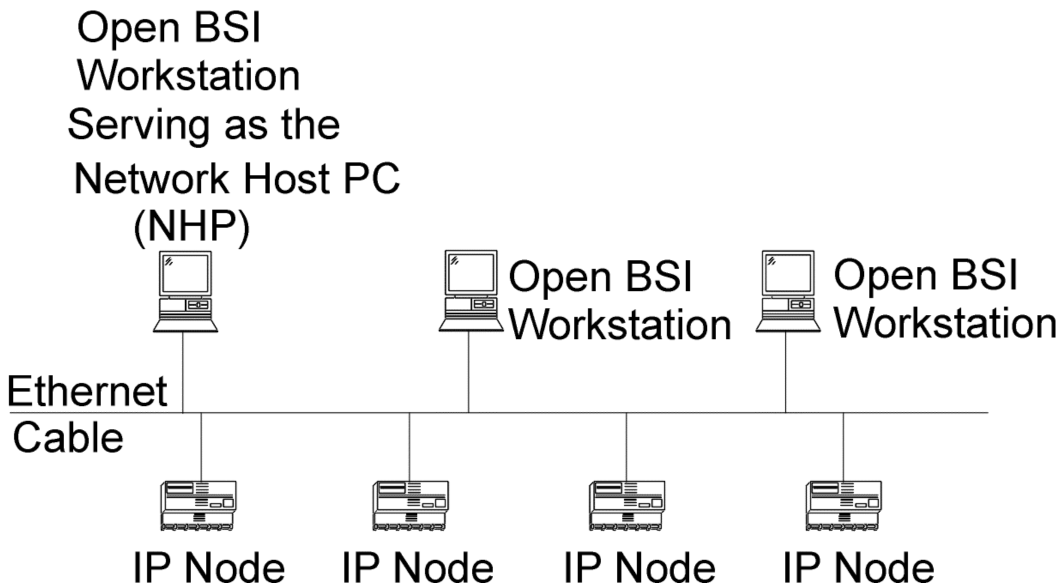
IP allows Network 3000 controllers to communicate using **Ethernet**, a standard type of **local area network (LAN)** originally developed by Xerox Corporation.

[5] *This manual will discuss certain aspects of IP as part of communications configuration for your Bristol Babcock controller. The details of how IP works are beyond the scope of this manual, however. One book on the subject we recommend is Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture by Douglas E. Comer, published by Prentice Hall.*

(Other types of LANs may be supported in the future, however, Ethernet is currently the only choice available.)

A Bristol Babcock network using only IP nodes does NOT follow a hierarchical structure. For certain applications, this is a significant advantage over BSAP, because all nodes in a given section of the network are essentially on the same level. This means that peer-to-peer communication is simplified; there is no need to use more than a single pair of IP_Client/IP_Server modules[6] to get a message from one node to any other node because all nodes are 'siblings' on the same level.

Information on the structure of the network is stored in a Network Definition File (NETDEF) at an Open BSI workstation called the **Network Host PC**, which is abbreviated as **NHP**. A program called **NetView** is used to create the NETDEF file.[7]



If an IP node or an Open BSI workstation needs to communicate with *another* IP node or Open BSI workstation, and it doesn't know the address of the IP Port for that node or workstation, it obtains the necessary addresses and routing information from the NETDEF file at the NHP.

---

[6] *IP_Client and IP_Server provide IP the same node-to-node data transfer capabilities as the Master and Slave modules do for BSAP networks.*

[7] *Although any Open BSI 3.0 workstation can potentially serve as an NHP, the 'host function' refers to Open BSI 3.0 workstations which contain detailed routing information (in their NETDEF file) for particular RTU's, and therefore are 'hosting' them. Open BSI 3.0 workstations which do NOT store detailed routing information for a particular RTU are considered to be* ***proxy*** *workstations with respect to that RTU because they initially establish communication with it by requesting access through that RTU's NHP.*

The concept of the NHP is easier to understand if you consider an analogy to the public telephone system. Most people remember a certain set of phone numbers for people they call frequently, but occasionally, they need to call someone whose number they don’t know, so they call directory assistance and ask for the correct phone number. The NHP performs the exact same function as the directory assistance operator; except instead of giving out phone numbers, it provides IP address information, on request, for connections to any node in its section of the network.

What Type of Data Gets Transmitted in an IP Network?

The type of data transmitted in an IP network includes alarms, RBE signal changes, audit trail, or array data, etc. The various methods of IP network data collection are described below:

Data Collection Method:	What is this used for?	What sort of configuration activities are required?
Alarm Data Collection	Collecting analog alarm and logical alarm signal data, when these signals enter or leave an alarm state.	Signals must be declared as alarms in the ACCOL load. Up to four different destination IP addresses may be specified for receiving alarm data from a particular node. Additional configuration may be required, however, within the HMI/SCADA package in order to retrieve the alarms.
Report By Exception (RBE) Data Collection	Collecting RBE logical signals which change status, or RBE analog signals which have a value change exceeding a specified deadband.	Signals must be declared as RBE signals in the ACCOL Load. An RBE Module must be configured in each ACCOL load containing RBE signals. Up to four different destination IP addresses may be specified for receiving RBE data from a particular node. Compatible RBE Manager software must be available for the particular Network Master HMI/SCADA software.
Remote Data Base Access (RDB) Data Collection	Collecting signal, array, signal list, archive or audit trail data. RDB access is used by Open BSI and other programs specifically written to use RDB commands. Often this type of collection occurs based on user request, i.e. calling up a DataView Window, etc.	Configuration varies depending upon the program being used.
Peer-to-Peer Communication	Sending signal and/or array data from one controller, to another controller, so the other controller can perform some processing on it.	Users must configure IP_Client and IP_Server Modules in their ACCOL loads.

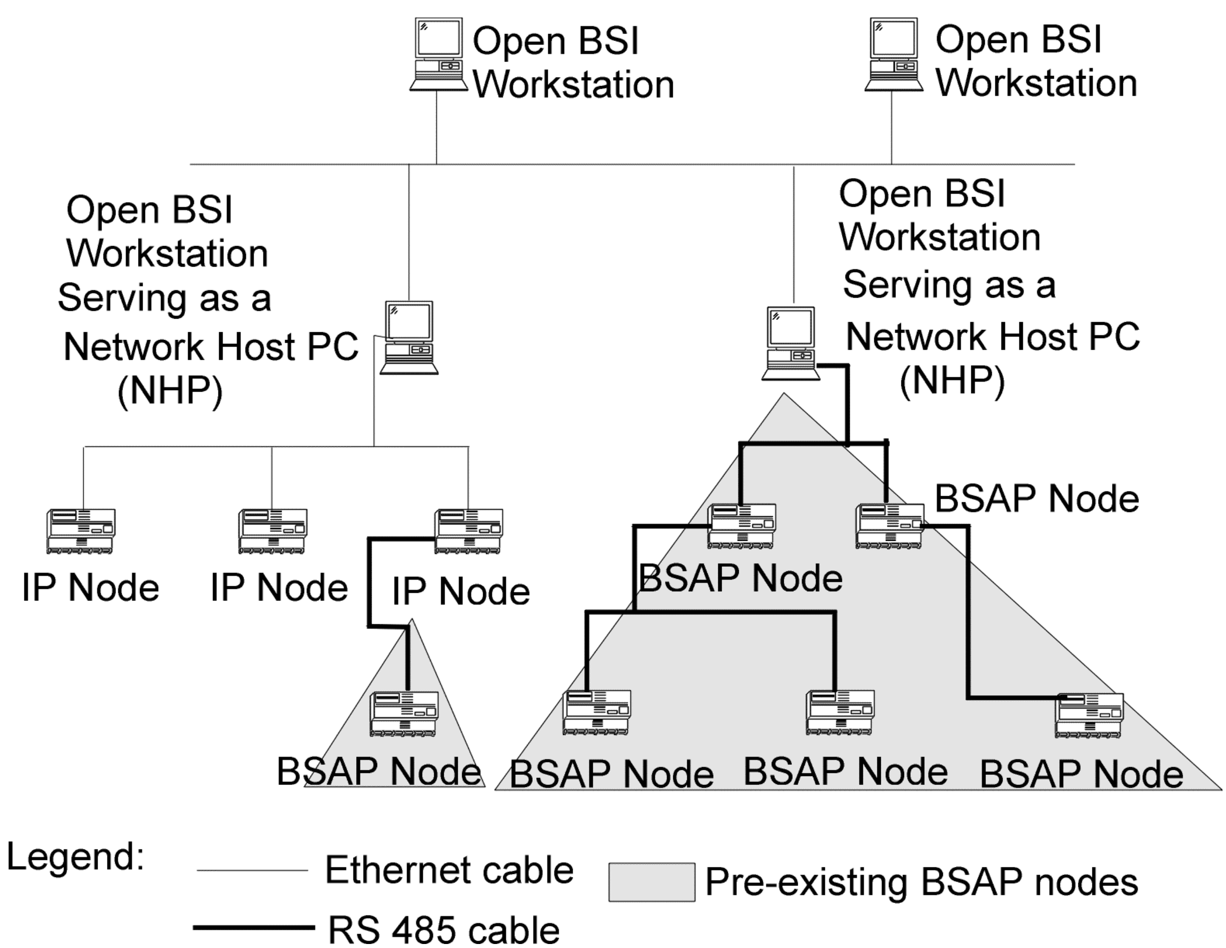
Converting NETTOP Files for Use Below an NHP

It is possible to add a Network Host PC (NHP) on top of an existing BSAP network, replacing the existing Network Master node. To preserve the network configuration data, the existing NETTOP files can be used to generate a single NETDEF (*.NDF) file, using the NETCONV utility.

Any BSAP nodes which are to exist underneath IP nodes (instead of NHPs) must be explicitly added in NetView, or added manually to the NETDEF file, using a text editor.

The figure, below, shows a typical configuration which mixes both IP nodes, and BSAP network(s).

Only the single BSAP node underneath the IP node would have to be manually defined in NetView; the other BSAP node definitions (under the NHP) would be extracted from the existing NETTOP file, and added to a new NETDEF file, using the NETCONV utility.



Overview of Configuration

Whether you are using BSAP, IP, or a combination of the two, the goal of configuring network communications is to achieve maximum throughput of data. To obtain this performance goal, however, the user must have an understanding of how Network 3000 communications work. We can think of the Network 3000 architecture as being made up of three major parts. These parts are the **data link**, the **network routing** information, and the **application**s which use the data link.

Each of these parts is configured independently, however, they all work together to allow communications to function. Failure to properly configure any of these parts can degrade system performance (slow down communication throughput) or can even prevent any communication from occurring. It is important, therefore, to understand how these three parts of the Network 3000 architecture are configured, and how they interact.

Data Link

The **data link** or communication line handles the actual physical transmission of data messages from node to node in the network. It includes the communication ports, as well as the connections between them such as cables, modems, or radios. It simply moves data messages; without regard to content of the messages. There are numerous software configuration parameters which control how the data link functions. These subjects will be discussed in detail in Chapters 4 and 5.

Network Routing Information

Network routing information is used to determine where data messages should be sent within the network. This is primarily determined based on entries made in the **NETDEF** (or **NETTOP)** file including node names, addresses, etc. These subjects will be discussed in detail in Chapters 2 and 3.

Applications

Applications are the programs or system functions which *use* the data link. They send messages out on the data link, and wait for responses to come back. Applications include Open BSI utilities such as DataView, associated SCADA or HMI packages such as OpenEnterprise, Intellution® FIX® or Iconics Genesis, as well as Master Module communications in the controller.

There are several things to be aware of when creating your ACCOL load, as well as certain application level parameters, which must be configured. These subjects will be discussed in detail in Chapter 6.

Is this still unclear?

If these concepts are still unclear to you, it might help to think of the communications system as analogous to using a postal delivery service. Let's say you want to order some merchandise by mail. You place a letter (data request) in an envelope, and you write an address on the envelope indicating where you want it delivered. (This is equivalent to defining network routing information.) Next you go to your local post office, and drop your letter in the outgoing mailbox.

The postman retrieves your letter, along with lots of other peoples' letters, and they get sent out through the postal system to many different post offices (nodes) in the network. The mechanism of transit could involve delivery trucks, trains, planes, etc. (This transportation system is similar to the concept of the data link, except instead of trucks and planes the data link uses modems, cables, and radios.)

Finally, if all goes well, the mail order company will pick up your letter, from its local post office, and will respond by sending back a package, addressed to you, containing whatever merchandise you ordered. The response might actually be sent as one package, or several packages at different times, if not all of the requested items were ready at one time. The post office then delivers the return package(s) to you. Both you and the mail order company are users (applications) which make use of the postal system (data link).

The actual time it takes to complete the full transaction may vary, somewhat. For instance, while the postal system might be able to guarantee that, under normal circumstances, your request will reach the post office nearest the mail order company within three days, there's no guarantee that the mail order company (application) will pick it up on time, that it will understand your request, or that it has the items you want in stock. This is entirely out of control of the postal system (data link).

Of course, if the postal truck carrying your letter gets into an accident (data link failure) your request might not get delivered at all, and so you will have to send another one. Similarly, if you write the wrong address (network routing information) on the outgoing letter, the postal system doesn't know the difference, and will try to deliver it to the non-existent address. Eventually, you'll receive a note from the post office saying that your letter (data request) was undeliverable.

If this analogy didn't clarify the distinction between the data link, application, and network routing information, don't worry. We'll talk more about these subjects in the next several chapters.

Chapter 2 - Guidelines For Setting Up BSAP Networks

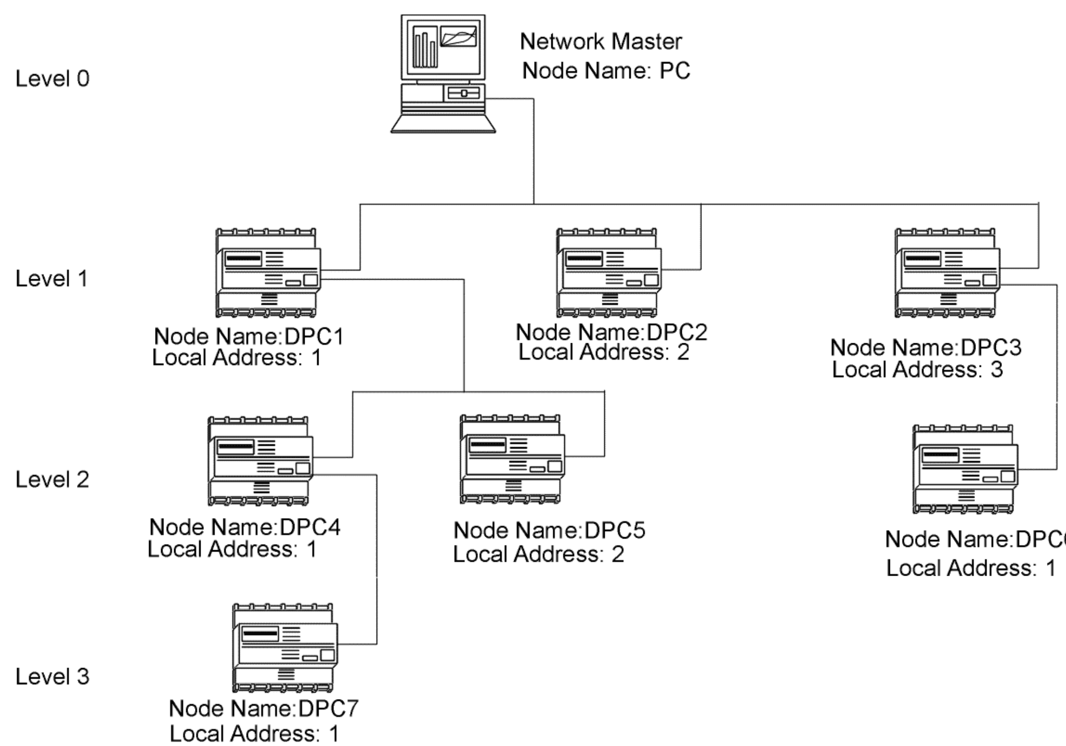
In this section, we will discuss, briefly, the terminology used in BSAP network configuration, and then go over some important items, which, if overlooked, can cause problems. For full details on using the Network Topology (NETTOP) program as well as the alternative NETBC and NETREV compilers, see the *NETTOP and NETBC User Manual* (document# D4057). If you are using Open BSI Utilities Version 3.0 (or newer) you MUST use the Network Wizard in NetView, *instead of* the NETTOP, NETBC, and NETREV programs. NetView is discussed in the *Open BSI Utilities Manual* (document# D5081).

What is BSAP?

BSAP stands for Bristol Synchronous / Asynchronous Protocol. All ACCOL-based Network 3000-series controllers can communicate using BSAP.

Each node (RTU) in a BSAP network is assigned a 7-bit local address (from 1 to 127) and a 15-bit global address, based on its location in the network. The local address you configure in the NETTOP or NETDEF files for a particular RTU must match the local address hardware switch setting (or FLASH parameter) configured at the RTU.

A Quick Overview of BSAP Network Terminology



Node Name:	Network Level:	Local Address:	This node is a master of:	This node is a slave of:
PC	0	Not applicable	DPC1, DPC2, DPC3	Not applicable
DPC1	1	1	DPC4, DPC5	PC
DPC2	1	2	Not applicable	PC
DPC3	1	3	DPC6	PC
DPC4	2	1	DPC7	DPC1
DPC5	2	2	Not applicable	DPC1
DPC6	2	1	Not applicable	DPC3
DPC7	3	1	Not applicable	DPC4

At the top of a BSAP network is a computer called the **network master**. Typically, this is a PC Workstation.[1] Going out through the communication port (e.g. COM1: or COM2) of the network master is a communication line. The communication lines could be direct connections, as shown in the drawing, or they could use modems (dial-up), or they could use radios. It doesn't matter for purposes of this discussion.

Below the PC workstation are controllers (**nodes**), each of which has a unique **node name**. The nodes are organized into a hierarchical structure of 1 or more **levels**. A node's level indicates how many communication lines there are between it, and the network master. The number of levels varies depending upon the requirements of your particular system, and how you decide to lay it out.

Any data message from one node in the network to another node in the network is considered to be **local** if there is only one communication line involved. If more than 1 communication line is involved, a message is considered **global**.

The nodes on level 1 are called **top level nodes** because they are directly below the network master. Another term for top level node is **first level slave**. The top level nodes, in turn, are called **master** nodes to the nodes immediately below them on Level 2. The Level 2 nodes are **slaves** of the top-level nodes, and are masters to any nodes connected to them on Level 3, and so on. NOTE: Any slave node can only have 1 master.

Up to six levels (1-6) are allowed in the network, not counting the Network Master on Level 0.

Each slave node *of a given master* is assigned a unique **local address**, which ranges from 1 to 127.[2]

[1] If this BSAP network is a sub-network of an IP node, the Network Master would be the Network Host PC (NHP), and the IP node would be the only level 1 node.
[2] A node's configured local address may be viewed on-line through the #NODEADR.. system signal.

Things to Be Aware of When Defining Your Network

Every node in the system, as well as software in the network master, and any pseudo masters (to be discussed later), depends on the data defined in NETTOP (or NetView) to route messages. It's essential that you plan out your network carefully, *before you attempt to define it in the Network Topology Program (NETTOP)*, or in NetView.

Ideally, you should define your NETTOP/NetView information once at the beginning of your system configuration, and then leave it alone. Although a small network may not be a major problem, if you have a large complicated network, and attempt to modify its layout late in the configuration process, you are likely to endure a heavy cost in time and effort. Such a change may require you to re-generate data bases, re-download ACCOL loads, etc.

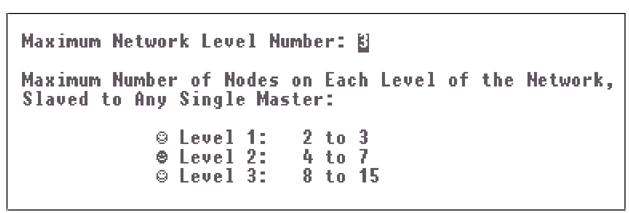
The best way to plan it out is to draw a picture of your network. You can draw one similar to the figure at the beginning of this chapter, except with simple boxes representing each node. You should show the communication lines connecting the nodes, the network levels, and each node's local address.

The following pages discuss some items to consider when setting up your network:

General Considerations - Applicable to Both NETTOP & NetView Users:

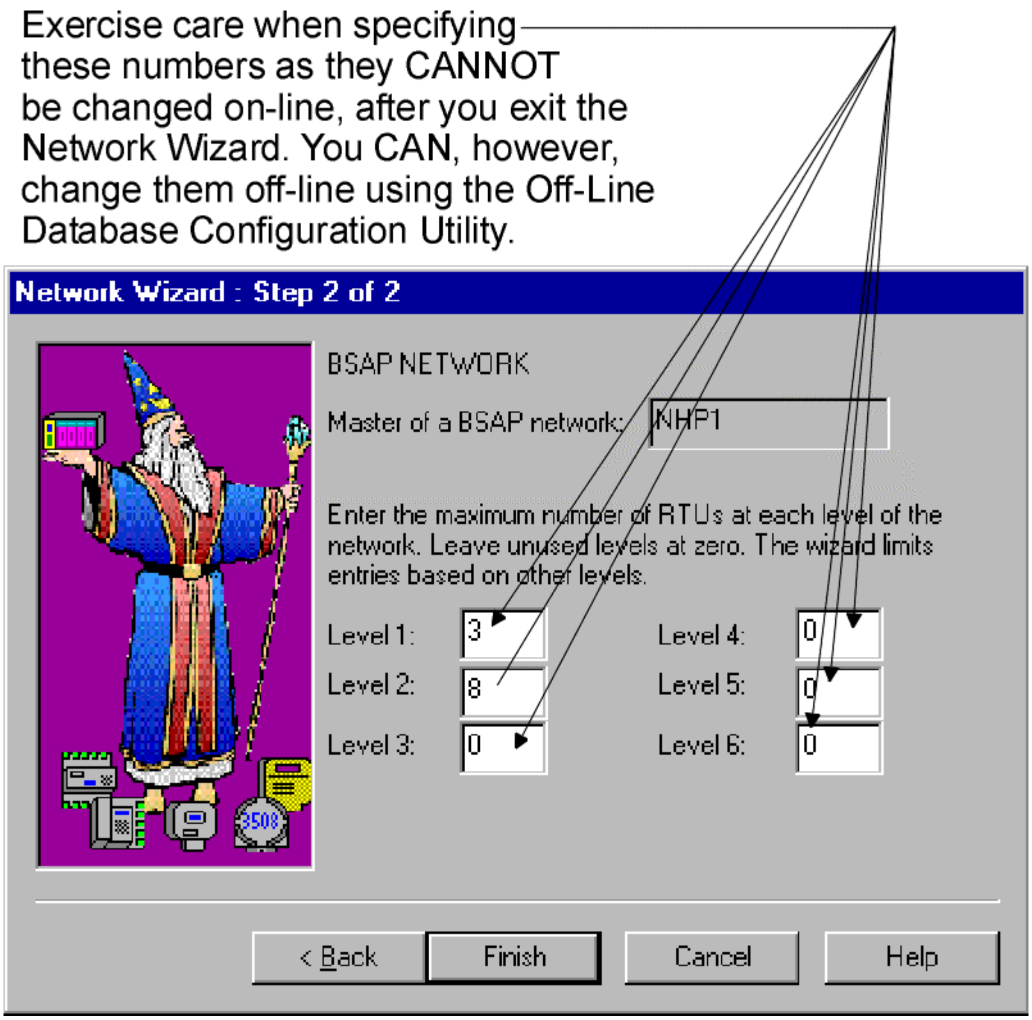
- Some HMI/SCADA packages (such as Iconics Genesis for DOS) DO NOT support peer-to-peer communication (discussed later in this manual) between level 1 nodes. Therefore, if peer-to-peer communication is required between two such nodes, they cannot both be on level 1, or a different HMI/SCADA package must be used.

- Try to define the maximum expected size of your network, allowing room for any planned future expansion. Defining some extra nodes at the end which are currently non-existent will not affect system performance, provided polling for them is turned off.[3]



Defining Network Size in NETTOP

[3] ***Polling*** *will be discussed in a later chapter. Note that program size in the master node of non-existent slaves* <u>*will be affected*</u> *because extra #NODE.nnn signals will be created.*

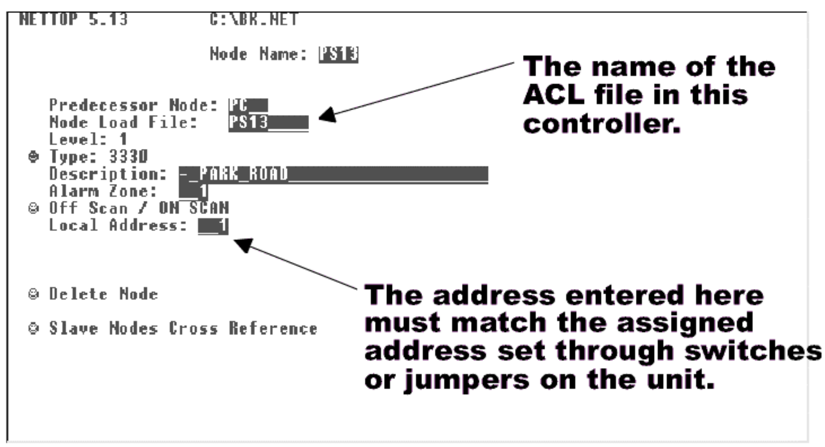


- Choose your node names carefully. NETTOP users have only four characters to work with. Because most software in the network master computer (including Open BSI, and SCADA/HMI packages) use node names in data base generation, you must avoid changing the node names later on, or else you will have to re-generate these data bases, or re-do other configuration. If you are using NetView, up to 16 characters may be used for the node name, however, be aware that not all HMI/SCADA packages, nor earlier versions of ACCOL and Open BSI tools support the longer node names, therefore, if your system includes a mixture of older and newer software, you should not exceed the 4 character limit.

- A master node can have (using any or all of its ports) a total of 127 slave nodes on the level below it. If more than this number is needed, the user can consider using a technique called expanded node addressing (also known as expanded BSAP or EBSAP) which is discussed fully in the *ACCOL II Reference Manual* (document# D4044). Expanded node addressing however, introduces certain complexities into the network which must be dealt with carefully, and is not appropriate for all systems. The user should fully understand the implications of using expanded node addressing, before attempting to use it. In particular, if you are defining an EBSAP communication line within Open BSI 3.0 (or newer) the virtual nodes on that communication line must be on Level 1 of your network, and the expanded addressing slave nodes must be on Level 2 of your network.

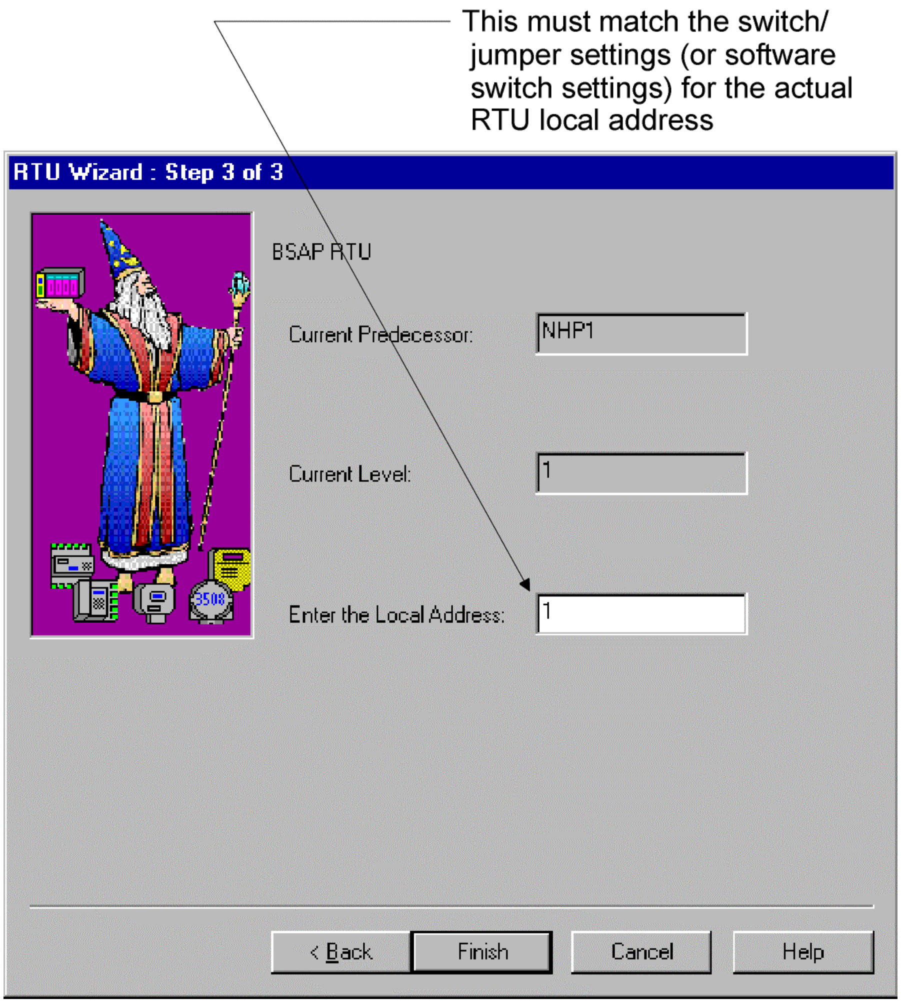
- As you plan your network, you will need to decide whether or not you want to use data concentrator nodes. Data concentrator nodes are master nodes which take data from one or more slave nodes, using peer-to-peer communication (discussed later in this manual) and perform some sort of calculations on that data; they may or may not have their own local process I/O. Data concentrators are useful, for example, if control decisions must be made upon data from several slave nodes. In the illustration on page 2-1, there are 3 nodes which could potentially serve as data concentrator nodes: DPC1, DPC3, and DPC4. NOTE: If not configured for peer-to-peer communication, these nodes would perform whatever local I/O and control they have been configured for, and simply serve as communication relay devices for their slave nodes, without making control or processing decisions based on slave node data.

- The local address for a node is assigned using jumpers/switches on the unit (or for some models, software switches.) That same address must be entered for the node in NETTOP or NetView. DO NOT use the local address as a means of identifying a station name.



Assigning Local Address in NETTOP

For example, if you are designing a network which has a node at each of three pump stations called pump station #13, #25 and #36, don't set the addresses of your three nodes to 13, 25, 36. Local addresses should be sequential, and should always start with address 1. Assigning these values randomly, with gaps in between, or using your own numbering scheme, imposes unnecessary overhead in the master node (unusable extra #NODE.nnn. signals are created). Start with address 1, then use address 2, 3, etc.

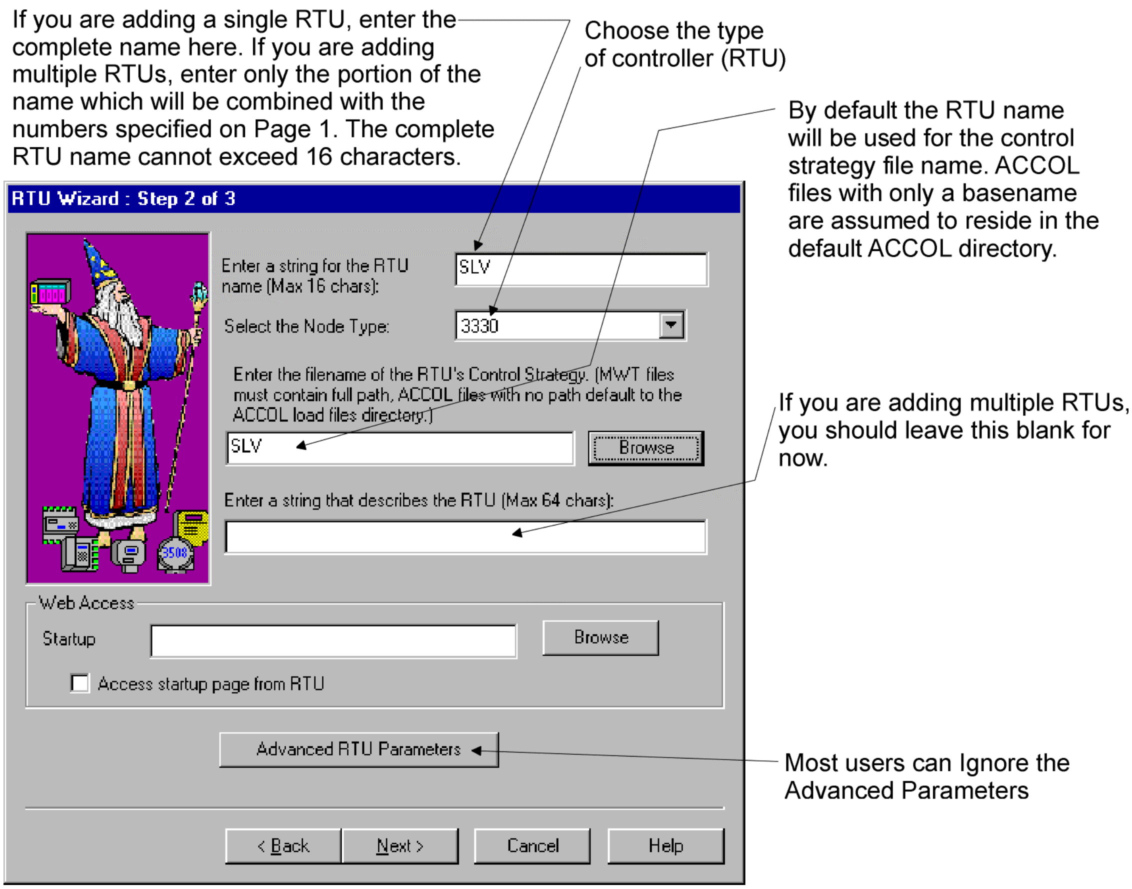


Entering the Local Address in NetView

- Each and every node[4] in your network must be specified in NETTOP/NetView. Be sure to specify the correct .ACL file base name in NetView or in the **"Node Load File"** field in NETTOP. This must be the actual .ACL file running in the controller. Although not required, it might help you to use the same name as the node name. That way, you won't forget which file goes in which node. NOTE: Entering an incorrect node load file name will prevent Open BSI, and certain other HMI/SCADA packages from functioning correctly.

---

[4] *If your system includes redundant controllers, each redundant pair is considered to be a single node, since only 1 of the 2 can be the on-line unit at any one time.*



Defining the ACCOL load file name and other parameters in NetView

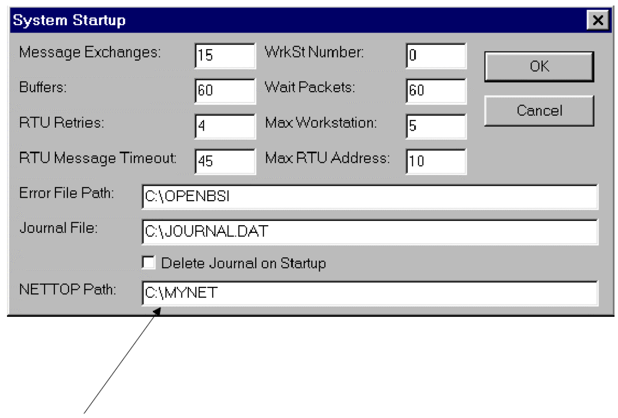
Considerations For NETTOP Users or Users With Open BSI 2.3 (or earlier):

- When you define the Network Master node, in NETTOP, the choice for the type of Network Master is simply a textual description. It doesn't matter what you choose. The Off Scan/On Scan field is currently unused, and the Alarm Zone is only applicable for certain HMI packages such as Enterprise Server, or the older Trolltalk VAX system.

- Additional PC's (laptops running Open BSI or UOI, for example) are often connected at lower-levels of the network, through a controller's Pseudo Slave Port, or Pseudo Slave with Alarms Port. These PC's are called pseudo masters. They are NOT considered to be part of the network and so they MUST NOT be defined in NETTOP. Pseudo Masters must, however, have current identical copies of the NETTOP files in order to perform global communication.

- When you release a new network file using NETTOP, you actually generate three files, named NETFILE.DAT, RTUXREF.DAT, and GLADXREF.DAT. These are your NETTOP files, which will be referenced by Open BSI, and other HMI/SCADA software in order to route data through your network. In addition, data from these files is used to create a structure called a **Node Routing Table**. The node routing table is transferred to a node through a time synchronization operation, or through a download.

These files are critical to the operation of your system; they must exist on your Network Master computer (the operator workstation). If you have other computers (Pseudo Masters) connected to nodes, they must also have <u>identical</u> copies of these NETTOP files. On an Open BSI workstation (Version 2.3 or earlier), the location where the system should look for these files must be specified in the Open BSI Setup Tool.[5] (See picture at right.)



Path where NETFILE.DAT, RTUXREF.DAT, and GLADXREF.DAT are stored must be specified here.

- If you are communicating with a portion of the network, i.e. you are not connected to a top-level node, but are instead connected locally to a Slave Port of a lower level node, be sure you specify the correct node name when establishing communications in preparation for a download. In Communication Setup Menus for DOS-based tools, if you specify the local address instead of the node name, prior to attempting a download, the wrong Node Routing Table will be sent. Similarly, if you are using Open BSI 2.x for this same purpose, configure the line as “Local” in the Communication Line Parameters dialog box and be sure to specify the correct node name in the Downloader. In both cases these problems can be avoided by downloading globally from the top of the network.

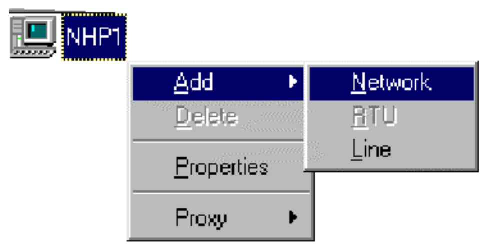
[5] *If you are using UOI, or the older DOS-based tools (on-line AIC, Toolkit, Taskspy) the location where these programs look for the NETTOP files is whatever directory serves as the default ACCOL directory, usually C:\ACCOL. The default ACCOL directory for DOS-based tools may be changed using the command ‘SET ACCOL=\directory_spec\’ where directory_spec is the DOS directory containing the network files. This command can be entered directly at the DOS prompt, or may be included in the AUTOEXEC.BAT file of your PC.*

Chapter 2 - Guidelines For Setting Up BSAP Networks

Considerations For NetView Users ONLY (Open BSI 3.0 or newer):

- Additional PC's (laptops running Open BSI, for example) are often connected at lower levels of the network, through a controller's Pseudo Slave Port, or Pseudo Slave with Alarms Port. These PC's are called pseudo masters. They are NOT considered to be 'nodes' in the network and so they MUST NOT be defined in NetView. If they are connected to BSAP nodes, and these PC's must perform global communication, they must have current identical copies of the NETDEF file. If local communication only is required, then LocalView may be used instead. If the PC's are instead connected to IP ports on IP nodes there is no need for the NETDEF file since the PC can request it from the Network Host PC (NHP).

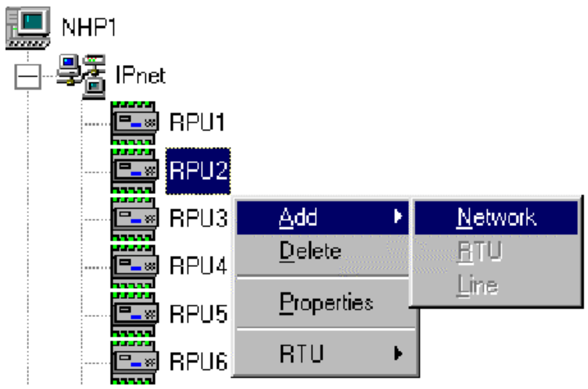
- In Open BSI 3.0 (or newer), you can add a single BSAP network underneath the Network Host PC (NHP). The NHP then serves as the BSAP Network Master node.



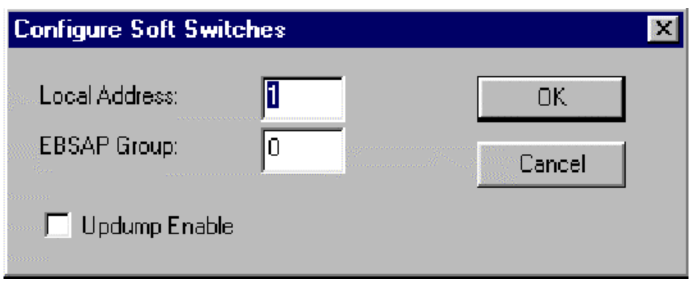
Adding A BSAP network

- If you have an existing BSAP network defined via NETTOP, and you plan to upgrade your Open BSI to Version 3.0 (or *newer*), you must run the NETCONV utility to convert the NETTOP files into a Network Definition File (*.NDF) and then run the NetDef to Database utility to generate NETDEF database file(s) compatible with Open BSI 3.1 (or newer). The BSAP network will then be known to the Network Host PC (NHP). (Otherwise, you would have to re-enter all of the NETTOP information within NetView.) Instructions for using NetView, NETCONV, and the NetDef to Database utility are included in the *Open BSI Utilities Manual* (document# D5081).

- In Open BSI 3.0 (or newer) you can only have one single BSAP network directly underneath the Network Host PC (NHP). Additional BSAP nodes however, can be added underneath one or more IP nodes, by the addition of one or more BSAP sub-networks. In a BSAP sub-network, the NHP is the Network Master node, and the IP node to which the BSAP sub-network is added is *the only level 1 node*. Purely BSAP nodes can only exist on levels 2 through 6 of the BSAP sub-network. It is important, therefore, that when you specify a local address for the level 1 IP node, you take care not to define a large number, since that would reduce the number of bits available to represent addresses for the BSAP nodes on level 2 through 6.



Adding A BSAP Sub-Network underneath an IP node



Setting the Local Address For the IP node on Level 1

Chapter 3 - Guidelines For Setting Up IP Networks

In this section, we will discuss, briefly, the terminology used in IP network configuration, and then go over some important items, which, if overlooked, can cause problems. Full details on using the NetView and LocalView programs are included in the *Open BSI Utilities Manual* (document# D5081).

A Discussion of IP Network Terminology

Open BSI 3.0 was the first version of Open BSI to support IP. **IP** is a standard communications protocol for transmitting data over a computer network. It also allows computers on *different* networks to exchange information with one another.

IP is *also* the standard communication protocol used on the **Internet,** a constantly changing set of computer networks that millions of people around the world use for business, government, and educational purposes. The 'IP', in fact, stands for **Internet Protocol**.[1]

Any Bristol Babcock Network 3000-series controller which supports IP communications will be referred to as an **IP node**.[2] Each IP node must be configured with at least one network connection called an **IP port** which will allow communications via Ethernet. Information on defining IP ports is included in Chapter 5.

Each IP port has an associated **IP address**. Because a particular IP node can have more than one IP port, it can also have more than one IP address.

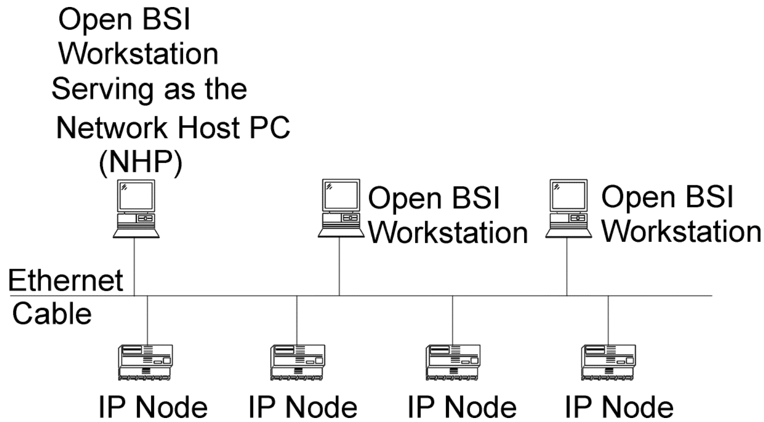
Each IP node also has an associated **sub-net mask** which is used in communications routing decisions. These subjects will be explained later in this chapter.

---

[1] *This section will discuss certain aspects of IP as part of communications configuration for your Bristol Babcock controller (RTU). The details of how IP works are beyond the scope of this manual, however. One book on the subject we recommend is <u>Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture</u> by Douglas E. Comer, published by Prentice Hall.*

[2] *Currently, the only Network 3000 series controllers to support IP communication are the 386EX Protected Mode versions of the DPC 3330 and DPC 3335 with PES03/PEX03 or newer firmware.*

There is no hierarchical structure enforced in a pure IP network; all nodes may be on the same level.



