This document is under formal review of the NMRANET Working Group. It has not been approved by the WG, reviewed by the Manager, S&C Dept., nor approved by the NMRA Board of Directors.

NMRA TECHNICAL NOTES	
S-9.6/OpenLCB NMRAnet Proposal	
All Scales	
06/06/10	S-9-6

S9.6/OpenLCB Goals and Measures

Interoperability

Goal: Products from one company can work with products from any other company, and with legacy systems.

Measure: Manufacturers can easily test their nodes for interoperability using standard tests. Any device passing these tests should be able to work with other devices passing these tests. Likewise, it should be easy for manufacturers to build bridges to systems like DCC or Loconet which allows NMRAnet turnouts to be controlled from DCC throttles, and DCC engines/accessories to be controlled from NMRAnet.

The S9.6 protocols are being documented at a high level, such that any proper implementation will interoperate with any other. At the same time, proper functioning of low-level detail is being insured via test and example implementations.

Care is being taken to define a protocol that can be used by large and small manufacturers without giving anybody an undue advantage. For example, the test and example implementations are available for manufacturers to study and, subject to licensing, use as the basis for their own products.

(Non-normative) examples of connections to DCC, LocoNet?, and eventually other systems, such as C\MRI, Xpressnet, S88, and NCE Cab Bus, are being created to demonstrate interoperability.

Features

1. S9.6 nodes have world-wide unique node IDs, and therefore there is no risk of clashes between products from different companies.
2. The S9.6 generous 48-bit node ID allows legacy IDs to be encapsulated when bridged onto S9.6.

Issues

S9.6 has not yet specified any specific conformance testing criteria or methodology. We are developing an extensive series of tests for development purposes, which may in the future form the basis of conformance tests, but the process for doing that has not been specified by the NMRA, as yet.

No Central Control

Goal: Products can interact with other products without the need for a central processor.

Measure: Once a layout is set up, it should run perfectly and completely without a central processing system for basic and advanced functionality.

The S9.6 protocols provide both direct node-to-node communications (datagrams, streams), and broadcast notifications (events). No central facility is required for any of these.

Features

1. S9.6 does not mandate the existence of either a Network Manager nor Configuration Manager, although these can be used if desired.
2. On CAN, S9.6 node ID aliases are negotiated between the nodes present, and a NM is not required for this.

Issues

No entry

Optional PC Control

Goal: A PC can provide higher-level functionality, such as a CTC Interlocking, and system functions such as monitoring, testing, and debugging.

Measure: A computer attached to the bus can provide more advanced functionality, such as CTC, and system-control functions such as: monitoring, testing and debugging

Features

- S9.6 allows multiple connections for multiple computers.
- S9.6 allows multiple Network Managers, Configuration, and other programs to be present and operational at the same time, which will be important for large layouts, such as museums and large modular meets. These can come and go as needed to do their job; they do not need to be present at startup or shutdown.
- S9.6 has multiple ways to connect a computer either directly or via various bridges:
    - Ethernet directly
    - Wirelessly, for example by Wifi via commodity Wifi-Ethernet routers
    - USB to CAN bridge

Issues

S9.6 is in the process of specifying testing and monitoring protocols. The data transfer protocols (datagrams, streaming, event notification) are in place. A protocol for remote logging is under discussion, but there is not yet a reference implementation. Only preliminary work has been done on protocols for remote testing and diagnosis.

Simple

Goal: Make it easy for a novice customers to install and configure without technical knowledge (should be easier than DCC systems).

Measure: A customer can change the configuration of any device at any time with a simple interface that includes text descriptions of settings, thus greatly reducing the need for a manual and required level of learned expertise. This mechanism should also reduce the support burden to the manufacturer. In addition:

** You can disconnect a node, reconnect it somewhere else on the same net and have is still work without having to make any configuration changes.*

Features

Configuration methods:

- S9.6 defines a push-button configuration method that will allow simple configuration without the need for attached PC or configuration device*. (This is defined so that it will inter-operate, but manufacturers may omit it from their products if they wish) In this simplest form, a few button pushes on two boards will associate them so an input on one triggers the output on another, e.g. a pushbutton operates a distance turnout.
- For more complex configuration, S9.6 nodes all include detailed self-descriptions in a defined format. This permits the construction of stand-alone configuration tools that will just connect and work, no separate PC needed. We anticipate that limited tools of this kind will be available for about the current cost of a DCC throttle.
- Programs in personal computers are full S9.6 NMRAnet partners, and will be able to provide configuration assistance limited only by the ingenuity and motivation of their programmers.

