Flexible Peer-to-Peer Networking for Recloser Controls and Relays

by Jack McCall, Director, Protective Relays

Most protection applications in a distribution system rely upon time-graded coordination between protective devices where downstream devices operate more quickly than upstream devices. Even automated recloser loop schemes depend upon the coordination of timing delays as an integral part of their control logic. Direct communication between two protective devices to enhance protection, security or control has been used for many years in transmission relaying. There, communication-aided-tripping schemes provide for more rapid tripping of faulted transmission lines by sharing a couple of status points, traditionally via relay contacts.

Enhancing the peer-to-peer nature of communications between relays has taken a few steps forward, most notably via MIRRORED BITS™ found on some relays from Schweitzer Engineering Laboratories, where up to eight "virtual" contact pairs may be passed between a relay pair. A more comprehensive status sharing system is found in the so-called GOOSE messaging structure defined in the UCA 2.0 protocol specification.

However both of these systems have some restrictions.

- Both are limited to sharing binary status-type of data.
- MIRRORED BITS is limited to two devices, though communication between more devices is possible with added hardware, expense and programming complexity. The added hardware required for more than two devices also leads to the possibility of a common point of failure bringing down the entire system.
- GOOSE messaging can allow for essentially unlimited numbers of devices to communicate, but its dependence upon Ethernet connectivity makes it impractical for easy installation on overhead lines outside of the substation. Furthermore, if a LAN does not already exist in the substation, additional costs must be incurred.

PeerComm™ Protocol Flexibility

PeerComm communications protocol was developed to address the real world need of having small networks of protective devices that span both sides of the substation fence sharing data to make protective decisions. PeerComm is an exceptionally flexible, compact peer-to-peer communication network.

	Device 1	Device 2	Device 3	Device N
Binary 1	1	0	0	1
Binary 2	0	0	0	0
Binary 3	1	1	1	1
Floating Point 6	12.456	16.222	0.000	14.441

Figure 1 Sample memory array

PeerComm protocol may be implemented in a variety of configurations. The user can choose the best one for the situation. The initial deployment of PeerComm protocol permits up to five devices[1] on the network to share one of three sets of data:

- 15 binary (status) points
- 15 binary points, plus 3 short integers
- 15 binary points, plus 3 short integers, plus 6 floating points numbers.

1 PeerComm communications protocol will eventually be expandable to 32 devices.

Short integers permit sharing of information such as counters, percentages, and so forth. In the event that more than 15 status points per device need to be shared, software tools are provided to convert a short integer into an additional 16 status points. In this fashion, up to 63 binary status points may be shared.

The floating point values permit the communication of voltage or current magnitudes, angles, or other high-resolution data. Each device on the network shares its own data with every other device on the network. Since the less data that is being transmitted results in faster communications, the several forms of PeerComm network described above (and future forms) allow a PeerComm network to be optimized for any application, ensuring that performance is maximized, and communications bandwidth is not wasted.

Each device running PeerComm protocol allocates an array in memory that is populated with the data from all of the devices on the network. See Figure 1. The devices in a PeerComm network take turns sharing their data by broadcasting it on the network. All of the non-actively broadcasting devices are in listening mode during this broadcast and update their images

of the shared data accordingly. Once Device 1's data is broadcast, Device 2 broadcasts its data, and so forth. See Figure 2.

