

# Belcarra Technologies USB Networking

## White Paper

Stuart Lynne  
Richard Pitt  
February 20, 2003

## 1. Overview

The rise of both Ethernet (and other 802.3 framed technologies) and the Internet has paralleled the rise of USB as an interface to the microcomputer's I/O bus. In some respects, this has made little difference. In others, it has complicated the implementation of solutions to what on the surface may seem to be straightforward problems. A specific case in point is the typical PC, LAN/WAN and portable device scenario where at least one or more of the portable devices is connected via USB networking while the LAN/WAN connection might be either by hard-wired Ethernet or another USB device such as Ethernet dongle, cable modem or xDSL modem. The requirements for each of the various types of USB networking involved in such a scenario are the subject of this paper.

Embedded USB Devices will typically match one of the following types:

- PDA type devices (smart hand-held, cell phone, GPS, MP3 player, data collection device, test/analysis devices, telematics, etc.)
- Bridge type devices (cable modem, xDSL modem, etc.)
- Router type devices (consumer firewall, SOHO router/firewall, edge router, WLAN router, etc.)

A PDA-device will require a virtual network connection so that it can be networked to another system. Typically this will be a desktop or other workstation, and there will be applications on both the PDA and Desktop that need to interact. In some cases the host system may in turn act as a router or proxy to provide access to another network for the PDA-device.

A Bridge-device will implement an 802.1d bridge between another network and the system that the Bridge-device is connected to by USB. The other system, again typically a desktop or similar system, is using the Bridge-device to connect to some external network. For example the Bridge-device could be a cable or xDSL modem and the desktop system is mainly interested in networking to the outside world, not in interacting with the Bridge itself.

A Router-device will implement an Edge router between another network and the system the Router-device is connected to by the USB. This is similar to the Bridge-device but instead of implementing an 802.1d bridge, the device acts as an IP router. This might for example be a cable or xDSL modem with built-in masquerading firewall software. In this case the host or desktop system wants to interact with the outside network, but this interaction is being protected by the firewall and may be using services provided by the Router-device such as DHCP, DNS proxy, caching, etc. It also may be controlled/configured by the host system via a web browser for example.

---

Each of the above involves implementing a virtual LAN connection over the USB bus, but in each case the particulars of how this needs to be done are different. Specifically, the design for each device needs to look at specific USB protocol requirements, host class implementation requirements, network address assignment, and user interface peculiarities; all in order to create a supportable and reliable end-product.

## 2. USB Network Protocol Requirements

While all are based upon the USB standards group's Ethernet Networking Control Model, there are three categories of USB network protocols in common use:

- USB CDC (USB forum standard)
- Microsoft RNDIS (de facto standard for Windows)
- Other proprietary protocols (required by Device hardware unable to implement one of the others. See Belcarra's CDC/RNDIS discussion paper at [http://technologies.belcarra.com/belcarra\\_usb\\_rndis.htm](http://technologies.belcarra.com/belcarra_usb_rndis.htm))

The CDC and RNDIS protocols are required for some devices, for example the cable modem DOCSIS standards. These protocols have some extended facilities that allow the device to pass additional information to the host, such as MAC address to be used (MAC borrowing or spoofing used to register a specific PC with the ISP's gateway for authorization purposes.)

Some of the proprietary protocols use variations on CDC/RNDIS, or implement similar or reduced subsets of them. Belcarra has a simple configuration, useful for products (such as PDA-devices) that do not require the more extensive configuration provided by the RNDIS or CDC protocols. This protocol is called SAFE and also implements an optional 32 bit CRC to provide additional protection against corrupted data (for example where the UDC – CPU interface has speed-related problems or other hardware deficiencies.)

Finally, some devices that need to implement the CDC protocol are unable to satisfy the CDC requirements for USB resources. For example, some USB UDCs cannot implement the required SET ALTERNATE INTERFACE command. To circumvent this problem, a variation of the CDC configuration called MDLM is available.

## 3. USB Networking Host Requirements

There are several requirements for USB networking placed on the Device by the host OS:

- Unique serial number (MS Windows)
- MAC address (all OSs)

For Windows hosts, each USB Device must have a unique serial number. This is used to manage which drivers are loaded and associated with the device. Windows will not work properly if each device does not

have a unique serial number. In other words, it may not recognize an existing device if it is reconnected, particularly at a different USB port, and it may falsely recognize a new device as an existing one.

Depending on the type (PDA, Bridge, Router) of Device, the MAC address the host uses can be either a locally administered address or a unique, globally administered address.

When used with PDA-devices, the host can have a locally administered address that is always the same. The PDA-devices should use locally administered randomly generated MACs.

When used with a Bridge-device, the host must have a unique, globally administered address provided to it by the Bridge-device. The Bridge-device itself can optionally have it's own globally administered address (if it needs to be accessed from the public network) or can use a locally administered address (if it only needs to be accessed from the private network.)

## 4. Network IP Address Assignment

The host system and in some cases the device itself, may need to be assigned one (or more) IP addresses. Typically this is done by using DHCP (or in some cases RARP.) The location of the DHCP (or RARP) server varies depending upon the type of device being implemented.

For PDA-devices, the DHCP/RARP server must be implemented in the host system they connect to.

For Bridge-devices, the DHCP server will only be used by the host system and will reside somewhere on the outside network the device is bridging traffic to (the ISP's network.)

For Router-devices, the Router will use DHCP to configure the public interface by using the DHCP server on the public network. It will in turn implement a DHCP server for use by the host that is connected to it by USB.

