Section 2 System Description

2.1 How VersaNet2 Operates

2.1.1 Nodes & Networks

VersaNet2 modules are connected together to form nodes. Each node is configured to handle its Input and Outputs (I/O) and communicate with other nodes. The way in which different nodes communicate depends upon the functionality required and the available radio paths.

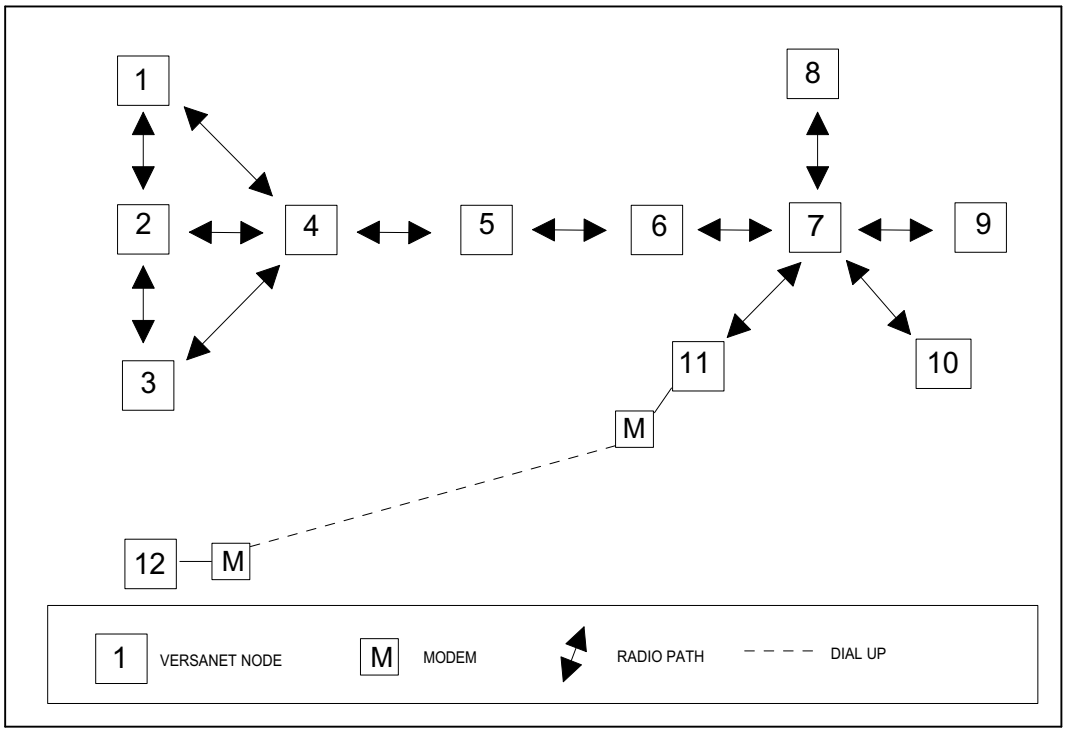

Figure 1 Example VersaNet2 System

The diagram shows an example VersaNet2 network. All nodes have a Communication Controller containing the radio and node intelligence, as well as storing the node configuration in non-volatile memory. Every VersaNet2 Node has the same capability so they communicate using a 'peer-to-peer' architecture rather than 'master-slave'.

The main function of Nodes is to send and receive I/O. They operate on a single radio channel and may be used to form a simple point-to-point system or a network of nodes with each Node being capable of communicating with any other Node within radio range. Any Node can function as a ' repeater' ór relay' which extends the effective system range and/or permits the circumnavigation of obstacles.

In the above example, Node 12, which is outside of the normal radio coverage, is fitted with a GSM modem for a dialup connection. This Node can send information to any other Node in the system via Node 5, which also has a GSM modem. In addition, any Node fitted with a GSM modem can send SMS messages.