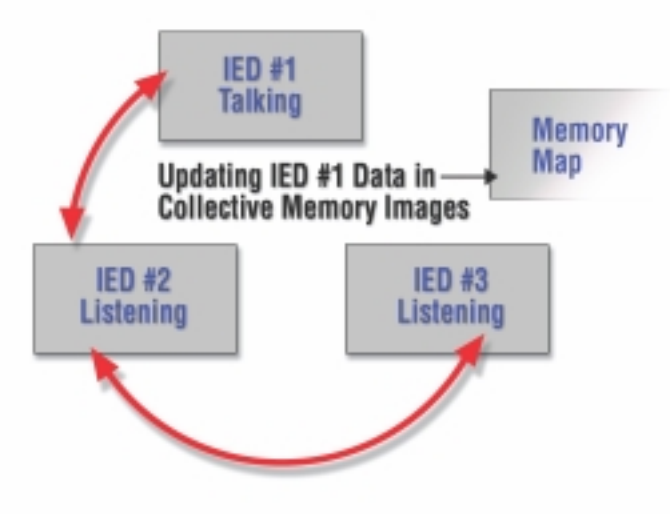

Figure 2 Data sharing for PeerComm network

PeerComm communications protocol also takes care of itself in case one or more relays stops communicating. It has a timeout feature, which allows the rest of the network to recognize a device has stopped communicating and to ignore its data, leaving the rest of the network intact. The user can easily implement logic to alter the behavior of the protection system in case of such an event.

The physical connection between the devices on the network requires no hardware beyond the RS-232 or RS-485 interfaces provided standard on the Form 6™ recloser control, Idea™ relay, and IdeaPLUS™ relay protection platforms. Alternative physical media are optional using external modems or other adapters. Typical physical configurations include:

- Use of a standard RS-485 port running up to 115kbps. This requires only a single twisted pair running in a drop serial fashion between devices.
- Use of standard RS-232 ports. Note that use of serial port to fiber converters on either the RS-232 or RS-485 ports allow for longer distance, noise free communications.
- Use of radios connected to RS232 or RS485 ports.

The transmission speed of the system will vary depending upon the selected baud rate, the selected transmission media, and the amount of data being transmitted. The fastest operation will be for a hardwired, two device network sharing the minimal 15 binary status point data set. In this configuration, one device can change a status value, share it with the other device, and have the other device act on the information in under 1/2 cycle. Five device networks sharing the maximum data set over radio may take between 100 - 500 msec (depending on the radio type) to share all data.

Once the common memory array has been populated, the values of any of the array members may be accessed in the relay or control and used to implement custom protection and control schemes.

PeerComm Protocol

PeerComm communications protocol is distributed as a software module for use in IEDs that have the Idea Workbench™ software programming feature. Presently, this includes Cooper's Form 6 recloser control and the line of Idea and IdeaPLUS protective relays.

The Idea Workbench software feature provides the ability to load new features or functions into the relay or recloser control in a fashion similar to the plug-ins for popular Internet Web browsers. Once a few files are installed on the user's PC, the full range of PeerComm protocol programming tools is available in the Workbench environment.

To set up a PeerComm network, the user selects the appropriate network and data model block from the Add-Ins button on the Advanced Tools palette in the Idea Workbench environment. See Figure 3.

For this example, the 2 device, 15 binary network will be used. Therefore, the 2x(15b) block will be chosen. Idea Workbench software uses a graphical drag-and-drop programming technique, which is much more intuitive than command or equation based programming methods. Here, the user will simply click on the 2x(15b) line, and drag that network object into the Idea Workbench environment. Figure 4 shows the end result of this operation.