Simplicity:

- Configuring events in S9.6 is simplified by the world-unique event ID numbers. Configuration tools don't have to keep records of event IDs used, avoid conflicts, or generally have to worry about them at all.
- Since each node has a unique node ID, they can be easily be cross-referenced on the Internet so that manufacturers can supply support, including configuration tools and manuals.
- Because the S9.6 48-bit node IDs are world-wide unique, free relocation of nodes without risk of node collisions is guaranteed. Nodes can be moved around, and

working modules interconnected, with no reprogramming required to retain their existing function.

Issues

- S9.6 48-bit node IDs and 64-bit event numbers incur a moderate penalty in terms of space and search time, however, this is mitigated by:
    - This space/time penalty is far outweighed by the benefit in terms of ease of use, safety, and reduced software costs.
    - The full node ID only needs to be saved by its owner and bridges, and not other nodes.
    - On CAN, a shortened 12-bit alias, the NIDa, is used for the full node ID.
    - The price of memory is falling, and memory other than EEPROM can be used to store events, such as flash memory.
    - Search algorithms can be chosen that allow event matching with only a minor speed penalty over shorter event numbers.

Footnote

* We are only beginning to prototype the mechanism for push-button configuration of nodes (e.g. without an attached PC). We're sure it will work; we're not so sure whether model railroaders will like it. We've got a PC-based simulation you're welcome to try. It simulates several nodes, their pushbuttons & LEDs, and their input/output connections on the screen so you can play with the push-button protocol. If you want a copy, ask on the development mailing list or build it from SVN.

Expandable

Goal: Allow very large layouts (a higher-level of technical knowledge required is OK for large layouts).

Measure: A very large museum layout might be divided into distinct segments that work together as a single network. You can use a device, such as a hub, repeater, or switch, to connect multiple segments to form a single network.

Features

- S9.6 is designed to not only handle large layouts, but to minimize the problems inherent in a large heterogeneous layout.
- S9.6 supports Modular Layouts and large Modular Meets, by minimizing the integration tasks required when joining multiple independent layouts and net segments into a cohesive whole. To this end, it has these features:
    - 48-bit node IDs ensure there are no node conflicts.
    - 64-bit event numbers with allocation criteria that eliminates event number conflicts and re-programming tasks other than those required to integrate adjacent sub-layouts by configuring their mutual edge events.
    - Multiple integrated transports including, but not limited to Ethernet and CAN. This allows local area CAN nets that can be linked by a higher-speed backbone bus.
    - Intelligent bridges with automatic routing and traffic-limiting - messages are transferred to the network segments containing nodes that need them, and they don't appear where not needed, all automatically.
    - The ability to have multiple computers running configurations tools simultaneously.
    - The ability to define virtual nodes and run testing across the network.
- S9.6 supports museum and other large layouts by having these features:
    - Use of off-the-shelf components, such as CAN-USB bridges, Ethernet routers and wireless routers.
    - Industrial strength noise resistant local network hardware, including Ethernet and CAN.
    - Allows use of multiple hard-, firm-, and software solutions for the control and display of the net. Multiple glass-panels, configuration tools, and testing tools can be used simultaneously.

Flexible

Goal: Customers can easily connect many-to-many devices, and connect to legacy systems.

Measure: You can buy devices from any manufacturer that were designed without knowledge of the other devices and make them work together.

Features

- Many-to-many:
  - S9.6 uses a clean Producer/Consumer (P/C) Event Model which allows for many-to-many interactions between devices.
- No knowledge of other devices:
  - S9.6 uses large node and event IDs which are world-wide unique*. This means that there will be no conflicts between devices or their events.
  - The S9.6 P/C model lets devices from two different manufacturer work seamlessly together, without either device having knowledge of how the other works internally.
- Legacy Systems:
  S9.6 is able to connect to legacy systems through the use of intelligent gateways, which make the legacy system look like a fully functional set of S9.6 devices. For example:
  - MERG RPC is a serial bus that transfers raw data to and from a 'stack' of input/output modules. An intelligent gateway stores the RPC stack's state, and reports state changes as S9.6 events and responds to S9.6 events by transferring the changed state to the RPC stack.
  - DCC commands are directly translated into S9.6 events by encapsulating the DCC command and DCC-address into S9.6 64-bit event numbers. Similarly, S9.6 events are translated to their corresponding DCC packet by extracting the command and DCC-address information from the S9.6 event number.