An IP network can, however, have multiple sub-networks, including BSAP **sub-networks**; the BSAP portions of which, must follow the standard BSAP concept of network levels. (For more information on BSAP networks, see Chapter 2.) There are numerous other possible IP configurations including routers, multiple networks, etc., however, these will not be discussed here.

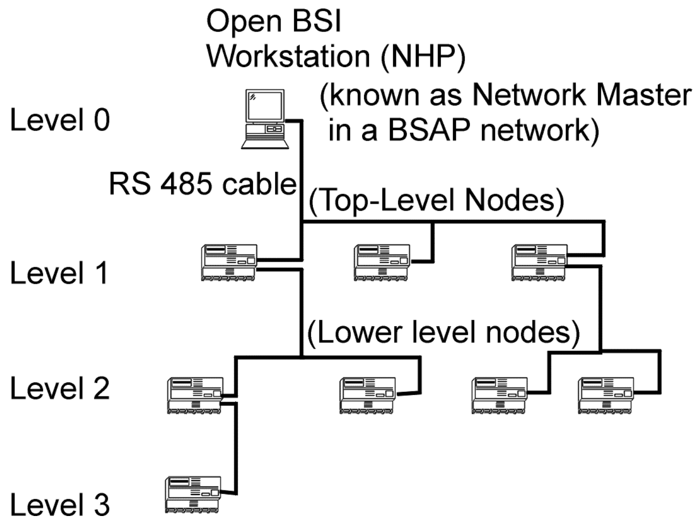
Every IP network must have at least one **Network Host PC (NHP)**.

What is A Network Host PC (NHP)?

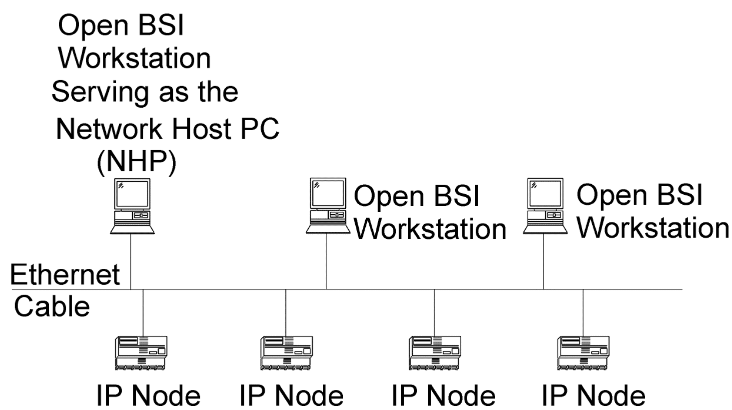
A Network Host PC is any Open BSI 3.0 (or newer) workstation. Typically, this workstation has some Network 3000-series remote process controllers (RTUs) connected to it (it serves as the 'host' for those RTUs). Those RTUs must be defined in a Network Definition (NETDEF) File at this NHP. Any other NHP can only gain access to these RTUs if access is granted by this NHP. NOTE: An Open BSI 3.0 Workstation without attached RTUs is still technically considered to be an NHP, even though it isn't 'hosting' any RTUs.

How Do NHPs Work?

In a BSAP or EBSAP network, the Network Host PC (NHP) performs the function of the network master node or level 0 node; it polls all Level 1 controllers (RTUs) for data.



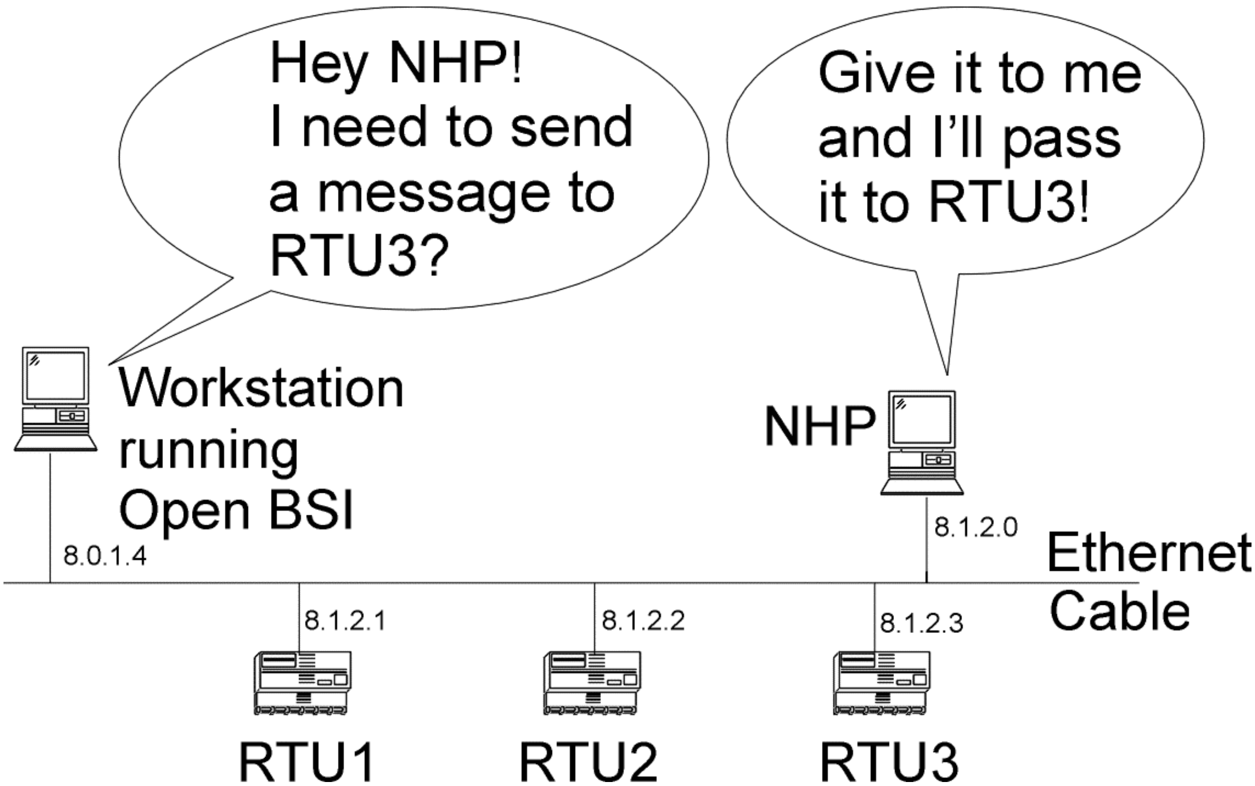
For an IP network (and also BSAP or EBSAP networks) the Network Host PC (NHP) serves as a central location for obtaining address information about RTUs and workstations in this portion of the network. This information is stored in the Network Definition (NETDEF) File, generated by NetView.



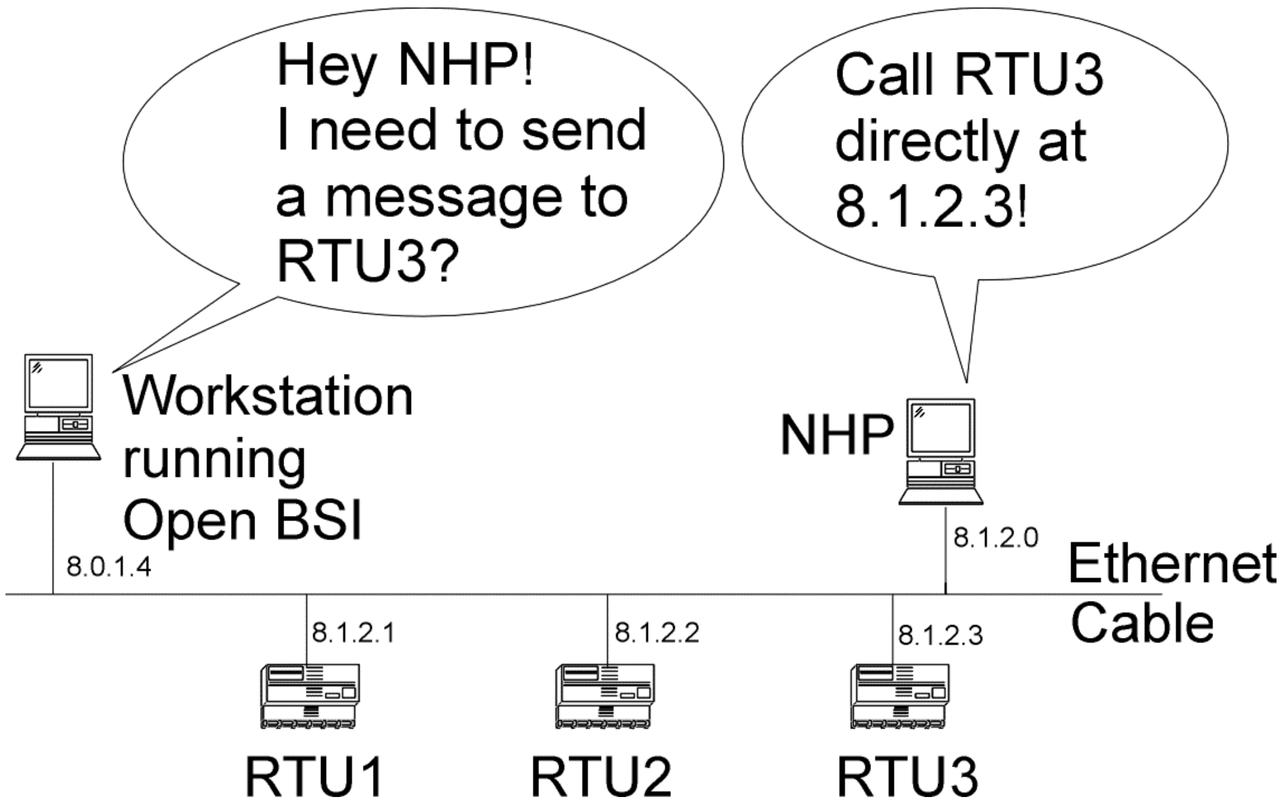
Any Open BSI 3.0 workstation, whether or not it has attached RTUs of its own, can communicate with any RTU in the network, provided it has been granted access to the RTU, by that RTU's NHP.

The concept of the NHP is easier to understand if you consider an analogy to the public telephone system. Most people remember a certain set of phone numbers for people they call frequently, but occasionally, they need to call someone whose number they don't know, so they call directory assistance and ask for the correct phone number. The NHP performs the exact same function as the directory assistance operator; except instead of giving out phone numbers, it provides address information, on request, for connections to any node in its section of the network.

In addition, following our same analogy, the directory assistance operator can generally do one of two things when you call for information. If you are unable to call someone, one thing the directory assistance operator can do is establish a connection for you. This same concept applies with respect to NHPs. If a particular Open BSI Workstation needs to communicate with an RTU associated with another NHP, the other NHP can relay messages between its RTU and the inquiring workstation. This is called proxy access, and is depicted in the figure, below:



The second thing that can happen when you call for information is that the directory assistance operator just gives you the number of the person you want to call and says "call them yourself, directly!" This also applies in the case of NHPs. An NHP can grant proxy direct access to a workstation requesting access to one or more of its RTUs. That workstation can then contact the RTU directly, without having to pass messages through the RTU's NHP. NOTE: Proxy direct access is only possible in IP networks, and requires the definition in the Comm Line Wizard (at the workstation requesting access) of an IP communication line for the proxy RTUs. Proxy direct access is depicted in the figure, below:



What Are the Advantages of Using NHPs?

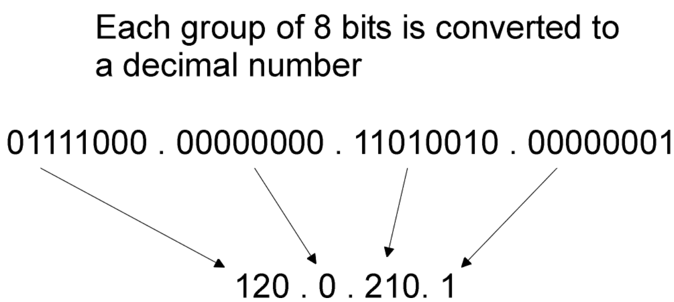
All any Open BSI 3.0 (or *newer*) workstation in the network needs to know in order to communicate with an RTU is the IP address of the NHP for that RTU, and the RTU name. This simplifies network configuration because if the address of an RTU should change, for any reason, only its NHP needs to be notified. Any other workstation can find out from the NHP.

Also, because you can have multiple NHPs, each of which is responsible for a portion of your network, your supervisory control is truly distributed among multiple sites. Through proxy access, any Open BSI 3.0 (or *newer*) workstation can communicate with RTUs belonging to any NHP in the network, provided that it knows the name of the RTU, and the IP address of the NHP for that RTU.

What is the Format of IP Addresses?

Each network connection from an IP node has an **IP address** which is unique within the network. It is important to note that the IP address is associated with the *network connection* (IP Port) on the node, NOT the node itself. This allows a single IP node to have *more than one* IP port, and consequently, more than one IP address.

IP addresses consist of 32 **bits** (1's and 0's) which are divided up into 4 groups of 8 bits each. A period is used to separate each group. Each group of 8 bits is then converted from binary to a decimal number from 0 to 255. The resulting IP address is said to be in **dotted decimal** notation.



Each of the numbers in the address generally has a specific meaning. The IP address is generally divided up into a **network portion** which must be common to each node in the network, and a **local portion** of which some part *must be unique* to a particular node.

How is the Specific Meaning of Each Part of the Address Defined?

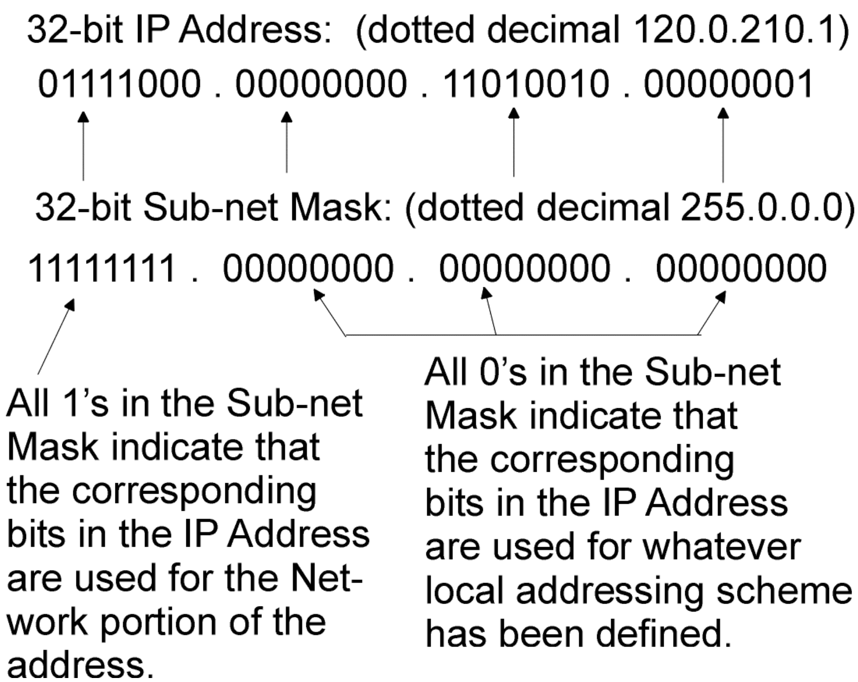
Addresses must be assigned to be consistent with whatever conventions have been established for your system.[3] In addition, there are certain rules to defining addresses, which will be discussed later.

The specific meaning of each part of the address is defined in something called the **sub-net mask**. The sub-net mask is simply another set of 32 bits (which must also be converted to dotted decimal notation). Each bit in the sub-net mask corresponds

[3] *For example, if this network has connections outside the plant (i.e. a connection to the real world-wide Internet), then the choice of this network number is assigned by an Internet governing body called the Network Information Center (NIC) or whatever Internet service provider you are using.*

to a bit in the IP address. If a bit in the sub-net mask is set to 1 (ON), then the corresponding bit *in the IP address* is considered to be part of the **network portion** of the IP address. The network portion can be ignored (or '*masked*') when performing communications to *nodes within the same network*, because by definition, all nodes in the same network have identical network portions. Any bit in the sub-net mask which is 0 (OFF) is considered to be part of the local addressing scheme.

The figure, on the next page, shows the IP address and corresponding sub-net mask for an IP address of 120.0.210.1 and a sub-net mask of 255.0.0.0.



As we said before, a '1' in the sub-net mask indicates that the corresponding IP address bit is part of the network portion of the address. Because the first part of the IP address '01111000.' has a corresponding sub-net mask of '11111111' we know that '01111000' (120 in decimal) is the network portion of the address.

The remaining parts of the IP address '00000000.11010010.00000001' have a corresponding sub-net mask of '00000000.00000000.00000000'. These bits are used as part of the local communications addressing scheme.

RULES FOR CREATING A LOCAL ADDRESSING SCHEME:

When you are creating your IP address, the network portion of the address must appear *first*. For example, if the network portion is 200, you CANNOT define an IP address as 0.200.14.1. The network portion must appear first. This means that when creating the sub-net mask, the masked portion (i.e. all 1's) must appear first.

The organization of the remaining bits can follow any local communications scheme you choose to devise, except that each group of bits that represents something *must be contiguous*.

For example, let's say the first 16 bits have been 'masked out' to define the network address, i.e. there is a sub-net mask of:

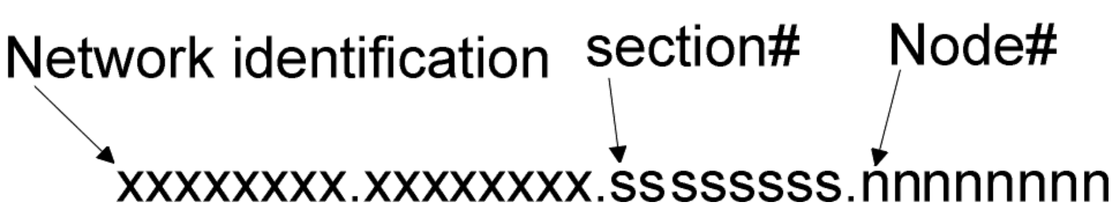
11111111 . 11111111 . 00000000 . 00000000

which in dotted decimal format is:

255 . 255 . 0 . 0

That leaves 16 bits (indicated by the 0's) for devising a local communications scheme.

You might want to use the first 8 bits to indicate a section or area number for a section of your network. 8 bits will allow up to 256 sections to be defined. Another 8 bits (remaining out of the 16 available) can be used to indicate a node number, allowing up to 256 IP controllers (RTUs) and Open BSI workstations, in a given section.



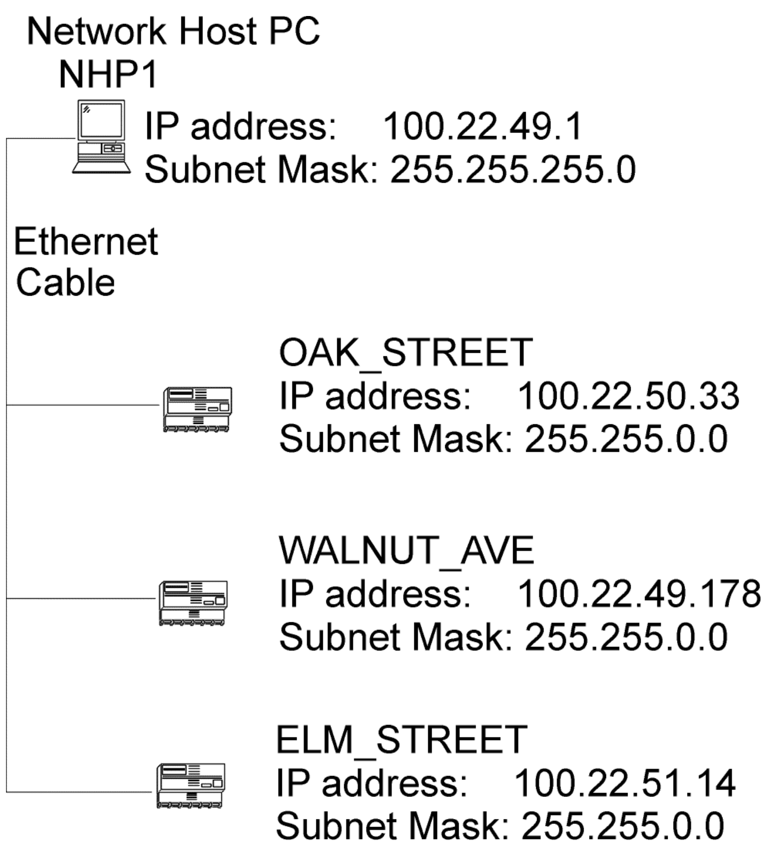
Sub-net Masks Determine Which Nodes Are Reachable From This Node:

So far, we have been talking about the mechanics of creating IP addresses and sub-net masks. The aspect we have not discussed is why IP addresses and subnet masks are so important.

A node's IP address, and its sub-net mask, define the range of acceptable addresses with which the node can communicate. For example, if one node has an IP address of 4.3.2.1 and another node has an IP address of 100.100.0.1, there is no common network portion between the two addresses. For that reason, there is NO way these two nodes can communicate with each other directly - - they are each part of *different* networks. Any messages between these nodes would have to pass through one or more **router** computers.

For two nodes to communicate directly, the network portion of their addresses (as specified by the sub-net mask) must match exactly.

To illustrate this concept, look at the figure, at right. The network shown has one Network Host PC (NHP) called NHP1, and 3 controllers (RTUs) named OAK_STREET, ELM_STREET, AND WALNUT_AVE.



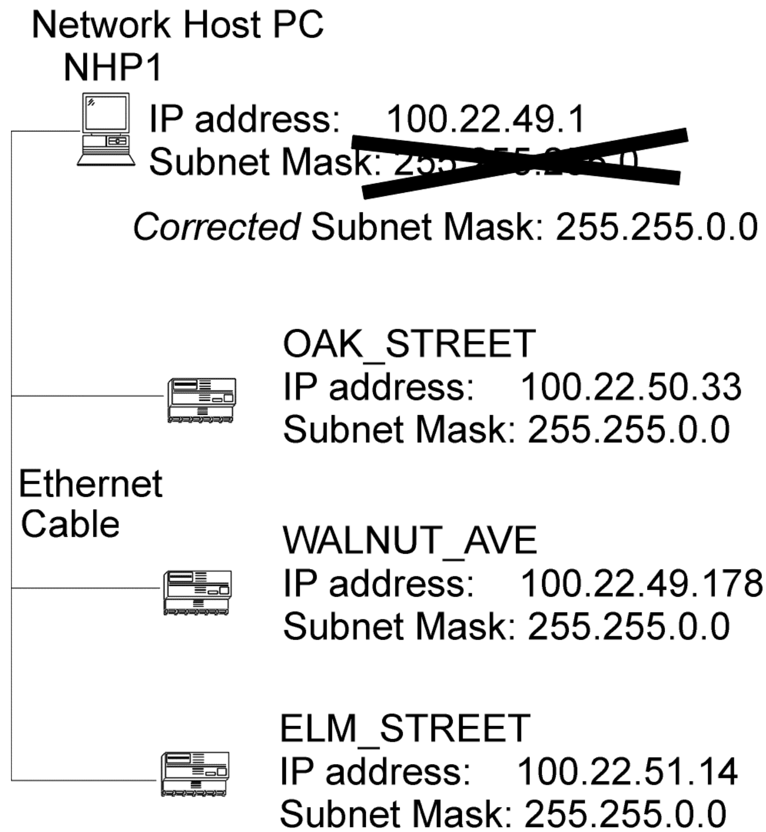
The table, on the next page, however, reveals a problem with the configured sub-net masks.

NODE NAME	IP ADDRESS, SUBNET MASK:	MASK SAYS THIS NODE CAN SEND MESSAGES TO ALL NODES WITH ADDRESSES:
NHP1	IP ADR: 100.22.49.1 MASK: 255.255.255.0	100.22.49.*yyy* where *yyy* is an integer from 0 to 255.
WALNUT_AVE	IP ADR: 100.22.49.178 MASK: 255.255.0.0	100.22.*yyy*.*zzz* where *yyy* and *zzz* are integers from 0 to 255.
OAK_STREET	IP ADR: 100.22.50.33 MASK: 255.255.0.0	100.22.*yyy*.*zzz* where *yyy* and *zzz* are integers from 0 to 255.
ELM_STREET	IP ADR: 100.22.51.14 MASK: 255.255.0.0	100.22.*yyy*.*zzz* where *yyy* and *zzz* are integers from 0 to 255.

Based on their specified IP addresses and sub-net masks, OAK_STREET, ELM_STREET, and WALNUT_AVE can all communicate with each other. They can also send messages to NHP1.

There is a problem, however. NHP1 has a sub-net mask which specifies that it can only send messages to nodes with addresses 100.22.49.*nnn* where *nnn* is an integer from 0 to 255. The only node which it can send messages to, therefore, is WALNUT_AVE.

To remedy this situation, NHP1's sub-net mask *should be changed to* 255.255.0.0 so that it can also send messages to OAK_STREET and ELM_STREET. The corrected sub-net mask is reflected in the figure at right.



Things to Be Aware of When Defining Your Network

- If you have a small network which will never be connected to the world-wide Internet, your choice of IP addresses is largely unrestricted. Even if you have no plans to connect your network to the global Internet, however, the Internet Engineering Task Force *recommends*, as per, *RFC 1918*[*] that IP addresses for private networks should be assigned from the following ranges:

  10.0.0.0 to 10.255.255.255
  172.16.0.0 to 172.31.255.255
  192.168.0.0 to 192.168.255.255

  These particular ranges of Internet addresses have been set aside for private networks. Any messages coming from these addresses can be recognized by most Internet Service Providers (ISP) as coming from private networks, and so can be filtered out. This helps avoid addressing conflicts should an accidental connection occur between a private network, and the global Internet.

[*] *RFC 1918 - Rekhter, et al, Best Current Practice memo - Address Allocation for Private Internets, Internet Engineering Task Force, February, 1996. Please see http://www.ietf.org for complete text of this memo.*

- Once you have an outside connection, to another network, or the worldwide Internet, your IP addresses MUST CONFORM to whatever conventions have already been established. If you are intending to connect your Open BSI network directly to the global world-wide Internet, you must obtain a range of IP addresses from your Internet service provider (ISP) or from an Internet governing body such as the Internet Assigned Numbers Authority (IANA).

- If you have a small network, it may simplify your configuration to use the *same* sub-net mask throughout each sub-net.

- The examples presented in this book generally involve small numbers of IP nodes in order to better illustrate networking concepts. Other than the caveats and rules discussed earlier, regarding addressing and sub-net masks, there is no restriction on users creating much more elaborate networks. Be aware, however, that *it is the user's responsibility* to configure and support such networks, and to fully understand potential problems associated with large or unorthodox network configurations. The examples and trouble-shooting notes in this book do not attempt to encompass every possible user configuration.

WARNING

We are discussing the Internet Protocol (IP) for use in communication between Bristol Babcock controllers. The normal, intended application is for a 'closed circuit' internet (LAN) of controllers and workstations in a company plant or industrial site. While, there is no built-in restriction against connecting an IP network of Bristol nodes to the world-wide Internet, customers must be aware that any external IP connection (no matter what brand of RTUs and software you are using) poses potential risks. *Customers are urged to change default passwords, as well as default UDP/TCP socket numbers, to lessen the possibility that an unauthorized intruder could gain access to your internal company process control data.* These Bristol Babcock security features are designed to prevent accidental access by plant operators, and should NOT be considered protection against intentional malicious activity by a sophisticated intruder, i.e. professional 'hacker'. Customers should consider purchasing commercially-available "firewall" software to gain a further degree of protection against such malicious intrusion.

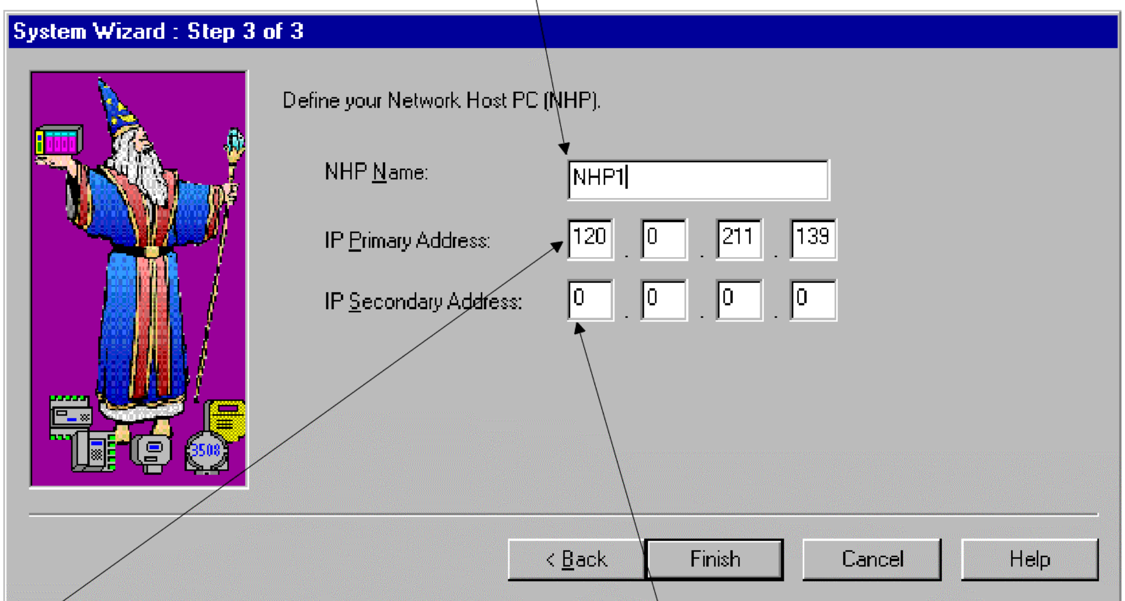
- If you have a redundant pair of controllers; each must have a *unique* IP address for its IP Port.

- If your IP network includes a BSAP sub-network underneath an IP RTU, then the Network Host PC (NHP) must serve as **network master** of the BSAP sub-network, and the IP RTU must be the only BSAP level 1 node.

- Currently, IP nodes do NOT include support for dial-up modems, or radios, though support for these may be included in a future release.

The IP address of the Network Host PC (NHP) is initially defined in the System Wizard of NetView.

Name for the current PC (or pair of PCs if you are using redundant NHPs)

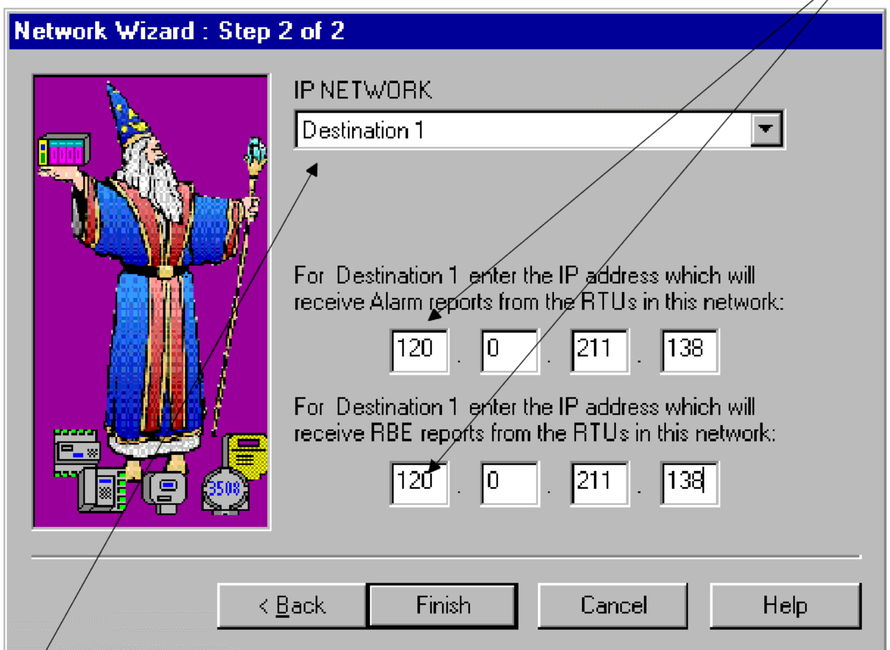


IP Primary Address is the IP address (in dotted decimal format) for the current workstation, or if this workstation is part of a redundant pair of workstations, the IP address of the “A” unit.

IP Secondary Address is left blank unless this workstation has two IP connections, or it is part of a redundant pair. If it is part of a redundant pair this is the IP address of the “B” unit.

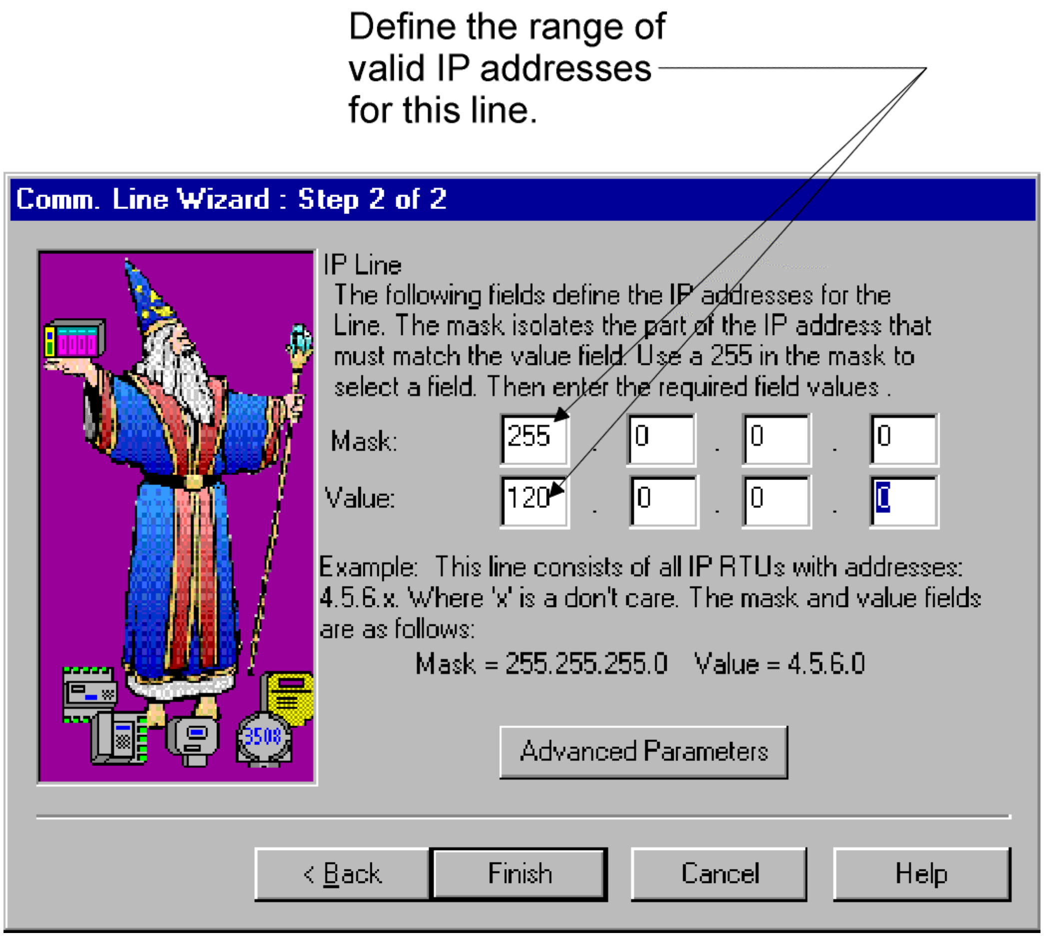
- Alarm and RBE (Report By Exception) data is sent to destination IP addresses defined in the Network Wizard of NetView.

You can specify the IP addresses of up to four different destinations which will receive alarm data, and up to four different destinations which will receive RBE data from the RTUs in this network. (These addresses would be PC work-stations running Open BSI 3.*x or newer*) By default, Destination 1 is THIS NHP.



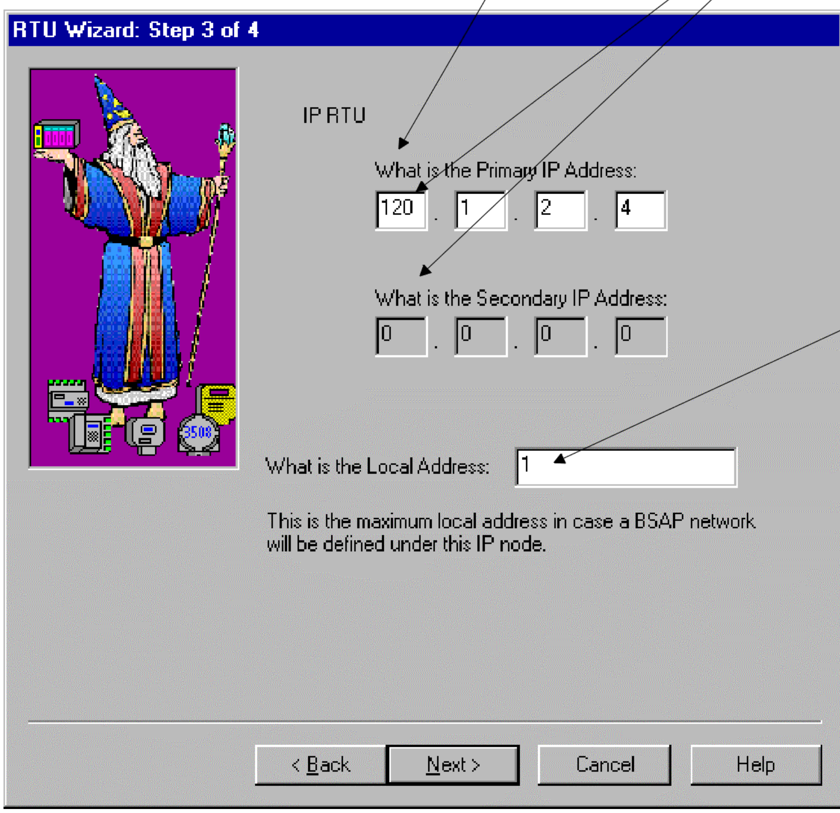
Use the list box to select Destination1 through Destination4

- The range of valid IP addresses, and the sub-net mask for the communication line are defined in the Comm Line Wizard of NetView.



- The IP address for an RTU's IP port is defined in the RTU Wizard of NetView.

Define the IP Address for this RTU's IP Port. This must match the address defined during RTU communications configuration. If you are defining multiple RTUs, enter the address of the first RTU here; addresses of the remaining RTUs will be assigned sequentially based on the first address.



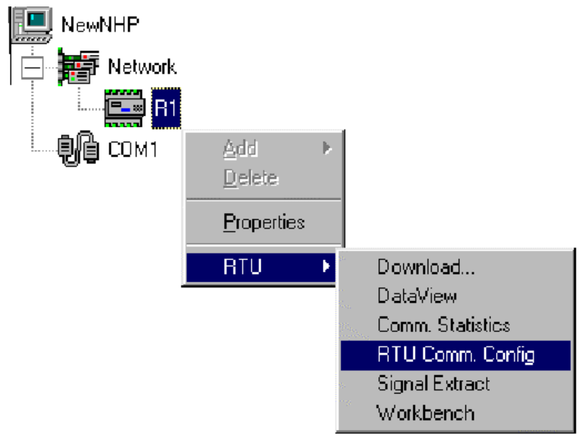
NOTE:
If this RTU is part of a redundant pair of RTUs the "A" unit's address goes in the primary field and the "B" unit's address goes in the secondary field.

The local address is important if one or more of these RTUs will have a BSAP sub-network underneath it. If not in a BSAP network or sub-network, it must still be set to match the local address set at the RTU (via jumpers or soft switches) to ensure alarm and RBE messages are routed correctly.

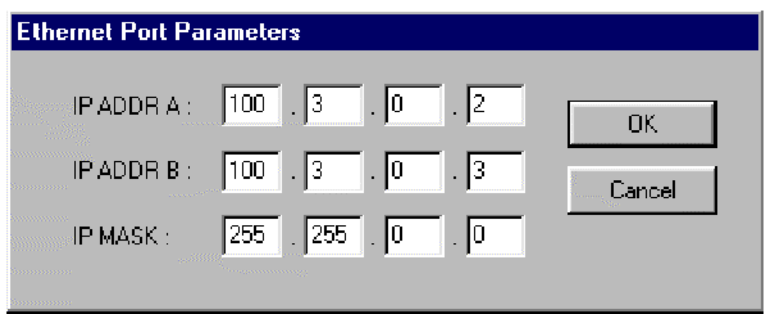
Chapter 3 - Guidelines For Setting Up IP Networks

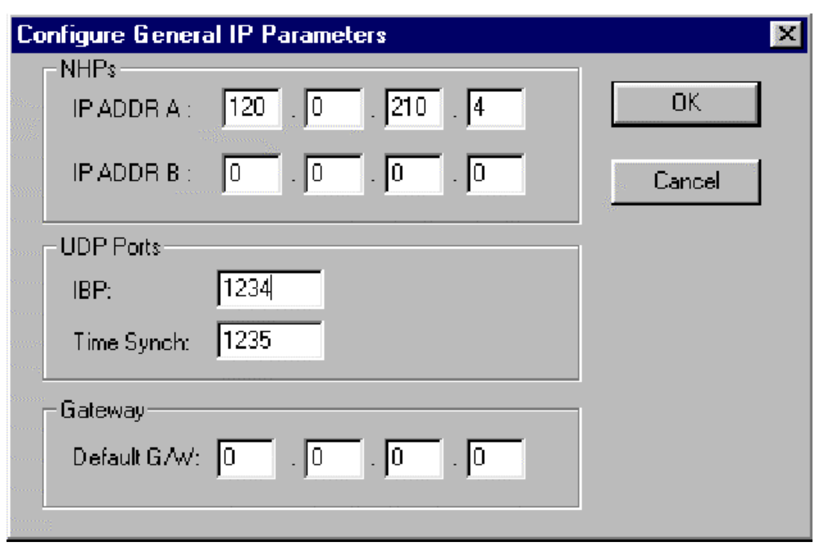
- In the ACCOL load, each IP node must be configured with an **IP Port**.

- Although the IP address information is defined in NetView, the same information MUST be configured in the FLASH memory of each individual RTU, using LocalView. LocalView is used to assign an IP address for each Ethernet IP Port on the IP node, and to specify the IP address for the RTU's associated NHP.



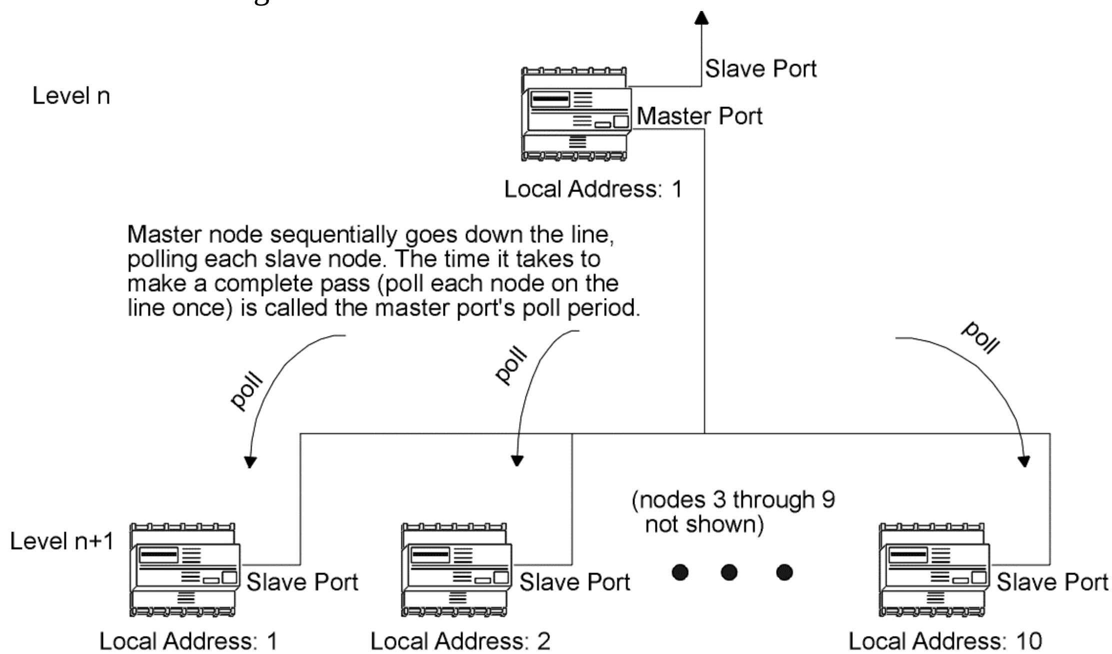
Once LocalView has been used to configure this information, the RTU must be reset (or downloaded with a new ACCOL load) for the new parameters to be activated. NOTE: For more information on IP address configuration, see '*Chapter 5 - Guidelines For Configuring the IP Data Link*'.