A Node can be fitted with either a UHF radio, a GSM modem, or both, as in the case

of Node 11 in the above diagram. Note that a Wire Line modem can also be used in place of the GSM. See Section 8 for details of the GSM and Wire Line Modems supported.

All Nodes within the same system must have a unique Node Number and the same Network Name.

2.1.2 Communications

A Network is formed by a number of Nodes, with the simplest being a two Node chain.

Nodes operate entirely on a single configured RF channel. When no data is being exchanged, the receiver is switched to the RF channel and waits for an instruction. For example, in a three node network, if Node 1 needs to send data to Node 2, it first listens to the RF channel to check it is not in use (Listen before Transmit) and then sends a message to Node 2. Node 2 acts on the message and sends an acknowledgement back to Node 1.

If Node 3 also wanted to send data to Node 2 at the same time it would have encountered activity on the RF channel and thus would have waited until it was clear. Care should, therefore, be taken to ensure your system can handle potential delays that may occur when channel activity is high.

Every transmission from one Node to another is acknowledged. If a transmission is unsuccessful then the Node will re-try up to a maximum number of 9 times. The number of re-tries is configurable between 0 and 9. If after the programmed number of retries, communication is still unsuccessful and no acknowledgement is received, a 'comms-fail' alarm is activated. It is strongly recommended that this alarm is programmed to activate only after at least 2 TX intervals i.e. 2 attempts to send the data. This is because there are a number of reasons why occasionally radio communications may fail, interference, weather conditions etc. Setting the alarm delay to twice the transmission interval reduces false alarms. See section 4.4, programming.

Data to be sent from one Node, (e.g. Node 1), to another Node, (e.g. Node 6), is programmed in a connection list. A 'Primary Route' is also programmed. This gives information such as send data via Node 2, where Node 2 acts as a repeater. It is also possible to programme a 'Secondary Route'. If the primary route fails or is unavailable for any reason the system automatically defaults to the secondary route. This route may go via Node 3, where Node 3 is the repeater. The secondary route may also be a wire-line or GSM Modem.

2.2 Data Handling

The previous pages explained how VersaNet2 operates as a communications system. This section explains what types of data VersaNet2 can handle and how it uses this communications system to send securely coded data over sometimes complex paths.

The explanation breaks down into the following components:

2.2.1 Input and Output Types

Data gets into and out of a VersaNet2 Node via I/O modules, or the I/O on the Communications Controller module. These I/O modules are connected to the Communication Controller via T2-BUS interconnections to form a complete Node. The modules listed in the table below are the "prime" I/O modules and up to sixteen

of each may be used in a single Node. (NOTE: To a maximum of 192 inputs and outputs per Node).

Module Name	No of Inputs/Outputs
IRDN301 Digital Output Module	8 Outputs per Module
IRDN302 Digital/Pulse Input Module	8 Inputs per Module
IRDN307 Analogue Input	4 Inputs per Module
IRDN308 Analogue Output	4 Outputs per Module
IRDN314 Alarm Output	8 Alarm Outputs per Module
IRDN207 Analogue Input Module	8 Inputs per Module
IRDN209 Pulse Output Module	8 Outputs per Module
IRDN212 Low Power Pulse Input Module	8 Inputs per Module

Table 1 – Prime I/O Modules

Note that each channel on the IRDN302 module is software configurable to accept either Pulse Counting Inputs or Digital Inputs (volt-free contacts).

The next table lists the two modules having a mixture of I/O types.

Module Name/Code	I/O Configuration
IRDN310 Combination Output Module	4 Digital Input + 2 Analogue Output
IRDN311 Combination Output	4 Digital Input + 2 Analogue Input

Table 2 – Mixed I/O Modules

Only one of each of the modules in the above table can be used in a Node since they each have a fixed address. See Section 2.2.2b below.