Issues

- S9.6 connections to legacy systems are in the demonstration-phase.

* No formal mechanism has been established for the distribution of S9.6 IDs. However, IDs have already been assigned via NMRA manufacturer numbers, NMRA membership, and MERG membership numbers. We would expect that the controlling body would set-

up some mechanism, possibly by an automated Internet form, that will distribute IDs to other manufacturers, organizations, groups, and individuals on a as-needed basis.

Extensible

Goal: Allow additional functionality to be added easily, whether by the NMRA, or manufacturers.

Measure: The NMRA can add Standards and/or RPs on top of the bus to define new capabilities. Additionally, manufacturers can add their own proprietary functionally between their own devices.

Features

- S9.6 provides three basic communications mechanisms (event notification; datagrams; stream transfer) that provide a strong basis for higher-level functionality.
- The use of a pure Producer/Consumer Model for global notification in S9.6 provides excellent opportunity to add functionality while maintaining compatibility.

Issues

The S9.6 development team have had only preliminary discussions about adding proprietary protocols, however, they see no technical impediment to this goal.

S9.6 defines how the Node and Event universal identifiers are allocated by a combination of delegation and central (NMRA) record-keeping. A similar mechanism for future protocol identifiers is yet to be defined.

Extensible

Goal: Allow additional functionality to be added easily, whether by the NMRA, or manufacturers.

Measure: The NMRA can add Standards and/or RPs on top of the bus to define new capabilities. Additionally, manufacturers can add their own proprietary functionally between their own devices.

Features

- S9.6 provides three basic communications mechanisms (event notification; datagrams; stream transfer) that provide a strong basis for higher-level functionality.
- The use of a pure Producer/Consumer Model for global notification in S9.6 provides excellent opportunity to add functionality while maintaining compatibility.

Issues

The S9.6 development team have had only preliminary discussions about adding proprietary protocols, however, they see no technical impediment to this goal.

S9.6 defines how the Node and Event universal identifiers are allocated by a combination of delegation and central (NMRA) record-keeping. A similar mechanism for future protocol identifiers is yet to be defined.

Bi-directional Support

Goal: Supports bi-directional exchange of information.

Measure: All products should be able transmit and receive information to/from the bus. Feedback of information such as Block Occupancy Detection should be fundamental to the bus. As such, success here would be that a more sophisticated BOD could report not only occupancy, but optional additional information, such as the locomotive or rolling stock unique number, as well as yet-to-be determined information.

Features

- S9.6's clean Consumer/Producer Event Model allows painless transfer of stateless data.
- S9.6's Datagrams and Streams will allow ample room for transfer of state information.

Issues

- The S9.6 has only had preliminary discussions regarding stateful data, and in particular regarding location state, such as will be required with RFID, Bi-Di, bar-coding, IR and other train/block detection methods. However, we do not see any technical impediment to implementation.

Free IP

Goal: The Standards and/or RPs should be free from intellectual-property restrictions from parties other than the NMRA.

Features

NMRAnet S9.6 uses standard communications links (CAN, Ethernet, USB, etc) without modification. Manufacturers who use commercial implementations of those, and comply with the conditions of their suppliers, will have no IP issues. Manufacturers who want to create their own "almost-Ethernet", "almost-CAN", etc might possibly encounter IP issues, but their solutions won't be NMRAnet S9.6 compliant in any case.

The NMRAnet S9.6 documents, prototype code and examples, are all being developed and made available under Open Source licenses. Once an agreement on NMRA standardization has been reached, they will be freely and generally available to all. Depending on their business model, manufacturers will have to write their own manuals and software to implement NMRAnet S9.6 or comply with the license terms of other developers who provide solutions.

Issues

The NMRA's own licensing terms are inconsistent with this point.

Transport Agnostic

Goal: Allow different transport mechanisms for messages.

Measure: If a company has another method for connecting devices, such as wireless or another wired method, they can substitute their own transport mechanism and still use all the other standards/RPs and automated test suites.

Features

- S9.6 is designed to be transport agnostic, and is being prototyped specifically on TCP/IP (Ethernet, wireless) and CAN networks. It can be implemented on transports such as multi-drop RS-232, RS485, serial phone lines, cell-phones and others not yet designed by using the S9.6 concept of a "wire protocol" to map specific NMRAnet S9.6 messages onto a specific transport mechanism.
- Use of a CAN bus imposes specific constraints and features, however, S9.6 mitigates these in the CAN-specific wire protocol:
    - CAN frame size is limited
        - S9.6 uses 29-bit headers in order to maximize CAN frame length to 13-bytes.
        - S9.6 48-bit node IDs are converted to shorter local12-bit aliases.
        - S9.6 Datagrams allow transfer of data greater than 8-bytes.
        - S9.6 Streams allow large quantities of data to be transferred.
    - CAN frame-headers are required to be unique
        - S9.6 node ID aliases are present in every frame header and guarantees it is unique, collisions cannot occur.