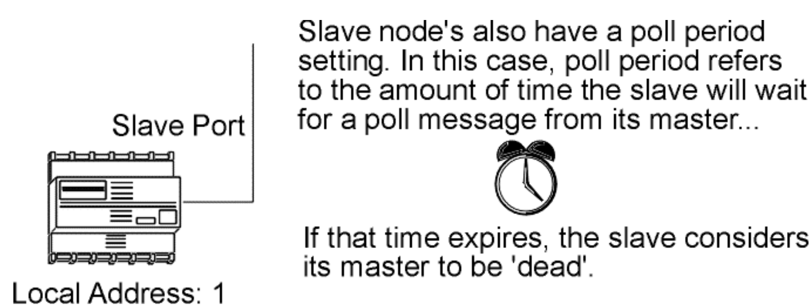
Chapter 4 - Guidelines For Configuring the BSAP Data Link

The **data link** refers to the actual mechanism by which data flows through the network. This includes the port and communication line itself, and for purposes of our discussion, also includes the software configuration for the ports and the communication line. The functioning of the data link is completely independent of the applications which use the link (HMI/SCADA packages, Master Modules, etc.) In order to understand how the **data link** operates, it is necessary to understand how data moves through a BSAP network.



At a specified interval, called the **poll period**, each master node requests data from all of its slave nodes. This request for data is called a **poll** message. So, for example, if a node on level 2, has a Master port defined, with ten slave nodes on level 3, this node will sequentially ask each of its ten slaves for data. If it makes this request once every second, that means that the poll period for this Master Port is 1 second.

Each slave node has a slave port, through which it receives poll messages from its master, and it responds to the poll messages with data. The slave port also has a poll period defined, however, in this case, *the poll period has a totally different meaning.*



For slave ports, the poll period refers to the amount of time the slave node will wait to hear from its master. If the slave node does not receive a poll message from its master within this period of time, it discards data messages for the master (except for alarms, which are saved in buffers), because the master is assumed to be ‘dead’.

The actual character-by-character movement of data is governed by the **baud rate**. The baud rate specified on a master node's master port must match the baud rate for each slave on that communication line.

When all of these data link parameters are configured correctly, data moves from a slave node, to its master, to the master's master, etc., until it reaches the network master computer, where it may viewed by the operator, exported to HMI/SCADA packages, etc.

This section will discuss data link parameters which specify how this movement of data will occur. These parameters include:

- Proper configuration of the Master Port or Expanded Addressing Master Port(s) if this is a Master node including: baud rate, high slave addresses, timeouts, as well as setting the proper poll period for the port(s), and defining the node array (#NDARRAY..)

- Proper configuration of the Slave Port, if this is a Slave node, including the baud rate and the poll period for the slave.

- Proper configuration of the Network Master (usually a PC workstation) including baud rates, retries, etc.

- Special considerations concerning radios, modems, etc.

Setting Up the Master Port(s)[1]

Master Port configuration should be performed prior to configuring other port types. It involves defining the port in the ACCOL source file, setting the poll period for the port, and setting up the #NDARRAY to control polling of slave nodes.

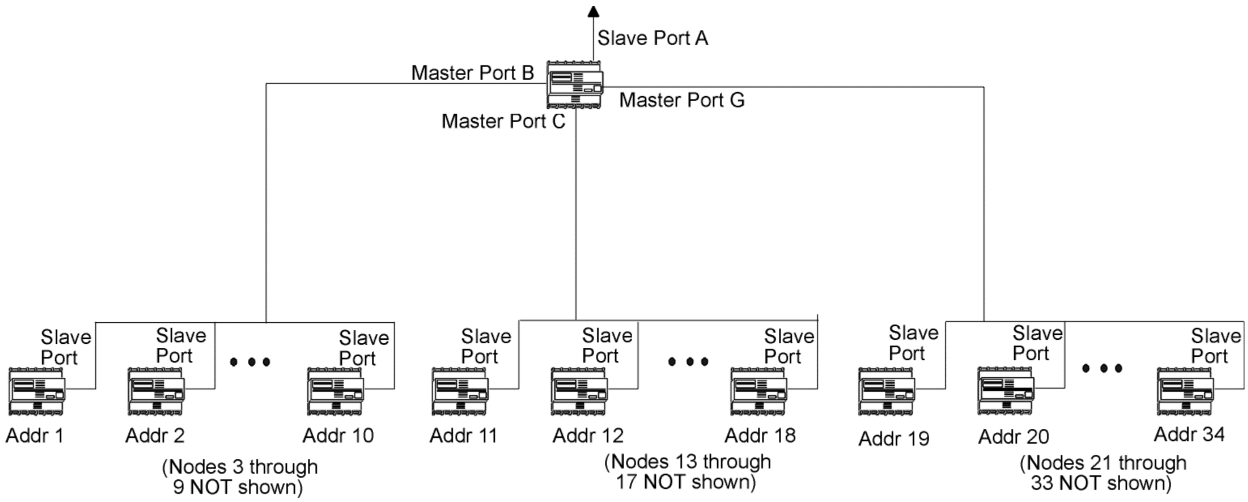
Master Port Definition in the ACCOL source File

A controller's Master Port(s) are configured within the ACCOL source file.

[1] *If you are using one or more Expanded Addressing Master Ports, the same instructions generally apply. Specific differences will be noted in footnotes.*

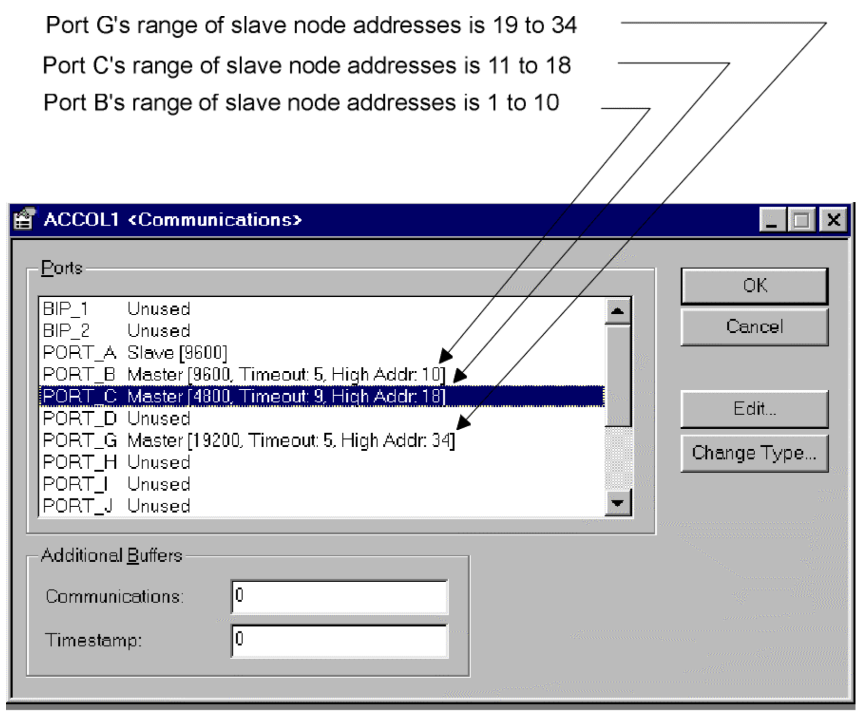
Choice of High Slave Addresses:

One of the first things to be aware of is that if a controller will have more than 1 master port, *the range of addresses for each master port's slave nodes must be in ascending order and MUST NOT OVERLAP.*



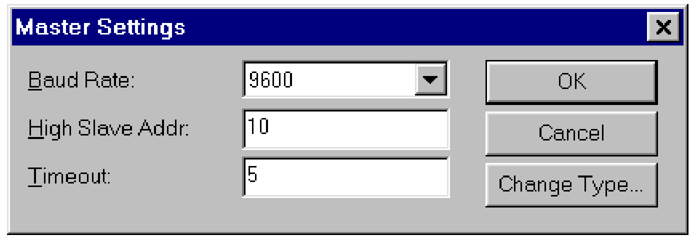
For example, let's say we have a 386EX Protected Mode DPC 3330 which is a Master to 34 Slave nodes.

If Ports B, C, and G of this controller are to be configured as Master Ports (see figure, above, and at right) then the **"High Slave Addr"** parameters for each port must be configured properly.



NOTE: Port address ranges must be defined in ascending order, and must not overlap. Port definitions should be done starting from the top, and going down towards the bottom of the window, i.e. going from BIP1, BIP 2, Port A, down through Port J.

For Port B, the **"High Slave Addr"** parameter is 10, this means that the range of slave node addresses for Port B is 1 to 10.

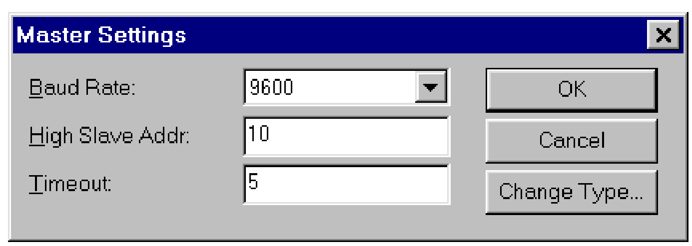


Similarly, if Port C's **"High Slave Addr"** parameter is 18, then Port C's range of slave node addresses is 11 through 18, and if Port G's **"High Slave Addr"** parameter is 34, then its range of slave node addresses is 19 through 34.

Note also, in this configuration, that once these ports have been set up, and slave connections established, etc., the user CANNOT, during future expansion, convert Ports BIP1, BIP2, A or D to Master Ports without changing the local addresses for all of the intervening slave nodes. Ports H, I, and J, however, could be used for this purpose.[2]

Choice of Baud Rate:

"Baud Rate" is the character by character rate at which data can be sent out through this port. The available baud rate depends upon the type of port, as well as the characteristics of the underlying link (cable, radio, dial-up modem, etc.)



The most critical aspect of setting the baud rate is making sure that the baud rate of each Master Port matches the baud rate of its associated slave node's slave ports. That is, if the Master Port is set at 9,600 baud, the Slave Port in each slave node on that port must also be set at 9,600 baud. Bristol Babcock controllers *do NOT* 'fall back' to lower baud rates if they are unable to communicate at the specified rate.

NOTE: Not all ports support all baud rates.[3] The '*Communication Ports*' section of the *ACCOL II Reference Manual* (document# D4044) contains details on which

[2] *The same rules apply for Expanded Addressing Master (EAMaster) Ports, except that the "**High Slave Addr**" includes virtual slave nodes, and a "**High EASlave Addr**" field appears for specifying the high slave address of a virtual node's slave nodes. See 'Expanded Node Addressing' in the ACCOL II Reference Manual (document# D4044) for details.*

[3] *Although some ports configured as Masters support synchronous communication, synchronous communication is NOT supported on any Expanded Addressing Master (EAMaster) Ports.*

ports support which baud rates, as well as whether synchronous communication is supported.

Choice of the Timeout value

The **"Timeout"** field for the Master Port specifies the amount of time (in tenths of seconds) for which the Master Port will wait for a particular slave node to begin transmission of a response message to the Master's poll. The default value is 5 (0.5 seconds). If the slave node does not begin responding within that period of time, a **response timeout** is said to have occurred.[4]

For example, if the **"Timeout"** is set to 7, it means that from the time the Master issues a poll to a slave node, it expects the slave node to begin transmitting a response within 0.7 seconds. If the slave node does not begin responding within 0.7 seconds a response timeout is logged. The Master will make two *additional* attempts (for a total of three attempts) to communicate with the unresponsive slave node.[5] If there is still no response, that slave node is considered to be 'dead' and is added to the list of 'dead nodes.'[6] During each polling cycle, the system will choose *one* node from the list of dead nodes and try to communicate with it, in case it has been 'revived'. To completely turn off polling for a node, its entry in the node array must be turned off. (See '*Setting up the Node Array (#NDARRAY) to Control Slave Node Polling*' later in this section.)

If your data link uses radios with RTS-CTS delays (to allow time to key the radio) then the **"Timeout"** value must be *greater than* the RTS-CTS delay time. If a repeater is being used, you must add timeout time equivalent to two additional delays (one delay for transmit, another delay for receive.) If you are using a satellite or cellular phone system to dial-up nodes, you will also typically require an extended value for the **"Timeout"**.

If you need to have a dynamic "**Timeout**" value, which may be changed on-line, you may make use of the advanced master polling parameters, discussed later in this chapter. See '*Configuring Advanced Master Polling Parameters*'.

[4] *For information on seeing whether response timeouts are occurring, see 'Chapter 7 - Troubleshooting Communications' later in this manual.*
[5] *If your firmware revision supports it, and you have configured a Master Poll Control Signal List, the number of attempts can be changed by altering the DATA MSG ATTEMPTS list entry. See 'Configuring Advanced Master Polling Parameters' on page 4-10 for details.*
[6] *For a technical description of the polling algorithm, and the handling of 'dead nodes' see the Network 3000 Communications Application Programmer's Reference (document# D4052).*

Setting the Master Port Poll Period

As discussed, previously, each port has an associated poll period. For a Master Port, the poll period should be set to the time it takes, under normal operating conditions, for this Master Port to poll each and every slave node on its line for data - *once*. Note, also, that Master Port poll periods should always be defined *prior* to defining Slave Port poll periods.

The poll period is set using #POLLPER.*nnn* system signals in the ACCOL load. The choice of which poll period signal to use depends upon which port you are configuring. The table, below, shows the correspondence between #POLLPER.*nnn*. signals and their associated ports.

If you are configuring the poll period for this port…	You must use this poll period signal:
Port A	#POLLPER.000.
Port B	#POLLPER.001.
Port C	#POLLPER.002.
Port D	#POLLPER.003.
BIP 1	#POLLPER.004.
BIP 2	#POLLPER.005.
Port G	#POLLPER.006.
Port H	#POLLPER.007.
Port I	#POLLPER.008.
Port J	#POLLPER.009.

One way to specify the Master Port poll period, is to use the worst case poll period calculation, discussed below, as a starting point, and then tune the poll period, on-line, to achieve the optimum system throughput.

<u>Worst Case Poll Period Calculation:</u>

A ‘worst case’ calculation of the poll period is as follows:[7]

Master Poll Period = 7,650 * (number of nodes on the line) / (baud rate)

So, for example, if a master port has 12 slave nodes, and the baud rate used is

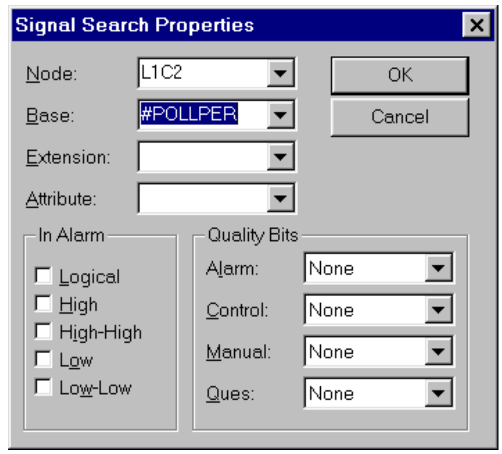
[7] *For a more accurate calculation of the poll period, which takes into account message size, delays, etc., users can obtain the POLLPER.BAT program from the on-line bulletin board of Bristol Babcock's Application Support Group. Bristol's Application Support Group can be contacted using the phone numbers listed on page ii of this manual.*

9,600, then a 'worst case' calculation of the poll period is:

7,650 * (12) / (9600) = 9.56 seconds

If line turnaround time is a factor (due to modems, or radios), this must be taken into account when calculating the poll period. For example, if a modem/radio combination requires 0.2 seconds after RTS assertion before CTS is asserted, then 0.8 * (number of nodes on the line) should be added to the poll period calculated by the formula above. 0.8 was chosen because it represents 4 times the RTS/CTS delay; this amount of time is needed because there can be up to four messages per data transaction.

The 'worst case' poll period can be entered as the initial value for the poll period signal when creating the ACCOL load. The value can then be altered, on-line, using DataView, based on the tuning technique described below.



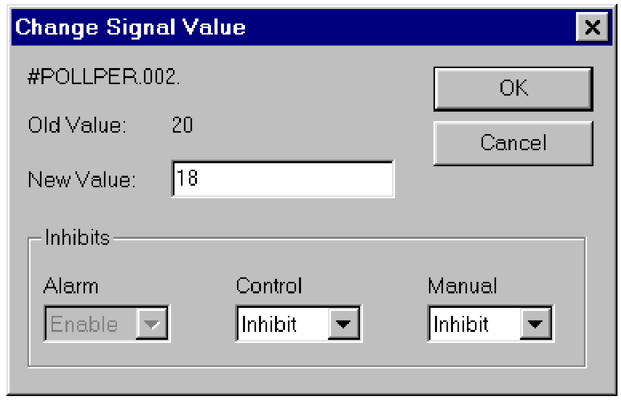
Technique For Fine-Tuning the Poll Period

Once the system is 'up and running' with the initial 'worst case' poll period, the poll period *may* be able to be shortened based on observed performance of the network. This must be done carefully, however, because making the poll period too short may have a negative impact on performance.[8]

To perform this 'fine tuning', the system must be operating normally. All nodes on the line must be on-line and communicating, and there must be no non-existent nodes.

[8] *Making the poll period too short will cause the system to go as fast as it can, but can reduce throughput by eliminating preferred polling and other aspects of the polling algorithm.*

Go to the Master node which has the port being tuned, and observe the transmit (TX) and receive (RX) LEDs which correspond to the particular Master Port. Under normal operating conditions; these LEDs should both be OFF simultaneously between 30% and 50% of the time.



If they are OFF more frequently than that, the poll period is unnecessarily long, and there is too much 'dead time'. Call up the associated poll period signal in DataView, and enter a smaller value to shorten the poll period.[9] This process may be repeated until the OFF time for the LEDs is between 30% and 50%.

If the LEDs are ON almost continuously, the poll period *is too short* and must be lengthened.

IMPORTANT
The Master Port poll period tuning process, described above, is only appropriate when all slave nodes on the line are operating. Attempting to use this technique when there is polling for non-existent controllers (out-of-service, not yet installed, etc.) will generate incorrect results.

Measuring the Rate at Which Nodes Are *Actually* Being Polled:

One way to get a general idea of how often a particular slave node is polled is to establish communications with that node (from the level of its Master) via DataView (or Toolkit, if applicable) and call up the #TIME.001. or #TIME.007. signals. These represent time in seconds; the rate at which they are changing shows the rate at which polling is actually occurring.

If the rate at which the seconds values change is significantly slower than the Master Port poll period you defined, you probably set too small a value for the poll period (i.e. you are trying to poll your nodes faster than the system will allow.)

NOTE: Make sure the refresh rates for DataView are not set longer than the poll period or this technique will NOT work.

[9] *Some dead time is always required. It is used for processing alarms, download messages, etc.*

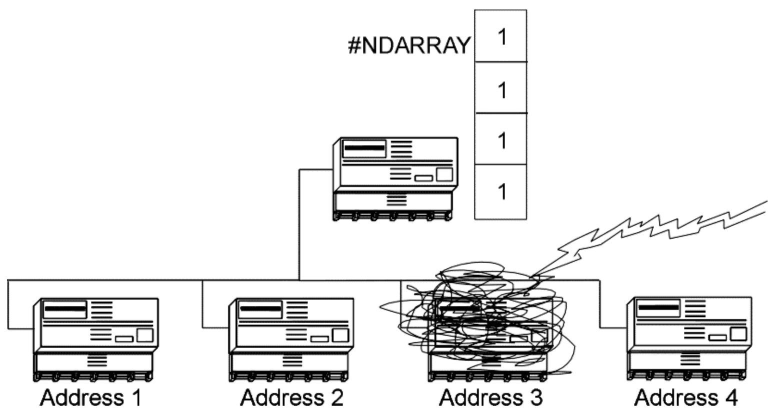
Setting Up the Node Array (#NDARRAY) To Control Slave Node Polling

Users should always define a node array to allow polling for slave nodes to be turned ON/OFF as necessary.

Turning ON/OFF polling is necessary for many reasons: If certain slave node addresses have been reserved for future expansion, or if an existing slave node has failed, or has been taken out of service for maintenance or repairs, polling for the affected slaves MUST BE TURNED OFF via the user-defined node array. Failure to configure and use the node array will result in the system believing that 'dead' or non-existent nodes are alive, and so it will continue to try to contact them. Unnecessary communication retries, as well as excess overhead in the master node, will degrade system performance significantly.

<u>Creating the Node Array:</u>

The ACCOL programmer must create a Read/Write Logical Data Array in the controller containing the Master Ports. The number of rows defined must be equal to (or greater than) the highest slave address among all the Master Ports in this controller.



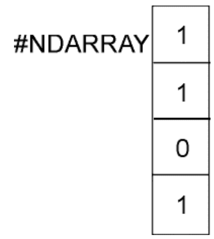
For example, if the ACCOL program we are creating is for a controller which has twenty slave nodes on two Master Ports, with Master Port A covering addresses 1 to 10, and Master Port B covering addresses 11 to 20, then a twenty row by one column logical read/write array should be created.

The number of the logical array (1 to 255) to be used is specified by the system signal #NDARRAY..[10]

Normally, the operator or ACCOL logic should leave each element set to ON, so that communication with slave nodes can occur. If, for whatever reason, one or more slave nodes are taken out of service (failure, maintenance, repairs, communication problems) or if there are certain node addresses reserved for future system

[10] *If you enter a negative value for #NDARRAY.., the absolute value of the number you enter refers to a signal list, which allows advanced users to define the node array and set additional polling parameters, See 'Configuring Advanced Master Polling Parameters', later in this chapter, for details.*

expansion, then polling for all of these non-existent or failed nodes should be turned off using the node array. This prevents unnecessary communication attempts by the master to a non-existent node.

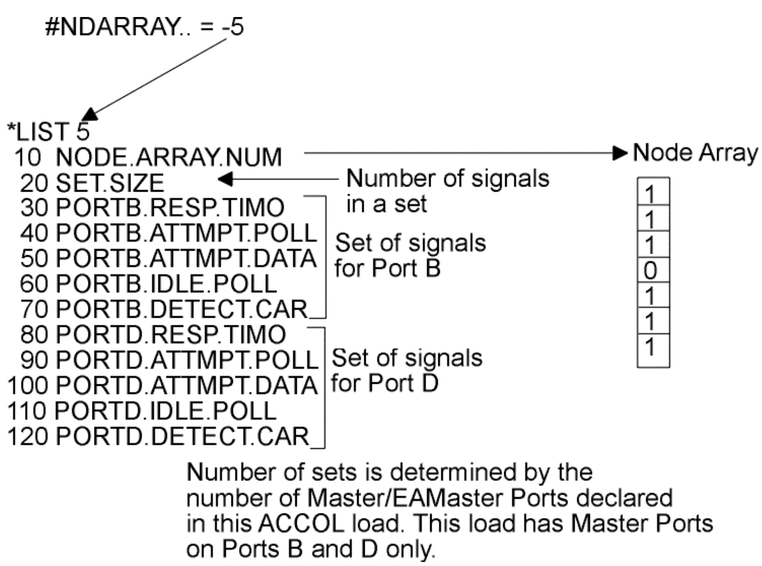


In the figure, shown on the previous page, the third slave node has been struck by lightning and so has failed. The master controller (which contains the node array) will continue attempts to communicate with it, until, the third element in the node array (corresponding to address 3) is changed from ON to OFF by the operator, as shown at right.

For a full description of how to configure the node array[11], see the #NDARRAY portion of the '*System Signals*' section in the *ACCOL II Reference Manual* (document# D4044).

Configuring Advanced Master Polling Parameters (PES03, PEX03, PLS03, PLX03, LS501, RMS04, or newer only)

Advanced users may choose to utilize certain additional parameters, which provide greater control over Master/EAMaster Port polling. In order to use these advanced parameters, a negative number must be entered on the #NDARRAY.. system signal. The absolute value of this number is used to designate a signal list which will serve as the Master Poll Control Signal List, instead of directly referencing the node array.



[11] *If your Master node includes Expanded Addressing Master (EAMaster) Port(s) the #NDARRAY includes virtual nodes as well as real nodes. Also, an additional node array, similar to #NDARRAY must be created for each virtual node, to control polling for its slave nodes. See the 'EAStatus Module' and 'Expanded Node Addressing' sections of the ACCOL II Reference Manual (document# D4044) for details.*

The Master Poll Control Signal List contains the following items; the choice of signal names used is entirely that of the ACCOL programmer.

NODE ARRAY number — An analog signal containing the number of the node array used to turn ON/OFF polling of Slave nodes. (This is normally specified by the #NDARRAY.. system signal, however, it is already used to reference the Master Poll Control Signal List.) For instructions on configuring the node array, see '*Creating the Node Array*' on page 4-9.

SET SIZE — An analog signal containing the number of configuration signals per Master Port in *this* list. Not all of the configuration signals need to be used; the number used is determined based on the value of SET SIZE. If SET SIZE is 7, all seven configuration signals must be configured for each port (RESPONSE TIMEOUT, POLL MSG ATTEMPTS, DATA MSG ATTEMPTS, IDLE POLL ENABLE, DATA CARRIER DETECT, RASCL LINK, RASCL SELECT); if SET SIZE is 2, only the first two configuration signals (RESPONSE TIMEOUT, and POLL MSG ATTEMPTS) must be configured, etc.

The remaining signals in the Master Poll Control Signal List are the configuration signals. Each Master/EAMaster Port should be configured with its own set of configuration signals, and they should appear in the list in the following order, based on which ports are defined (BIP1, BIP2, Port A, Port B, Port C, Port D, Port G, Port H, Port I, Port J).

RESPONSE TIMEOUT — this is the first configuration signal, and is available if SET SIZE is greater than or equal to 1. The RESPONSE TIMEOUT allows dynamic user control over the Master Port response timeout. When configured, this signal disables the response timeout value configured with the port, and allows it to be specified dynamically via this signal. In this configuration signal, the response timeout is the amount of time (in milliseconds, *not* tenths of seconds) that this Master Port will wait for a given Slave node to begin transmission of a response to a poll message. This value can range from 1 to 65,535 milliseconds. If the value is specified outside of this range, it will be ignored, and the standard response timeout, discussed on page 4-5 will be used *instead*.

POLL MSG ATTEMPTS — this analog signal specifies that number of times the Master Port will attempt to send a poll message to its slave nodes. The default is 1 attempt. With this signal, the user can specify from 1 to 255 attempts. If a value outside the 1 to 255 range is specified, the default will be used. POLL MSG ATTEMPTS is useful, for example, when using a system with radios. Since the window of time in which the radio is ON and 'listening' may be very short, it may be useful to issue several poll messages in order to increase the chances of the radio being on when the poll is sent.

DATA MSG ATTEMPTS this analog signal specifies that number of times the Master Port will attempt to send a data message to its slave nodes. The default is 3 attempts. With this signal, the user can specify from 1 to 255 attempts. If a value outside of the 1 to 255 range is specified, the default will be used. DATA MSG ATTEMPTS is useful, for example, when using a system with radios. Since the window of time in which the radio is ON and 'listening' may be very short, it may be useful to issue several data messages in order to increase the chances of the radio being on when the data is sent.

IDLE POLL ENABLE is a logical signal which, when set ON, causes additional polling to occur in a system where slaves dial into the Master. For this to function, the DIAL_UP_ACK enable signal in the Enhanced Slave Dial Control List must be ON in the Slave nodes (see page 4-34.) This additional polling occurs after all normal polling, and polling to see if 'dead' nodes have been revived has been completed. Under normal conditions, any extra time would leave the port idle. When the IDLE POLLING ENABLE signal is ON, however, the Master will issue a poll to Node 0 (a non-existent node; there is no Node 0). Even though Node 0 is non-existent, a node attempting to dial in can respond to the poll via the DIAL_UP_ACK, and transmit its data.

Another usage for IDLE POLL ENABLE is in a radio system where radios at the slaves are periodically turned on to listen for messages from the Master. IDLE POLL ENABLE allows radios to have a shorter listening time, because messages are sent more frequently, and the slave nodes can therefore quickly determine whether or not a message is intended for them, and if it is not they can shut off the radios to conserve power.

DATA CARRIER DETECT ENABLE — is a logical signal which, when ON, causes the Master Port to hold off sending poll messages until the modem's carrier detect is ON. This is useful because issuing a poll to a Hayes-compatible modem while it is still 'negotiating' the line would cause the modem to hang up, thereby terminating the connection. DATA CARRIER DETECT ENABLE can prevent this problem from occurring.

RASCL LINK — reports which RASCL link is currently in use.[12] If a logical signal is used, OFF (0) indicates that RASCL Link 1 is in use, and ON (1) indicates that RASCL Link 2 is in use. If an analog signal is used, '1' indicates Link 1 is in use, and '2' indicates Link 2 is in use. This signal is only applicable if a baud rate of 1MEG RASCL is configured for this port.

RASCL SELECT — is an analog signal which allows user control over which RASCL link (1 or 2) is currently in use. Entering '1' selects link 1; entering '2' selects link 2. Any other value allows automatic link control. Note that the signal will automatically revert to automatic link control after a single execution. If the user wants to keep the selection on a particular link, this signal must be control inhibited, thereby keeping RASCL in manual mode. This signal is only applicable if a baud rate of 1MEG RASCL is configured for this port.

An example Master Poll Control Signal List is shown below. It uses only the RESPONSE TIMEOUT configuration signal for each port (because SET SIZE equals 1).

```
#NDARRAY = - 7
SET.SIZE. = 1

*LIST 7
10  NDARRAY..
20  SET.SIZE
30  PORTA.RESP.TIMO
40  PORTC.RESP.TIMO
50  PORTD.RESP.TIMO
```

[12] *RASCL stands for Redundant Automatic Switching Communication Link, and is available on certain ports and versions of the DPC 3330 and DPC 3335.*

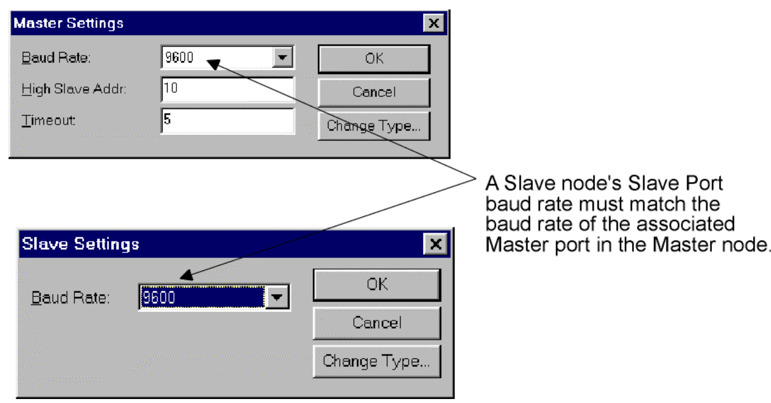
Setting Up the Slave Port(s)[13]

Slave Port configuration should be performed *after* the Master node of this Slave node has been configured. Configuration of the Slave Port includes defining the port in the ACCOL source file, and setting the poll period for the port.

Slave Port Definition in the ACCOL Source File

A particular ACCOL source file can have *one and only one* Slave Port. This is because, a particular Slave controller can only have one Master.[14]

The first parameter to define for the Slave Port is the **"Baud Rate"**. The value chosen must match the baud rate value for the Master Port, in the associated Master node.



So, for example, if node DPC3 is a Master to node DPC5, and DPC3's Master Port has a baud rate of 9,600, then DPC5's Slave Port baud rate must also be 9,600.

Setting the Slave Port Poll Period

The Slave Port poll period *has a totally different meaning than the poll period on a Master Port*, discussed previously. For a Slave Port, the poll period refers to a period of time that the Slave Node will wait to receive a poll message from its Master Node. If this period of time expires without a poll request from the Master, the Slave Node assumes its Master is 'dead' and so will discard all data messages (other than alarms) collected up to that time, for the Master.

---

[13] *There are two other types of ports which can be substituted for Slave ports under certain circumstances. Serial CFE Ports (instead of Slave Ports) should be used if this is a top-level node in an Enterprise Server system. Another alternative is VSAT Slave Ports, however, these ports require a corresponding VSAT Master which is currently only supported by Open BSI workstations. For more information on Serial CFE and VSAT Slave Ports, see the 'Communication Ports' section of the ACCOL II Reference Manual (document# D4044).*

[14] *One Pseudo Slave with Alarms Port, and several Pseudo Slave Ports may be defined for use by Pseudo Master devices, but these ports do NOT serve the same function as the Slave Port. See Appendix A for summary information on port types.*

Under most circumstances, the Slave Port poll period should be set to at least 3 to 5 times the rate at which the Master node's Master Port is *actually* *polling this slave.* If the poll period for the Master Port was set accurately, this would be 3 to 5 times the Master Port poll period. To get an idea of how fast polling is actually occurring, see '*Measuring the Rate At Which Nodes Are Actually Being Polled*' on page 4-8.

So, for example, if the Master Port poll period in the Master node is 20 seconds, the Slave Port poll period in the Slave node should be set to 100 seconds. The reason the poll period is set this way is to allow for any communication difficulties or retries which the Master node encountered while conducting its poll of all of the other nodes.

If you set the Slave Port poll period to *too short a period of time*, it will cause #NODE.*nnn*. alarms to be generated, in the Master node, and data messages destined for the Master node will be discarded. (A count of the number of messages discarded can be viewed as the 'Messages Aborted for Transmit Queue' statistic in the Portstatus Module, or in the Open BSI Remote Communication Statistics Tool.)

The poll period is set using #POLLPER.*nnn* system signals in the ACCOL load. The choice of which poll period signal to use depends upon which port you are configuring. The table, on page 4-6, shows the correspondence between #POLLPER.*nnn*. signals and their associated ports.

There is one instance in which the Slave Port poll period should be set to a small value. This occurs if the Slave node has a critical control scheme which requires timely updates of data from elsewhere in the network. In this case, the ACCOL program must include logic which monitors the Slave Port's #LINE.*nnn* signal to identify a communication loss, and perform some sort of conditional logic based on that occurrence. In this case, the Slave Port poll period would be set 'tightly' since the fresh acquisition of data is critical.

Using Advanced Poll Period Parameters (PES03, PEX03, PLS03, PLX03, LS501, RMS04, or newer firmware only)

Advanced poll period parameters are available to:

- Compensate for the turn-around time of messages in a 186-based 33XX controller.
- Provide quicker response from Slave nodes.

The advanced poll period parameters are accessible by placing a negative value on the #POLLPER.*nnn* signal corresponding to the port you want to configure, and setting up a signal list with signals representing the advanced parameters. The absolute value of the #POLLPER.*nnn* signal value is the number of the signal list.

For example, if Port C is to be configured using advanced parameters, and the parameters are to be in signal list 7, then a value of '-7' should be assigned to the #POLLPER.002. signal.

The parameters must be entered in the signal list in the order in which they are presented below. Each signal must be analog; the choice of signal names is entirely up to the user's discretion.

POLL PERIOD — This is whatever the poll period for this port should be, depending upon whether it is a Master Port or a Slave Port.

386 DELAY TIME — At higher baud rates, 186-based controllers sometimes have problems responding quickly enough to a message from a 386EX-based controller. This signal specifies a delay time (in seconds) to allow a 386EX-based controller to compensate for the message turn-around time in a 186-based controller. This may be used in either a Master Port (386EX node is master to a 186 slave node) or on a Slave Port (386EX node is a slave to a 186 master node). This delay can range from 0 to 65.535 seconds with resolution in milliseconds. A typical value for 386 DELAY TIME would be 0.010 seconds (10 milliseconds).

IMMEDIATE RESPONSE DELAY — This parameter may only be used on Slave Ports; a Master Port will ignore it. The IMMEDIATE RESPONSE DELAY specifies a period of time (in seconds) that a Slave node will wait, before responding to a request for data from its Master. This delay can *actually speed up* *communication throughput* because it gives the Slave node time to get data ready for the Master, thereby reducing the number of 'ACK NO DATA' messages to its Master, and the subsequent additional data requests from the Master. (An 'ACK NO DATA' is simply the Slave telling the Master that it received the request, but it has no data yet to give.) This value can range from 0 to 2.55 seconds (with a resolution in milliseconds).

An example is shown below:

```
#POLLPER.002.= -7

*LIST 7
10  PORTC.POLL.PER
20  COMPENS.DPC3.186
30  IMMED.RESP.DLAY
```

Notes For Remote Process I/O Users

If you are using Remote I/O Racks (RIO 3331) connected to a DPC 3330, DPC 3335 or RTU 3310 host, an RIOR Port must be created. Configuration information on these ports is included in the '*Communication Ports*' section of the *ACCOL II Reference Manual* (document# D4044). For details on setting the poll period for an RIOR Port, see the discussion on poll periods in the '*System Signals*' section of the same manual.

Setting Up Data Link Parameters At the Network Master

The previous sections of this chapter dealt entirely with setting up ports and polling between controllers. This section of the chapter deals with ports and polling at the Network Master device.

In a pure BSAP network, the Network Master device is the host computer (usually a PC workstation) at the top of the network (Level 0). Its responsibility is to collect data from the **top-level nodes** (controllers on Level 1). In order to perform this collection, many of the same types of configuration issues for controller Master Ports apply. The Network Master has to perform polling, it has to do this at a certain baud rate, etc. This section will highlight certain aspects of the configuration of Network Master communications for the two Bristol Babcock supplied Network Master software packages - - Open BSI Utilities (either Version 2.*x* or 3.*x* and *newer*), and the Universal Operator Interface (UOI). For full details on installing and configuring these programs, please see the *Open BSI Utilities Manual* (document# D5076 or D5081), the *UOI Configuration Manual* (document# D5074), and the *UOI Operator Manual* (document# D5075).

Open BSI (Version 2.3 or *earlier* 2.x) Data Link Parameters

Open BSI is configured via the Open BSI Setup Tool. System Startup parameters are entered in the System Startup dialog box, then the user configures a communication line using the Communication Line Parameters dialog box.

The figure, on the next page, reproduces the System Startup dialog box for the Open BSI Setup Tool. Of the various parameters in this dialog box, the ones which have a direct affect on the data link are the **“Max RTU Address”** and the **“RTU Retries”**. **“Max RTU Address”** is similar to the ‘High Slave Addr’ parameter for a Master Port, except it encompasses all top-level (level 1) nodes. (The address range for each communication line is defined in a different dialog box.)

You may remember that when a Master Port polls a slave node, and receives no response within a specified timeout period, it will make two additional attempts for a total of three attempts.[15] For the **“RTU Retries”** parameter in Open BSI, you have the option of changing the number of attempts made.

[15] *It is possible to change the number of retries made, if your firmware supports advanced master polling parameters. See ’Configuring Advanced Master Polling Parameters’ on page 4-10.*

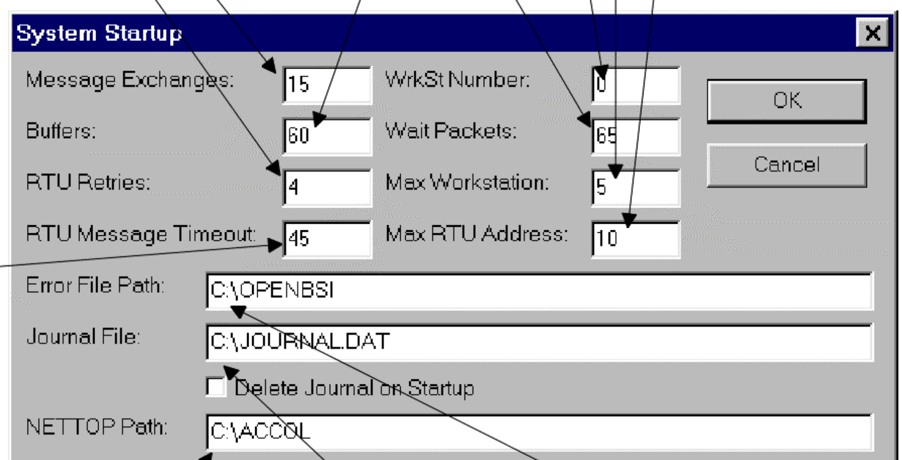
Message exchanges specify how many different applications can be run simultaneously. For example, if you are running BSISetup, DataView, and the Downloader at the same time, you are using 3 message exchanges. A normal value for this is between 15 and 50. The maximum is 100.

Buffers and Wait Packets are used for handling communication messages. Both of these values should be greater than 50, and there should be more wait packets than buffers. In general, the more active Network 3000 nodes in your network, the more Buffers and Wait Packets you need. For an Intellution FIX system, 150 to 200 is a reasonable range for these values.

This is the Workstation Number for this PC. It cannot exceed Max Workstation which is the highest allowed workstation number for all the PC's in the system.

This is the maximum local address of Top-Level Network 3000 nodes in the entire system. (Top-Level means it has a direct connection to the PC.)

If a message response is not received from a top-level node before expiration of the Timeout value defined in the Communication Line Parameters dialog box, that attempt is said to have failed. RTU Retries is the total number of attempts Open BSI will make to communicate with this Network 3000 (3xxx) node. If this total number of attempts fails, this top-level node is declared 'dead'.



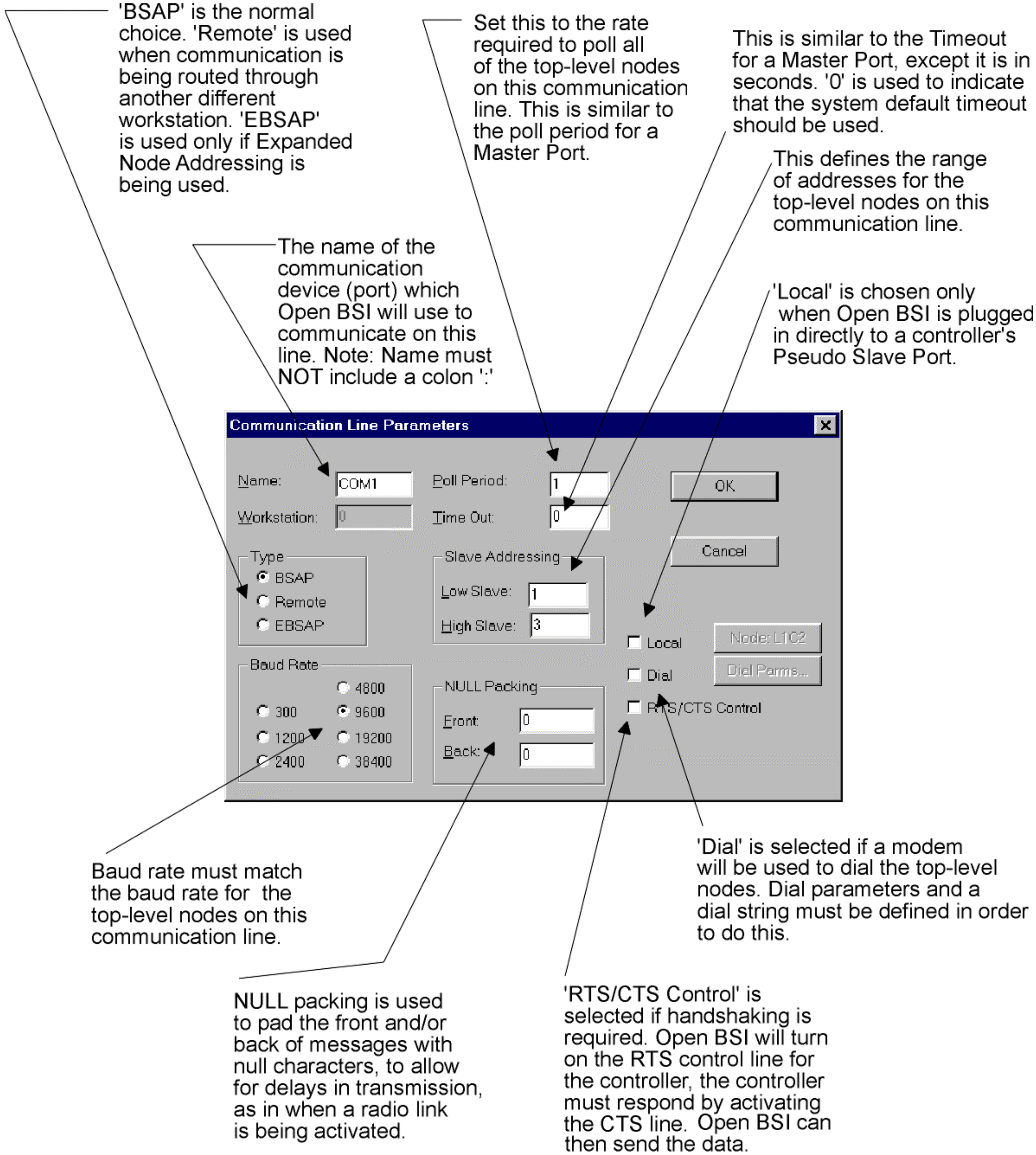
A Network 3000 node must respond to a program's request for data within the specified RTU Message Time Out period. If no response is received within that time, the response is said to have 'timed out.' In specifying this value, the baud rate, number of network levels, poll periods and the number of Network 3000 nodes must be considered. This value should not be less than the sum of the poll periods for each level of the network, but should not be made too large since that would delay initiation of a retry (if the application suports automatic retries.)