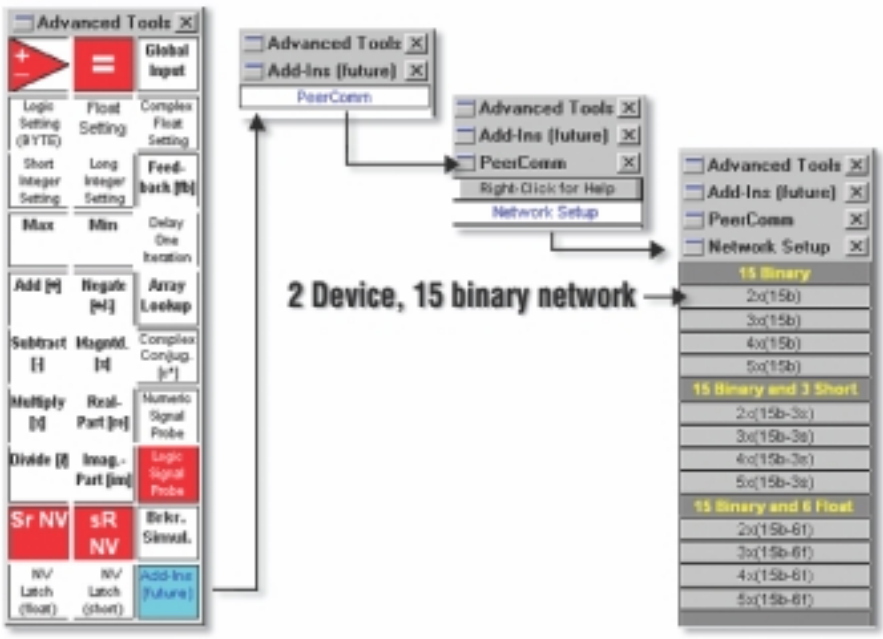

Figure 3 Defining the PeerComm network set-up in the Idea Workbench environment

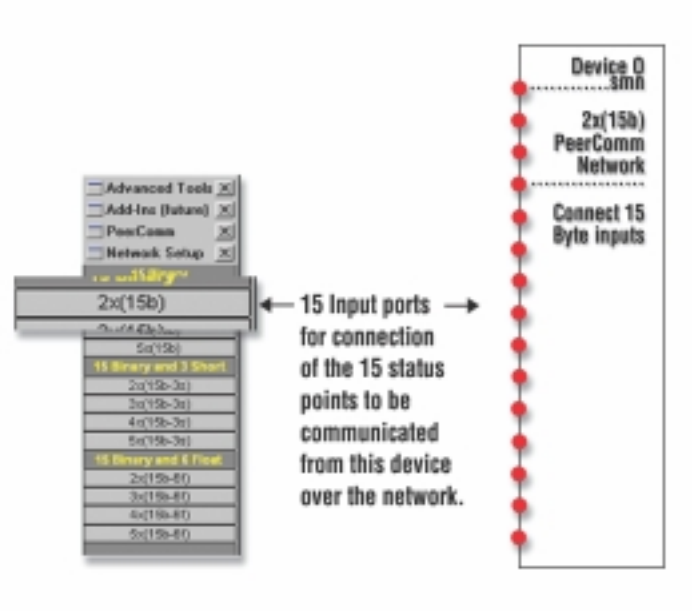

Figure 4 The 2x(15b) PeerComm network module placed in the Idea Workbench

This particular PeerComm network object has 15 input ports on its left hand edge. The user connects to each of these input ports one binary status signal such as open/close status, reclosing mode, protective element pick-up or trip status, etc. The user configures the other devices on the network in the same fashion. PeerComm protocol allows each input port to be given an ASCII name, such as "Breaker101:52a," so that when developing custom logic, the programmer can work with meaningful signal names. This makes the programming and debugging process proceed much more rapidly.

Because of the extensive programmability of Idea Workbench software, it is easy to implement backup logic that can direct the operation of the devices in the event the network goes down.

Another advantage of the system is that, should an additional device be desired to be added to the network, or should more data be desired to be exchanged over the network, only the network block as shown in Figure 4 need be replaced. All remaining programmed input signals and logic remains intact. Only new logic desired resulting from the addition of the new device need be constructed.

Uses for PeerComm Protocol

PeerComm communication protocol can be used to implement whatever type of distributed control and protection system the application may require. Possibilities include:

- High-speed overhead loop scheme reconfiguration.
- Simple implementation of zone interlocking schemes.
- Implementation of automatic backup schemes where one backup device can selectively trip one or more breakers.
- Classical POTT, PUTT or other communication-aided tripping schemes without using contacts.
- Pseudo bus differential schemes.
- Automatic load shedding or bus reconfiguration to match transformer loading.

Conclusion

PeerComm communication protocol is a simple to use, flexible and powerful peer-to-peer communication protocol for protective relays and recloser controls, allowing them to work together over radio, fiber or copper wire as a team. The ability to configure the network in the Idea Workbench programming environment makes for rapid implementation.

THE LINE