## 5. USB DEVICE – PDA Configuration

For PDA-devices that need to internetwork with a USB Host system and other devices using the USB Bus

USB Device considerations (Linux):

- SAFE configuration is recommended
- Unique serial number is required
- Random, locally administered local MAC address is OK
- DHCP client or RARP client (*dhcpc/rarpd*) should be used to configure interface (usb10)
- CRC encapsulation is available if required

In addition to the above, any device designed to only be intermittently connected to the USB bus or to be connected to another portable, battery-conserving Host, must also correctly implement Device suspend (when for example a laptop Host goes into power-save mode) and GPIO-controlled pullup of

Windows USB Host considerations:

- USBLAN driver required (see Belcarra's USBLAN white paper)
- Default use of DHCP to request address should work to get address from remote (public network) DHCP server.
- *dhcpcd/rarpd* is run locally (on Windows) as part of the USBLAN package to co-ordinate IP address assignments on the virtual LAN

Linux USB Host considerations:

- *usbnet* may be used
- Configure usbN into 802.1d bridge
- Offer *dhcpcd/rarpd* on all usbN interfaces (stand alone server, not part of *usbnet*)

## 6. USB DEVICE – Bridge configuration

In this configuration, the device will implement a transparent IEEE 802.1d bridge to an external network fabric such as Data over Cable (DOCSIS) or RFC 2684 (for xDSL modems). As such, the device will normally not be addressable by the PC at all. The device will likely have a management IP address which is not accessible from the PC host.

For Bridge-devices that will act as an invisible 802.1d type bridge between USB Host system and another network allowing the local hosts to look like they are directly on the public network.

USB Device considerations:

- RNDIS with one of the CDC/MDLM configurations required
- Unique serial number required
- Random, locally administered, local MAC address may be OK
- Unique globally administered remote MAC address is required
- Device node usbN should be added to 802.1d bridge to public network
- CRC encapsulation is available if required (only with CDC/MDLM)
- Firewall may be configured to protect inside network (device node usb0) from outside traffic on public network using 802.1d support for *ipchains*

Windows USB Host considerations

- USBCDC or RNDIS drivers must be used
- Default use of DHCP to request address should work to get address from remote (on public network) DHCP server

Linux USB Host considerations

- *usbnet* may be used
- *dhcpc* can be used to get address from remote (on public network) DHCP server

## 7. USB DEVICE – Router Configuration

For Router-devices that will act as an edge router between a USB Host system and another network, typically implementing a routed, NATed or masqueraded network for locally connected hosts.

USB Device considerations:

- SAFE protocol may be used. RNDIS with one of CDC/MDLM configuration is preferable
- Unique serial number required
- Random, locally administered local MAC address may be OK
- Random, locally administered remote MAC address may be OK
- Device node *usbbN* should be statically configured
- CRC encapsulation is available if required (only with CDC/MDLM)
- *dhcpcd/rarpd* should be configured to respond to *usbbN*
- *dhcpc* should be used to configure public network interface
- Firewall should be configured to protect inside network (*usbb0*) from outside traffic on public network interface

Windows USB Host considerations:

- USBCDC or RNDIS drivers may be used
- Default use of DHCP to request address should work to get address from local (on USB Router) DHCP server

Linux USB Host considerations

- *usbnet* may be used
- *dhcpc* or *rarpd* can be used to get address from USB router

---

## 8. Configuration Selection Summary

- RNDIS
  - RNDIS requirements
    - The device must provide the Windows host a unique IP address and a unique MAC address
    - The UDC must meet certain requirements (extended enumeration, INTERRUPT IN endpoint)
  - RNDIS benefits
    - Microsoft-maintained Windows drivers
- CDC/Ethernet
  - CDC/Ethernet requirements
    - UDC must implement “select alternate interface”
    - The device must provide the Windows host a unique IP address and a unique MAC address
  - CDC/Ethernet benefits
    - Vendor neutral *de jure* standard published by the USB Forum. The DOCSIS standard (for cable modems) requires either RNDIS or CDC/Ethernet
    - Framing is simpler than RNDIS (possible performance enhancement)
- SAFE
  - SAFE requirements
    - SAFE can be implemented by all known UDC’s
  - SAFE benefits
    - when used with USBLAN driver, can support bridging and IP address management of several devices (PDA’s).
    - Options exist to cope with all known design limitations and errata of popular UDC’s.

- Devices without unique serial numbers can be assigned unique MAC addresses which can then persist as long as the PDA remembers them, facilitating recognition of PDA's by host synchronization software.

## 9. Conclusions

The implementation of USB networking must take into consideration the end-use of the device. Consideration of the nature of the device dictates what 802.3 and IP network facilities must be or should be available. The most complex, the PDA-device, requires special consideration on the Host end to enable easy to use and reliable end-user products.

While we work with all manner of USB Network connected devices, our specific goal is to enable products such as PDAs, Notepad computers, wearable computers, automotive systems and other "smart" devices to use USB in the manner of a peer to peer network. The result of our work, Belcarra's USBLAN, is a set of products designed to allow a USB Host to implement a Local Area Network (LAN) comprised of various USB Devices. Belcarra's USBLAN allows the USB Host to use TCP/IP protocol and services to communicate with each of many USBLAN connected Devices. It also allows each Device to communicate with other USBLAN Devices on the bus in a full peer to peer fashion.

## 10. Authors

Stuart Lynne is a founder and Senior Engineer at Belcarra Technologies division of BMC. His background includes the creation of all manner of commercial and open source Internet and communications software. Most recently he has been the architect of the Linux USB device facilities currently in the Linux kernel tree.

Richard Pitt is a founder and CEO of Belcarra Messaging Corp. His background includes design and marketing of Internet devices and facilities from SOHO routers and multi-media jukeboxes to continent-spanning ISP Internet Infrastructure.

Bruce Balden is a founder and Senior Engineer of Belcarra Technologies. His background includes extensive work with embedded devices.