The final module capable of handling I/O directly is the Communication Controller. It has the following data ports:

Communications Controller Data Ports
1 Digital Input, 1 Digital Output 1 Analogue Input, 1 Analogue Output 1 Pulse Counting Input, 1 Pulse Output 1 Alarm Output 1 Serial Data Highway 1 Configuration/Monitoring Port

Table 3 – I/O Capability of Controller Module

2.2.2 Addressing

All data inputs and outputs are allocated a unique address enabling data to be transferred anywhere within a system. Addressing breaks down into two areas:

a. Node Address

Every Node has a unique address, constructed as follows:

[NETWORK NAME] [NODE NUMBER]

For example, Node 3 in network ABC has a Node address of ABC3.

b. Data Address

Each data point is allocated a data address, constructed as follows:

[I/O TYPE] [MODULE NUMBER] [CHANNEL NUMBER]

[I/O TYPE]	A Analogue D Digital P Pulse
[MODULE NUMBER]	0 Communications Controller 1-16 "Prime" I/O Modules 17-29 Virtual Memory 30 Combination Output 31 Combination Input 32 Low Power Input 33-256 Virtual Memory
[CHANNEL NUMBER]	1-N Where 'N' is the total number of that I/O type on a module

Table 4 – Data Address Definition

For example, an Analogue Input Module on a card that has been set to Card Address 1, will have the following valid data addresses, corresponding to the eight input channels:

A1.1, A1.2,……. to A1.8

If a second Analogue Input Module is added to the same Node and set to Card Address 2, the additional addresses available are:

A2.1 to A2.8

A Combination module always takes a unique Module Number of 30 (Output), 31 (Input) and 32 (Low Power Input). The Communication Controller is always Module Number 0.

Note that it is possible to have an input and an output with the same data address and Node address. For example, a Node fitted with analogue inputs and outputs will have the following valid data address:

Inputs A1.1 to A1.8
Outputs A1.1 to A1.4

VersaNet2 will interpret the addresses correctly based on their position in the message packet, as described in the next section.

c. Virtual Memory (Virtual Address)

From table 4 it can be seen that addresses (channel numbers) from 0 to 16 are normally used for hardware I/O modules. These addresses are set by DIL Switches on the modules. It is possible however to use addresses 17 - 29 and 33 - 256 as Virtual Memory locations (or Virtual Outputs). Note that if no hardware is fitted, addresses 1 – 16 may also be used as virtual outputs.

When a Node receives data for a Virtual Output, there is no hardware (output module) associated with this address. Instead, the Node stores the data in memory, which can be accessed by the SCADA. This data can also be accessed through VNMGR using the Monitor facility

2.2.3 Message Address Construction

The Node Number and Data Addresses are used to configure VersaNet2 by constructing messages defining what data is sent to where. Only Nodes with the same Network Name will communicate with each other. The message packet construction is as follows:

SOURCE	DESTINATION	ROUTE
Node Number: Data Address	Node Number: Data Address	Node Number

In normal use, this message is entered in parts by responding to questions during configuration. For example, Data Input A1.3 on Node number 1 is to be sent via Node 2 to Node 4 and output on Data Output A1.4

The SOURCE and DESTINATION parts of the message are entered into the ' Connections' screen within the VersaNet2 Node Manager Software (VNMGR). The ROUTE is entered into the ' Routing' screen.

SOURCE	DESTINATION	ROUTE
A1.3	4A1.4	2

NOTE: It is not necessary to enter the Node number (as part of the address) in the source, since this is the Node being programmed.

Data inputs may be sent to more than one output by simply entering additional connections and routes. Similarly, more than one input may be sent to a single output, but care should be taken to avoid unexpected results. Pulses, for example, must only have a single input sent to a final destination, otherwise the pulse count will be incorrect.

2.2.4. Over the Air Protocol.

VersaNet2 uses a proprietary ‘over-air’ protocol to ensure secure sending and receiving of data messages. A brief description of the message structure is as follows:

A message is transmitted as a series of fixed length packets, each 23 bytes long. A message will contain a string of these packets in the sequence, preamble, header, data. The number of data packets depends on the amount of data in the message. Each packet takes approximately 75mS to transmit.