Path where system error files are kept. These files cannot be viewed by the user.

NETTOP path specifies the drive and directory where Open BSI will look for the currently released NETTOP files. This is a critical parameter.

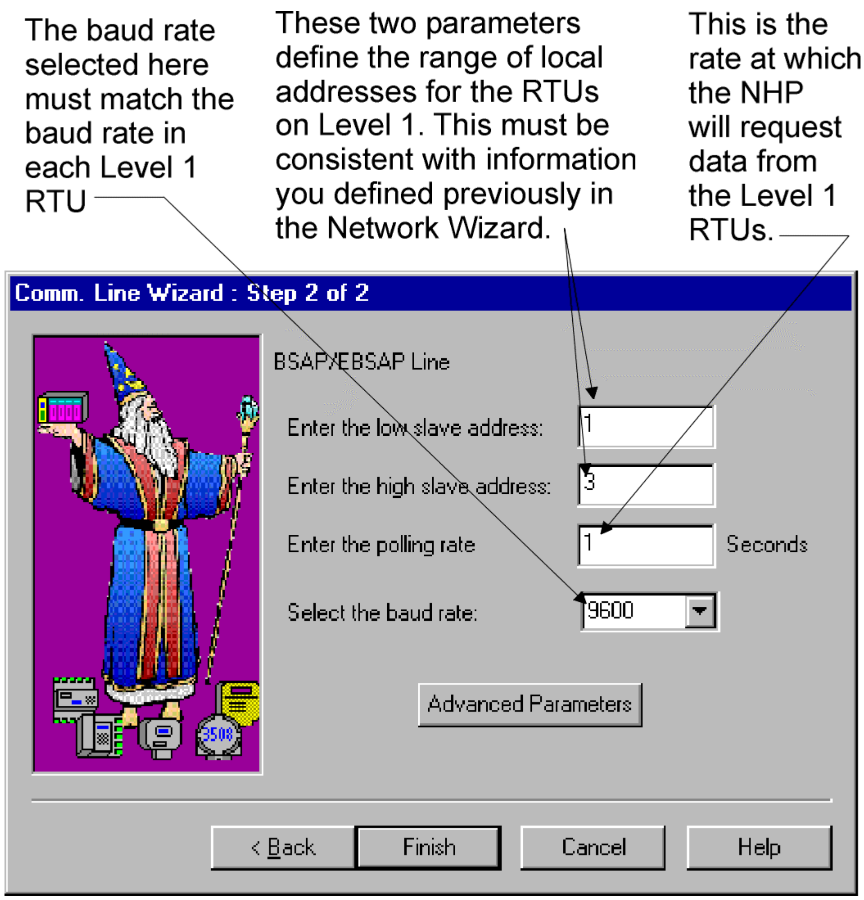
Path and file name for journal messages. This file will always be appended to, unless deleted on startup.

Once the System Startup parameters have been defined, the Active Communication Lines dialog box must be activated and a new communication line must be configured via the Communication Line Parameters dialog box. All parameters within this dialog box are related to configuration of the data link. Many serve similar functions to the parameters used when defining a Master Port. (See the figure below.)



Open BSI (Version 3.0 or *newer*) Data Link Parameters

In Open BSI Version 3.0 (or *newer*), the Network Host PC (NHP) serves as the Network Master of the BSAP network. Open BSI Version 3.0 (*and newer*) data link parameters are configured in the Comm Line Wizard of NetView:

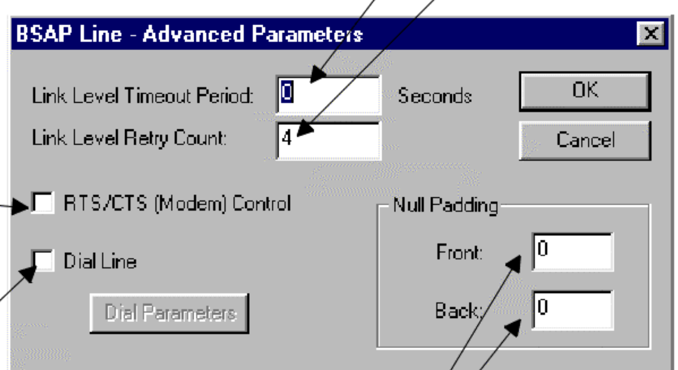


Adjustments to the link-level timeout period, and dialing parameters may be made via the **[Advanced Parameters]** push button.

This defines the maximum amount of time (in seconds) that Open BSI will wait to receive a response to any one data link transaction. If 0 is entered as the link level timeout period, the system will use a default timeout calculated based on the baud rate of the line.

This box is checked if the RTUs on this line require RTS/CTS hand-shaking in order for messages to be sent. The NHP turns on the Request to Send (RTS) control line for the RTU, which must respond to the NHP be turning on the Clear to Send (CTS) control line, at which point, the data can be sent.

If the number of Link Level Timeouts associated with a particular RTU reaches this value, that RTU is declared 'dead'.

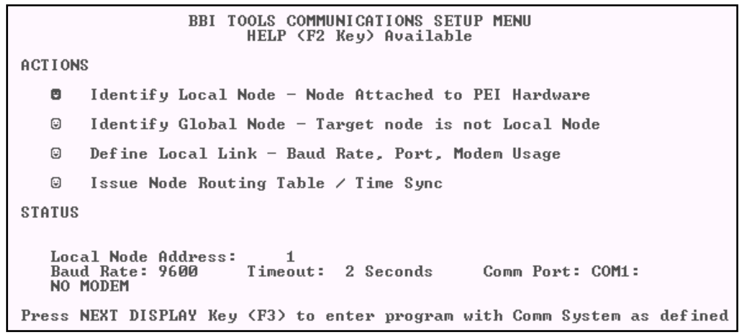


This box is checked if the NHP must communicate with the Level 1 RTUs on this communication line via a modem. When this box is checked, dialing parameters must also be configured. Use the Dial Parameters dialog box, accessible via the **[Dial Parameters]** push button.

These fields specify the number of null characters to insert at the beginning (front) or ending (back) of a message. Null characters may be useful in situations where there may be a momentary delay which could cause the start of a message to be missed, for example, while a radio link is being activated. To determine the delay caused by null packing, perform the following calculation:

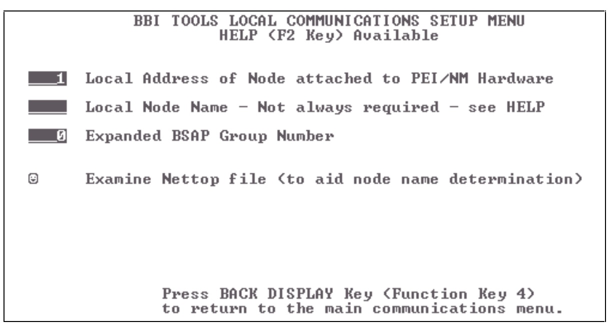
$$\text{seconds of delay} = \frac{\text{number of null characters} * 10}{\text{baud rate}}$$

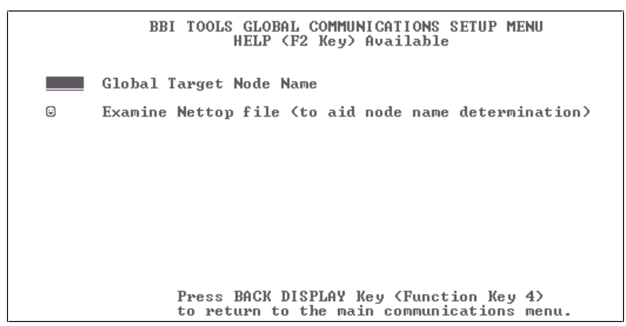
UOI/TMS Data Link Parameters



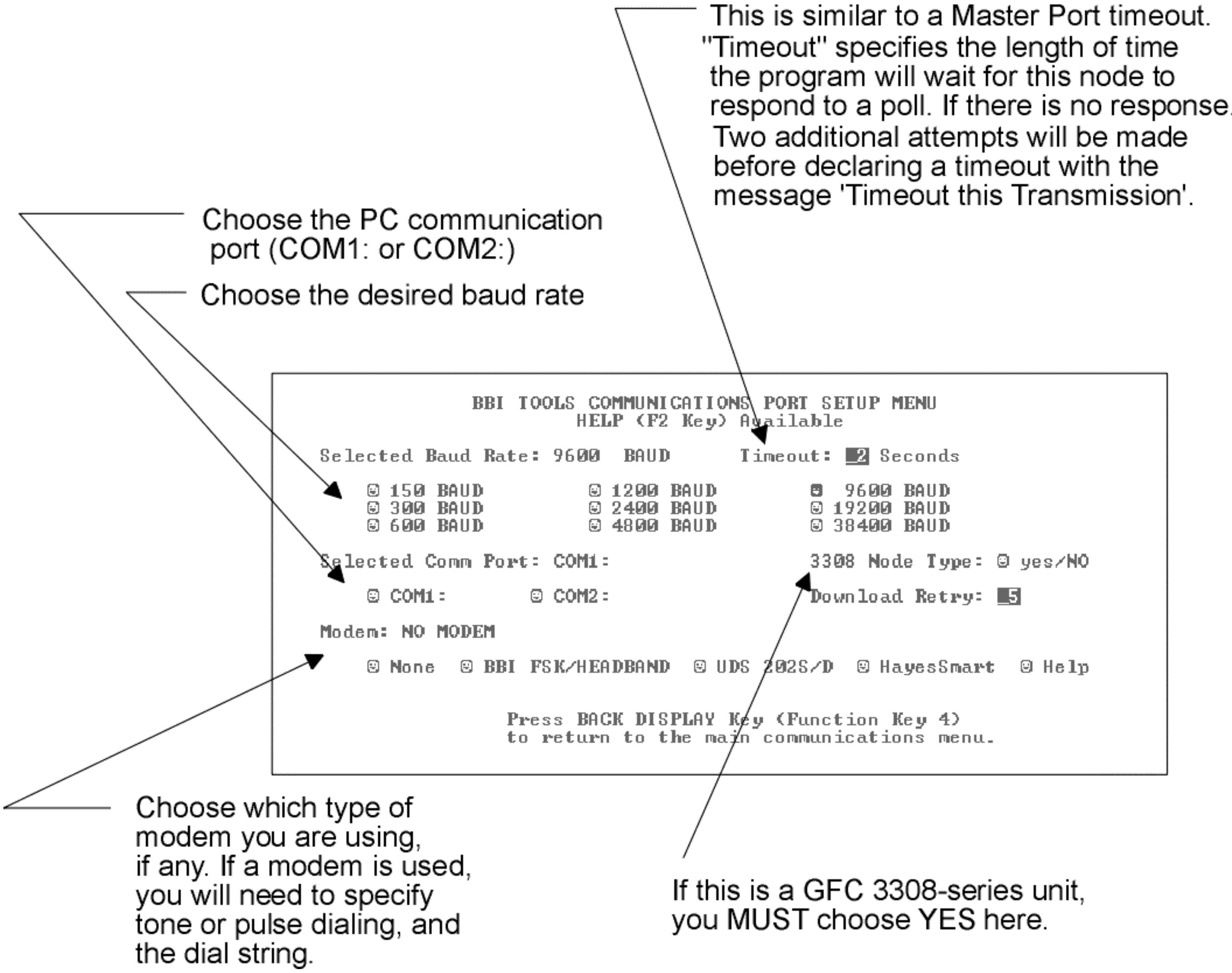
If you are using UOI or TMS software, or a DOS-based version of the ACCOL Tools, a common set of Communication Setup Menus is used to configure the data link. Optionally, much of the information normally entered in these menus may be specified, off-line, in the ATOOLS.INI file, which is automatically read at system startup.

Users must choose either local communication with the attached node, or global communication to another node which exists in the NETTOP files. If local communication is used, the node address and/or node name must be entered, as well as the Expanded Node Addressing Group Number (if applicable). If global communication is used, only the node name (as defined in NETTOP) needs to be entered.





Once the decision concerning local or global communication has been made, the user must enter the actual data link parameters including the baud rate, timeout, etc.



POLL_RATE Parameter in ATOOLS.INI

The default rate at which UOI/TMS or the DOS-based ACCOL Tools request data from their attached top-level nodes is 0.3 seconds. This value may be modified by changing the POLL_RATE parameter value in the ATOOLS.INI file. Valid rates which may be entered are: 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, or 1.0. For full details on the ATOOLS.INI file, see the *ACCOL II Reference Manual* (document# D4044).

BSAP Local Lines (Used for special applications only - Open BSI Version 3.1 and newer)

Beginning with Open BSI Version 3.1 (and newer), Open BSI Workstations can include a **BSAP local line**.

There are two common uses for the BSAP local line:

- Users running Open BSI on a laptop/notebook PC want to plug into the pseudo slave port of a Network 3000 controller which is at a lower level of the BSAP network (network level 2, 3, 4, 5, or 6) to view data from various network nodes. This is useful during testing and debugging or any time a technician is visiting a portion of the network that is geographically distant from the Network Host PC.

- A communication line has failed, so operators need to establish a temporary, backup communication line. For example, if the regular hard-wired communication cable becomes disconnected from an RTU, but the RTU also has a Pseudo Slave Port connected to a modem, and auto-answer is configured, a BSAP local line could be configured at the NHP to dial into that RTU and obtain data, until the normal hard-wired connection is restored.

IMPORTANT: Communication traffic from a particular workstation to a particular RTU can only occur via one communication line at any one time. Therefore, if a BSAP Local Line is active (as a temporary backup line) and then the regular communication line is re-activated, RTUs reachable via the BSAP local line will be inaccessible via the regular line as long as the BSAP local line is active.

For more information on BSAP local lines, see '*Overview of Supported Network Architectures*' in Chapter 1 of the *Open BSI Utilities Manual* (document# D5081), as well as '*Comm Line Wizard: Step 2 of 2 (Local BSAP Line)*' in Chapter 6 of the same manual.

Configuration Issues For Modems

Modems may be used to allow data to be transferred over telephone lines. Data can be transferred either:

- between two Network 3000-series controllers

  -or-
- between a Network Master (PC) and a Network 3000-series controller

A pair of modems is required for each setup; i.e. there must be one modem on each end of the telephone line. Depending upon the Network 3000 series controller(s) you are using, they may have an internal modem already installed. In other cases, external modems must be purchased.

The telephone line can be either a **private line** or **public switched network line**. A private line means there is a single dedicated phone connection between the two modems, eliminating any need for dialing. A public switched network line means that the modem must dial a number using the public telephone system.

Which Modems Are Supported?

Bristol Babcock currently offers three different modems for use with our Network 3000 product lines. These are shown in the table, below:

Modem:	Baud Rate:	Supports Dialing?	For details on hardware set up of this modem please see the Bristol Manual listed below:
BBI 1200 bps Modem	1200	YES	CI-1200
BBI 1200 bps - PL Modem	1200	NO	CI-1200-PL
BBI 9600 bps - PSTN Modem	9600	YES	CI-9600A

Note: Older Bristol modems such as the SNM are no longer sold by Bristol, and so will not be discussed.

Because of the large number of modems available on the market, it is NOT possible for Bristol Babcock to test and support non-Bristol Babcock modems. Bristol Babcock's Application Support Group, however, has successfully tested current Bristol equipment with U.S. Robotics modems.

The table, on the next page, shows how modem characteristics should be set when connecting a U.S. Robotics modem to an Open BSI workstation.

The first column describes what modem settings are required.

The second column shows corresponding initialization codes which would be included in the initialization string for US Robotics modems. If you are starting with a modem which is at its factory default settings, the E1, &C1 and &D2 codes shown in the table are unnecessary.

If you have a different type of modem, *use the equivalent initialization code from the manual accompanying that particular modem*; not the code shown here:

A modem connected to an Open BSI Workstation must have the following settings:	**Corresponding US Robotics Code:**
Turn ON local echo of codes.	E1 *NOTE: E1 is a factory default*
Disable ARQ result codes.	&A0
Set modem's serial port rate to variable; to follow the connection rate.	&B0
Set Carrier Detect (CD) to normal operation.	&C1 *NOTE: &C1 is a factory default*
Set Data Terminal Ready (DTR) to normal.	&D2 *NOTE: &D2 is a factory default*
Disable transmit data (TD) Flow Control.	&H0
Disable data compression.	&K0
Disable error control. NOTE: This setting is NOT required if you have an identically configured modem at the RTU end.	&M0
Ignore RTS for receive data (RD).	&R1

Configuration Tips

If you are trying to configure modems for use with Bristol equipment, the following configuration tips are offered, however there is NO GUARANTEE that these will allow compatibility with modems other than those listed under the *'Which Modems Are Supported?'* section.

If your Network Master is a PC running Bristol DOS-based software tools such as UOI, TMS, on-line AIC, Toolkit, Taskspy, etc, which will be using a modem to dial out to a particular Network 3000-series node:

- The ATOOLS.INI file may be used to configure modem initialization strings. Modems should be set to their factory defaults, typically using an 'AT@F' command. The user should then examine the factory default profile of the modem. (For information on ATOOLS.INI, see the *ACCOL II Reference Manual*, document# D4044.)
- If the factory default profile of the modem uses *data compression* this *must be DISABLED*.
- If the factory default profile of the modem uses *error correction* this *must be DISABLED*.
- Flow control should be set to RTS/CTS.
- Result codes should be enabled and set to NUMERIC.
- *All devices on the line must share the same baud rate*. That is, the baud rate specified in the Communication Setup Menu (or ATOOLS.INI file) for the Bristol software program (Toolkit, UOI, etc.) must be same baud rate as the modems on both ends of the line, and the Slave/Pseudo Slave ports in the Network 3000-series controllers. Bristol Babcock controllers do NOT dynamically change their baud rates.

If you are configuring a Network 3000-series controller to dial *another* Network 3000-series controller, or an Open BSI Workstation:

- The 'Auto-dial' feature must be configured. It allows controllers to dial out. Note, however, that the auto-dial feature is not truly automatic. For example, if a message to a slave node must be transmitted, that, in itself, will not trigger the port to dial the slave. Dialing is activated based on entries in the Dial Control Signal List.
- Master Ports, Slave Ports, and Pseudo Slave Ports can be configured for dialing. If Slave or Pseudo Slave Ports are used, however, dialing will only

establish the connection; the Master must still poll the slaves in order for data to be transmitted.

- Any modem used must go to command mode when Data Terminal Ready (DTR) is OFF for 50 milliseconds. The modem must also use a Carriage Return code of 13 (0DH).

- The ports and modems at both ends must be set to the same baud rate.

Normally, the standard Dial Control Signal List must be configured. If, however, you are using Slave auto-dialing (i.e. the Slaves dial into the Master) and your Network Master and controllers support it, you should consider using the Enhanced Slave Dial Control Signal List, instead, which can significantly improve throughput if you have a large number of Slaves.

- The number of the Dial Control Signal list (or Enhanced Slave Dial Control Signal List) is specified for a particular port using the port's corresponding #DIAL.*nnn* system signal. The table below shows this correspondence between ports and #DIAL signals.

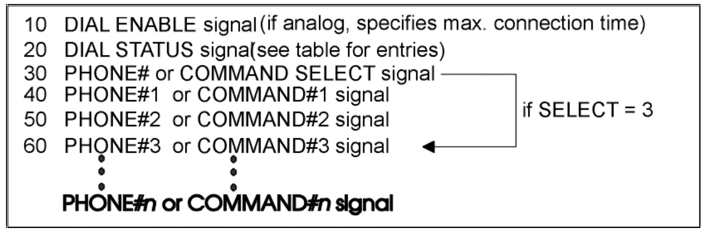
If you are configuring auto-dialing for this port….	You will need to use this system signal to specify the Dial Control List
Port A	#DIAL.000.
Port B	#DIAL.001.
Port C	#DIAL.002.
Port D	#DIAL.003.
BIP 1	#DIAL.004.
BIP 2	#DIAL.005.
Port G	#DIAL.006.
Port H	#DIAL.007.
Port I	#DIAL.009.
Port J	#DIAL.010.

A positive value for the #DIAL.*nnn*. signal specifies that the Dial Control Signal list will be used.
A negative value for the #DIAL.*nnn*. signal specifies that the Enhanced Slave Dial Control Signal List will be used.

- Auto-dialing is described in the '*Auto-Dial Modem Interface*' section of the *ACCOL II Reference Manual* (document# D4044). Highlights from that section are presented below.

Standard Dial Control Signal List

The standard Dial Control Signal List allows dialing to be activated, reports the status of dialing, and specifies the phone numbers, and any special commands which are sent to the modem.



For example, if #DIAL.000. is set to 1, a Dial Control Signal List must be created, similar to the list shown below. The choice of signal names is entirely up to the user.

```
*LIST 1
10  PORTA.ENABLE.DIAL
20  PORTA.STATUS.DIAL
30  PORTA.PHONE.SEL
40  PHONENUM.CALL.RTU1
50  PHONENUM.CALL.RTU2
60  PHONENUM.CALL.RTU3
```

The standard Dial Control Signal list has the following elements:

DIAL ENABLE signal — if a logical signal, this signal enables dialing when ON, and disables dialing when OFF. For a Slave or Pseudo Slave Port, this should be defined as an analog signal, in which case, it indicates the maximum connection time, in seconds. Once a connection has been established, a timer starts which terminates the connection when the specified time has elapsed. Setting this signal to 0 or OFF disables dialing or terminates the connection, if it is in progress. Typically, ACCOL logic would be used to turn ON this signal and activate dialing.

DIAL STATUS signal — is an analog signal which reports error and status information concerning auto-dial operations.

Possible values for this signal are shown in the table, below:

Error/Status Code:	Description:
2	Dialing or Special Command completed successfully
1	Dialing or Special Command in progress
0	Phone hung up
-1	Dialing complete but line was Busy

Error/Status Code:	Description:
-2	Dialing failed (no dial tone when required, no answer, or no response from modem) or Special Command failed (no response from modem or unrecognized response from modem)
-3	Dialing not possible
-4	Special Command failed (“ERROR” response from modem)
-5	Invalid dial control signal list
-6	Invalid Phone #/Command Select (value=0 or > number of entries in Phone#/Command list)
-7	Invalid Phone # / Command (signal not a String)
-8	Invalid Phone # / Command length (String is NULL or > 32 characters)

PHONE#/COMMAND SELECT signal — specifies which command or phone number should currently be used, from those specified beginning with entry 4 in this signal list.

PHONE#/COMMAND signals 1 to *n* — each of these string signals consists of a phone number or (for Hayes-compatible modems) a special modem command.

If you are using a phone number it can include:

- 0-9 any digit from 0 to 9
- W wait for 4 seconds, then proceed
- T change to Touch Tone dialing
- P change to pulse dialing (default is Tone). Pulse remains in effect for only a single dialing sequence.
- \- dashes may be added for readability; they are NOT sent to the modem.

Example: T-9-555-1212

If you are specifying a special command for a Hayes-compatible modem, you must precede it with an exclamation point ‘!’ in the first character position. The remainder of the string is sent, and automatically terminated with a carriage return (13 or 0DH, not shown below):

Example: !AT &D2 &R1 SO=1 &W0

Enhanced Slave Dial Control Signal List (PES03, PEX03, PLS03, PLX03, LS501, RMS04, or newer firmware ONLY)

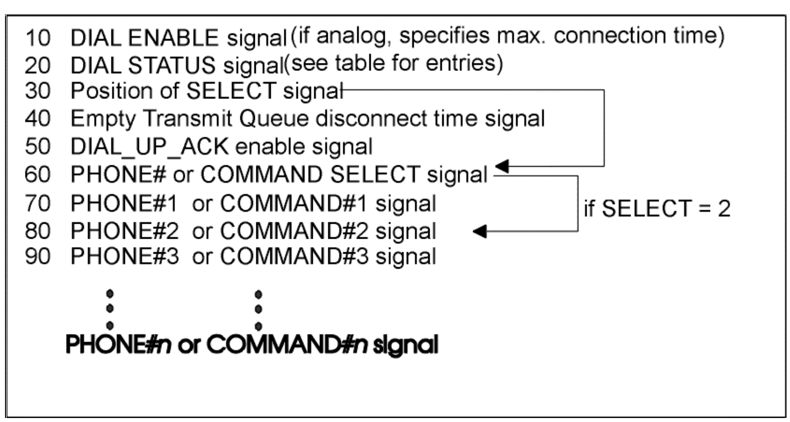
The Enhanced Slave Dial Control Signal List may be used *instead of* the standard Dial Control List, for Slave Port dialing only. In addition to the types of entries in the standard list, it allows the user to specify a delay in disconnecting if the slave runs out of data to send, and a dial-up acknowledgment signal.

When standard Slave dialing is used, a slave node dials into its Master. The Master *has no way of knowing which slave is dialing in*, therefore, it begins polling *all* of its slaves, one by one, until it polls the slave dialing in, which can then answer. This is equivalent to the situation of hearing a ‘knock’ on your door, but the person knocking cannot speak until you call out *their* name. The only way you can find out who it is, then, is to call out the names of everyone you know, until you reach the name of the person doing the knocking, who can then answer, as in ‘Is that you Alice?’, ‘Is that you Bob?’, ‘Is that you Carol?’, and continuing this until you say ‘Is that you Margaret?’ and Margaret answers ‘Yes it’s me!’ Depending upon the number of acquaintances you have, this could take a long time.

Enhanced Slave dialing is a much more efficient method. In this situation, when the Master begins to poll its first slave node, the slave dialing in will answer (even if it is not the first slave) and the Master will then accept *its* data immediately, rather than polling its other slaves. To follow the previous analogy, when you hear a knock on the door, and start asking ‘Is that you Alice?’, the person knocking will answer ‘No! it’s not Alice, it’s Fred' (or whatever their name is). Now you can say 'Hi Fred, what do you have for me?' and Fred responds with data.

This enhanced method, which uses the DIAL_UP_ACK enable signal, greatly increases throughput in a system with many slave nodes, by eliminating extraneous polling to nodes that aren’t dialing in.

The figure, at right, shows the organization of the Enhanced Slave Dial Control Signal List. It is similar to the Dial Control Signal List except it has some additional signals which include the Empty Transmit Queue disconnect time signal, and the DIAL_UP_ACK enable signal.



For example, if #DIAL.003. is set to '-5', an Enhanced Slave Dial Control Signal List must be created, similar to the list shown below. The choice of signal names is entirely up to the user.

```
*LIST 5
10  PORTD.ENABLE.DIAL
20  PORTD.STATUS.DIAL
30  POSITION.OFSEL.SIG
40  MAXEMPTY.QUEUE.TIME
50  PORTD.DIALUP.ACK
60  PORTD.PHONE.SEL
70  PHONENUM.CALL.RTU1
80  PHONENUM.CALL.RTU2
90  PHONENUM.CALL.RTU3
```

The Enhanced Slave Dial Control Signal List has the following elements:

DIAL ENABLE signal — (see description under Dial Control Signal List)

DIAL STATUS signal — (see description under Dial Control Signal List).

POSITION of SELECT — this analog signal specifies the position in the list of the PHONE#/COMMAND SELECT signal. (The position of this signal has been made variable to allow for special purpose signals to be added to the list, such as the Empty Transmit Queue Disconnect signal, the DIAL_UP_ACK Enable signal, and any signals related to future enhancements.)

In List 5, shown on the next page, POSITION.OFSEL.SIG is the POSITION of SELECT signal.

```
*LIST 5
10  PORTD.ENABLE.DIAL
20  PORTD.STATUS.DIAL
30  POSITION.OFSEL.SIG
40  MAXEMPTY.QUEUE.TIME
50  PORTD.DIALUP.ACK
60  PORTD.PHONE.SEL
70  PHONENUM.CALL.RTU1
80  PHONENUM.CALL.RTU2
90  PHONENUM.CALL.RTU3
```

In this case, it should be set to 6 because there are two special purpose signals (MAXEMPTY.QUEUE.TIME and PORTD.DIALUP.ACK) which pushes the PHONE#/COMMAND SELECT signal (PORTD.PHONE.SEL) to position 6.

If there are no special purpose signals to be included in the list, this signal should be set to '4'.

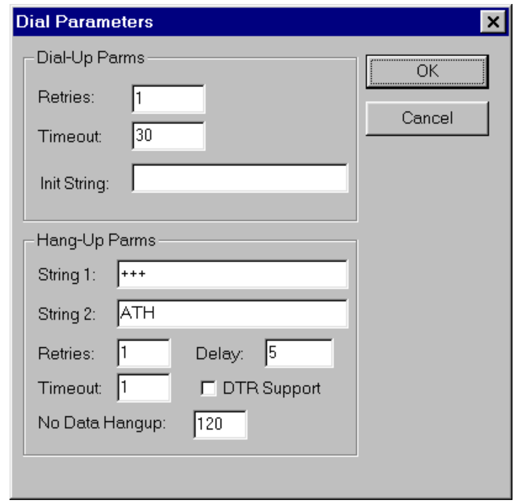
EMPTY TRANSMIT QUEUE DISCONNECT TIME	this is an analog signal which specifies how long, in seconds, a connection should be maintained after the slave node's transmit queue becomes empty (i.e. the slave has no more data messages to send to its master). If this signal is not included, is of the wrong type, or its entry is less than or equal to zero, the default disconnect time will be 10 seconds.
DIAL_UP_ACK Enable	this is a logical signal. When set ON, it allows the Slave node to acknowledge the first poll message issued by its Master, whether or not the poll is for this slave. This allows the slave node dialing in to immediately send data to the master, rather than wait for the master to poll all of its nodes until it reaches the correct slave node. This feature significantly increases throughput on Slave dial-up.
PHONE#/COMMAND SELECT signal	(see description under Dial Control Signal List).
PHONE#/COMMAND signals 1 to *n*	(see description under Dial Control Signal List).

<u>If your Network Master is a PC running Open BSI (2.x) which will use a modem to dial Top-Level Nodes:</u>

You must be running the Windows 95/NT version of Open BSI. Windows 3.1 users CANNOT use Open BSI to dial.

Open BSI can be configured to send an initialization string to the modem. Any other modem configuration must be performed using any modem configuration software provided by the modem manufacturer, or if applicable, through switch settings on the modem.

Dial parameters are configured using the Dial Parameters dialog box, which is accessible from the **[Dial Parms]** push button of the Communication Line Parameters dialog box, when the **"Dial"** option is selected for the line.



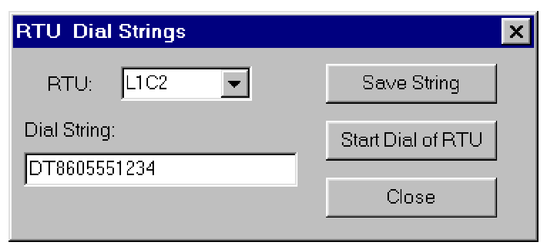
The *carrier detect* line from your modem to the Open BSI workstation (PC) must operate consistently with the modem's carrier detect setting (i.e. it CANNOT be left constantly on.)

The modem's connection baud rate must be constant, and must be equal to the baud rate selected for the workstation in the Open BSI Setup Tool's Communication Line Parameters dialog box, and the baud rate selected for the controller's slave port.

The modem's transmit and receive *flow control* settings, and *data compression* settings must be consistent with those at the RTU's modem.

The modem should be set to ignore the Ready to Send (RTS) signal from the workstation.

Local echo of codes should be turned ON.



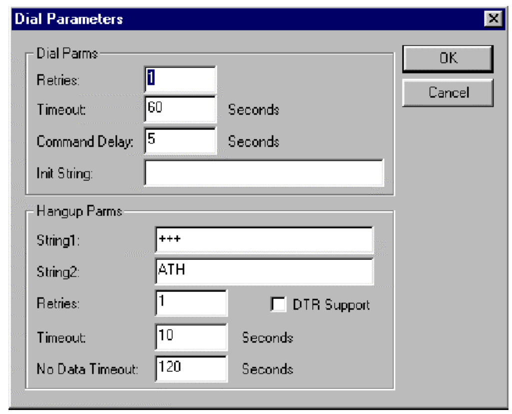
When configuring the dial string in the Open BSI Setup Tool's RTU Dial Strings dialog box, include a 'DT' or 'DP' modem command in the string.

Open BSI immediately precedes the dial string with the 'AT' modem command.

If your Network Master is a PC running Open BSI (3.0 or newer) which will use a modem to dial Top-Level Nodes:

Open BSI can be configured to send an initialization string to the modem. Any other modem configuration must be performed using any modem configuration software provided by the modem manufacturer, or if applicable, through switch settings on the modem.

Dial parameters are configured using the Dial Parameters dialog box, which is accessible from the **[Dial Parameters]** push button of the BSAP Line - Advanced Parameters dialog box, when the **“Dial”** option is selected for the line.



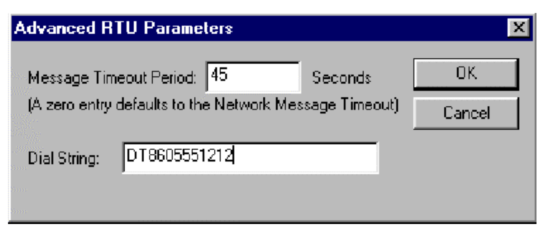
The *carrier detect* line from your modem to the Open BSI workstation (PC) must operate consistently with the modem’s carrier detect setting (i.e. it CANNOT be left constantly on.)

The modem’s connection baud rate must be constant, and must be equal to the baud rate selected for the workstation in the Comm Line Wizard (NetView) and the baud rate selected for the controller’s slave port.

The modem’s transmit and receive *flow control* settings, and *data compression* settings must be consistent with those at the RTU's modem.

The modem should be set to ignore the Ready to Send (RTS) signal from the workstation.

Local echo of codes should be turned ON.



When configuring the dial string in the Advanced RTU Parameters dialog box (accessible from the third page of the RTU Wizard in NetView), include a ‘DT’ or ‘DP’ modem command in the string.

Open BSI immediately precedes the dial string with the ‘AT’ modem command.

Configuration Issues For Radios

Bristol Babcock Network 3000 equipment has been used successfully with several different brands and types of radios.

The configuration of the radio interface varies depending upon the type of radio, particularly with respect to cabling and timing. A full discussion of radio configuration is beyond the scope of this book, however, here are some items to be aware of:

- Some radios have fixed speed internal modems and some synthesized. Most radios today are configurable in several different ways including frequency, baud rate, and pre and post data delays. As far as cabling, all radios use the full RS-232 compliment except the Spread Spectrum type including Microwave Data Systems (9810) and Freewave brands.

- Spread Spectrum radios only use TxD, RxD, and Signal Ground. The RTU at each of these sites must have either an internal jumper on the connector between RTS and CTS on the RTU end or the loopback switches on the communications engine board enabled.

- The MDS licensed radios (2310, 4310, 4710, 9710, etc.) and DataRadio (formally EF Johnson) use TxD, RxD, RTS, CTS, DSR, DCD, DTR, and Signal Ground and no loopback enabled. The master radio companions of these radios also use the full compliment. Timing parameters of these radios are different too. MDS radios are faster as far as key speed and receive speed than DataRadio are.

- If there is a repeater in the system, some configuration parameters can change. A tail-end-link setup in a system is another scenario which will include a modem sharer in addition to two radios working together to send the data further down the road similar to a repeater. The modem sharer is necessary in the data line if there is a co-located RTU at the site.

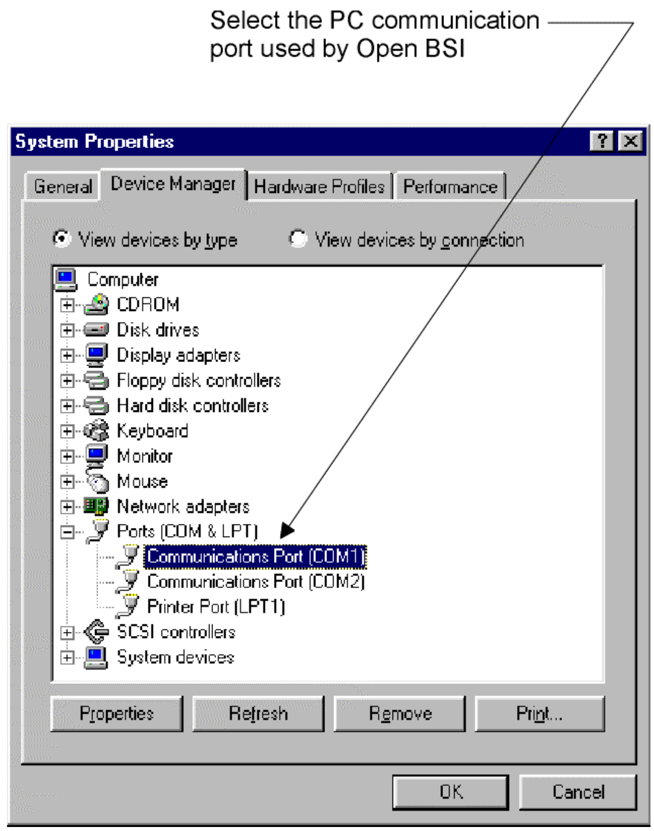
For additional technical assistance in using radios with Bristol equipment, contact our Communication Technologies division in Orlando, FL at the number listed on page ii.

Radio Tip - If messages are truncated when using RTS/CTS, try this...

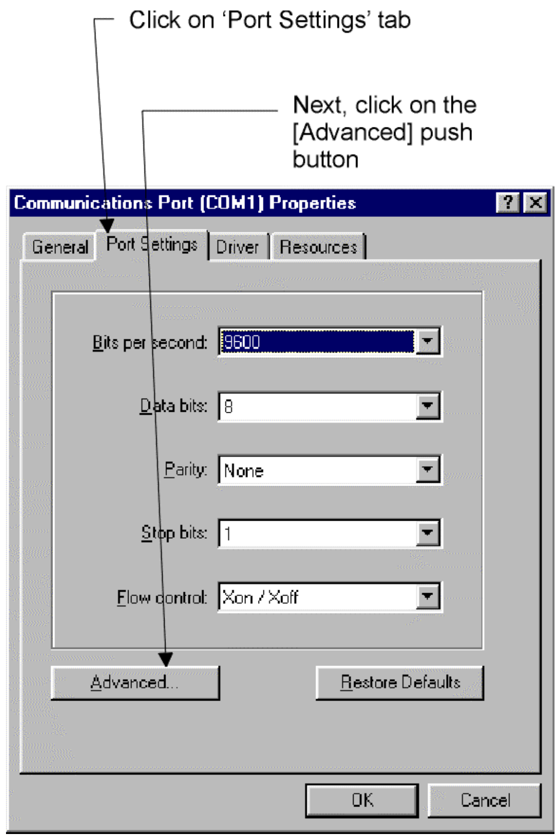
If you are using RTS/CTS with radios, and encounter a problem where the Open BSI workstation can transmit, but RTUs are unable to respond, it could be related to PC port configuration in Windows 95 which results in messages being truncated.

If this problem is occurring, go to "**Start**" then "**Settings**" and "**Control Panel**" to call up the Windows 95 control panel. Next double-click on the 'System' icon. The System Properties dialog box will appear. Click on the 'Device Manager' tab.

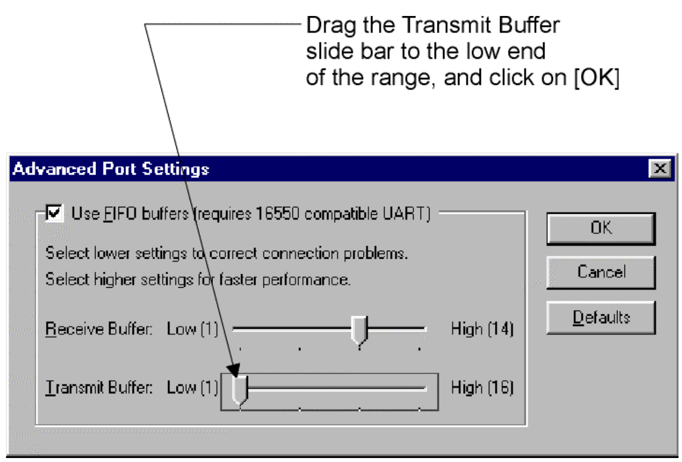
Click on the plus sign '+' next to the 'Ports (COM & LPT)' selection. A list of ports will appear. Click once on the port used for Open BSI communications (typically COM1 or COM2), then click on the **[Properties]** push button.



The Communication Port Properties dialog box will appear; click on the 'Port Settings' tab, then click on the **[Advanced]** push button to call up the Advanced Port Settings dialog box.



In the Advanced Port Settings dialog box, drag the 'Transmit Buffer' slide bar to the low end of its range, and click on the **[OK]** push button.



Then choose the **[OK]** push button in both the Communication Port Properties dialog box, and in the Device Manager page, to exit the control panel and save the settings.

Reboot your PC for the new settings to take effect.

Chapter 5 - Guidelines For Configuring the IP Data Link

The **data link** refers to the actual mechanism by which data flows through the network. This includes the port and communication line itself, and for purposes of our discussion, also includes the software configuration for the ports and the communication line. The functioning of the data link is completely independent of the applications which use the link (HMI/SCADA packages, IP_Client/IP_Server Modules, etc.)

A Bristol Babcock network using the IP protocol operates like any other IP network.[1] Data messages from a node are divided up into small pieces called **packets**. The packets are transmitted through the communication line (such as an Ethernet cable) and are then re-assembled into the full data message when they reach the destination node. Currently, the Ethernet port is the only type of IP connection supported by Bristol Babcock networks.

Just as many BSAP network terms do NOT apply in IP networks, many of the concepts associated with BSAP data link configuration (polling, etc.) do NOT apply in IP data link configuration. (These terms still apply, however, within a BSAP sub-network underneath an IP node.)

The main configuration activities involved in IP data link configuration involve setting up communication ports. These include:

- Setting up the Network Host PC (NHP) for this network. (This was discussed as part of Chapter 3.)
- Defining a Master port in any IP node which serves as a **network master** to a BSAP sub-network. (If applicable, see Chapter 4; if you are not including a BSAP sub-network, this activity is unnecessary).
- Declaring one or more IP ports in the ACCOL load of each IP node, using ACCOL Workbench.
- Specifying the characteristics of the IP port(s) in the RTU's FLASH memory using the "**RTU Comm Config**" feature of LocalView.

---

[1] *The specific type of IP protocol Bristol Babcock uses is called the User Datagram Protocol (UDP).*

Declaring an IP Port in ACCOL Workbench

An IP Port is declared in the *COMMUNICATION PORTS section of the ACCOL source file.

The 386EX Protected Mode units contain two Built-In Ports (BIP)s, four to eight serial ports (Ports A through D, and, optionally, G through J) as well as an optional Ethernet Port. Any of these ports can be configured as the IP port.