As a consequence CAN nodes are full participants in the larger NMRAnet network. Similar adaptions should allow NMRAnet to work over any reliable transport mechanism.

Train Agnostic

Goal: Support controlling trains when the network is connected to a train-control system.

Measure: The NMRANet should be able to connect, via bridge products, to any train control system, such as DCC, DC, or PWM, to allow interaction with each other. A hand-held throttle on NMRANet can control a locomotive on DCC, DC, or PWM. Similarly, you can have a bridge between a DCC command station and the NMRANet that will allow the DCC hand controller, for example, to throw a turnout motor attached to NMRANet.

Features

- The S9.6 group is fully committed to meeting this goal. Prototyping has started on LocoNet and DCC connections.

Issues

- Only very preliminary steps have been taken to achieve what is hoped is a solution that is simple, elegant, unifying, and comprehensive.
- We note that this criteria is inconsistent with the NMRA charge to the NMRAnet Working Group. This must be resolved at some higher organizational level.

Discoverable

Goal: Allow a user to find out what devices are connected and how they're configured on a layout.

Measure: You can connect a laptop to a layout it has never seen before. The configuration software can query the layout, provide a list of all the nodes attached to the layout, display how each node is configured, and show how nodes are interconnected and interact with each other. In addition, you can save the configuration of an entire layout and later restore it (in case you want to make some changes, and then decide to go back to what you had).

Features

- Connection and discovery:
    - Because each S9.6 device has a factory set unique ID, devices will never conflict with preexisting devices on the layout, and many can be added to a layout simultaneously without concern about duplicate addresses.
- Configuration and device teaching:
    - S9.6 has well defined protocol to retrieve P/C event use from nodes.
    - S9.6 nodes contain compressed XML containing a description of the device. This is sufficiently complete that a *de novo* node can be added to a layout and then queried as to its characteristics such that it can be programmed/configured.
    - Collateral information will be obtainable via files or online, including test suites, manuals, and configuration tools.
    - In addition, at the manufacturer's discretion, devices may be programmable using the S9.6 protocol for local programming, or programmed by a standalone programming device.
    - To get the full benefit of a device, it is likely a dedicated or generic software program, the equivalent of open-source JMRI project, will be preferable, as it will allow the use of user descriptive terms for all of the programming parameters.
- Saving and restoring device content and inter-device relationships:

  S9.6 will have a number of ways to back-up device content and inter-device relationships.
    - As described above, there is a well-described S9.6 protocol to retrieve device relationships.
    - S9.6 device content can be read and written using the Datagram protocol.

- In addition, device content is serializable and is downloaded, copied and stored using the S9.6 Streaming protocol.
- The stored content can then be restored to original device, or another device if the user wants to duplicate or replace a defective device. The restored or replacement device will be a full participant in the original device's relationships.

Issues

- Preliminary specification of the XML schema has been made, but further development is necessary.
- Preliminary adaptation of JMRI tools to S9.6 has been made, and requires roll-out once licensing issues have been resolved with the NMRA.

Self Describing

Goal: Devices should describe themselves.

Measure: You can connect a computer to a type of node that it has never seen before. The computer can display the manufacturer of the node, the number of devices inside the node, the configuration variables, actions, reaction, etc. All of this information can be used to generate a user-friendly interface on the fly that allows configuring a node with text prompts.

Features

- Because each S9.6 device has a factory set unique ID, devices will never conflict with preexisting devices on the layout, and many can be added to a layout simultaneously without concern.
- S9.6 has well defined protocol to retrieve P/C event use from nodes.
- S9.6 nodes contain compressed XML containing a description of the device. This is sufficiently complete that a *de novo* node can be added to a layout and then queried as to its characteristics such that it can be programmed to some basic level.
- Collateral information will be obtainable via files or online, including test suites, manuals, and configuration tools.
- In addition, at the manufacturer's discretion, devices may be programmable using the S9.6 protocol for local programming, or programmed by a standalone programming device.
- To get the full benefit of a device, it is likely a dedicated or generic software program, the equivalent of JRMI, will be preferable, as it will allow the use of user descriptive terms for all of the programming parameters.

