



# 1 Belcarra Technologies USBLAN Windows™ Network Class Driver with EEM sub-class White Paper

Stuart Lynne  
Richard Pitt  
April 21, 2006

## 1. Overview

Belcarra Technologies' USBLAN class driver for Windows is a Microsoft Windows networking-over-USB solution compatible with both the older CDC-ECM and the more recent and higher performance CDC-EEM. USBLAN is available for Windows 2000 and Windows XP with Vista versions expected soon.

Belcarra has followed the progression of USB's use for connecting network peripherals of all types. As a pioneer in the provision of driver software both for smart peripherals as well as for the desktop operating systems that support them, Belcarra has produced USBLAN, which enables smart peripherals to share data across the USB fabric as if they were connected to an otherwise normal TCP/IP network via Ethernet.

Belcarra's USBLAN Network Class Driver implements an Ethernet-based Internet network function using either the CDC-EEM (Ethernet Emulation Model) sub-class for transporting Ethernet frames over USB or the CDC-ECM (Ethernet Communications Model).

A typical desktop Windows networking environment that might be called upon to use USB to connect to network devices meets two distinct types of USB networking devices:

1. **Smart** (PDA) devices which will source and sink network data
2. **Infrastructure** (Bridge/Router) devices that will act as an Ethernet packet bridge between the Windows system, its own local devices, and another 802.3 (Ethernet) based network, and/or route Internet Protocol packets to another IP based network

Each type of device implements a distinct type of network connection and must interact with the host in a correspondingly distinct fashion. The USBLAN driver recognizes the type of device by the type of configuration presented by the device during enumeration.

### 1.1 Smart Devices

When one or more smart (PDA) devices are connected to the Windows system they each need to receive an IP address in order to interact with each other and the Windows host, or through the host to other Internet connected devices. While the Windows host itself may have received an IP address from its upstream host, it cannot be presumed that its own downstream (USB connected) devices can or should

---

get addresses in the same fashion. A separate, locally controlled and hosted DHCP/RARP server must be used. Since Windows does not generally supply these with most of its versions, the USBLAN driver does.

As smart devices are attached to the USB bus, the USBLAN driver implements a 802.1 style bridge allowing them to access both the Windows system and the other smart devices on the USB bus. The internal DHCP server (which may be turned off in the configuration if not required<sup>1</sup>) coordinates IP address assignment between a single Windows NDIS network interface and the smart USB-connected device network interfaces. The address assignment can be integrated into the USB enumeration process or handled in normal DHCP fashion. Similarly, smart device time-of-day may be synchronized to the Windows host during USB enumeration or through the integrated Network Time server.

## 1.2 Infrastructure Devices

When one or more infrastructure devices are connected concurrently (via USB) to the host system they each need to be treated as separate and independent connections. Windows needs to get a MAC Address from the device and to be able to get an IP address for its own interface by sending a DHCP request to or through the device.

When an infrastructure device connection via USB is detected, the USBLAN driver implements a separate Windows network interface for each such device. The MAC address for the interface will be assigned to the address given to it by the Bridge or Router using the USB CDC protocol. DHCP requests from Windows are sent over the network to/through the Bridge or Router device. IP address assignment is the realm of the device or externally connected network.

## 2 Background

The progression of USB from its beginnings as a simple way to add peripheral ports<sup>2</sup> to a PC without the need of opening the case, to its current extended status as a secure, fast, desktop connection facility for all manner of network aware devices, has progressed through a number of iterations.

Connecting smart devices such as PDAs and cellular telephones (cellphones) started out as a serial-protocol function. The proprietary nature of the typical protocol used on the serial link meant that devices didn't interact at all other than with their own specific desktop program. The possibility of being able to share calendar scheduling information or music between a PDA and a cell phone (even if the cell phone had the software to handle it) via connection with the PC was simply not there.