> IMPORTANT
>
> Although ACCOL Workbench will allow you to configure serial ports BIP 1, BIP 2, Port A through D, and G through J as Internet (IP) ports, PLS03/PLX03/PES03/PEX03 firmware WILL ONLY SUPPORT IP COMMUNICATION VIA THE ETHERNET PORT; the serial ports will be available for IP communication in a *future* firmware release.
>
> NOTE: Although an IP node can contain more than one IP Port, once it has an IP Port defined, it CANNOT include any of the following port types: Slave, VSAT Slave, or Serial CFE.

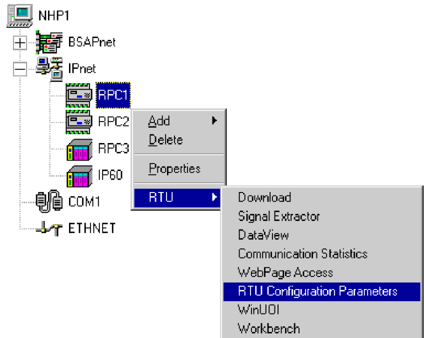
Examples of syntax for the ACCOL source file are shown below:

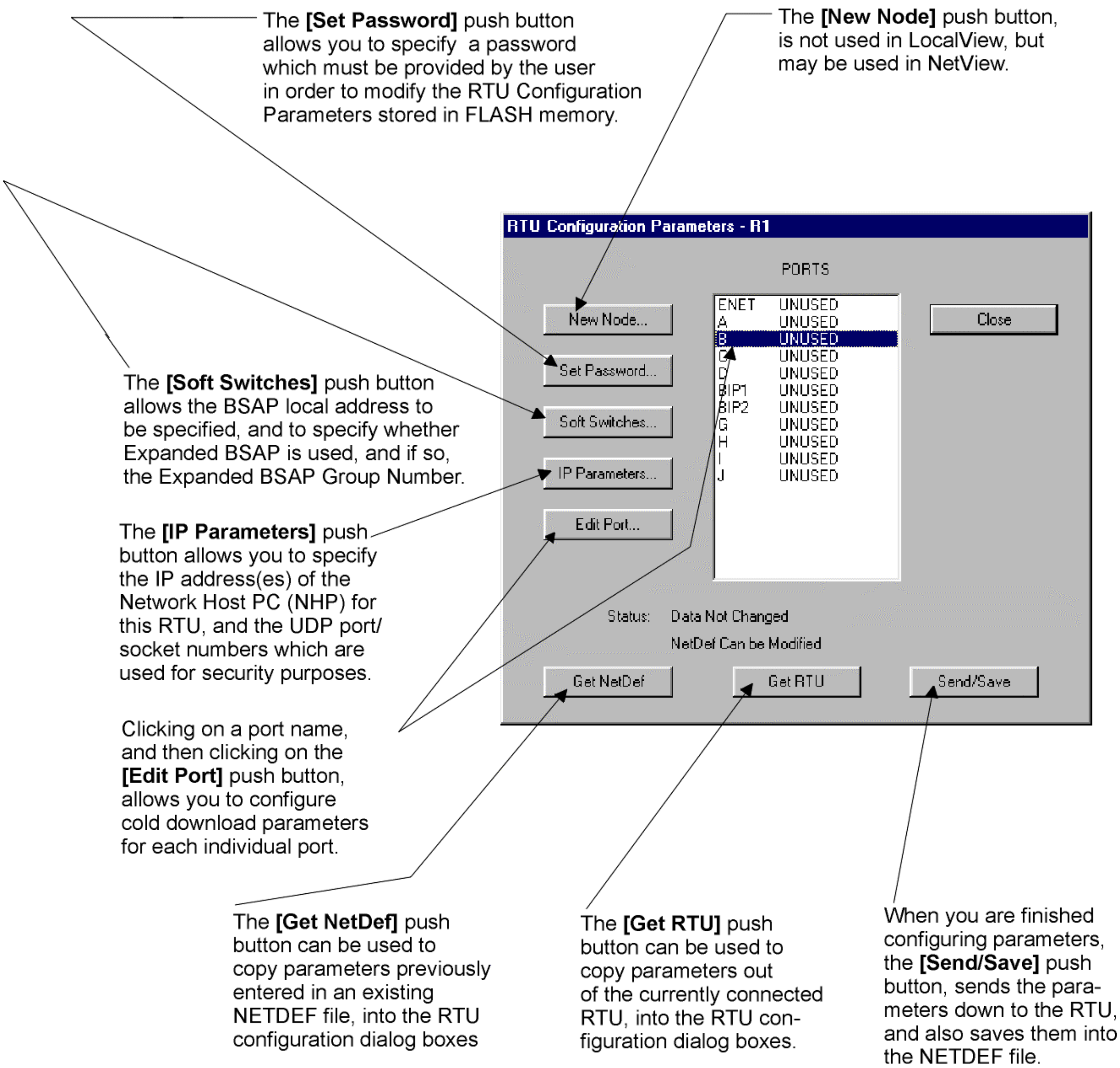
```
BIP_1     IP
BIP_2     IP
PORT_A  IP
PORT_B  IP
PORT_C  IP
PORT_D  IP
PORT_G  IP
PORT_H  IP
PORT_I  IP
PORT_J  IP
ETHRNT IP
```

No other configuration is necessary within the ACCOL source file. IP Port baud rate and other characteristics must be defined using the "**RTU Comm Config**" feature of LocalView.

Specifying the Characteristics of the IP Port(s)

The characteristics of the IP ports are specified using the "**RTU Comm Config**" feature of LocalView[2]. (NOTE: LocalView must be used for initial IP configuration; once global communications have been established, the same feature may be used in NetView to view/modify the IP port characteristics.)





---

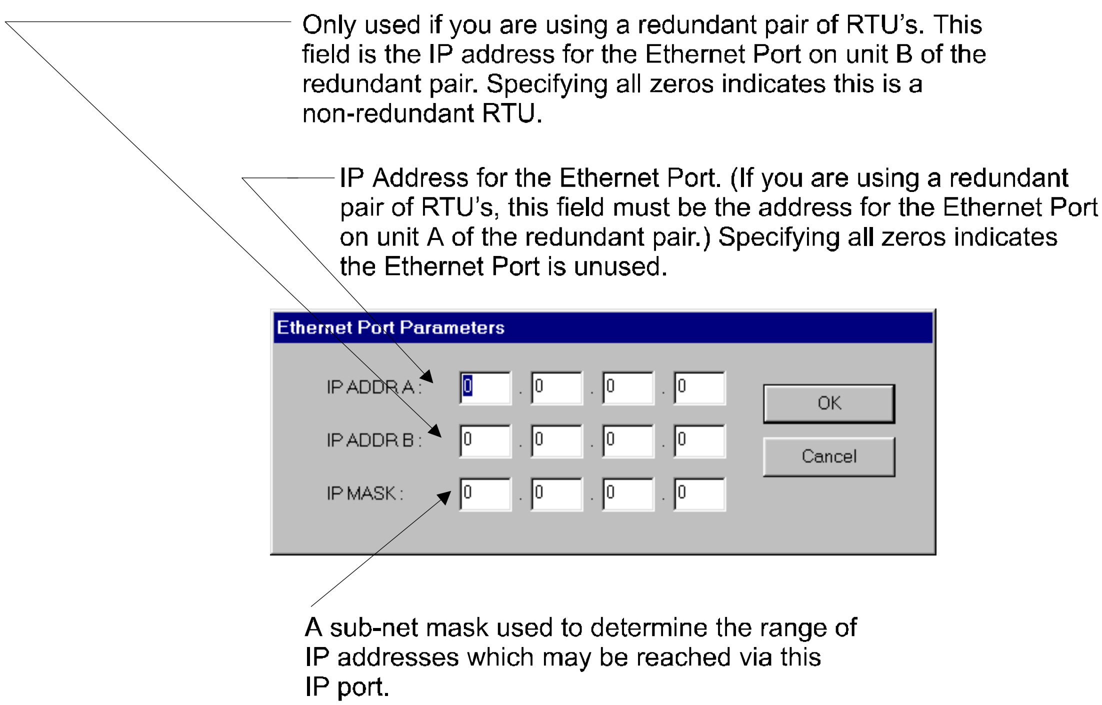
[2] *LocalView always uses BSAP to communicate, even though you are configuring IP port parameters with it. A full description of the RTU Comm Config feature is included in the Open BSI Utilities Manual (document# D5081).*

Your IP port can either be the Ethernet Port, or one of the serial ports. If you are using the Ethernet Port as your IP port, please read '*Setting Up the Ethernet Interface*'. If you will be using serial port(s) for your IP port, please read '*Setting Up the Serial Ports*'. In either case you must also configure the general IP parameters for the system. These are described under '*Defining the IP System Parameters*'.

> IMPORTANT
>
> The IP port characteristics, soft switches, etc. which you set or change using LocalView's "**RTU Comm Config**" feature do not come into effect until *after* the RTU has been reset or downloaded with a new ACCOL load. Once initial configuration has been completed using LocalView, and global communications have been established, RTU parameters may then be altered in the same way, using NetView. For more information on using LocalView and NetView, see the *Open BSI Utilities Manual* (document# D5081).

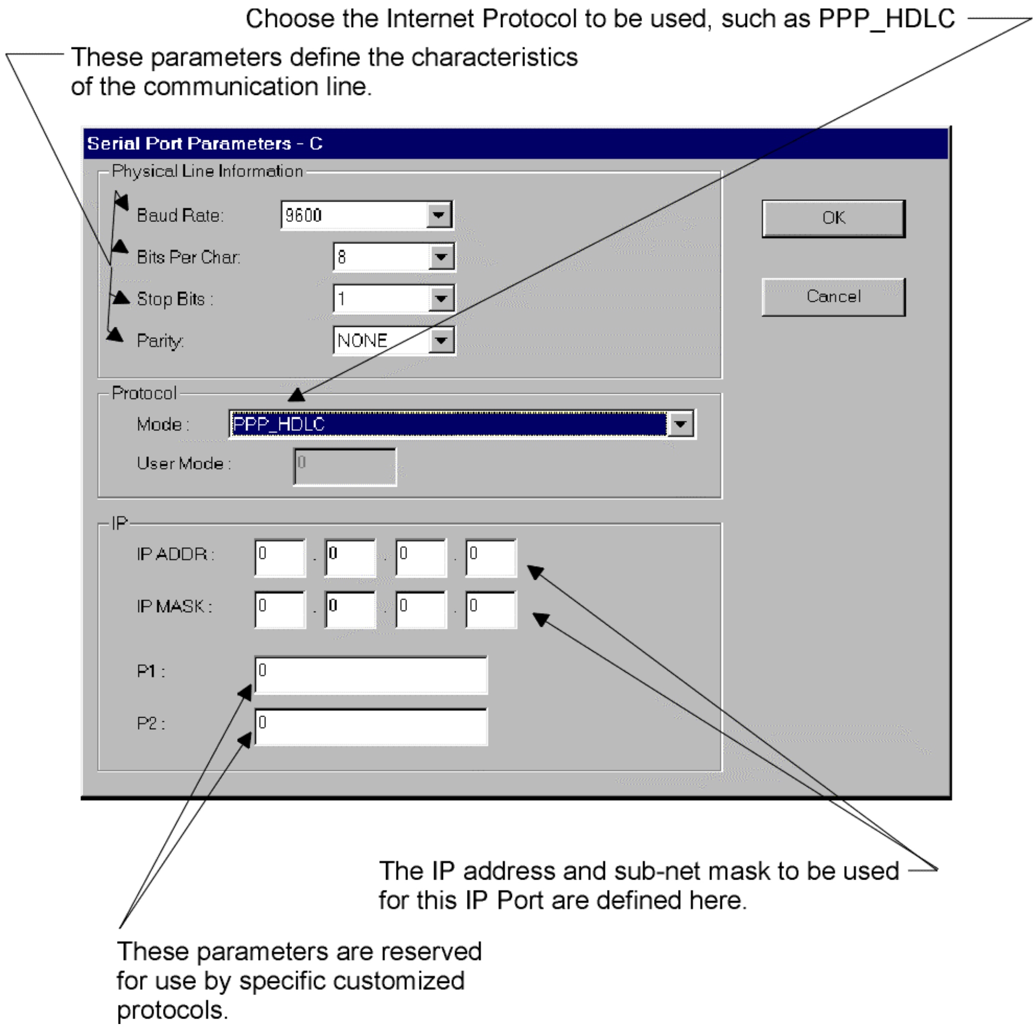
Setting Up the Ethernet Interface

Ethernet parameters are only configured if you are using the Ethernet Port as your IP Port; they do NOT apply if you are only using serial IP ports.



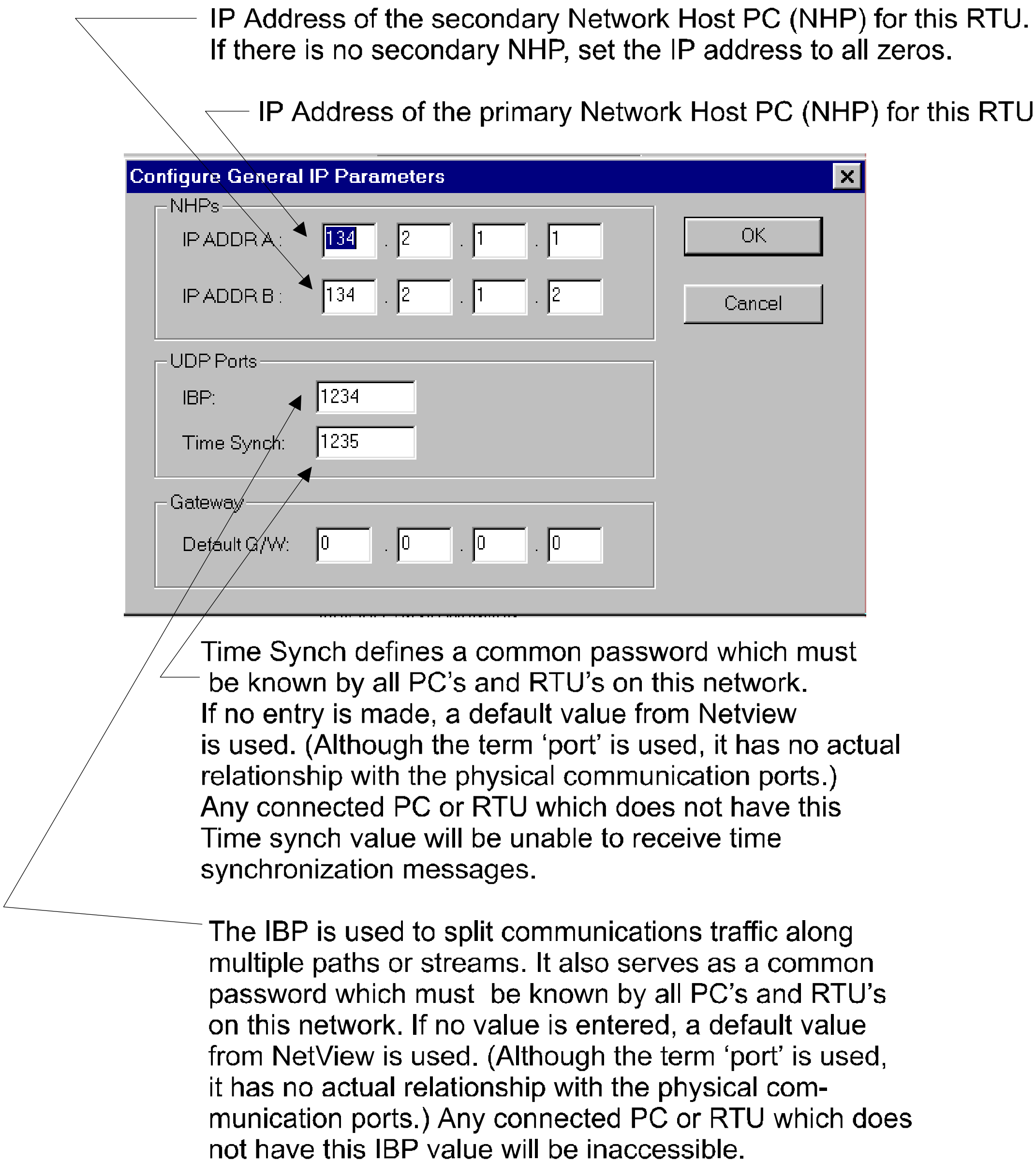
Setting Up the Serial Ports (NOT supported in PES03/PEX03 PLS03/PLX03 firmware)

Be sure to select an Internet Protocol (such as 'PPP_HDLC' or 'User Mode' if you have created your own Internet Protocol implementation) for the port; choosing either 'BSAP' or 'EBSAP' will disable the Internet parameters.



Defining the IP System Parameters

Once your IP ports have been defined (either Ethernet or Serial) you must also define certain IP System Parameters.



Chapter 6 - Application Notes

As discussed earlier in this manual, an **application** is a program or function that makes use of the data link to transmit messages. Among the commonly used applications are elements of the ACCOL load itself such as Master Modules used for peer-to-peer communication. Open BSI Utilities or DOS-based tools such as UOI are applications because they make use of the data link to collect data from controllers in the network. HMI/SCADA packages such as OpenEnterprise, Intellution® FIX®, and Iconics Genesis are also applications.

NOTE:
This section will discuss various applications, and provide configuration hints and tips. *Not all information presented is strictly related to communications*, but it is nonetheless important when trying to improve system performance.

Peer-to-Peer Communication (BSAP Master/Slave Modules)

Peer-to-peer communication is the transfer of signal list or data array data from one Network 3000 controller to another. This transfer is performed using Master Modules (or EMaster Modules) and Slave Modules.

Peer-to-peer communication is typically used in situations where one controller must make some logical decision based on data from another controller.

Another reason to use peer-to-peer communication is if greater user control is required over when communications occur. The user can create ACCOL logic to activate the Master Module only at certain times or when certain data is ready.

IMPORTANT
Do NOT confuse Master/EMaster Modules and Slave Modules with Master nodes/Ports and Slave nodes/Ports. They are NOT the same things.

The following are certain things the user should be aware of when setting up peer-to-peer communication:

- Users must designate which signal list or data array must be transferred. Signal lists on both ends must have the same number of entries, and data arrays on both ends must have identical row/column dimensions. Peer-to-peer communication transmits data values only; it does not, for example, transmit the signal names, inhibit/enable status information, etc.

- A given controller can participate in peer-to-peer communication with its Master node, any of its slave nodes, and any sibling nodes (nodes that share the same Master).

- It's recommended that Master Module(s) be separated into a dedicated task; this allows easy tuning of the task rate, without affecting other modules.

- The EMaster Module is necessary for peer-to-peer communication with Expanded Addressing Slave nodes (slaves of 'virtual nodes') but can also be used in standard BSAP networks to communicate with the slave of a slave node.

- For maximum throughput, it is recommended that Master Modules (but NOT EMaster Modules) be placed in *lower-level* nodes with respect to their associated Slave Modules. That is, the Master Modules should be placed in the Slave *nodes* and the Slave Modules should be placed in the Master *nodes*. The main reason for this is to distribute the Master Modules among the slave nodes; thereby eliminating the situation of having several Master Modules competing for a single node's resources (CPU time, buffers, etc.) If, instead, the Master Modules are put in the Master node, the recommended task rate for the task containing them is *half* the poll period for the corresponding Master Port.

- It is recommended that *at least* one additional communication I/O buffer be declared in the ACCOL load *for each Master/EMaster Module* in the load.

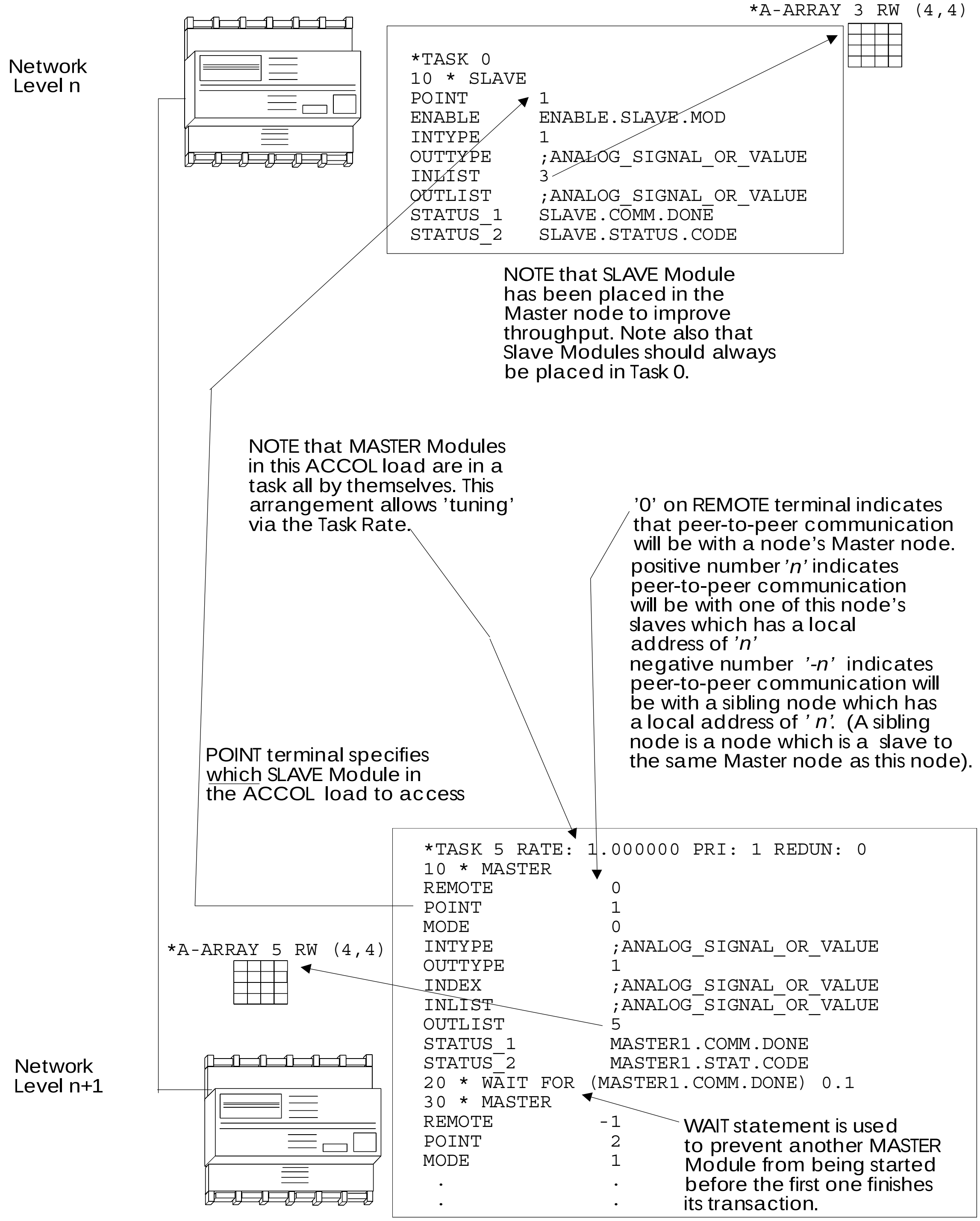
- Master/EMaster Modules function differently from other modules in that their communication transactions may still be underway when execution of the ACCOL Task proceeds to the next module in sequence. Because of this, if an ACCOL module (say a Calculator) follows a Master Module because its calculations depends on data being collected by the Master Module, appropriate ACCOL logic must be added to ensure that the data collection has been completed *before* execution of the Calculator; *otherwise the Calculator will be using OLD DATA*. This ACCOL logic would consist of a WAIT FOR (DONE) statement after the Master Module, with "DONE" being defined as a logical signal, on the STATUS_1 terminal of the Master Module. This signal is turned ON when communications are complete, and is OFF when communications are in progress or off. DO NOT INCLUDE A TIMEOUT VALUE in the WAIT statement; the Master Module has its own built-in timeout of two minutes, which may be modified in ACCOL.[1] This same WAIT FOR logic should be used in situations where multiple Master Modules are in the same ACCOL load; execution of a Master Module should not be started until communications have been completed by the previous Master Module.

- It is recommended that the STATUS_2 signal on the Master Module be made an analog alarm signal, with a high alarm limit of 2, and a low alarm limit of -1; whenever this signal is in an 'alarm state', the user will know that peer-to-peer

[1] *To make this timeout modification, see instructions under the MODE terminal description in the 'Master/EMaster' section of ACCOL II Reference Manual (document# D4044).*

data is suspected of containing errors. See the '*Master/EMaster*' section of the *ACCOL II Reference Manual* (document# D4044) for details on interpreting particular error values.

The figure, below, shows a typical peer-to-peer communication setup:



Peer-to-Peer Communication (IP_Client / IP_Server Modules)

Peer-to-peer communications using IP_Client / IP_Server modules allows data from ACCOL structures such as signal lists and data arrays to be transferred from one IP-capable remote process controller to another IP-capable remote process controller. Peer-to-peer communication is typically used in situations where one controller must make some logical decision based on data from another controller.

Another reason to use peer-to-peer communication is if greater user control is required over when communications occur. The user can create ACCOL logic to activate the IP_Client Module only at certain times or when certain data is ready.

NOTE
Currently, the only IP-capable Network 3000 RTU's are 386EX Protected Mode versions of the DPC 3330 or DPC 3335, with PES03/PEX03 or newer firmware. To perform peer-to-peer communication in non-IP RTU's you must use BSAP Master/EMaster/Slave modules.

The IP_Client Module initiates all such communication transactions by contacting an IP_Server Module in another IP-capable controller (the target RTU). The IP_Client Module requests either read-only, or read-write access to various arrays, signal lists, etc. in the target RTU. The IP_Server can grant those requests for access, or deny them, based on how its module terminals are configured.

The IP_Client Module can request access to entire structures (an entire data array, or an entire signal list) or it can request access to only a set of contiguous signals in the signal list, or contiguous elements in a data array based on the entries in its SERVR_INDEX and CLNT_COUNT terminals.

The IP_Server Module in the target RTU can restrict access to the structures under its control using its LIST_DB, AARRAY_DB and LARRAY_DB terminals. It can also restrict access to IP_Clients whose node names or addresses appear in the KNOWN_IP_NODES list; any unlisted nodes will be denied access by the IP_Server.

An example of IP_Client/Server communication is as follows:

There are two remote process controllers, with IP node addresses of 120.0.0.1, and 120.0.0.3, respectively (see figure on page 6-6). The user wants to read the first three entries of Signal List #5 in the 120.0.0.3 unit (the server), into positions 8, 9, and 10 of Signal List #3 of the 120.0.0.1 unit (the client). This example outlines the configuration required to perform this data transfer.

Configuring the IP_Client Module:

When configuring the IP_Client Module, the target RTU for which access is desired must be identified on the REMOTE terminal signal. In this case, the target RTU will be identified by its address of 120.0.0.3.

Because read access to that RTU is required, the ACCESS_MODE signal must be set to 1 (DBREAD).

Because the type of structure being read is a signal list, the STRUCT_TYPE signal must be set to 1 (Signal List).

Since data must be read from Signal List #5 in the target RTU, the SERVR_STRUCT_NO terminal signal must be set to 5. Since only the first 3 entries in Signal List #5 are to be read, then the SERVR_INDEX terminal signal must be set to 1 (i.e. start with the first position in the list), and the CLNT_COUNT terminal signal must be set to 3 (i.e. read only the first 3 entries of the list).

In addition, the signal list data being copied in from the target RTU is to be stored in positions 8, 9, and 10 of Signal List #3. Therefore, the CLNT_STRCT_NO must be set to 3 (for Signal List #3), and CLNT_INDEX must be set to 8 (because position 8 is the first position where data must be stored).

NOTE: Always check for successful completion via the STATUS terminal(s) before attempting to use peer-to-peer data.

Configuring the IP_Server Module

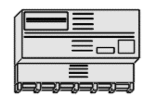
IP_Server configuration is somewhat simpler. If the LIST_DB and KNOWN_IP_NODES terminals were left unwired, there would be almost no configuration. To illustrate how the IP_Server Module allows restrictions to be placed on access, these terminals will be wired.

The LIST_DB terminal will specify an analog data array which will identify which signal lists are to be made accessible to IP_Clients. In this case, we must include Signal List #5 in that array, and specify that it will be available for read access.

The KNOWN_IP_NODES terminal will specify a signal list which will contain signals that identify the RTU's which will be granted access via IP_Client requests. In this case, we must identify the RTU with address 120.0.0.1.

NOTE: Always check for successful completion via the STATUS terminal(s) before attempting to use peer-to-peer data.

120.0.0.1



IP_Client Module Definition

TERMINAL NAME	SIGNAL NAME	SIGNAL TYPE	SIGNAL VALUE
REMOTE	TARGET.RTU.	String	120.0.0.3
RESOLV_NAME	RESOLVE.NOW.	Analog	0
SERVR_ID	IPSERV.NUM.	Analog	1
ACCESS_MODE	ACCESS.MODE.	Analog	1
RESP_TMO	RESPONSE.TIMOUT.	Analog	1200
STRUCT_TYPE	STRUCT.TYPE.	Analog	1
SERVR_STRUCT_NC	SERVER.STRUCT.NO.	Analog	5
SERVR_INDEX	SERVER.INDEX.	Analog	1
ACCESS_TYPE	ACCESS.TYPE.	Analog	1
SERVR_SELECT	*leave unwirec*		
CLNT_STRCT_NO	CLIENT.STRUCT.NO	Analog	3
CLNT_INDEX	CLIENT.INDEX.	Analog	8
CLNT_SELECT	*leave unwirec*		
CLNT_COUNT	NUM.OF.ELTS	Analog	3
STATUS_1	COMM.DONE.	Logical	*set by module*
STATUS_2	COMM.STATUS.	Analog	*set by module*

Signal List #3

1
2
3
4
5
6
7
8
9
10

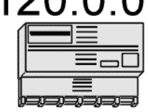
Signal List #5

1
2
3
4
5
6
7
8
9
10

IP_Server Module Definition

TERMINAL NAME	SIGNAL NAME	SIGNAL TYPE	SIGNAL VALUE
SERVR_ID	SERVER.NUM.	Analog	1
LIST_DB	AVAIL.LISTS.ARE	Analog	43
AARRAY_DB	*not needed for this example*		
LARRAY_DB	*not needed for this example*		
ARCHIVE_DB	*not needed for this example*		
KNOWN_IP_NODES	ALLOW.ACCESS.TO	Analog	1
RESOLV_NAME	RESOLVE.NAMES.NOW	Analog	0
STATUS_1	COMM.DONE.	Logical	*set by module*
STATUS_2	COMM.STATUS.	Analog	*set by module*

120.0.0.3



Signal List #1

1	OUR.CLIENT.1	String	120.0.0.1

Analog Array #43

5	1

Setting Task Rates

The **task rate** specifies how often (in seconds) the controller will attempt to start execution of an ACCOL task.[2] This determines how fast the modules in a particular ACCOL task are executed, and how fast data is processed by that task. If the task includes process I/O modules (ANIN, ANOUT, DIGIN, DIGOUT etc.) it also affects how fast data is collected from (and sent to) process instrumentation.

To maximize throughput, tasks should be set to execute as quickly as possible, *so long as they do NOT result in task* ***slippage***. Slippage is a condition in which a task cannot run at its scheduled time because it is still attempting to complete the previous execution. Slippage is reported via the #RCNT.*nnn* system signals (where *nnn* is the task number).

If peer-to-peer communication is being configured in a Master Node (and Master Modules are placed in the Master node) the task rate for the task containing the Master Module(s) should be set to one half the poll period of the Master port.

Task rates can be tuned dynamically (via DataView, Toolkit, etc.) using the #RATE.*nnn*. system signal (where *nnn* is the task number.)

Setting the task rate to 0 stops execution of the task.

The amount of time it takes for a task to execute can be measured using the #RTTIME.000. and #RTTIME.001. system signals. See the '*System Signals*' section of the *ACCOL II Reference Manual* (document# D4044) for details.

Setting Task Priorities

In ACCOL, multiple tasks execute concurrently. If two tasks are scheduled to run at the same time, the one with the higher priority will be executed ahead of the lower priority task. Priority is set using the #PRI.*nnn*. system signals (where *nnn* is the task number) and can range from 1 to 64.

The aspect of task priority most relevant to network communications is to make sure that your ACCOL tasks are NOT given a priority higher than system-level tasks which handle communications.[3] To ensure this, do NOT assign an ACCOL task a priority greater than 32.

---

[2] *For more information on task rates, see 'An Introduction to ACCOL' (document# D4056) for details.*
[3] *For a list of system tasks priorities, see the 'Task' section of the ACCOL II Reference Manual (document# D4044). For more information on task priority see 'An Introduction to ACCOL' (document# D4056).*

ACCOL Load Design Issues

There are numerous factors involved in the design of your ACCOL load, which can potentially affect communications throughput.

For example, if you are doing time-critical calculations, for example, massive array manipulations) at the same time when communications are to occur, a conflict over system resources can result, causing the calculations to be delayed. If possible, schedule the calculations to occur at some time other than when data collection is scheduled to occur.

As another example, if there are many signals in your system which change infrequently or are of lesser importance, you may want to consider collecting them via Report By Exception (RBE), which is supported by OpenEnterprise, Enterprise Server or Intellution® FIX®. RBE reduces the amount of data flowing on the data link at any one time.

A full list of these sorts of design issues is beyond the scope of this document.

Remote Process I/O

Remote process I/O involves connecting RIO 3331 Remote I/O Racks to a DPC 3330/DPC 3335 or RTU 3310 model controller in order to allow I/O to be physically separate from the controller itself. Users should make sure the I/O point count from all of the boards in the RIO 3331s *and* in the host unit, does NOT exceed that of a fully populated host. Numerous other factors can affect RIO system operation as well. Users are urged to test their desired system configuration to verify that it provides adequate performance.

One technique which may *slightly* improve throughput in a unit using remote process I/O is to group the remote input modules (RANIN, RDIGIN, etc.) together in the task. Because all remote I/O data from all racks is requested simultaneously, keeping the remote I/O modules close together in the task may allow a single request for all data to satisfy all the modules, without causing an additional request due to stale data.

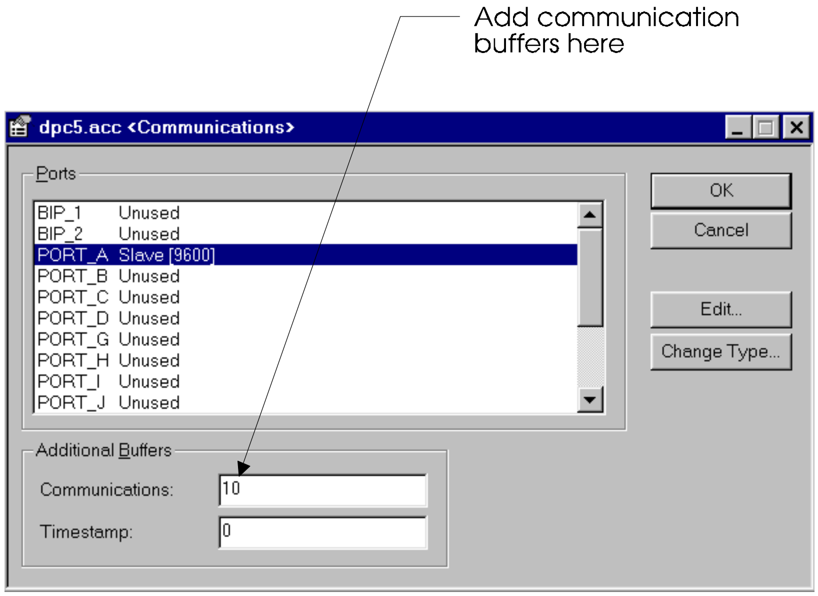
Communication I/O Buffers

Buffers are pre-allocated blocks of memory which are used to hold data messages. The system automatically allocates a certain number of buffers in your Network 3000 controller; additional buffers should be added by the user, depending upon the requirements of your application.

As a general rule, it is better to have too many buffers than too few. A shortage of buffers will cause messages to be delayed; a severe shortage can cause communications not to function at all. Having too many buffers expends extra memory, but otherwise, does not impact system performance.

At least 1 additional buffer should be added for each Master Module in the load. At least 10 additional buffers are the *minimum* recommended configuration.

The number of buffers is changed during ACCOL load creation / modification in ACCOL Workbench.



If you are using Open BSI for communications, buffers must *also* be allocated at the PC. This is done in the Open BSI Setup Tool (Open BSI 2.3 or *earlier* versions) or in the System Wizard of NetView (Open BSI 3.0 or *newer* versions.)

Report By Exception (RBE)

Report by exception (RBE) allows data to be collected only when it changes. For analog RBE signals, *users must carefully choose deadband values*, in order to determine what range of changes can be ignored, otherwise, even the most minimal fluctuations will cause the signal to be collected, even if it has not changed significantly, thereby defeating the purpose of RBE.

In general, alarm signals should NOT be declared as RBE signals, because alarm data transmission always takes priority over other methods of collection. Declaring an alarm signal to be an RBE signal allows the possibility for older RBE reports to arrive after a newer alarm report, thereby giving the user old data.

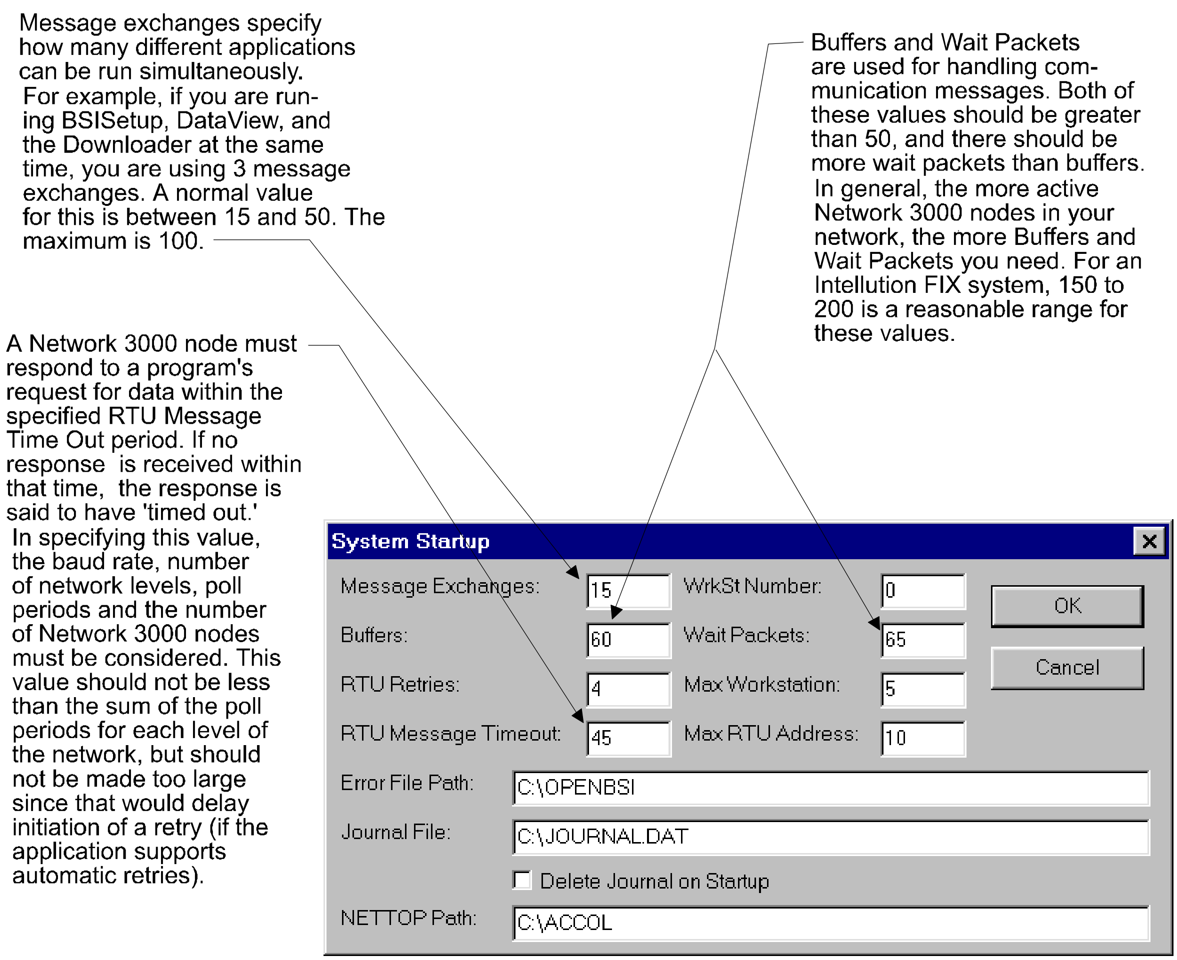
In order for RBE data to be viewed by the user, the central software (i.e. Network Master HMI/SCADA package) must include RBE Manager software. Not all HMI/SCADA packages support this. For more information on RBE, see the '*RBE*' section of the *ACCOL II Reference Manual* (document# D4044).

Open BSI Utilities (Version 2.3 or *earlier*)

The most critical application level parameters for Open BSI 2.*x* are defined in the System Startup dialog box of the Open BSI Setup Tool.

"Message Exchanges" determines how many PC applications can be using Open BSI simultaneously. **"Buffers"** and **"Wait Packets"** are structures in PC memory which are used to hold communication messages. As in the case of buffers in the controller, it is better to have too many of these structures than too few.

"RTU Message Timeout" represents the length of time (in seconds) that Open BSI will wait for a Network 3000 controller to respond to an application's specific request for data. To determine a minimum value for this number, find the longest Master Port poll period for each level of the network (including the Network Master level.) The **"RTU Message Timeout"** should be greater than or equal to the sum of these poll periods. Note, however, that setting this value too long will cause delays in those applications which initiate automatic retries.



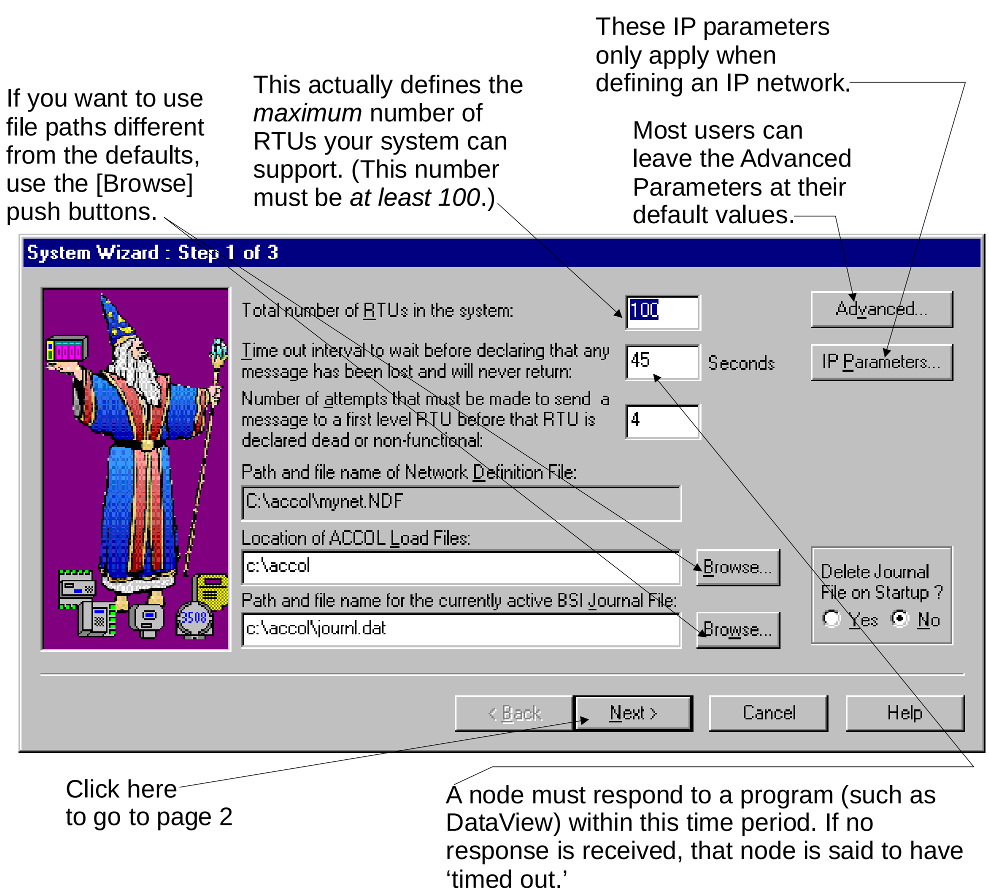
Some other application-level parameters, of lesser importance, which affect communications include:

The **"Comm Retries"** and **"Modem Retry Interval"** parameters in the 'Collection' page of the Open BSI Setup Parameters dialog box relate to the Data Collector's Array and Audit Trail collections. These are discussed in detail in the *Open BSI Utilities Manual* (document# D5076).