Issues

- Preliminary specification of the XML schema has been made, but further development is necessary.
- Preliminary adaptation of JMRI tools to S9.6 has been made, and requires roll-out.

Testable

Goal: Allow components to be easily tested for compliance.

Measure: Components can be easily tested for compliance using available tools and standard test suite.

Features

- S9.6 documents a basic set of messages, which can be automatically tested by a test suite. It also documents three specific protocols (event exchange; datagrams; streams) that can be individually tested. In general, testability has been a criterion all along.
- Testing suggestions are being written into each draft Standard and RP. These form a starting point for a test suite. Having them right in the S&RP document ensures that product developers are aware of them.
- As part of developing prototype implementations, suitable tests are being developed in parallel.

Issues

- The development of a S9.6 test suite is only in preliminary stages. As of this writing on June 6, 2010, there are 5005 lines of testing code implemented and available in SVN.

Compliance

Goal: A name protected by a trademark that NMRA can use to help ensure compliance.

Measure: A manufacturer should be able to obtain a license to use the NMRANet name and logo when their product successfully passes a test suite. To comply with these standards a product should be capable of being connected to a real layout or a "white-box" simulation of a layout and pass all designed tests. These tests would examine the products ability to keep up with bandwidth requirements as well as make 100% compatible protocol exchange. Using standardized busses such as CAN, USB, Ethernet, ZigBee or 802.11 wireless would negate the need for a low level compliance test.

Features

- S9.6 uses industry standard connections. By not modifying these standards at all, commercial-off-the-shelf implementations require only limited testing at the electrical level.* The first version of S9.6 specifically includes:
    - Ethernet -- A ubiquitous high-speed high capacity network, which is available on almost all computers and many cellphones. Common usage network ancillary devices are available at low commodity prices including hubs, routers, and wireless routers.
    - CAN -- An industrial strength differential two-wire bus, designed by Bosch for the automobile market, commonly used in industry and academia for interconnection of equipment, including high electrical-noise areas such as factory floors and accelerators. Again, bridges, extenders, and relays are easily available.
- S9.6 can also be applied to other buses, such as I2C, multi-drop RS232 and RS485, and to wireless, such as 802.54.2 and WiFi?. By staying with standard communications links, additional capability can be added without requiring the development of communications test suites.

Above the level of the inter-node connections, NMRAnet S9.6 operates with the exchange of well-formatted messages and state-machine based protocols. Conformance with these can be explicitly tested. NMRAnet S9.6 documents clearly list which parts are mandatory (must be implemented and tested), which components are optional, and which parts of the documents are just non-normative discussion.

Issues

- Only preliminary work has been done on a specifying a test suite. It will likely take considerable work to make a robust but simple testing suite.

- There are issues associated with permission to use NMRAnet for development before product certification; timing of testing vs product announcement; fair-use and comparative use of trademarks; etc that are not S9.6-specific issues, and need to be resolved by higher level within the NMRA. Until those are resolved, it's not possible to say whether the NMRAnet S9.6 approach is compatible.

Footnote

* A separate decisions to be made about testing of non-standard implementations of the low level connections. Are they forbidden, just require extra testing, or are they permitted? As an example, consider a manufacturer who wanted to develop their own CAN controller, instead of using pre-existing silicon.

First Time Use

Goal: A new user can buy a device and have it work.

Measure: A user new to NRMAnet can buy a single device, take it home and have it do something useful and interesting without having to buy any other special hardware, or another device.

Features

- S9.6 has world-wide unique node IDs, which means that new users do not need to worry about conflict between nodes, in fact they do not need to concern themselves with IDs at all.
- S9.6 will have a comprehensive push-button programming model*, notionally called the "blue/gold algorithm" after the recommended color of the buttons, which allows new owners to connect two nodes out of the box and make them fully able to interact via event messages.

Issues

- We need to put some work into creating introductory articles, convention clinics, etc to help model railroaders become comfortable with NMRAnet S9.6. The excellent work done by the NMRA DCC WG to get DCC accepted is an example of what's needed.

Footnote

* We are only beginning to prototype the mechanism for push-button configuration of nodes (e.g. without an attached PC). We're sure it will work; we're not so sure whether model railroaders will like it. We've got a PC-based simulation you're welcome to try. It simulates several nodes, their pushbuttons & LEDs, and their input/output connections on the screen so you can play with the push-button protocol. If you want a copy, ask on the development mailing list or build it from SVN.

Table of Contents