Every packet in a message contains synchronization bits, the number of packets in the message, network name, originating Node and destination Node. Each packet ends with a checksum, which allows for error checking of each packet at the receiving Node. The Header packet contains additional information including

message number and routing details. The data packet has 7 bytes reserved for the message content. Longer messages are split over a number of packets.

If the message is received and decoded correctly with no errors, the receiving Node sends back an acknowledgement of a single packet.

A full description of the over-air protocol is available from RDT if required.

2.3 Software

There are 3 software modules associated with VersaNet2:

Node Software	filename: NODE 2-XX.A20	Approx size: 325Kb
Flash Download	file name: VNFUD 2-XX.EXE	Approx size: 672Kb
VersaNet Manager	file name: VNMGR 2-XX.EXE	Approx size: 760Kb

These are all supplied on CD and must be loaded into your own PC. See below for details of loading software. (NOTE: Floppy disk versions are available on request) The PC should be running Windows 95 or later (including NT and XP)

Software version control is by the two digits shown in the above file names as XX. For example NODE 2.01, NODE 2.02 etc. To check the version of software running on a Node use VersaNet Manager 'Test' facility. See Section 4.4.12 The version of VersaNet Manager in use is displayed in the title bar at the top of the open window.

NOTE: There is also V3.XX software available. This is exactly the same as V2.XX but with one added feature for Low Power GSM. If using V3.XX node software, note that V3.XX VersaNet Manager will be required. It is not compatible with V2.XX VersaNet Manager software.

Node Software

This programme is written in 'C' and then compiled down to machine code. The programme is loaded into the flash memory of the on-board microprocessor for each Node.

NOTE: VersaNet2 Nodes are shipped from the factory with the Node software pre-loaded. Normally therefore, it should not be necessary for customers to load Node software. If you need to load software, to update to a new version, follow the instructions later in this section.

Flash Download

This programme, written in Delphi, is required to download the Node software into the flash memory of the processor. If you need to update your Node software, follow the instructions later in this section.

VersaNet Manager.

The VersaNet Manager programme is the main customer interface to a VersaNet2 Node. It allows programming of all customer selectable parameters, data I/O connections, routing etc. and has powerful monitoring and test facilities. All the features of VersaNet Manager are described fully in section 4.

The CD Contents

As well as the three major software modules above, the CD has a complete copy of the VersaNet2 Manual and other useful information. See the leaflet supplied with the CD for more information.

Running the CD

Place the CD in the drive. Using Windows Explorer, copy or drag the files to a convenient folder. (It is advisable to first create a VersaNet2 folder and keep all information together in this folder). If required, create a shortcut for VNMGR and send to the desktop.

Downloading Node Software

Connect the Node, configuration port JP7, to your PC with an RS232 cable (see section 9.3) and power up the Node. Wait for the Node to initialize (Red power LED 'on', Orange RX LED 'on' and Green run LED 'on' and steady).

Open VNFUD.
Check that the correct COM port is selected in the box at the top of the screen.
Using the Browse facility, select the version of NODE software to be downloaded. Click on it in the VersaNet folder and 'open'. It will then appear in the Object File field.

Check that 'LINK 1' on the controller card is set to 'RUN' position and Click 'Start Transfer'.

The first status bar will indicate the progress of the Flash Upload. Once the initial upload is completed, you will be prompted to move 'LINK 1' to the 'PGM' (programme) position. The software will then continue with the downloading. The status bars indicate the progress.

When the download is completed, you will be prompted to move the 'LINK 1' back to the 'RUN' position. Wait a short time and you will be prompted that the download was successful and the new version number will be displayed.

The whole process takes a few minutes.

If you encounter any problem downloading, try aborting the download and checking the 'Use Low Speed' box on the initial screen. This will run the whole process slower, which may be required for some older machines, especially notebooks.