The **"Signal Data Rate"**, **"Array Data Rate"** and **"Comm Statistics Rate"** parameters in the 'Refresh Rates' page of the Open BSI Setup Parameters dialog box affect the refresh of data on the screen in DataView, and in the Remote Communication Statistics Tool. See the *Open BSI Utilities Manual* (document# D5076) for details.

The Write Parameters dialog box in SigWrite includes parameters which affect the rate at which SigWrite performs its signal writing operations. See the *Open BSI Utilities Manual* (document# D5076) for details.

Open BSI Utilities (Version 3.0 or *newer*)



For Open BSI 3.0 (or *newer*) to function correctly, each of four basic components must be properly configured:

- A Network Host PC (NHP) must be defined using the System Wizard
- A Network (either IP or BSAP) must be defined using the Network Wizard
- One or more RTU's (IP or BSAP as appropriate) must be added to the network using the RTU Wizard.
- A communication line (IP or BSAP as appropriate) must be defined in order to communicate with the RTUs in the network.

The most critical application level parameters for Open BSI 3.*x* (or newer) are defined in the System Wizard (shown on the previous page). For detailed instructions on configuring Open BSI see the *Open BSI Utilities Manual* (document# D5081).

Open BSI Scheduler and Open BSI Data Collector Array/Archive Collection (All versions)

When using Open BSI Scheduler, the Open BSI Data Collector, or any other program which collects data arrays or archives, do NOT schedule collection to occur at the same instant when array/archive manipulation is occurring, or the data collected may be erroneous.

For example, if new data is added into the array every hour on the hour, do NOT schedule collections from the Scheduler or Data Collector to occur at the same time; schedule them to occur a few minutes after the hour.

There are numerous other communication parameters and details related to these programs which are beyond the scope of this document. See the *'Open BSI Scheduler Manual'* (document# D5077 or document# D5082) and the *'Open BSI Collection/Export Utilities Manual'* (document# D5079 or document# D5083) for details.

OpenEnterprise

If you are using OpenEnterprise software, most communications configuration is performed within Open BSI itself, and not in OpenEnterprise. Certain items, however, such as RBE scan rates and data collection rates *are* defined within OpenEnterprise.

Enterprise Server

If you are using Enterprise Server, running on a VAX computer, many critical application and data link parameters reside in the COMMPORT.TXT file.

Be aware that all top-level nodes in this type of system must have expanded memory configured to hold template data. In addition, the top-level nodes must have a Serial CFE Port configured (instead of a Slave Port) in order to communicate with Enterprise Server.

There are numerous other configuration activities which must be performed, see the Enterprise documentation set for details.

Intellution® FIX®

If you will be using Intellution® FIX® software, in conjunction with Open BSI software, we recommend that you install all Open BSI programs in the \WDMACS directory. Otherwise, you will need to modify the PATH statement in AUTOEXEC.BAT to include the Open BSI installation directory, or else Intellution® FIX® will be unable to locate required dynamic link libraries (DLLs).

See the Open BSI documentation as well as the Intellution® FIX® documentation set for more information on installation.

Setting Parameters in the Intellution® FIX® BR3DIDW.INI File:

The following parameters that affect how Open BSI works with Intellution® FIX® are set in the BR3DIDW.INI file:

OutstandingMessagesPerDevice=*n*

n defines the maximum number of messages that can be outstanding at any one time to a single RTU. We recommend you set **OutstandingMessagesPerDevice** to a low number (start off between 5 and 10). Keeping **OutstandingMessagesPerDevice** to a low number allows the BR3 driver to communicate with more RTUs at startup instead of flooding a few RTUs with many messages and causing all others to time out.

BSIMexBuff=*m*

m this parameter defines the maximum number of buffers that are allocated from the pool of Open BSI buffers available for use by the

BR3 driver. This value cannot exceed 127. Example: If the total buffers allocated in the Open BSI driver are 100, and **BSIMexBuff** is set to 64, this means that the BR3 driver can peak at a maximum of 64 of these buffers. If Intellution® FIX® needs to request *more* data than can fit into 64 buffers during peak usage, refresh rates will slow and timeouts may occur until the system calms down.

BSIMexBuff can NEVER be greater than the total number of buffers in the Open BSI driver.

We recommend you set the total number of buffers available *in Open BSI* to 200 (TOTAL_BUFFERS=200 in the .NDF or .BSI file.) Once this has been done, set **BSIMexBuff** to 127. This will give you the maximum buffers available to the BR3 driver while keeping ample buffers free for Open BSI process communications (DataView, Remote Communication Statistics Tool, Downloader, etc.)

MaxOutstanding=*p*

p this defines the maximum number of messages out of the total count allocated for the BR3 driver (**BSIMexBuff**) that can be outstanding at any one time to all RTUs. We recommend that **MaxOutstanding** should be set to about 60 percent of the **BSIMexBuff** value. This will allow some of the buffers allocated for the BR3 driver from the Open BSI driver to be used for alarm acknowledgment, etc. in Intellution® FIX®.

Iconics Genesis

When using Genesis, make sure that the DEVICE block BRISTOL PARAMETERS show the correct address for the node to which the Genesis workstation is connected. Also, verify that the same NODE ID is used in NETTOP.

NOT all versions of Genesis support peer-to-peer communication between top-level (level 1) Network 3000 controllers.

Chapter 7 - Trouble-shooting Communications

Before you begin to trouble-shoot the communication system, it is important to follow a few simple rules:

- **Don't overlook the obvious.** With all the activity involved in setting up a control system, it is easy to accidentally unplug communication cables, incorrectly configure ports, or improperly set node addresses. Check those things *first*. (For a list of common problems, see the *'Checklist for Diagnosing Communication Problems'* at the end of this chapter.)

- **Adopt a systematic approach.** Don't try to solve the problem by changing several different things at once. Change one thing, see if it causes an improvement, then make notes about what you did, *then* you can try to make other changes. If you haphazardly begin swapping boards, re-routing cables, and changing software parameters, you may end up in worse shape than when you started, or you may end up masking symptoms of an underlying problem.

- **Try to isolate the problem.** If, for example, you cannot communicate globally with a particular controller, see if you can plug the same cable and pseudo master device in locally and talk through a Pseudo Slave Port. If so, this might allow you to narrow down the problem to the configuration parameters for the other port, or a failure in the port hardware. Those would be the next items to check. Similarly, if you have communication problems on a line which includes a radio or dial-up link, verify that you can communicate locally with a direct connection (without the radio or modem). If you can't, the problem may be with the port itself and not the radio or modem; if you can communicate, then check configuration of the modem, data link parameters, etc.

- **Use the hardware and software diagnostic tools provided with the product.** Bristol Babcock hardware often includes failure LEDs which can help identify problems. Many ACCOL software modules report error and status data which may be viewed on-line. In addition, Open BSI Utilities includes software tools for examining port usage, buffers, etc. All of these items may help you locate the source of a particular problem.

- **Collect and save as much relevant information as you can.** If possible, make notes concerning what steps you took leading up to the initial occurrence of the problem. Save printouts, screen dumps, error messages, crash block contents, etc. This information is often useful to Bristol Babcock support personnel, if you have to call for technical assistance.

The remainder of this section includes:

- Trouble-Shooting Techniques
- Frequently Asked Questions Concerning Communication Problems
- Checklist For Diagnosing Communication Problems

Trouble-Shooting Techniques

As discussed previously, useful information about a problem can often be obtained using the tools and error reporting facilities built-into the product.

Check Failure LEDs on Hardware

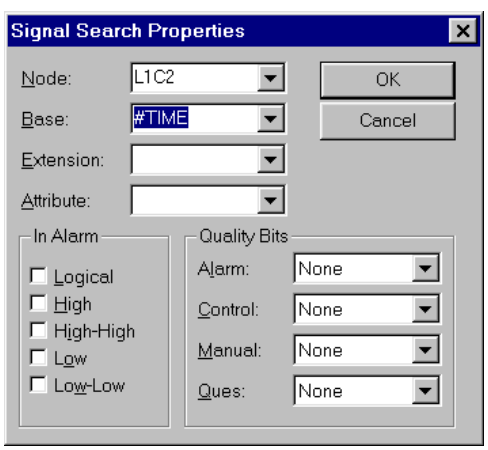
If your controller includes visible failure indicators (LEDs), look at them to see if a hardware problem has occurred. Instructions for interpreting what the failure LEDs mean is included in the hardware (CI-3*xxx*) manual. Some hardware failures can also be detected via off-line diagnostics. See the *33XX Diagnostics Manual* (document# D4041) for details.

If You Suspect A Problem in an ACCOL Task, Use A Debugger

To examine ACCOL logic, on-line, use ACCOL Workbench in on-line debugging mode, or for older units, use Taskspy. For information on Workbench, see the *ACCOL Workbench User Manual* (document# D4051). For information on Taskspy, see the *Taskspy Manual* (document# D4083).

For a Quick Check on Communications, Call Up #TIME Signals

A good way to test whether communications are operating properly with a particular node, is to view its #TIME.*nnn*. system signals, via DataView[1]. Look at the #TIME.001. and #TIME.007. signals (which are in units of seconds). The rate at which these signals change will tell you how fast data is being updated at the PC.



[1] *For help on using DataView, see the Open BSI Utilities Manual (document# D5076 or D5081).*

Set Up the Error and Process I/O Diagnostic Arrays in ACCOL

Certain errors in ACCOL module operations are detected and reported in a read/write array designated by the #ERARRAY.. system signal.

Certain errors in process I/O board configuration can be detected and reported using the #DIAG.002. system signal.

For help on setting up these arrays, see the '*System Signals*' section of the *ACCOL II Reference Manual* (document# D4044).

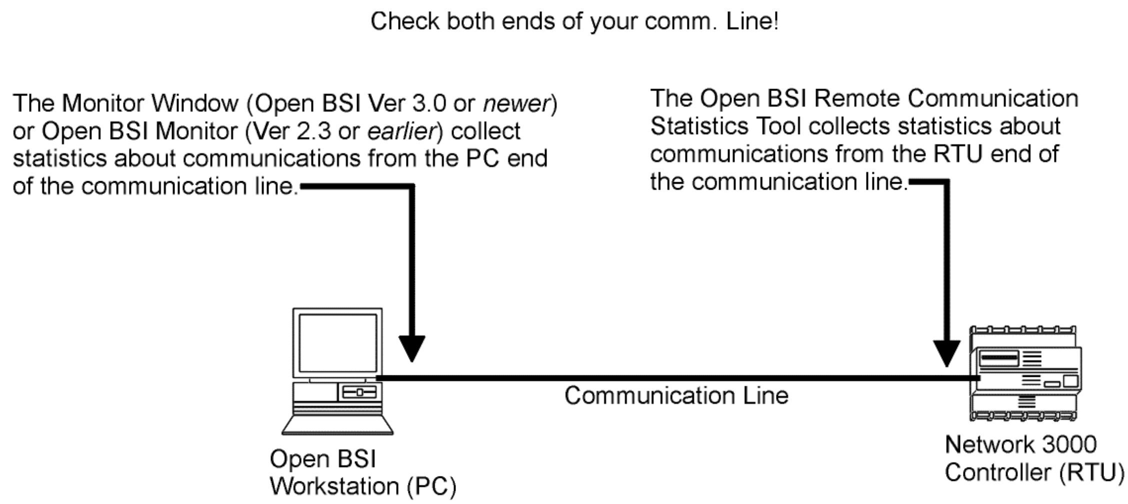
Set up Modules and Module Terminals Which Report Errors

For example, if you are using peer-to-peer communication, make sure you have an analog signal wired to the STATUS_2 terminal of the Master Module, so you can view any active error codes. A list of codes is included in the *ACCOL II Reference Manual* (document# D4044).

Similarly, if you are using remote I/O racks, be sure to set up the RIOSTATS module; port statistics can be viewed using the Portstatus Module. See the '*Error Reporting*' section of the *ACCOL II Reference Manual* (document# D4044) for more information.

If you are using IP communication, include ACCOL logic to check the value of the IPSTAT.. system signal. Negative values indicate errors. See the '*System Signals*' section of the *ACCOL II Reference Manual* (document# D4044) for details.

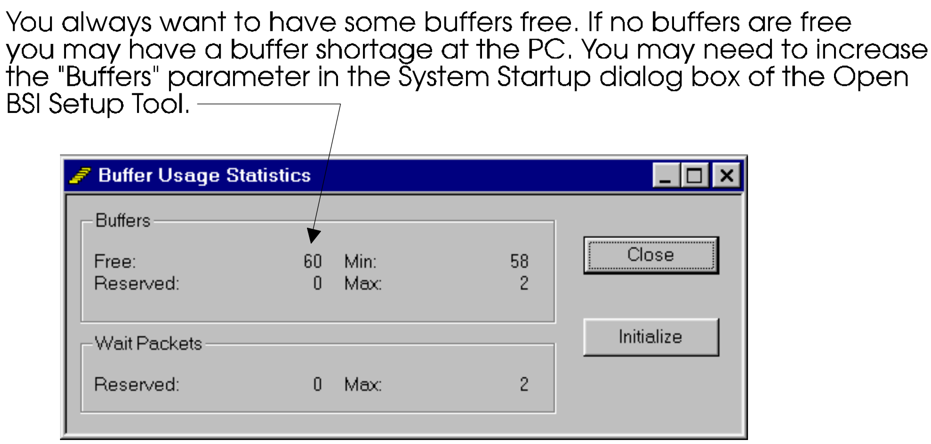
Check Both Ends of Your Communication Line Using the Open BSI Monitor (or Monitor Window in NetView) and the Open BSI Remote Communication Statistics Tool



The next troubleshooting techniques highlight certain problems which can be uncovered by using the Monitor Window of NetView, the Open BSI Monitor, and the Remote Communication Statistics Tool. Detailed information on starting and using these tools is included in the *Open BSI Utilities Manual* (document# D5076 or D5081).

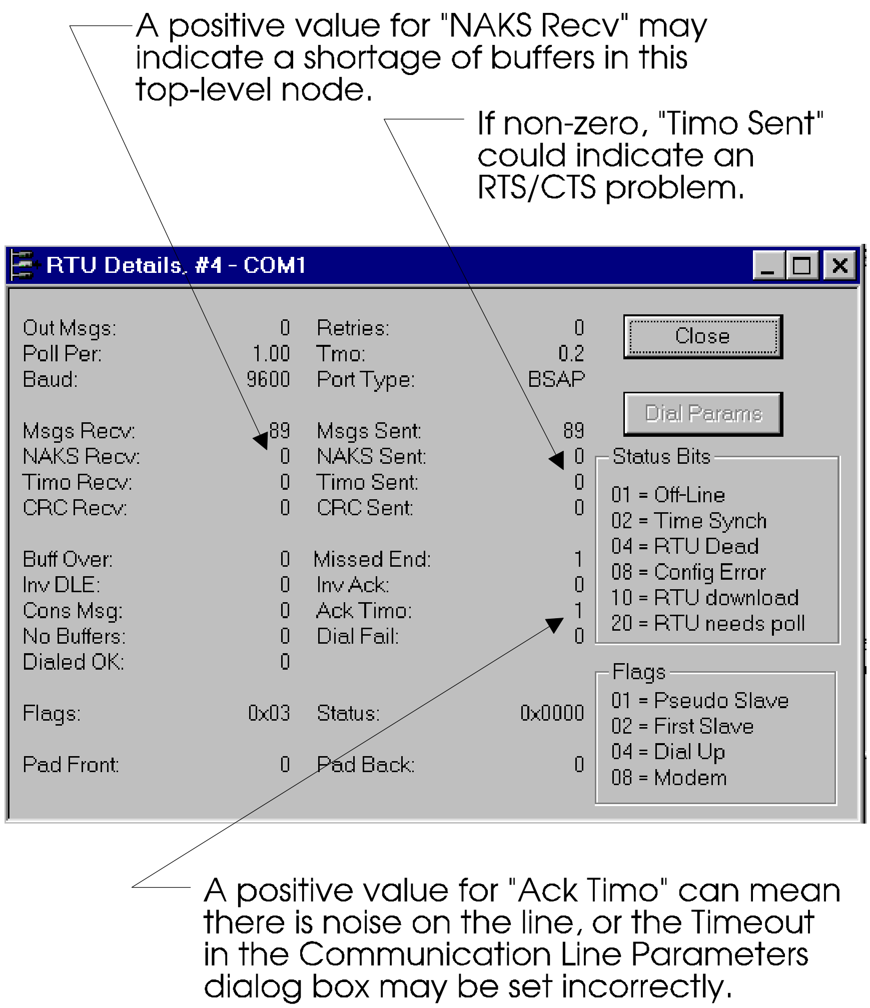
Use the Open BSI Monitor (Open BSI 2.3 or *earlier*) to View Statistics About Communication ’Health’ at the Workstation side of the Communication Line

If you are using Open BSI Utilities Version 2.3 (or *earlier*), the Open BSI Monitor can provide you with information on the operation of Open BSI communications, and the top-level nodes in the network. Check the Buffer Usage Statistics dialog box to see if you have enough buffers at the PC.



Another thing to check in the Open BSI Monitor is the statistics that Open BSI has collected about its communication with the top-level Network 3000 controllers. These are displayed in the RTU Details Window.

NOTE: The information displayed in the RTU Details window is collected at the workstation end of the communication line, it does NOT reflect statistics collected directly from the RTU. RTU statistics are available from the Remote Communication Statistics Tool, discussed later.

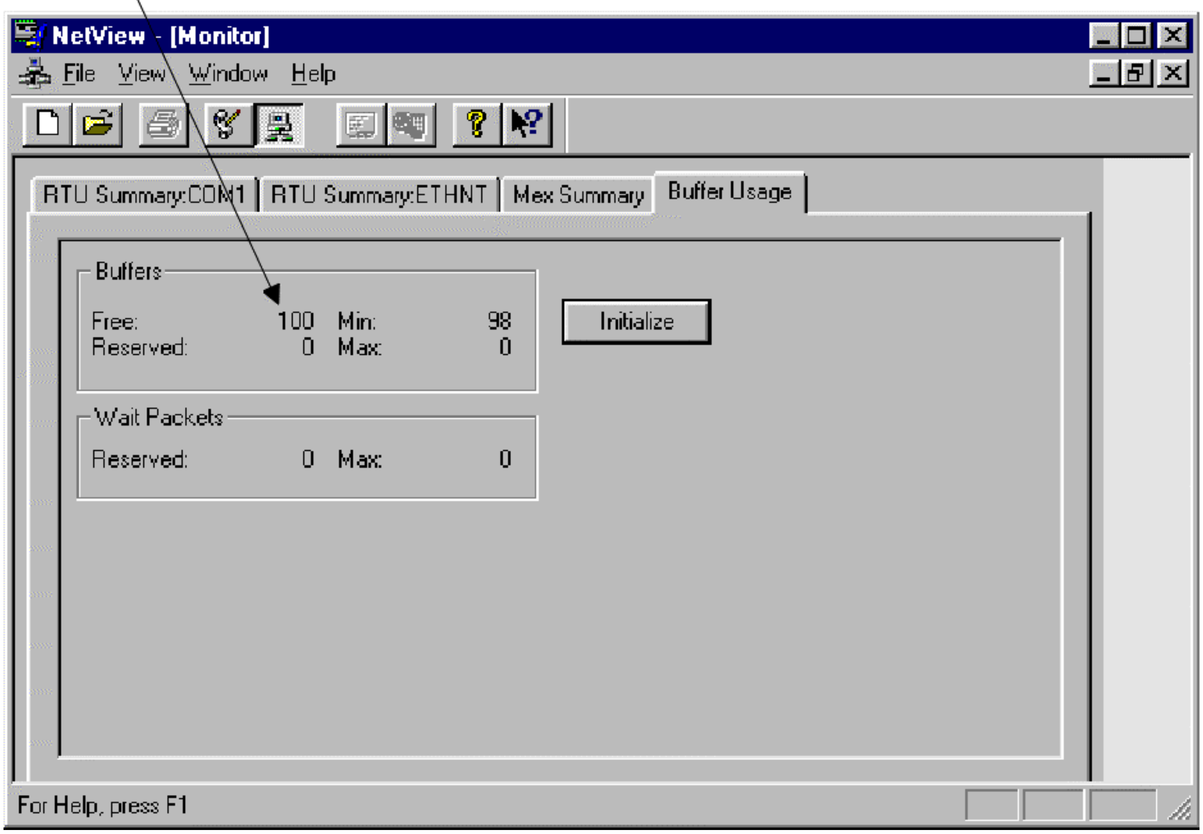


Use the Monitor Window (Open BSI 3.0 or *newer*) to View Statistics About Communication 'Health' at the Workstation side of the Communication Line

If you are using Open BSI Utilities Version 3.0 (or *newer*), the Monitor Window in NetView can provide you with information on the operation of Open BSI communications.

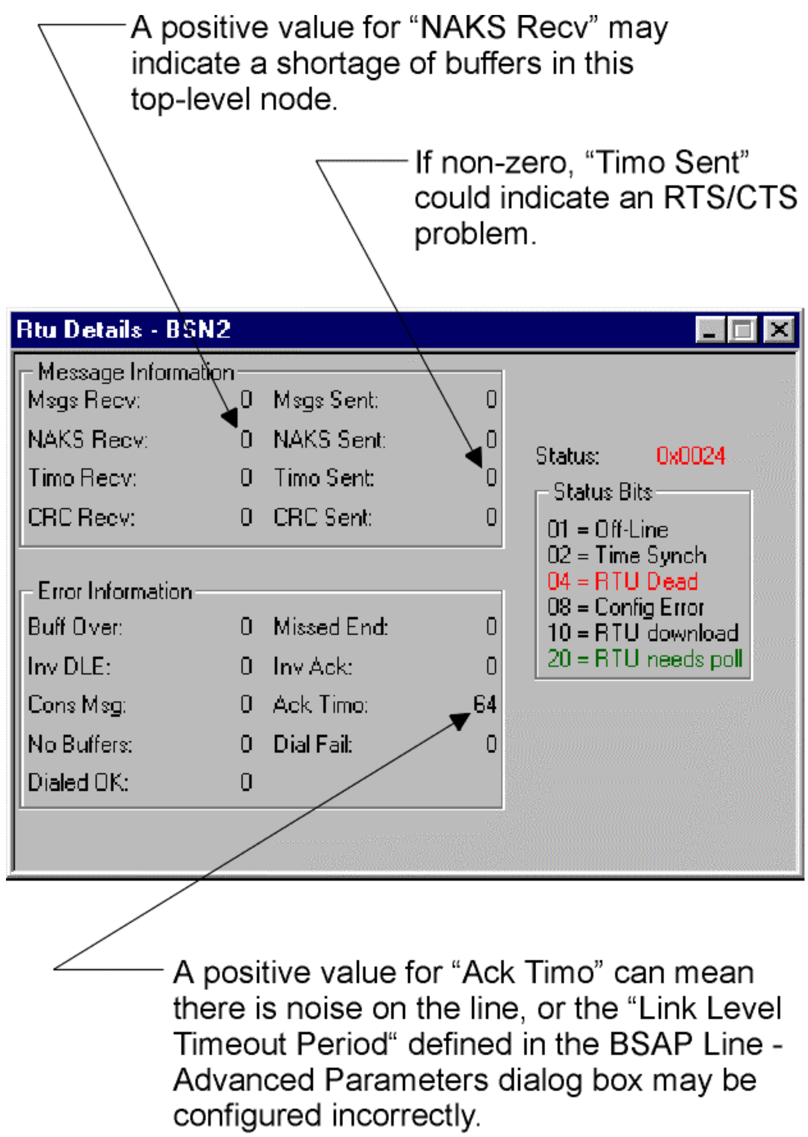
Check the Buffer Usage Statistics dialog box to see if you have enough buffers at the PC.

You always want to have some buffers free. If no buffers are free you may have a buffer shortage at the PC. You may need to increase the TOTAL_BUFFERS value in the [CONSTANTS] section of the NETDEF file, and re-start NetView.



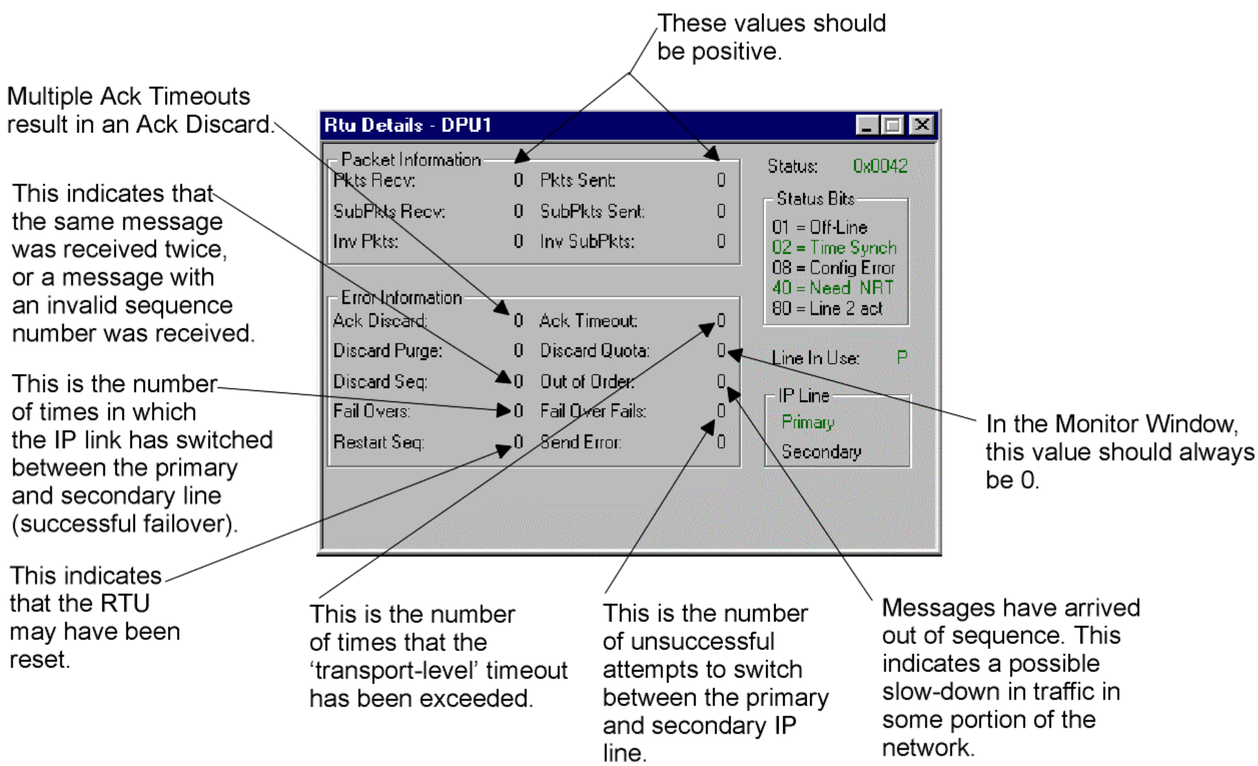
If you have a BSAP network, the Monitor Window allows you to examine statistics about the top-level Network 3000 controllers. These statistics are displayed in the RTU Details Window:

NOTE: The information displayed in the RTU Details window is collected at the workstation end of the communication line, it does NOT reflect statistics collected directly from the RTU. RTU statistics are available from the Remote Communication Statistics Tool, discussed later.



If you have an IP network, the Monitor Window allows you to examine statistics collected at the PC about the IP controllers. These statistics are displayed in the RTU Details Window.

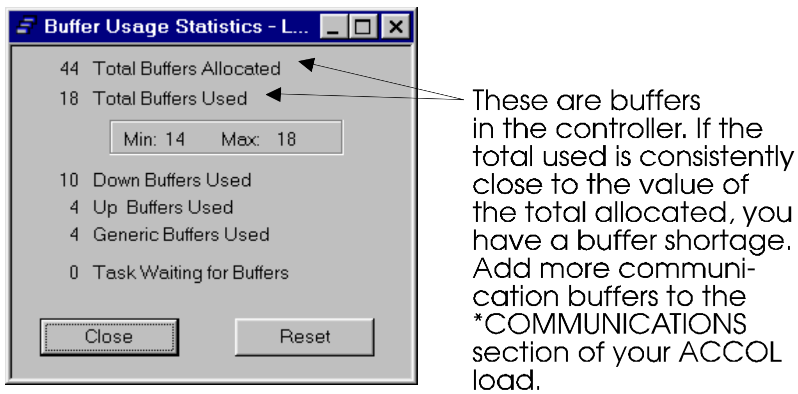
NOTE: The information displayed in the RTU Details window is collected at the workstation end of the communication line, it does NOT reflect statistics collected directly from the RTU. RTU statistics are available from the Remote Communication Statistics Tool, discussed later.



Use the Open BSI Remote Communication Statistics Tool to View Statistics About Communication 'Health' At the RTU Side of the Communication Line[2]

If you are using Open BSI Utilities, the Remote Communications Statistics Tool can provide you with current statistics on how communication ports and buffers in the controller are functioning.

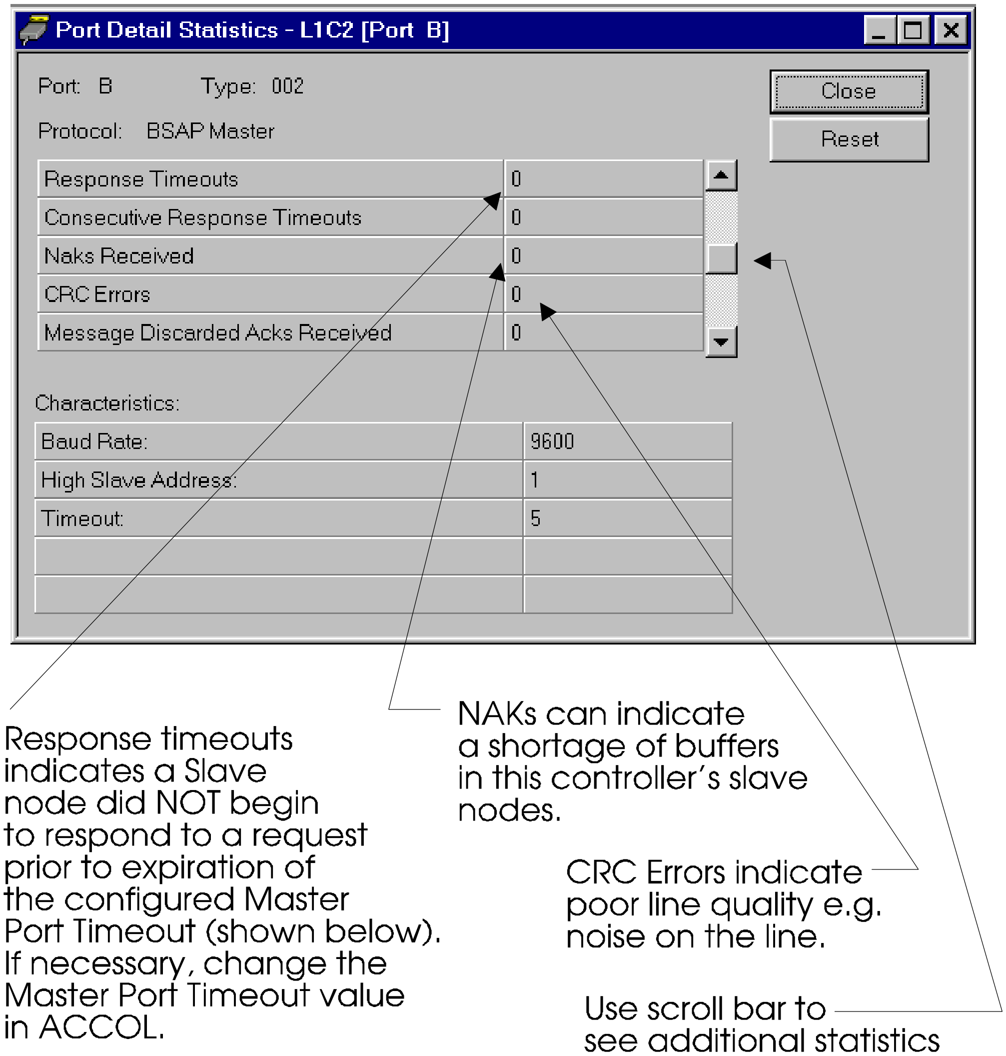
The Buffer Usage Statistics dialog box in the Remote Communication Statistics Tool relates to buffers in the controller (*not the PC*).



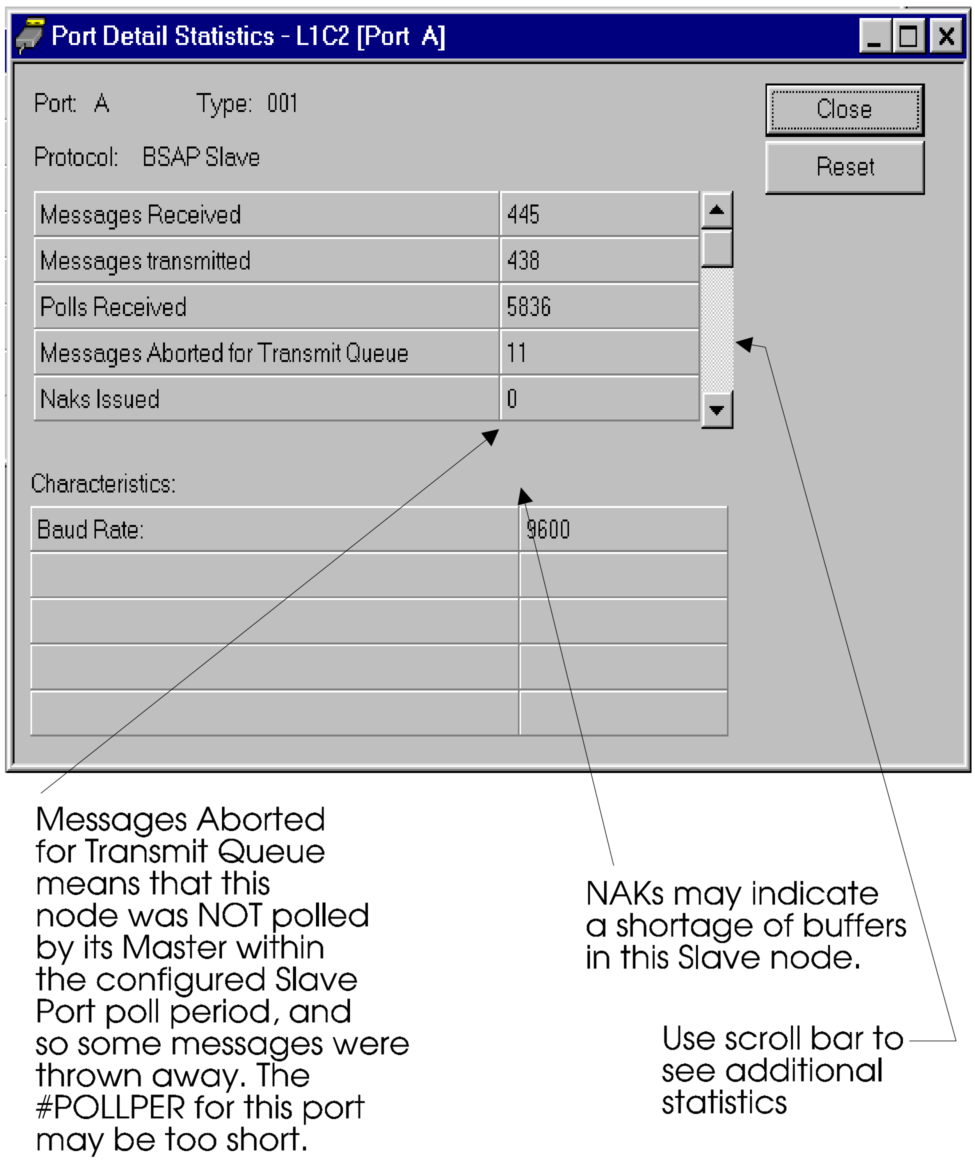
---

[2] *The EGM 3530 TeleFlow / RTU 3530 TeleRTU units do NOT store communication statistics, therefore, the Remote Communication Statistics Tools cannot collect and report communication statistics from these units.*

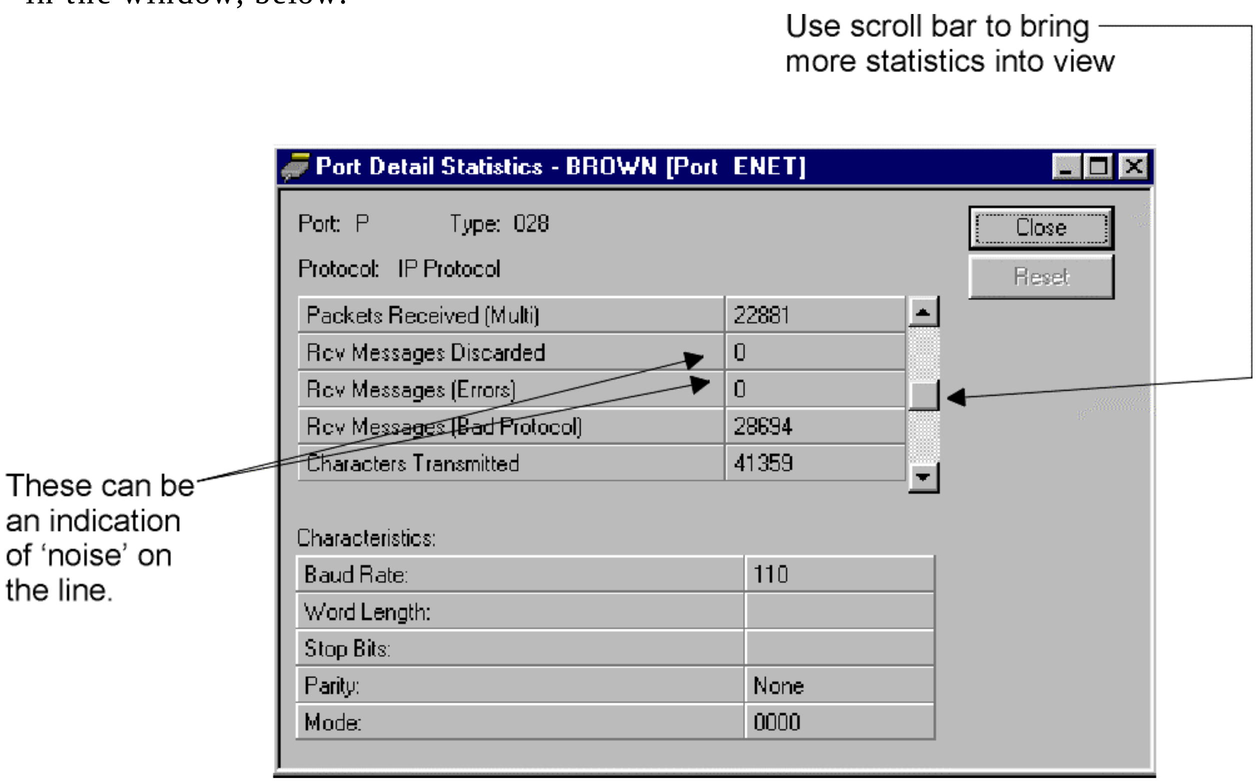
The Port Detail Statistics Window displays statistics collected on the operation of the communication ports in the controller. Master Port details are shown at right:



Information on Slave Port details is shown below:

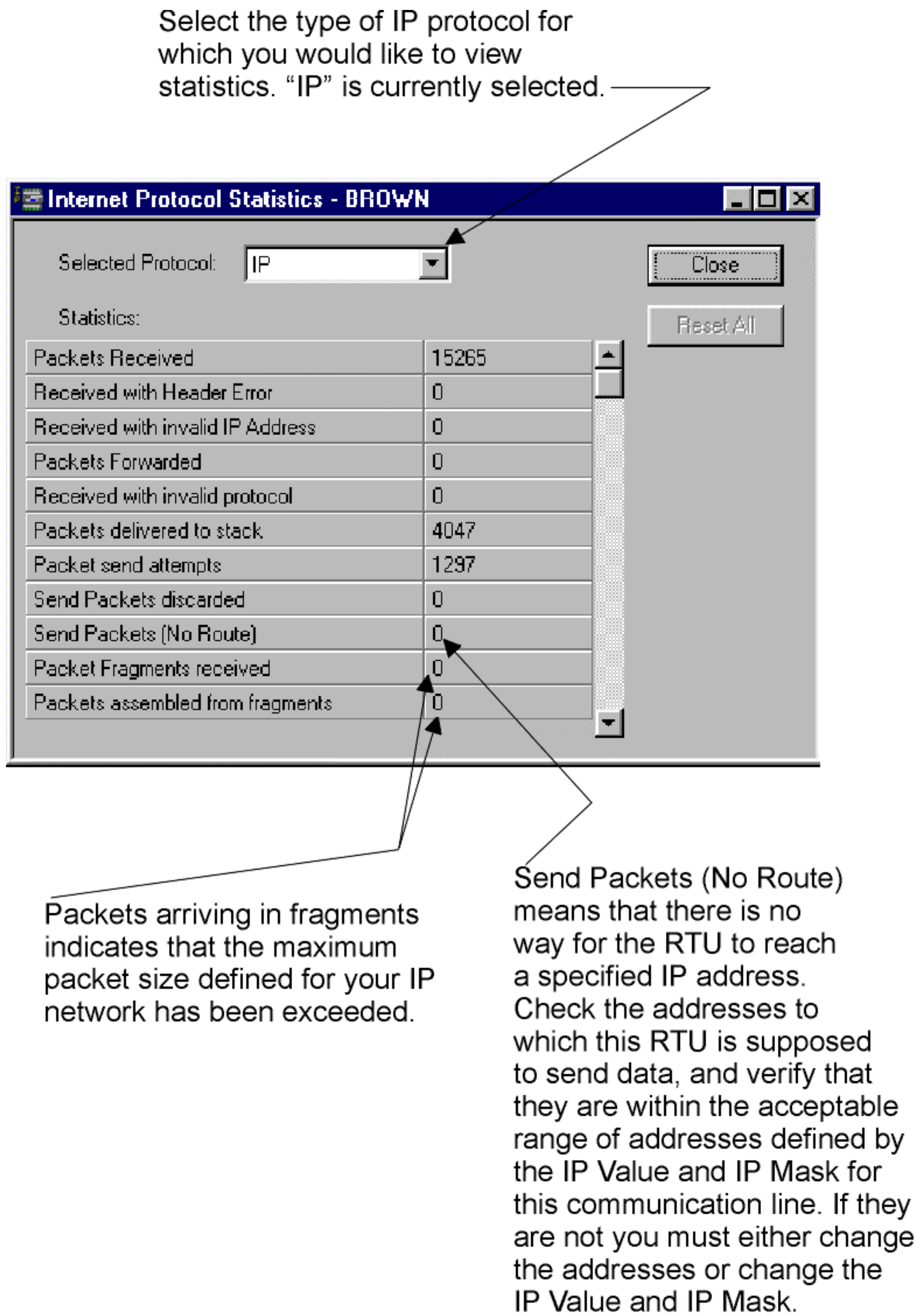


If you are using an IP Port, statistics are collected for the port itself, as well as the functioning of the protocol used for the port. Information on the port itself is shown in the window, below:



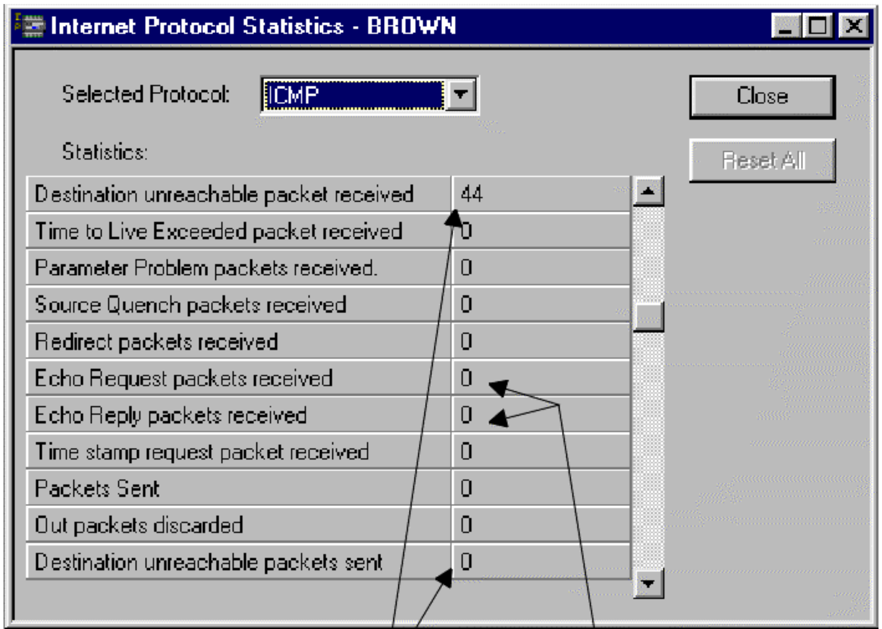
Protocol Statistics are also collected.[3] These are displayed in the Internet Protocol Statistics Window. The type of statistics displayed varies depending upon which IP protocol has been selected (IP, ICMP, UDP, or IBP).

IP Protocol Statistics:



[3] *Do NOT confuse 'Protocol' with the custom protocols which can be configured in ACCOL for communication with foreign devices. 'Protocol', in this case, refers to layered protocols which make up this Internet Protocol (IP) implementation - these layers are: IP, ICMP, UDP, and IBP.*

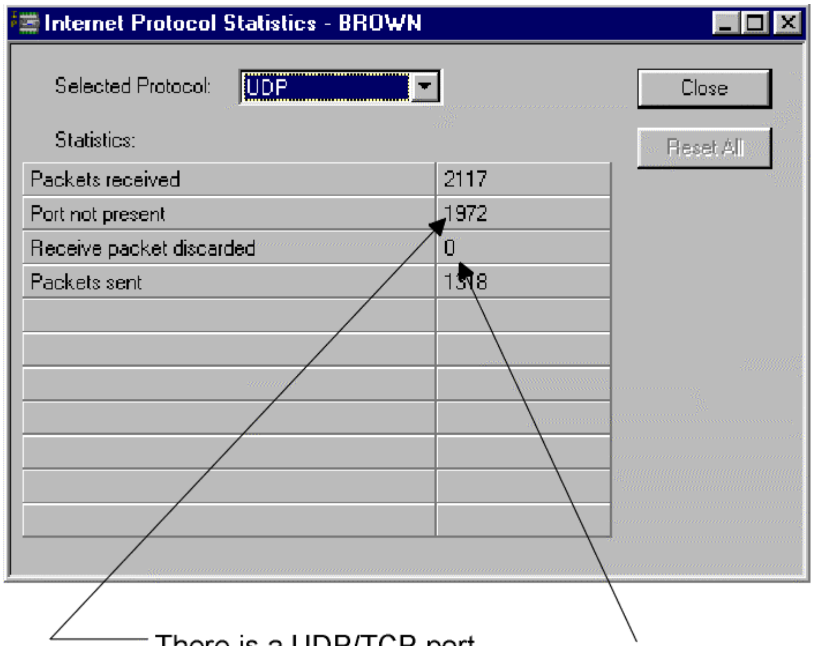
ICMP Protocol Statistics:



These are indications of network routing problems. Check the IP addresses and IP masks throughout your network, and verify that they are valid. Also, make sure the UDP Port number for the IP driver, the UDP Port number for Time Synchs, and the TCP Port number for the Router process are consistent throughout your network. (These were initially defined in NetView's System Wizard, and can be examined in the NETDEF file. If changes are necessary you may need to reconfigure them in the RTU(s) using LocalView).

These indicate usage of the 'PING' program

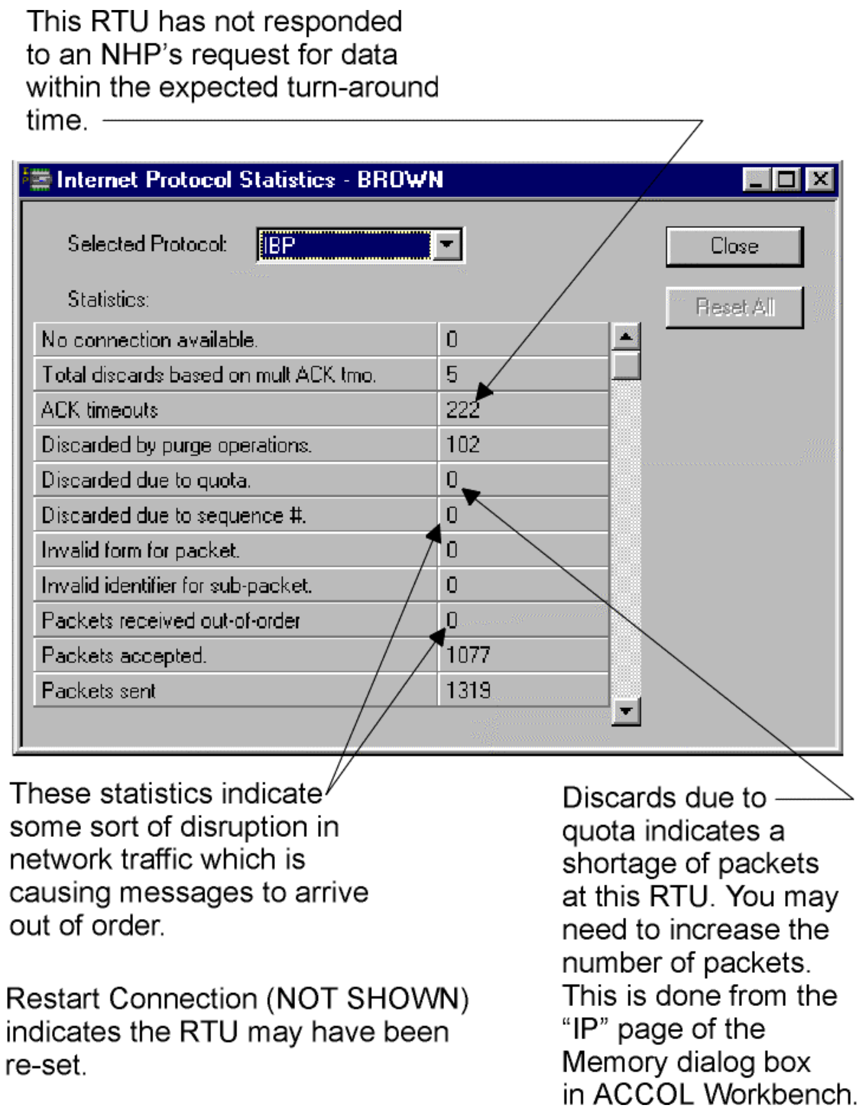
UDP Protocol Statistics:



There is a UDP/TCP port number discrepancy. Make sure the UDP Port number for the IP driver, the UDP Port number for Time Synchs, and the TCP Port number for the Router process are consistent throughout your network. (These were initially defined in NetView's System Wizard, and can be examined in the NETDEF file. If changes are necessary you may need to reconfigure them in the RTU(s) using LocalView).

A positive number here indicates line noise.

IBP Protocol Statistics:



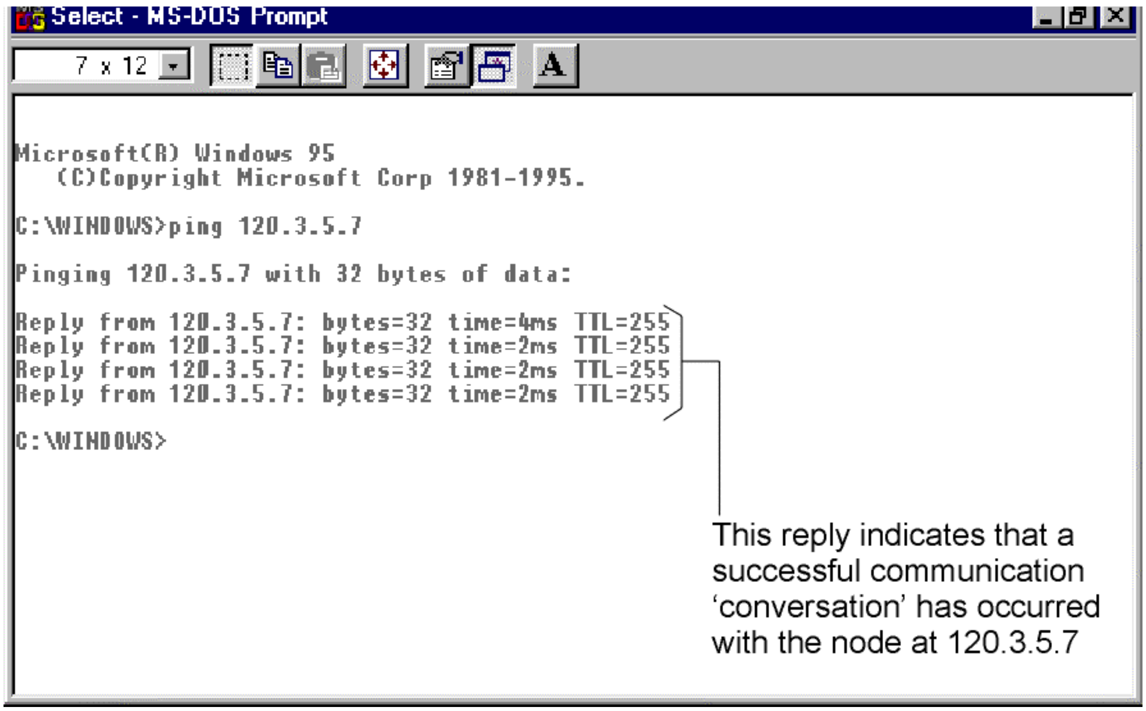
If You Are Using Older-model ACCOL Tools, Use Toolkit / Portstatus

If you do NOT have Open BSI, statistics on port and buffer usage in the controller may still be obtained via the Communication Buffers and Communications Line Statistics displays in Toolkit. NOTE: Toolkit does NOT function with 386EX Protected Mode controllers or 3530 TeleFlow/TeleRTU units. See the *Toolkit Manual* (document# D4053) for details.

In addition, Port statistics may also be viewed on-line using the Portstatus Module. See the '*Portstatus*' section of the *ACCOL II Reference Manual* (document# D4044) for details.

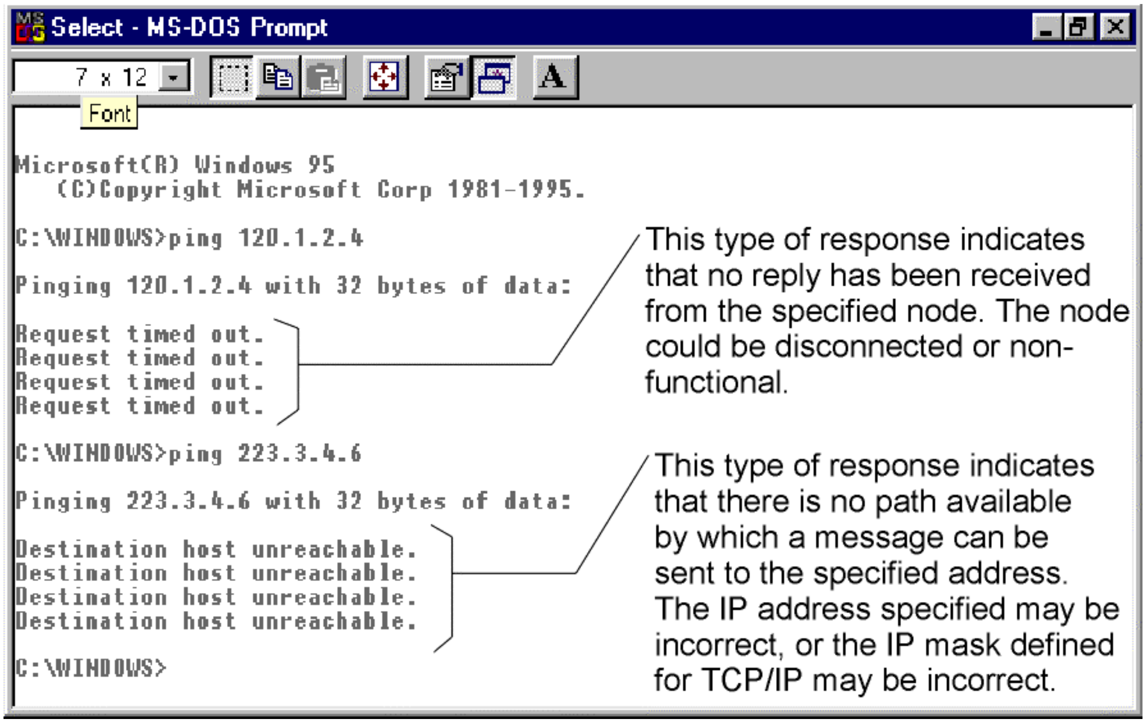
Use PING Command to verify that communication with a particular IP node is possible

In an IP network, you can verify that communication with a particular IP node is possible using the PING command at the DOS prompt. (PING is a program which sends a small amount of data to an IP node, and requests that the IP node echo that data back to its originating location).



If successful communication using PING is possible, yet you still cannot communicate using Open BSI, you can assume that IP addresses at the RTU are correct, and that the IP connection is plugged in. There is likely to be an Open BSI configuration problem. Have you configured all four necessary components (NHP, IP network, IP communication line, IP RTUs) in an Open BSI IP system? In particular, check in NetView to see that IP Mask and IP addresses are correctly defined. Check to see that the IP mask you configured is identical to the IP mask defined during TCP/IP installation.

If you cannot communicate using PING, there can be several reasons, including incorrect TCP/IP configuration (wrong IP mask, for example), the node is disconnected, the cable is bad, the IP address and IP mask at the RTU (configured via RTU Configure mode in LocalView) is incorrect or out-of-range. Once you have fixed any of these problems, communication using PING should be possible. If it is, but Open BSI still cannot communicate, start checking for Open BSI configuration problems.



Use the 'Virtual' Line Monitor At the PC (Advanced Users)

Open BSI 3.0 (or newer) can be configured (via settings in the BSBSAP.INI or BSIPDRV.INI files) to log the characters received through the communication line at the PC in an ASCII file. Advanced users who are knowledgeable about communication messages can then examine the file to see that the expected protocol messages are present.

In addition, certain Open BSI driver parameters may be modified.

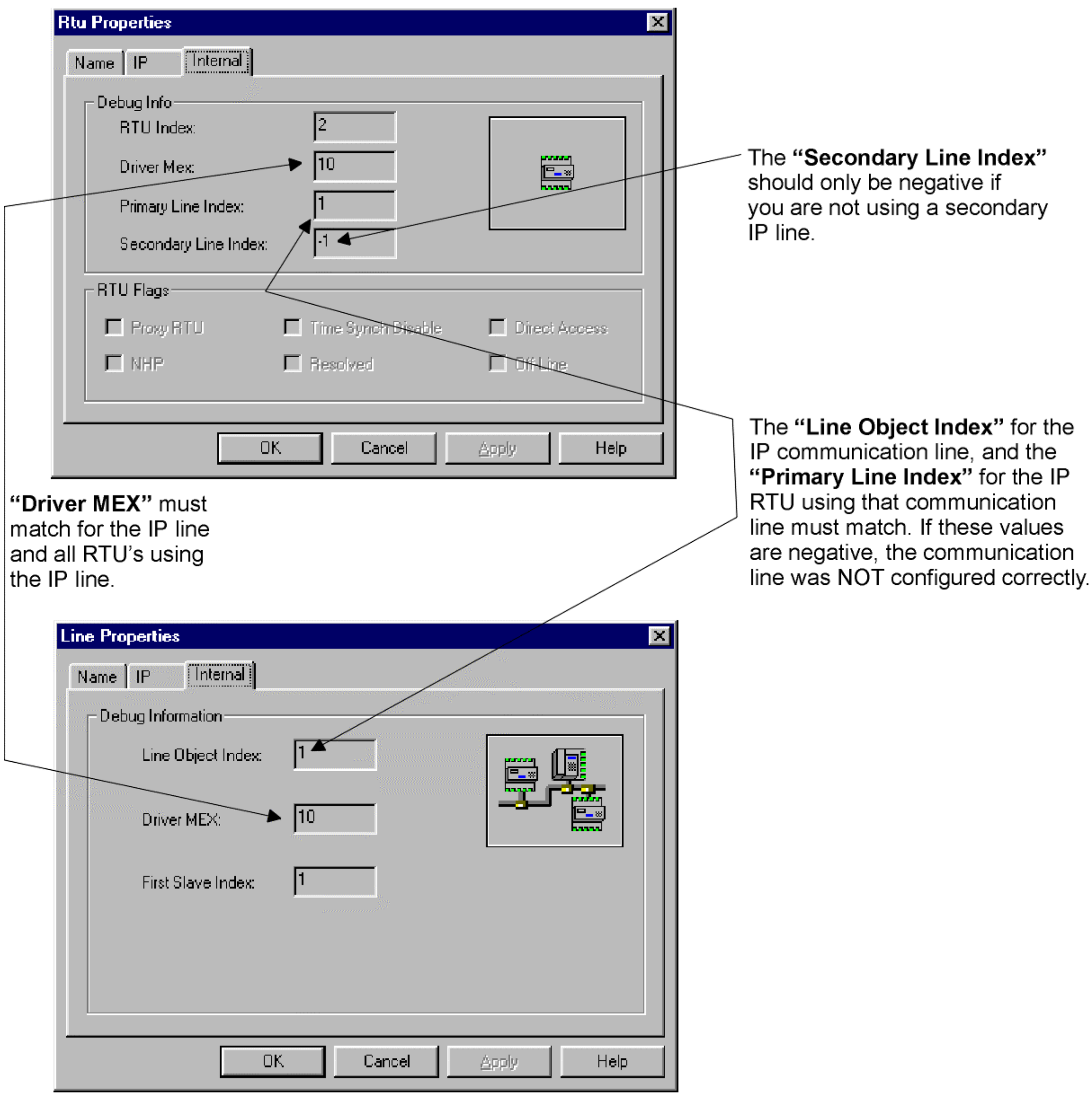
For detailed descriptions of the BSBSAP.INI and BSIPDRV.INI files, please consult *Appendix E* of the *Open BSI Utilities Manual* (document# D5081).

Examine Open BSI Internal Parameters

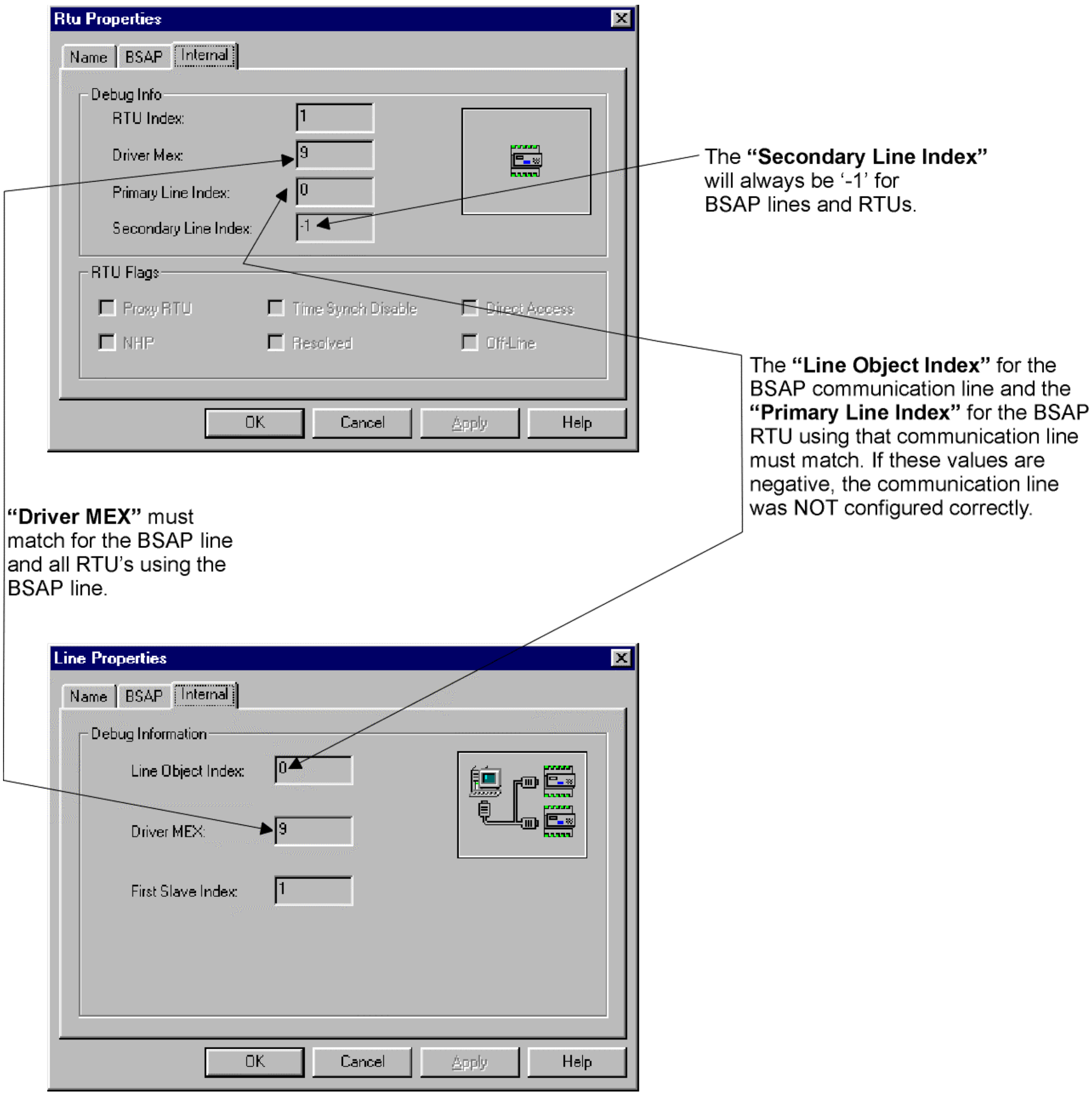
In NetView, you can check certain internal debug parameters from the "**Internal**" page of the Line Properties or RTU Properties dialog boxes. In general, a negative value for a debug parameter indicates an Open BSI configuration problem. NOTE: A negative value for a secondary line IP line for an RTU which has only one IP line should NOT be considered a problem.

IMPORTANT: You should also check to make sure that the RTU has not been marked "**Off-Line**". If it has been, de-select that flag.

IP Internal Debug Info:



BSAP Internal Debug Info:



IMPORTANT: You should also check to make sure that the RTU has not been marked "**Off-Line**". If it has been, de-select that flag.

Frequently Asked Questions Concerning Communication Problems

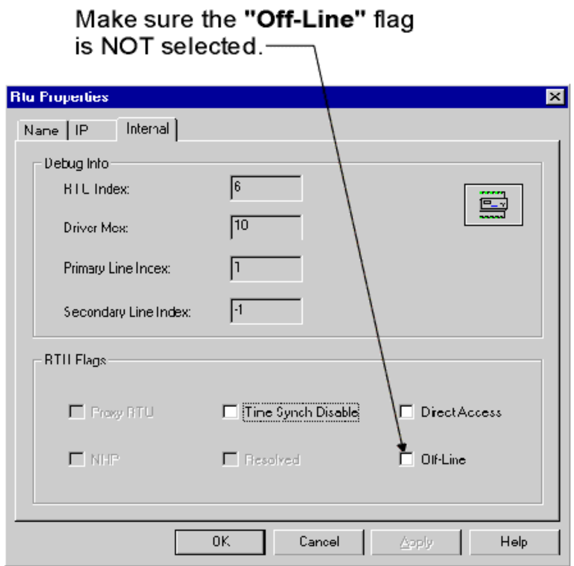
Problem or Question:	**Cause and/or Suggested Remedy:**
GENERAL PROBLEMS:	
"I downloaded my ACCOL load successfully, but then the controller immediately stopped talking. What should I do now?"	It is possible that the ACCOL load you downloaded had a port configuration which was different from the one previously in the controller. Look in the ACCOL source file which corresponds to the load you downloaded, and check to see that the port to which you are connected is defined as a Slave Port, and has the same baud rate as the computer you are using to talk with the unit. If the slave port is on a different port and/or the baud rates don't match, communication will be impossible through the port you are currently plugged into. Re-connect the cable to the correct port. *NOTE: If there is no properly defined Slave Port in the load, you will need to re-edit your ACCOL load to include one, and then reset the controller (by pressing the RESET button). This will allow a cold start to occur, which clears out the old erroneous load, and will allow you to download your revised ACCOL load at 9,600 baud on either Port A, C, G, I, or BIP 1.* This problem can also occur if the switches for Expanded Node Addressing are set incorrectly. See the '*Expanded Node Addressing*' section of the *ACCOL II Reference Manual* (document# D4044) for details, and variations on downloading rules for expanded node addressing.
"I keep getting **#LINE.***nnn*. alarms. What do they mean?"	These are communication line alarms. The *nnn* indicates which port is having the line failure: #LINE.000.= Port A #LINE.001.=Port B #LINE.002.=Port C #LINE.003.=Port D #LINE.004.=BIP 1 #LINE.005.=BIP 2 #LINE.006.=Port G #LINE.007.=Port H #LINE.008.=Port I #LINE.009.=Port J For a Slave, Pseudo Slave, or Pseudo Slave with Alarms Port, the #LINE.*nnn* logical alarm signal is turned ON if there are no poll messages on the

Problem or Question:	Cause and/or Suggested Remedy:
	(*continued…*) port for a period of time greater than the configured Slave Port poll period. The #LINE.*nnn*. alarm is turned OFF (generating a 'return to normal' alarm) when polling is restored. Review '*Chapter 4 - Guidelines for Configuring the BSAP Data Link*' to see if the Slave Port poll period might be set too short. #LINE.*nnn*. alarms could also be caused if the connection was broken with the master node, or with a PC workstation, either by the line being unplugged, or polling turned off (for example, by shutting down ACCOL Tools / UOI / Open BSI software). Check to see that the cable is plugged in, and that polling is active. If a #LINE alarm appears on a Master Port, it means all slave nodes on the line are classified as 'dead'. Check cabling, data link parameters, etc. If a #LINE alarm appears on an Expanded Addressing Master Port, the same reasons could apply, or it could occur if the EAStatus Module is improperly configured. For both Master and Expanded Addressing Master Ports, the #LINE alarm will issue a 'return to normal' message if at least one slave node is communicating on the line. If a #LINE alarm appears on an IP port, it indicates that there has been no communication traffic received within the time indicated by the #POLLPER signal associated with this port. If one half of the interval defined by the #POLLPER signal has passed, with no traffic received, the node will attempt to contact Network Host PCs (NHPs).
"What do **#NODE.***nnn*. alarms signify?"	#NODE.*nnn*. alarms occur in a Master node when communications have been lost with one of its slave nodes. The local address of the slave node is indicated by the *nnn*. This could be for many reasons, but is usually due to data link problems. Check cabling, baud rates, etc.

Problem or Question:	Cause and/or Suggested Remedy:
"What do **#OCTIME.ERROR**s mean?"	This error occurs upon receipt of a time synchronization message from the Master node. It indicates a time discrepancy of greater than 4 seconds between the node, and its Master node.
"No data is being collected and the controller's **'WDOG'** LED is ON."	The 'WDOG' LED indicates a serious failure, called a 'Watchdog' condition. To clear up this problem, the unit must be reset and re-downloaded. If the problem persists, it could indicate a serious hardware or firmware problem. Contact Bristol Application Support for assistance.
"Communications are operating, but data updates are very slow."	There are several possible causes of this, including, but not limited to, the following: 1) Poor communication line quality. Check for CRC errors on the port or ACK Timeouts. 2) Improper system tuning including Master Port poll periods set too long (BSAP networks only), or application-level timeouts set too long. 3)Task rates are set too slow. 4) Improper switch settings. If you are using Expanded Node Addressing verify that the proper group number has been set using switches/soft switches. Also verify that switches are set properly for the communication cable you are using i.e. if switches/jumpers are set for RS-485 but you are incorrectly using an RS-423 cable. Also check the port and buffer statistics in the Open BSI Monitor, Monitor Window, the Open BSI Remote Communication Statistics Tool, Toolkit, or the Portstatus Module to help isolate the error.
"Communications are operating satisfactorily, but certain signals are NOT changing and they should be."	There could be a problem with the ACCOL Task. Also, check to see that the signals affected are NOT control inhibited.

Problem or Question:	Cause and/or Suggested Remedy:
"Communications are operating, but the data I'm collecting doesn't change."	Check the ACCOL task rates (set using the #RATE.*nnn*. system signals where *nnn* is the task number). Make sure that they are non-zero. Any task rate set to 0 WILL BE DISABLED. Check to see that your ACCOL tasks are NOT suspended via task control statements. Also check to see that they are NOT in Step Mode (used for debugging in ACCOL Workbench or Taskspy).
OPEN BSI UTILITIES,	COLLECTOR, SCHEDULER OR WORKBENCH
FOR OPEN BSI 2.3 (or earlier) ONLY: "I get the message **'Another process has initialized the OpenBSI Comunications Driver. This process will now terminate.'** What does this mean?"	This message appears if Open BSI communications are already running, and you try to start them *again* by starting another copy of the Open BSI Setup Tool. Only one copy of the Setup Tool can be running at any one time. If no instance of BSISETUP is minimized, or on the Taskbar, depress [CTRL][ALT][DEL] and end tasks Resmon32 and BSBSAP32.
FOR OPEN BSI 3.0 (or newer) ONLY: "When I try to start LocalView, I get the message "**Some other process has already started the OpenBSI System. LocalView application cannot start.**"	This message means that NetView is already running. If you need to use LocalView, you must first stop NetView, then you can start LocalView.
FOR OPEN BSI 2.3 (or earlier) ONLY: "When I try to call up data from a node, I get a message in RED in the status bar which says **'Node Currently Off-Line'**. If I look at the On	First of all, the On Line Changes dialog box shows on-line if *polling* in Open BSI's node array is turned ON for this node; other than that, it has nothing to do with whether communications are *actually* working. The 'Node Currently Off-Line' message will appear in Open BSI if either 1) Open BSI cannot find the node you are looking for, *or* 2) it cannot

Problem or Question:	**Cause and/or Suggested Remedy:**
(continued) Line Changes dialog box in the Open BSI Setup Tool (accessible via the Active Communication Lines dialog box) it shows the node as on-line, and I know the controller is working. What's wrong?"	*(continued)* communicate with it. 1) If Open BSI cannot find it, it probably means that either the node name you are trying to access does not exist in the currently released NETTOP files, *or* the slave address or range of slave addresses is declared incorrectly in Open BSI. Also, make sure that the current, proper NETTOP files are in the path specified in the System Startup dialog box. 2) The other alternative is that there is a data link communication line problem. In this case, check the baud rate and other parameters in the Open BSI Setup Tool, particularly in the Communication Line Parameters dialog box.
FOR OPEN BSI 3.0 (or newer) ONLY: "When I try to call up data from a node, I get a message in RED in the status bar which says **'Node Currently Off-Line'**, and I know the controller is working. What's wrong?"	The 'Node Currently Off-Line' message will appear in Open BSI if either 1) Open BSI cannot find the node you are looking for, *or* 2) it cannot communicate with it. There can be several possible reasons for this; here are a few of them: 1) Incorrectly defined communication line (either BSAP or IP). Check BSAP address range for the line (or IP value and IP mask for an IP comm line). For BSAP lines, check the baud rate, etc. 2) For IP networks: Inconsistent UDP Port numbers for Time Synch or the IP Driver. Review the NETDEF file and see that these numbers are consistent throughout the IP network, and at NHP. 3) For IP networks: Incorrectly defined IP address, IP mask, or NHP addresses specified at the RTU. Use LocalView in "**RTU Configure**" mode to verify that these items are set correctly. 4) Node has been marked off-line. In the NetView tree, right-click on the icon for the RTU, and choose "**Properties**" then go to the 'Internal' tab. De-select the "**Off-Line**" RTU flag, if it is selected.

Problem or Question:	Cause and/or Suggested Remedy:
	
OPEN BSI 2.3 (OR earlier) ONLY: "I get the message **'Could not read NETTOP.'** What does this mean?"	This means that the NETTOP files cannot be found. Check to see that the files NETFILE.DAT, RTUXREF.DAT, and GLADXREF.DAT exist, and are in the "**NETTOP Path**" specified in the System Startup dialog box of Open BSI.
"I try to start DataView or one of the other utilities, and I get a **'Waiting for communications driver** message… then it says **'Failed to initialize communications system'**.	This message, and similar messages such as 'Could not initialize communications' or a message that says 'Exiting with status -115' are all indications that you have NOT started Open BSI communications. Open BSI communications are activated by starting the Open BSI Setup Tool (Open BSI Version 2.3 or earlier), NetView (Open Version 3.0 or newer), or LocalView (Open BSI Version 3.0 or newer). Without one of these three utilities running, no other Open BSI utility will function.
"In DataView, the search parameter fields don't show me which signals or data arrays exist in the load.	This occurs when DataView cannot find an (.ACO) file which matches the node load file specified for this node in the currently released network (NETTOP or NETDEF) file.
FOR OPEN BSI 3.0 (or newer) ONLY: "I set the local address	For Protected Mode RTUs with PES03/PEX03, PLS03/PLX03 or newer firmware, ONLY, you can set the Default Switch (SW1-3) to its OFF

Problem or Question:	**Cause and/or Suggested Remedy:**
(*continued)* of the IP RTU using soft switches in LocalView, but I forgot to write it down, and now I can't remember what it was. I can't get back in to read it because I need the local address to start communications? What do I do now?"	*(continued)* position. When the default switch is OFF, the unit will use whatever local address is defined via the hardware switches (SW2). You can use the address specified by SW2 to establish communication, and then view whatever information you previously entered for soft switches. NOTE: When the default switch is used in this way, the communication ports default to BSAP 9600 baud.
FOR OPEN BSI 3.0 (or newer) ONLY: "I made some changes to certain parameters in 'RTU Configure' mode (using LocalView) but they aren't reflected in my running RTU. How come?"	The changes you enter in "RTU Configure" mode are stored in FLASH memory *but* they are NOT activated until the RTU has either been re-downloaded with a new ACCOL load, or the unit has been reset by pressing the reset switch. Reset the unit (or re-download it) and the changes will become active.
"Open BSI communicates with the RTUs via radios. Open BSI seems to be able to send messages, but the RTUs don't answer."	There could be many reasons for this. One possible problem, however, could be that the port in Windows 95 is truncating the messages. See '*Configuration Issues For Radios*' in '*Chapter 4*' for information on how to correct this particular problem.
DOS-BASED TOOLS:	(UOI, TOOLKIT, on-line AIC, TASKSPY, DIAG)
"I get the message **'Communications cannot be established for ACCOL TOOLS functionality.'** What does this mean?"	There are several possible reasons for not being able to communicate, including: Incorrect port setup, incorrect baud rate, bad cable, the node address was specified incorrectly. Go to the Reconfigure Communications Setup Menu and change entries as necessary.
"I get the message **'Timeout this transmission'**. What does this mean?"	This means that the PC running the tools (UOI, Toolkit, etc.) has received no transmissions *from the slave node* (controller) connected to it, for at least 3 times the "**Timeout**" value specified on the Define Local Link portion of the Communication Setup Menus. If you are in the process of establishing communications when this occurs, there is probably a data link or network routing

Problem or Question:	Cause and/or Suggested Remedy:
	(continued) problem. Check to see that the node address and baud rate defined on the Communications Setup Menu are correct. Also check to see that the port in the ACCOL load is set up correctly, and that you are plugged into the correct port on the node. If communications *had* been working correctly, and have now failed (without any configuration changes) check to see that the cable is not loose or unplugged. If this occurred immediately after a download, see the first question on the first page of this table.
"During a download, I get the message **'Error during send extended address'**. What does this mean?"	The download information cannot be sent to the unit. Check to see that the port you are plugged into can accept a download, i.e. it is a Slave Port, and has the required baud rate configured.
"In the middle of a download, I get the message **'Error during send buffer'**. What does this mean?"	The download has been interrupted. Check to see that the cable is plugged in correctly, and that the port and baud rate have been correctly configured.
"I get the message **'Bad Message Received Via Communications'.** What does this mean?"	This is a possible indication of an invalid hardware configuration or improper memory configuration. It can occur, for example, if you try to download a 186-based ACCOL load into a 386-based unit, or vice versa. Check to see what type of unit you have, and make sure that it matches the target node type of the load you want to use.
"I get the message **'Overall Request Error'**. What does this mean?"	This indicates that there is a discrepancy between the ACCOL load running in the unit, and the .ACO/ACL files on the hard disk of your PC. You may need to re-link and re-download the unit to clear up this problem.

Checklist For Diagnosing Communication Problems

This checklist is adapted from information provided by the Bristol Babcock Application Support Group. These represent the most common problems customers encounter when configuring communications.

[ ] 1. DID YOU ACCURATELY DEFINE YOUR NETWORK AND ARE THE NETWORK FILE(S) ACCESSIBLE TO ALL NODES?

For DOS-based ACCOL Tools, and/or Open BSI 2.3 (or earlier):
All nodes in the network with the correct node names and node addresses must be defined in NETTOP. Identical copies of the NETTOP files released during this process (NETFILE.DAT, GLADXREF.DAT, and RTUXREF.DAT) must exist in the proper path on the Network Master (PC or VAX) and on every Pseudo Master in order to be read by the ACCOL Tools. For DOS-based ACCOL tools, this path is whatever directory is set as the \ACCOL directory. In Open BSI 2.3 or earlier, this path is specified in the "**NETTOP Path**" field in the System Startup dialog box in the Open BSI Setup Tool.

For Open BSI 3.0 (or newer):
Network Definition (NETDEF) files must be defined using NetView at the Network Host PC (NHP). The NETDEF files must contain the correct node names and addresses (either BSAP or IP) for all nodes connected to that NHP. Each IP RTU must belong to a configured IP network and have a configured IP communication line. Each BSAP RTU must belong to the configured BSAP network (or BSAP sub-network) and have a configured BSAP communication line. If your system includes proxy workstations (other Open BSI workstations which do NOT have a direct connection to these particular RTUs), then those proxy workstations must have a proxy file specifying the NHP's name and IP address, and the names of all RTUs for which access has been granted. For Proxy Direct Access to IP RTUs, an IP communication line must also be defined at the proxy workstation in order to communicate with the RTUs.

[ ] 2. FOR BSAP NETWORKS ONLY: DID YOU DOWNLOAD EACH TOP-LEVEL NODE THROUGH ITS *SLAVE* PORT TO ENSURE TRANSMISSION OF THE NODE ROUTING TABLE?

The top level nodes should be downloaded via their Slave port, not a Pseudo Slave Port, otherwise a Node Routing Table will not be transmitted to the unit, and any of its Slave nodes will be inaccessible for global communication. The top-level nodes must have Master Ports to communicate with their slaves, and the slave nodes must have Slave

Ports (NOT Pseudo Slave Ports) to communicate with their Master nodes.

If the top level node does NOT have a Slave port, then ACLINK (or the Workbench 'Build' command) will incorporate ANY existing node routing table into the load. As long as the high slave address values are correct throughout the network, peer-to-peer communication may work with an incorrect node routing table, since the Master Module actually uses local communications (although it will report a -23 error on its STATUS_2 signal if there is no node routing table present.) Attempts at global communications, however, will almost certainly fail.

If the top level node DOES have an active, connected Slave Port, it is permissible to download via the Pseudo Slave Port, however, global communications will not be available until a time synchronization with the master (which includes a Node Routing Table transmission) has occurred. To avoid this delay, it is best to follow the rule of downloading via the Slave Port.

[ ] 3. FOR BSAP NETWORKS ONLY: IF YOU HAVE A PSEUDO MASTER PLUGGED INTO A LOWER LEVEL NODE, MAKE SURE YOU DOWNLOAD TO THE CORRECT NODE:

<u>For DOS-based Tools (UOI, AIC, Toolkit, Taskspy):</u>
If you are working with a section of the network, and download via a Slave port, then you should have used the Communications Setup Menu to identify the NAME of the node to which you are connected. Otherwise, the system will "believe" that the current node is a Top-Level Node as defined in NETTOP and messages will almost certainly not be directed to the proper nodes.

<u>For Open BSI 2.3 (or earlier) Users:</u>
If you are working with a section of the network, and download via a Slave port, make sure the "**Local**" option is selected, and the correct node is selected in the Communication Line Parameters dialog box of the Open BSI Setup Tool. Otherwise, the system will "believe" that the current node is a Top-Level Node as defined in NETTOP and messages will almost certainly not be directed to the proper nodes.

<u>For Open BSI 3.0 (or newer) Users:</u>
Establish communications using LocalView, then activate the Downloader from within LocalView. This will ensure you are downloading to the correct node.

[ ] 4. FOR BSAP NETWORKS / SUB-NETWORKS ONLY: ARE BSAP LOCAL ADDRESSES SET CORRECTLY?

The node local address, set via jumpers/switches on the unit, must match the corresponding address defined for that node name in the NETTOP or NETDEF file, and the address defined in the Network Master software.

If Trolltalk-PC is being used, has TROLLSET been run with the correct address? If Genesis is being used, do DEVICE block BRISTOL PARAMETERS show the correct address for the node to which Genesis is attached, and the *same* node ID as used in NETTOP. For ACCOL tools, see item #3 above.

Are address DIP switches functioning properly? If dirty, the apparent address may not be active.

[ ] 5. FOR BSAP NETWORKS ONLY: HAVE MASTER PORTS BEEN DECLARED CORRECTLY?

Do the BSAP nodes defined in your network file which have slave nodes on the level below them, have Master Ports defined in them?

Are the Master Port "**High Slave Addr**" and "**Baud Rate**" fields configured correctly?

If using a radio system, especially, are Master Port "**Timeout**" values configured correctly (typically 0.6-0.8 seconds; maybe more if there is a repeater.)

[ ] 6. BSAP NETWORKS ONLY: ARE POLL PERIODS (#POLLPER SIGNALS) SET PROPERLY?

Are the Slaves *really* being polled at better than or equal to the assigned #POLLPER? If not, then the Slaves' time-out may be insufficient.

[ ] 7. ARE ACCOL TASK PRIORITIES SET TOO HIGH?

ACCOL task priorities should be set *below* 32 to avoid conflicts with communication task priorities.

[ ] 8. IS THE NODE ARRAY (#NDARRAY) CONFIGURED CORRECTLY?

It is recommended that the #NDARRAY be configured to turn OFF polling for failed, or non-existent nodes. To work correctly, however, it must reference a logical data array, whose elements must be ON for communications to proceed.

If you specify a non-zero value for the #NDARRAY, and do NOT follow through by properly configuring the array, all polling of slave nodes will be turned OFF.

[ ] 9. ARE #LINE AND OR #NODE ALARMS OCURRING?

These alarms indicate failures of the communication line, or slave node, respectively.

If you are using BSAP peer-to-peer communication, Master Modules will also report '-14' on the STATUS_2 terminal if the addressed node is in alarm.

[ ] 10. DO COMMUNICATION STATISTICS AND BUFFER USAGE LOOK O.K?

See '*Troubleshooting Techniques*' earlier in this chapter for details on interpreting communication port and buffer statistics.

[ ] 11. WERE DEBUGGING FLAGS OR LOGIC ACCIDENTALLY LEFT ACTIVE IN A TASK?

If you were using ACCOL Workbench in debugging mode, or Taskspy, all debugging flags (BREAKPOINTS, SKIPS, ABORTS) should have been cleared before exiting. If any ACCOL task is accidentally left in STEP mode, it will not execute properly.

For ACCOL Workbench users using on-line mode, you can check whether debugging flags remain active by clicking on "**View**" in the menu bar, "**Node Information**" in the pull down menu, and "**Debug Flags**" in the second pull down menu. See the *ACCOL Workbench User Manual* (document# D4051) for more information.

For Taskspy users, you can choose the 'Show Flags' poke point from the

Module Line Menu. See the *Taskspy Manual* (document# D4083) for more information.

Task control statements in ACCOL (e.g. WAIT FOR or SUSPEND) can also halt task execution.

[ ] 12. DOES THE NETWORK 3000 CONTROLLER HAVE ENOUGH COMMUNICATION I/O BUFFERS?

At least 10 additional I/O Buffers should be assigned. As a general rule, it is better to have too many buffers than too few.

[ ] 13. IS THE EXPANDED NODE ADDRESSING GROUP NUMBER SWITCH SET CORRECTLY?

If you are using Expanded Node Addressing, Switch Bank 1 Switch #8 on a DPC 3330/DPC 3335 or RTU 3310 must reflect the proper Group Number of the controller. OFF indicates Group 1 or above; ON indicates Group 0. See the '*Expanded Node Addressing*' section of the *ACCOL II Reference Manual* (document# D4044) for more information. If you are using a Protected Mode DPC 3330/DPC 3335 with PES03/ PEX03/ PLS03/ PLX03 or newer firmware, the Expanded Node Addressing Group Number must be set via a soft-switch by using LocalView in "**RTU Configure**" mode.

If you are NOT using Expanded Node Addressing, leave the switch ON.[4]

<u>THE FOLLOWING ITEMS APPLY STRICTLY TO OPEN BSI 3.0 AND NEWER USERS:</u>

[ ] 14. IF YOU ARE DEFINING AN IP NETWORK, DID YOU INSTALL TCP/IP ON EACH PC WHICH WILL BE RUNNING OPEN BSI 3.0?

If not, please consult the Windows™ on-line help files for instructions on how to do this.

In addition, the IP address you enter during TCP/IP configuration should be the primary IP address you use when defining the same PC as a Network Host PC (NHP), and the IP mask you enter during TCP/IP

[4] *In older manuals, this was incorrectly shown as OFF.*

configuration should be the same IP mask you enter for the IP communication line of this NHP.

[ ] 15. IN NETVIEW, DID YOU DEFINE ALL FOUR REQUIRED COMPONENTS FOR YOUR SYSTEM?

a. Did you create a Network Host PC (NHP) using the System Wizard?
b. Did you create a Network (either BSAP or IP as appropriate) using the Network Wizard?
c. Using the RTU Wizard, did you add one or more RTUs (either BSAP or IP as appropriate) to your network?
d. Did you define a Communication Line (either BSAP or IP as appropriate) using the Comm Line Wizard?

If you answered NO to any of these questions, Open BSI network communications will be impossible.

[ ] 16. IF YOU ARE USING AN IP NETWORK, DID YOU DEFINE AN IP PORT IN THE ACCOL LOAD FOR EACH IP RTU?

Each IP RTU must have an IP port defined in the *COMMUNICATION PORTS section of the ACCOL load. Currently, this must be the Ethernet port, however, serial ports *may* be supported in a future release.

[ ] 17. IF YOU ARE USING AN IP NETWORK, HAVE IP MASKS AND IP ADDRESSES BEEN DEFINED CORRECTLY, AND HAVE THEY ALSO BEEN SET CORRECTLY IN EACH RTU?

This is one of the most common sources of errors. In NetView, the Network Host PC (NHP) must have an assigned IP address, each RTU must be assigned an IP address, and the communication line must define a range of addresses (via the IP value and IP mask) which encompasses the addresses of the NHP and all RTUs which will use the line.

IMPORTANT: Besides assigning RTU IP addresses in NetView, you *also* need to set the IP address in the actual RTU using the "RTU Configure" option of LocalView, and *then* you need to reset or re-download the RTU for the IP address to become active.

[ ] 18. IF YOU ARE USING PROXY DIRECT ACCESS FOR IP RTUs, DID YOU DEFINE A COMMUNICATION LINE AT THE PROXY WORKSTATION?

Proxy direct access, which allows access to RTUs belonging to *another* NHP, without routing messages through that NHP, requires that an IP communication line for the proxy RTUs be defined at the proxy workstation.

[ ] 19. IF YOU ENTERED LONG NODE NAMES (GREATER THAN 4 CHARACTERS) IN NETVIEW, DID YOU VERIFY THAT THEY ARE COMPATIBLE WITH OTHER SOFTWARE DRIVERS USED IN YOUR SYSTEM?

Open BSI 3.0 supports long node names (up to 16 characters). Older versions of Open BSI, the DOS-based ACCOL Tools, UOI, TMS, as well as third-party HMI/SCADA packages using older Bristol drivers do NOT support long node names. If you are mixing old and new software, you should restrict your node names to 4 characters.

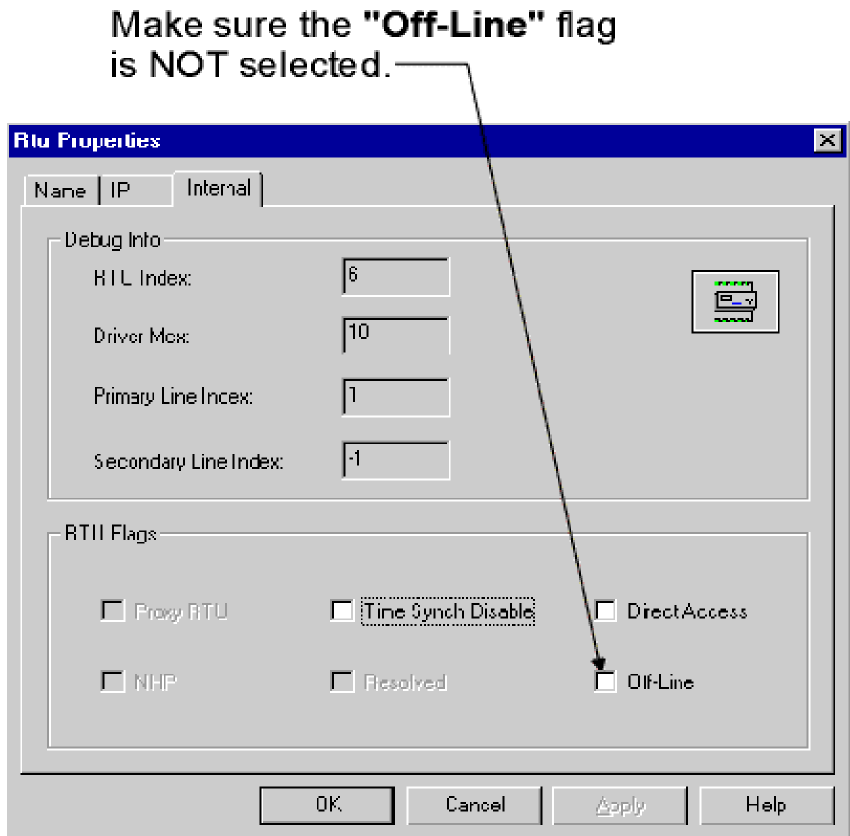
[ ] 20. ARE THE UDP PORT NUMBERS FOR THE IP DRIVER, AND THE UDP PORT NUMBERS FOR TIME SYNCHS CONSISTENT THROUGHOUT EACH WORKSTATION (NHP) AND RTU IN YOUR NETWORK?

These numbers must MATCH on all workstations and RTUs in your system. If they are inconsistent on a particular RTU or workstation, that RTU or workstation will be unable to communicate with the rest of the network. If necessary, use LocalView in "**RTU Comm Config**" mode to alter these numbers in the Configure General IP Parameters dialog box, then reset or re-download the RTU. (If they are inconsistent at the workstation, edit them in the **CONSTANTS** section of the NETDEF file and then stop and re-start NetView.)

[ ] 21. IS THE RTU's OFF-LINE FLAG SET, WHICH WILL PREVENT COMMUNICATIONS?

Certain port configuration problems can result in the RTU's off-line flag being set, to prevent communication attempts to a failed RTU. The user can also set the off-line flag accidentally from the 'Internal' page of the RTU Properties. If this happens, no communication with that RTU will occur. To check if the off-line flag is set, right-click on the RTU in the

NetView tree, and choose "**Properties**". Then select the 'Internal' tab, and de-select the "**Off-Line**" RTU flag, if it is selected.



Appendix A - Summary of Port Types

A full description of communication port types is included in the '*Communication Ports*' section of the *ACCOL II Reference Manual* (document# D4044). The table, below, provides a summary of various port types.

Port Type:	When would I need to use this type of port?
Master Port	This type of port is required in order to communicate with the slave nodes of the current node.
EAMaster Port	This type of port is required in order to communicate with slave nodes of the current node, ONLY when the Expanded Node Addressing technique is being used.
Slave Port	This type of port is required in order to communicate with the master node of the current node. Slave Ports are also used to receive downloads of an ACCOL load (ACL file), and to receive time synchronization/node routing table messages. A node can have ONLY ONE SLAVE PORT, because any node can be a slave to only one master node.
Pseudo Slave Port	This type of port allows a pseudo master device (for example, a PC or laptop running UOI or Open BSI) to be connected to this node. This type of port cannot collect alarm information.
Pseudo Slave with Alarms Port	This type of port allows a pseudo master device (for example, a PC or laptop running UOI or Open BSI) to be connected to this node. This type of port can collect alarm information.
Serial CFE Port	This type of port is ONLY used in top-level (Level 1) nodes which are connected to an Enterprise Server VAX computer.
RIOR Port	This type of port is ONLY used when a DPC 3330, DPC 3335, or RTU 3310 has one or more connected RIO 3331 remote I/O racks.
Custom Port	This type of port is used for communication with foreign devices (PLC's, third party controllers, etc.) via a configured Custom communication protocol.
VSAT Slave Port	This is a special type of Slave Port which was developed for use with satellite links. It allows a minimum and maximum response time for the controller to be specified. In order to function, the user must have VSAT Master software to support it. PC workstations with Open BSI Utilities (Version 2.1 and newer) can serve as VSAT Master devices, however, Network 3000-series controllers CANNOT serve as VSAT Masters.
IP Port	This type of port is ONLY supported in DPC 3330/DPC 3335 Protected Mode units with PES03/PEX03 or newer firmware. It allows communication using the Internet Protocol (IP) via Ethernet.

Port Type:	When would I need to use this type of port?
Logger Port	This type of port allows ASCII information to be output to an attached printer or terminal device.

Appendix B - BSAP, VSAT, and Immediate Response Message Traffic

Earlier in this manual, in the discussion of the data link, we discussed the subject of how a BSAP Master node (a node with a Master Port) polls its Slave nodes for data. We didn't bother to discuss the underlying message traffic involved in these requests for data because it is normally transparent to the user. There are certain situations, however, when someone configuring communications may need to know some of the details of this message traffic.

One such situation is when there is a heavy cost (in terms of money) associated with each message, as in using a satellite communications link. Satellite time is often very expensive, therefore, as the amount of underlying message traffic increases, so will the financial burden incurred by the user.

This appendix will briefly discuss (in greatly simplified terms) the underlying message traffic involved in three different polling methods, and the major advantages and disadvantages of each method.

These methods are:

- BSAP Transactions
- Immediate Response Transactions
- VSAT Transactions.

Appendix B - Message Traffic

BSAP Transactions

BSAP is a standard method of communication among Network 3000-series controllers. A Master node sends out messages on its Master Port, and the Slave node receives them, and sends out responses on its Slave Port.

For each *successful* transaction between a Master node and its Slave node, there are six different types of messages involved. These messages are described, below:

1) Data Request: The Master node says to its Slave node "Do you have *x* for me?" (Where *x* refers to a specific type of data, such as a set of certain signal values, etc.)

2) Acknowledgment (ACK): The Slave node replies "I received your request, and I'm working on it." (The slave node then has to start searching through its internal data base to find the requested data *x* so it can compose a response message.)[1]

3) POLL: The Master node now issues a poll to the slave saying "Give me any data".

4) RESPONSE: The Slave node sends a response message to the Master containing the requested data *x*.

5) UP ACKNOWLEDGMENT (UP ACK): The Master says to the Slave node "O.K. the response message made it up to me. You don't have to re-send it."

6) DOWN ACKNOWLEGEMENT (DOWN ACK): The Slave node says "O.K., I'm finished down here. Our transaction is over."

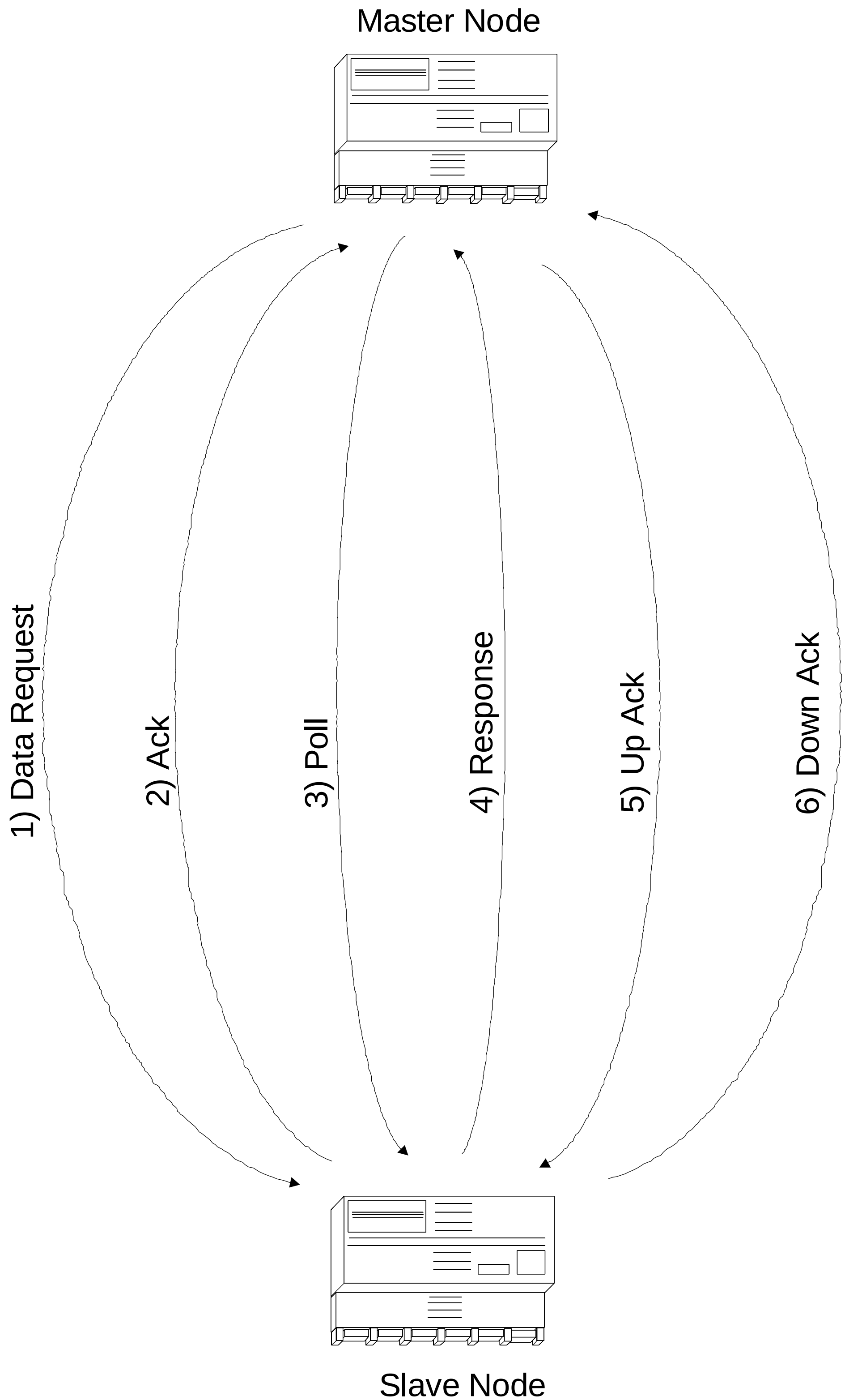
<u>Advantages:</u>

- Compatible with all Network 3000-series controllers.

<u>Disadvantages:</u>

- Up to six messages are involved in each transaction.

[1] *If for some reason, the Slave node doesn't have the requested data, it will send an 'ACK NO DATA' message which ends the transaction. Since we're talking about successful transactions, we won't go into that subject now.*



BSAP Transaction

Immediate Response Transactions

The immediate response concept was initially limited to only certain Network 3000-series devices such as the EGM 3530-10A and EGM 3530-50A TeleFlow, the 3508 TeleTrans Transmitter, the SLC 3740, and the older model RTU 3320. These devices were suitable for immediate response because each one always had a fixed set of data inside them, meaning that there could be a fixed set of messages, always 'ready to go'.

Network 3000-series controllers which are 'download-able' do NOT have a fixed set of data messages, because there is an essentially unlimited set of possible messages, based on how the user creates their particular ACCOL load. To implement immediate response in these devices, users must configure an Immediate Response Delay for the Slave Port. (See Chapter 4.)

An Immediate Response polling transaction can *potentially* use fewer messages than BSAP. A successful transaction uses only two messages. If the transaction is NOT immediately successful, however, immediate response transactions will revert to the standard six message BSAP transaction.

A successful immediate response transaction is as follows:

1) Data Request: The Master node says to its Slave node "Do you have *x* for me?" (Where *x* refers to a specific type of data, such as a set of certain signal values, etc.)

(A user configurable delay period then occurs, during which the Slave Node composes a response containing data *x*)

2) Response: The Slave node sends a response message to the Master containing the specific requested data *x*. The transaction is now considered over.

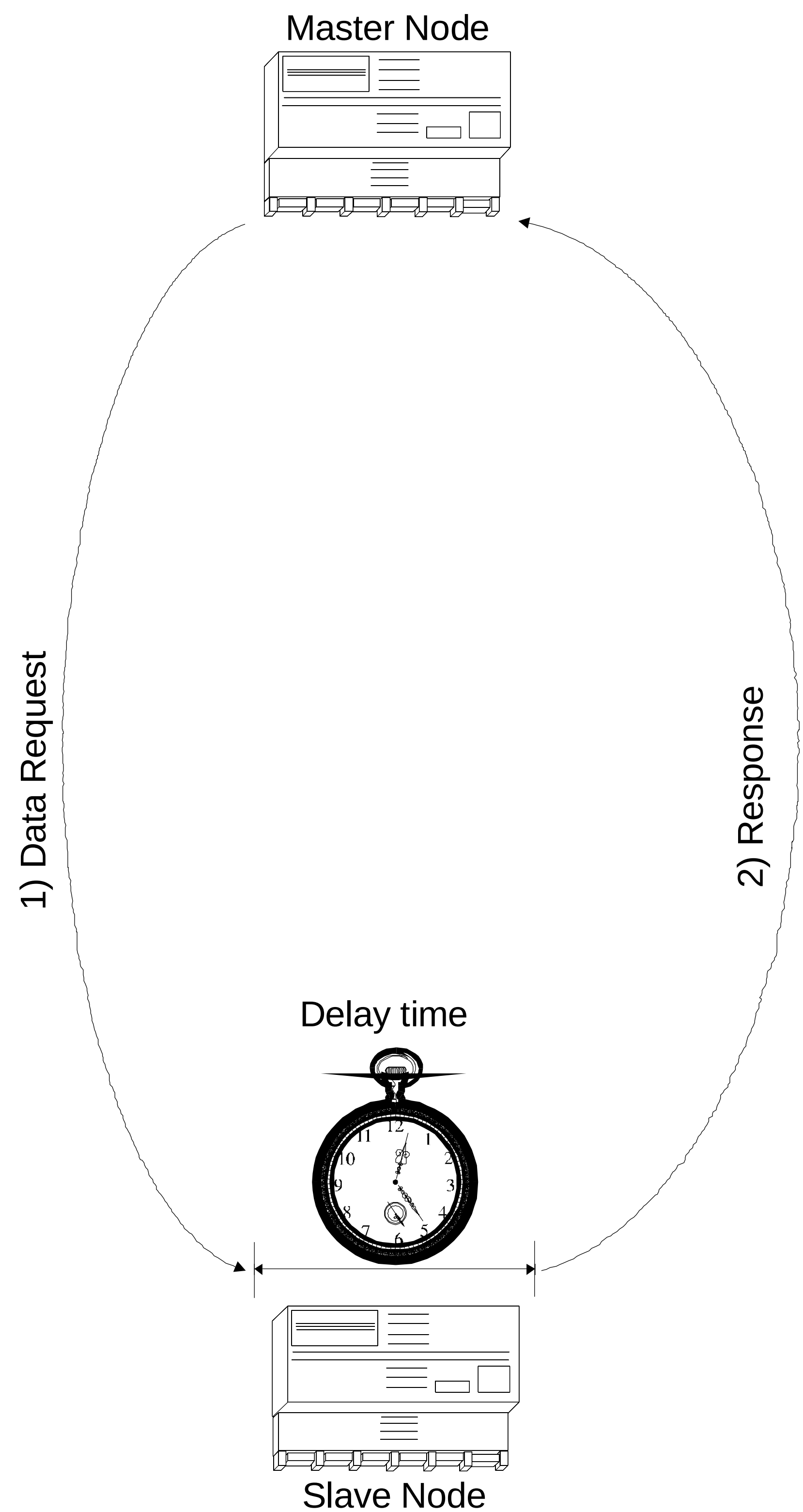
Note: If the Slave node was unable to obtain the specific requested data x within the required period, the polling method will revert to standard BSAP, with a DATA REQUEST, ACK, POLL, RESPONSE, UP-ACK, and DOWN-ACK.

<u>Advantages:</u>

- Message traffic can potentially be reduced by two-thirds.

<u>Disadvantages:</u>

- Not all Network 3000-series nodes support immediate response.



Immediate Response Transaction

VSAT Transactions

VSAT transactions differ from BSAP and immediate response transactions in that Network 3000 controllers CANNOT be configured as VSAT Masters. Open BSI Workstations, however, can communicate with top-level nodes configured as VSAT slaves.

VSAT Slave configuration involves defining a VSAT Slave Port. (This is discussed in the '*Communication Ports*' section of the *ACCOL II Reference Manual*, document# D4044). This involves, among other things, specifying a minimum response time, and a maximum response time, for the port. These values specify a 'window' of time during which the VSAT Slave Port must respond to its master.

A successful VSAT transaction is as follows:

1) Data Request: The Open BSI Workstation says to its top-level VSAT slave node "Do you have *x* for me?" (Where *x* refers to a specific type of data, such as a set of certain signal values, etc.)

(There will be a delay which will be somewhere between the minimum and maximum response times defined by the user for the VSAT Slave Port.)

2) Response: The VSAT Slave node sends a response message to the Open BSI Workstation which *may or may not contain the* specific requested data *x*. It may, in fact, contain data which had been requested during a *previous* transaction, and had not yet been delivered, or it could contain an alarm if one happens to be coming in at that time (alarms always take priority over other messages). For example, if a request for all #TIME signals is made, but a previous request for some data array values is ready first, and it is between the minimum and maximum response times, the data array values will be sent, and the #TIME signals will be sent during a subsequent transaction.

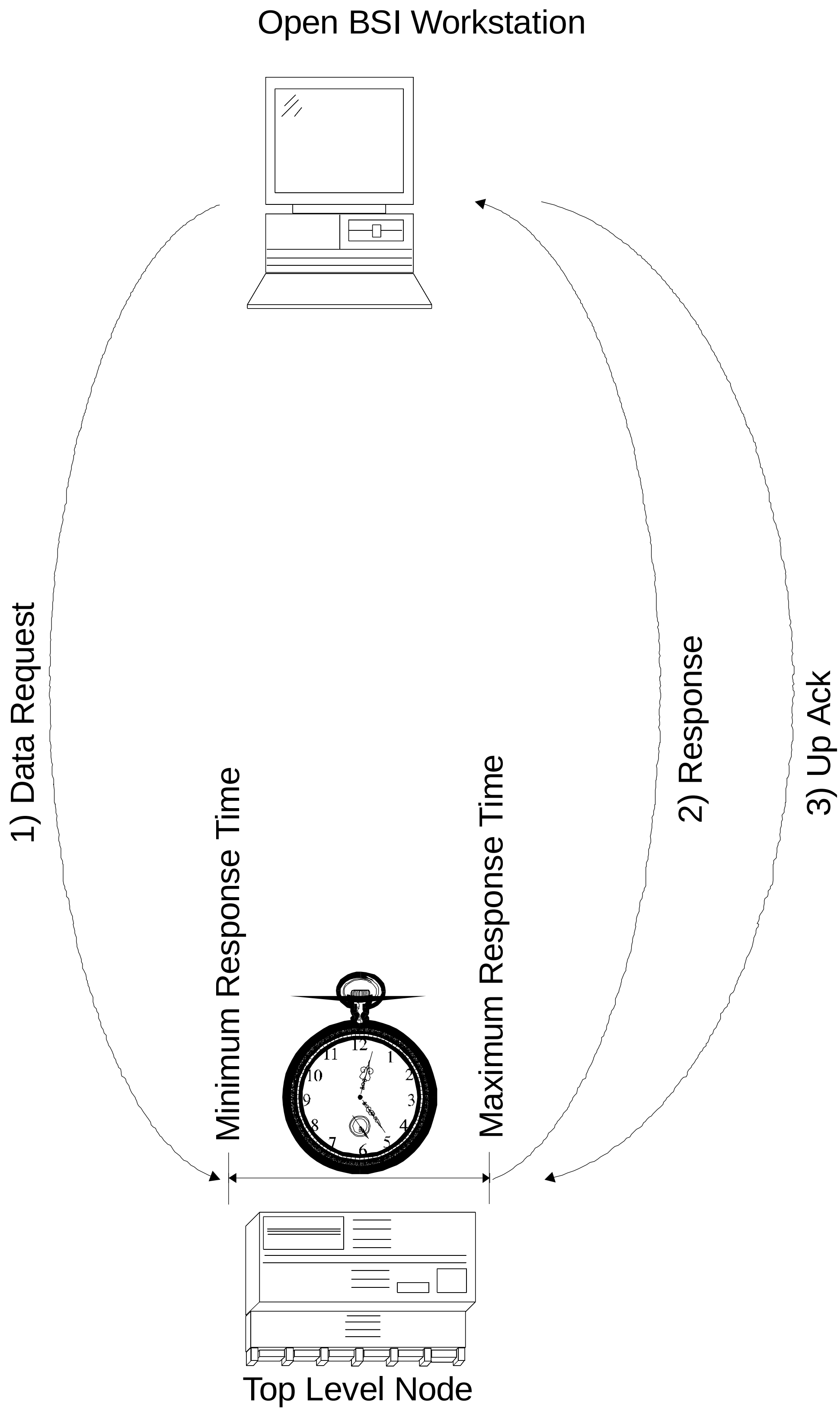
<u>Advantages:</u>

- Message traffic can potentially be reduced by half.

<u>Disadvantages:</u>

- Network 3000-series controllers CANNOT serve as VSAT Masters. Only Open BSI Workstations can be used for this purpose.

- Unlike immediate response, and BSAP, a given response message may not include the exact data requested by the master. In this case, whatever data is available will be provided. The available data could be data fulfilling a previous request, or it could be an incoming alarm message.



VSAT Transaction

Is this still unclear?

If this is unclear to you, let's think of a Slave node as a bakery, and the Master node, as a customer trying to buy jelly doughnuts:

<u>A BSAP transaction in the bakery would appear as follows:</u>

Customer: I'd like a dozen jelly doughnuts. (Data Request)

Cashier: OK! *Cashier starts walking over to where the jelly doughnuts are.* (ACK)

Customer: Do you have them? (POLL)

Cashier: *hands the jelly doughnuts to customer* (Response)

Customer: These look like good jelly doughnuts! (Up ACK)

Cashier: Thanks. (Down ACK)

At this point, the transaction is finished, and the customer goes on to the next errand, say a trip to the hardware store. (That is similar to the Master going on to the next node.)

<u>An Immediate Response transaction in the bakery would occur as follows:</u>

Customer: I'd like a dozen jelly doughnuts. (Data Request)

(*The cashier goes into the back room, without saying anything, and a few moments later, emerges with the jelly doughnuts*) This is equivalent to the delay time while the immediate response node composes a reply.

Cashier: *hands the jelly doughnuts to the customer.* (Response)

At this point, the transaction is finished.

<u>According to the VSAT Slave method, the transactions would go as follows:</u>

You: I'd like a dozen jelly doughnuts. (Data Request)

(*The cashier goes into the back room, without saying anything, and a few moments later, emerges with a box*)

Cashier: I don't have any jelly doughnuts right now, but here's the box of crullers you asked for the last time you were in. *(hands the crullers to the customer).* (Response). Note that the cashier must respond with something. This is equivalent to the port responding sometime between the minimum and maximum response times.

Customer: Okay. I'll take the crullers. (Up ACK)

At this point, the transaction is finished. The request for jelly doughnuts has still not been fulfilled, however, so a new transaction begins:

Customer: Do you have anything else for me? (Poll)

Cashier: Here are the jelly doughnuts. (Response)

Customer: Thanks. (Up ACK)

In some cases, the data request may be fulfilled during the first transaction. In all cases, the VSAT Slave will send whatever it has ready before the maximum response time expires.

Conclusions:

BSAP transactions are appropriate for most situations. If, however, the underlying message traffic of BSAP results in higher cost communication, one of the other methods (immediate response, or VSAT) may be a viable substitute.

Glossary

ACCOL Load(s)	also known as the .ACL file. It is called the ACCOL Load file because it is in a machine-readable format which is downloaded into the Network 3000-series controller. Once downloaded, the controller executes instructions in the ACCOL load file in order to measure or control a plant or process.
ACCOL Task(s)	is a series of modules and control statements which execute sequentially as a functional block. Task execution occurs at a user-specified rate and priority.
ACCOL Tools	is a set of software programs which are used to create and debug ACCOL load(s). The Windows™ based version of ACCOL Tools consists of the ACCOL Workbench program. The DOS-based version of ACCOL Tools consists of the ACCOL II Interactive Compiler, NETTOP tools, and optionally, the ACCOL II Batch Compiler (ABC), Toolkit, and Taskspy programs.
Address	see 'Local Address' for BSAP nodes, or 'IP Address' for IP nodes.
Alarm Message	is a communications message, generated when an analog or logical signal enters an alarm state. Alarm messages are passed up through the network to HMI software at the operator workstation, to notify the operator that an alarm condition exists.
Application	is a program or function which makes use of the data link to transmit messages. Among the applications supported in Network 3000 are Open BSI utilities such as DataView, Intellution® FIX®, and ACCOL Master Modules.
Baud Rate	the character-by-character rate at which data flows on a communication line.
Bit, Bit(s)	the binary representation used to represent data in a digital computer. A bit is either '0' (OFF or FALSE), or '1" (ON or TRUE).
BSAP	the Bristol Synchronous/ Asynchronous Protocol. This is a standard communication protocol used by Network 3000 devices. If expanded node addressing is used, a different version called expanded BSAP (EBSAP) is necessary.

Glossary

Buffers	are pre-allocated blocks of memory used to hold data messages. Buffers must be allocated in the controller, and if you are using Open BSI, buffers must also be allocated at the PC.
Communication driver(s)	these are software programs which facilitate communications between different devices.
Communication Lines	refers to a connection between two devices for transmitting data. (See 'Data Link').
Communication Ports	these are devices on the Network 3000 controller, which allow the controller to exchange data with other controllers and devices. They are named the following: BIP 1, BIP 2, Port A, Port B, Port C, Port D, Port G, Port H, Port I, and Port J, and Ethernet. Not all controllers support all ports.
Communication protocol	this is the method by which a particular device communicates. The standard communication protocol used by Network 3000 controllers is BSAP, (there is also a separate version of BSAP used for large networks called EBSAP). Some Network 3000 controllers can also communicate using Internet Protocol (IP). In addition, some Network 3000 controllers support the use of custom communication protocols, in which a controller port will be configured to communicate with a foreign device or PLC, e.g. the Modbus protocols.
Data Concentrator	This is a Network 3000 controller, in a BSAP network, which has a Master Port, through which it accepts data from one or more slave controllers. Data concentrators are typically used when ACCOL control decisions must be made based on data from more than one controller.
Data Link	the Data Link, for purposes of this manual, refers to the communication port, and the connection hardware (cable, radio, modem, etc.) The data link handles the actual physical transmission of data between nodes in the network, without regard to the content of the data messages. There are numerous data link software parameters which must be configured.
Dotted decimal	a notation used to make 32-bit IP addresses and sub-net

	masks more user friendly. Instead of representing an IP address as 32 ones and zeroes, it is separated into four 8-bit groups, separated by periods (dots). Each 8 bit group is then converted to a decimal number. For example, the 32 bit IP address: 00000001 00000011 00000010 00001001 is converted to: 1 . 3 . 2 . 9
Ethernet	an industry-standard local area network (LAN), originally developed by Xerox Corporation.
First level slave	same as Top Level Node. See 'Top Level Node'.
Global	a message between two Network 3000 nodes is considered 'global' if it must pass between one or more additional nodes on its journey. For example, if a message between DPC2 and DPC7 goes directly, without passing through any other nodes, it is considered 'local', and not 'global'. If the message must pass through DPC5 on its way to DPC7, then the message is considered 'global'.
Human-Machine Interface (HMI)	this is a software package, such as OpenEnterprise, Intellution® FIX®, or Iconics Genesis, which is used to display and report data for an operator. Formerly referred to as MMI.
Internet	a constantly changing collection of computer networks, used primarily for commercial and education purposes.
Internet Protocol	see IP
IP	Internet Protocol - a standard communication protocol used for transmitting data over a computer network, and *between* computers on different networks.
IP Address	A 32-bit binary number used to identify an IP port in an IP network. IP addresses must be converted to dotted decimal format in which the 32 bits are separated into 4 8-bit groups, and each group of 8 bits is converted to a decimal number from 0 to 255. Each group is then

	separated by a period. Example: 120.0.0.1 NOTE: Because IP addresses are tied to *ports* and NOT *nodes*, a particular RTU with more than one IP port has more than one IP address associated with it.
IP mask	see Sub-net mask
IP node	refers, in this context, to a Network 3000 controller (RTU) which supports IP communication. Currently, the only Network 3000 IP nodes are 386EX Protected Mode versions of the DPC 3330 and DPC 3335 with Ethernet hardware and PES03/PEX03 or *newer* firmware. NOTE: ControlWave supports IP communication, but is NOT part of the Network 3000 product line. 2nd Definition: IP node can also refer to an Open BSI 3.0 or newer workstation (NHP), because, like an RTU, it has an IP address, and communicates using IP.
IP port	In software: a port type, which must be specified in ACCOL Workbench, to identify that this ACCOL load will be running in an IP node. In hardware: The physical port (either Ethernet hardware or Serial port) used for IP communication at the RTU. An RTU can have more than one IP port, however, an RTU with an IP port CANNOT include a Slave Port, VSAT Slave Port, or Serial CFE Port.
Levels	refers to the hierarchical levels in a BSAP network. A node's level indicates the number of communication lines between it and the Network Master. There can be up to six levels in the network, not counting Level 0 (the Network Master).
Local	a local message is a message which travels directly from one node to another, without passing through any other nodes on its journey.
Local Address	is used to specify the location of a node within the network. In BSAP networks, a local address (an integer from 1 to 127) is set using jumpers/switches on the controller (or software switches). This address is then used to reference the controller within the network files (NETTOP or NETDEF) as a node under a given master node. Each unit also has a unique global address, defined automatically by the system.

Local Area Network (LAN)	a group of computers in a contiguous area which are connected together for purposes of exchanging data.
Local portion	the right-most part of an IP address. This part of the IP address must include some unique part which uniquely identifies a particular IP node. The local portion of the IP address is identified by all zeroes in the corresponding bits of the sub-net mask.
LocalView	allows local communication (BSAP only) with any Network 3000 series controller. It is also used to configure cold download parameters and IP addresses for IP controllers.
Master	refers to a BSAP node containing a Master or EAMaster Port. This port is used to request data from slaves (which have Slave Ports). 2nd Definition: Master also refers to the Master Module used in BSAP peer-to-peer communication; do NOT confuse the two uses of the term.
NETDEF	Network Definition File. Configuration information defined in NetView is stored in the NETDEF file. NETDEF files have the extension (NDF).
NETTOP	is one of the DOS-based ACCOL Tools - - also known as the Network Topology Program - - NETTOP5. It allows the user to define the structure of the BSAP controller network including network levels, addresses, etc. The NETTOP files containing this information are NETFILE.DAT, GLADXREF.DAT, and RTUXREF.DAT. If you have Open BSI 3.0 (or newer) you must use NetView instead of NETTOP.
NetView	is the program in Open BSI 3.0 (or newer) which starts communications, and allows the user to configure details about the characteristics of the communication network and RTUs. It also allows the user to monitor the 'health' of the communication network. (NOTE: NetView replaces the Open BSI Setup Tool, Open BSI Monitor, and NETTOP tools which are used with Open BSI 2.3 or earlier.)

Network 3000	a product name for a family of Bristol Babcock digital remote process controllers and related equipment.
Network 3000 controller	see RTU
Network Host PC	Abbreviated as NHP, this is any PC workstation running Open BSI 3.0 (or newer). If the workstation has attached RTUs, it is said to be 'host' to those RTUs, and they must be defined in its Network Definition (NETDEF) file. Any other Open BSI 3.0 workstation which wants to contact this NHP's RTU's must have proxy access to them. The other workstation (proxy workstation) does NOT need to know the address of the RTU it wants to communicate with, it only needs to know the address of the NHP which 'hosts' that RTU, and the RTU's name. NOTE: For BSAP networks, the NHP can be considered the Network Master device.
Network Master	is the host computer (usually a PC) which sits at the top of a BSAP network of Bristol Babcock controllers. The Network Master polls the controllers for data, and generally runs either Open BSI or UOI software. In addition, the Network Master usually includes HMI/SCADA software. The Network Master is at Level 0 in the BSAP network.
Network portion	is the left-most part of the IP address which identifies which network this IP RTU or workstation belongs to. Every workstation or RTU in a given network will have matching network portions in their address. The network portion is identified by all 1's in the corresponding bits of the sub-net mask. NOTE: The network portion must begin at the left-most part of the address, and must be contiguous.
Network Routing	the information necessary to identify where a particular data message should be sent, including node names, addresses, etc.
Network Topology Program	see NETTOP
NHP	see Network Host PC
Node	a Network 3000 controller which is part of a network of

	controllers. 2nd Definition: Node can also refer to a PC in the network.
Node Load File	the base name of the ACCOL load file (.ACL) running in a particular node.
Node name	a unique name, assigned for a particular node.
Node Routing Table	a portion of the network (NETTOP or NETDEF) file, customized for each node, which provides information to the node about its relative location in the network.
Open Bristol System Interface	see Open BSI
Open BSI	stands for Open Bristol System Interface. Open BSI is a set of software utility programs which facilitate data collection and communications with a network of Bristol Babcock Network 3000-series controllers. Among the utilities in the standard Open BSI set are either the Open BSI Setup Tool (Open BSI 2.3 or *earlier*) or NetView (Open BSI 3.0 or *newer*), the DataView utility, the Remote Communication Statistics Tool, and the Downloader.
Packets	these are data messages sent through a communication line.
Peer-to-peer communication	An application which allows the transfer of signal list or data array data values from one node to another. <u>For BSAP networks:</u> In BSAP networks, peer-to-peer communication is supported using the ACCOL Master /EMaster Module and Slave Modules. BSAP supports peer-to-peer communication between a given node and its master, any of its slaves, or any of its siblings (nodes which share the same master). <u>For IP networks:</u> Peer-to-peer communication is supported between any two nodes in a Bristol IP network using ACCOL IP_Client / IP_Server modules.
Poll, Polling	a request for data in a BSAP network.
Poll Period	Poll periods are specified for communication ports using the #POLLPER.nnn system signals. The exact meaning of

this term varies depending upon which type of port is being defined. For a Master/Expanded Addressing Master Port, the poll period refers to how often the port will request data from all of its slave nodes on the line. So, for example, if the poll period for a Master Port is 1 second, it means that each second, the Master Port will attempt to sequentially send a request for data to each and every slave node on this line.

For a Slave Port, Pseudo Slave Port, or Pseudo Slave with Alarms Port, the poll period has a totally different meaning. In this case, it is the length of time the Slave controller will wait to receive a poll from its master. If the slave controller does NOT receive a poll before expiration of the poll period, it discards its data messages because its master is considered ‘dead’.

For an IP port, the poll period refers to a timeout period after which the IP line is considered 'dead'. If one half of the timeout period has expired, with no traffic received on the line, the IP controller will attempt to communicate with NHPs.

Private Line — refers to a modem which does NOT dial a phone number. Instead, it has a direct, dedicated phone line.

Pseudo Master — a computer (usually a PC or laptop running UOI or Open BSI) which is plugged into the Pseudo Slave or Pseudo Slave with Alarms Port of a controller.

Public Switch Network Line — refers to a modem which dials a telephone number using the public telephone system.

Remote Process I/O — is used in situations where I/O needs to be physically separate from the controller. Remote I/O uses RIO 3331 remote I/O racks connected to a DPC 3330/DPC 3335 or RTU 3310 controller. It is also necessary if you have a redundant pair of DPC 3330s which need to have I/O cards; the I/O must be installed in RIO 3331(s).

Report By Exception (RBE) — is a method of data collection in which data is collected only when it changes.

Response timeout
: an error statistic which is reported in BSAP networks when a Slave node does NOT respond to its Master node within an expected period of time. After a certain number of response timeouts (default is 3) a Slave node is considered 'dead'.

Router
: this is a computer which allows data to be passed between two different networks, which cannot communicate directly.

RTU
: Remote Terminal Unit. This refers to any Network 3000-series remote process controller such as the DPC 3330, DPC 3335, RTU 3310, RTU 3305, EGM 3530, RTU 3530, GFC 3308. Often referred to by the terms 'node' or 'remote'.

Slave
: refers to a BSAP controller which contains a Slave Port. 2nd Definition: A Slave Module used in BSAP Peer-to-Peer communication; do not confuse the usage of the terms.

Slippage
: an occurrence of an ACCOL task not running at its scheduled time because it had not yet completed its previous execution. Slippage can be caused by setting the task rate too fast.

Sub-net mask
: (also known as the IP mask) is a 32-bit binary number which identifies which bits of a corresponding IP address are used to specify the network portion of the address, and which bits are used to specify the local portion of the address. If a 0 appears in the sub-net mask, all bits to right must also be 0. Like IP addresses, the sub-net masks must be converted to dotted decimal format in which the 32 bits are separated into 4 8-bit groups, and each group of 8 bits is converted to one of the following decimal numbers: 255, 254, 252, 248, 240, 224, 192, 128, or 0. Each group is then separated by a period. For example, 255.255.0.0

Sub-network
: this is a section of an IP network, as identified by a common sub-net mask for each RTU or workstation in the section. It is also possible to have a BSAP sub-network underneath an IP RTU. (In this case, the BSAP RTUs do

NOT have IP addresses, but the IP RTU at the top of the BSAP network does have both an IP address, and a BSAP address.)

Supervisory Control and Data Acquisition (SCADA) is a method of process control in which a supervisory computer, typically running HMI software, collects data from a network of remote process controllers, displays the data for an operator, and allows the operator to issue commands which are sent out to control the process.

Task Priority is a value, assigned by the ACCOL programmer, to each ACCOL Task. This value is used to determine which task should execute next. Priorities can range from 1 to 64. Tasks which perform important calculations should be given higher priorities.

Task Rate specifies how often an ACCOL task is scheduled to begin execution. If a particular task has a task rate of 1 second, for example, then it will begin executing every second. If a task cannot complete execution prior to its next scheduled execution, its execution will be delayed until the previous execution has completed; this situation is called ‘slippage’, and indicates that the task rate has been set too fast.

Timeout generically refers to some failure of a program or device to perform some function prior to the expiration of a preset time period. Timeouts are used in configuring both the data link, and the applications which use it.

Top Level Node(s) (used in BSAP networks ONLY): a Network 3000 controller which is on the network level immediately below the operator workstation running Open BSI. Also known as a ‘first level node’ or ‘first level slave’.

READER RESPONSE FORM

Please help us make our documentation more useful to you! If you have a complaint, a suggestion, or a correction regarding this manual, please tell us by mailing this page with your comments. It's the only way we know we're doing our job by giving you correct, complete, and useful documentation.

DOCUMENT NUMBER: D5080
TITLE: *Network 3000 Communications Configuration Guide*
ISSUE DATE: April, 2001

COMMENT/COMPLAINT:

Mail or FAX this page to:
Bristol Babcock Inc.
1100 Buckingham Street
Watertown, CT 06795
FAX#: (860) 945-2213
Attn: Technical Publications Group, Dept. 610

Bristol Babcock
an FKI company

1100 Buckingham Street
Watertown, CT 06795
Telephone: (860) 945-2200