The progression of the Internet in general and Internet Protocol (TCP/IP) in particular as a ubiquitous standard for communicating data between otherwise disparate devices, along with the standardization on

---

<sup>1</sup> Most configuration settings are done at compile time by the developer. Some are available as registry settings and may be manipulated by the end user if a suitable configuration tool is provided by the developer. In most cases the defaults work fine.

<sup>2</sup> serial, parallel, sound, human interface/keyboard/mouse, Ethernet, etc

formats such as Web, MP3, MIME, iCAL and other IETF RFC<sup>3</sup> standards has brought forth a myriad of devices that want to network to and through the typical desktop PC.

To this we add the virtual stranglehold that 802.3 (Ethernet) over various physical media has on the transport of IP.

USB steps up to this plate with a connection via a small form factor plug that by design provides power if necessary, and can handle data at high speed by emulating an Ethernet device. CDC-ECM was designed to support this scenario by including enough information in the device configuration to allow the host to correctly send and receive network frames (for example CDC-ECM has a mechanism for providing a MAC address to the host.)

CDC-EEM reduces the complexity, and offers the potential for better throughput as well.

Belcarra's goal with USBLAN is to facilitate the easy development and deployment of USB connected products in a fashion that takes best advantage of the Internet standards and capabilities embodied in today's Windows based desktop PCs. By allowing diverse smart USB devices to cooperate in a peer-to-peer network within the secure physical connection of the USB bus, USBLAN encourages synergies that otherwise would not typically be possible.

The Windows networking environment presents some unique challenges to the process, and Belcarra's USBLAN includes optional components that make the implementation easy and painless, especially for the end user. As well as the actual configuration requirements, some operational peculiarities in Windows' upper networking layers are areas where Belcarra's driver optimizations make large differences in throughput.

The addition of CDC-EEM to USBLAN extends our product to embrace the latest USB Ethernet Emulation facility. In our USBLAN Windows driver we continue our concept of enabling Windows users to easily and transparently network between their USB connected smart peripherals, and add the higher throughput and easier development path for new devices that EEM allows, while retaining backwards compatibility with earlier devices.

### **3 USBLAN – Belcarra's Solution**

Belcarra's solution to Windows-USB networking integrates functions in the typical smart USB peripheral with extended functionality of the Windows USB driver to achieve a unique level of integration and utility. The Windows driver system will also allow use of other USB peripherals such as infrastructure devices (bridge/router/DOCSIS<sup>4</sup> modem, etc.) and systems based upon their use for older smart peripherals.

---

<sup>3</sup> RFC - Request For Comments – a series of documents which together define the standards of the past, present and future of the Internet. An evolving set of documents. See also [http://en.wikipedia.org/wiki/Request\\_for\\_comments](http://en.wikipedia.org/wiki/Request_for_comments)

<sup>4</sup> DOCSIS - Data Over Cable Service Interface Specification for cable modems

---

Belcarra's USBLAN driver is implemented as a Windows NDIS Miniport driver with the upper half being a standard NDIS driver and the lower half a WDM driver to access the USB peripherals. For smart devices an integrated bridge supports implementation of a virtual LAN between multiple smart devices. The implementation also includes built in servers for DHCP, RARP and Time of Day which optionally allow for simple configuration of both the host and connected devices.

### **3.1 Smart Device Networking**

Smart (non-infrastructure) devices create, or are destinations for, data, as opposed to just passing it through. In order to do this, they must participate in the low level (Ethernet) transport and also in the higher level (IP) sessions so must have both a MAC address and an IP address in order to function in the network.

The typical USB connected smart device thus requires either a MAC address built in at manufacturing time, or one that is assigned and locally administered for the duration of its connection to the host PC – and the PC in this locally assigned case must ensure that the non-globally-unique MAC does not pass to the rest of the world!

Once the device has a MAC address, it must also be given an IP address in order to participate in the local area network. The address must be unique within the domain of the PC, and should be in a block that is not otherwise used on any other network interface besides the one associated with USB connected smart devices.

### **3.2 Infrastructure Device Networking**

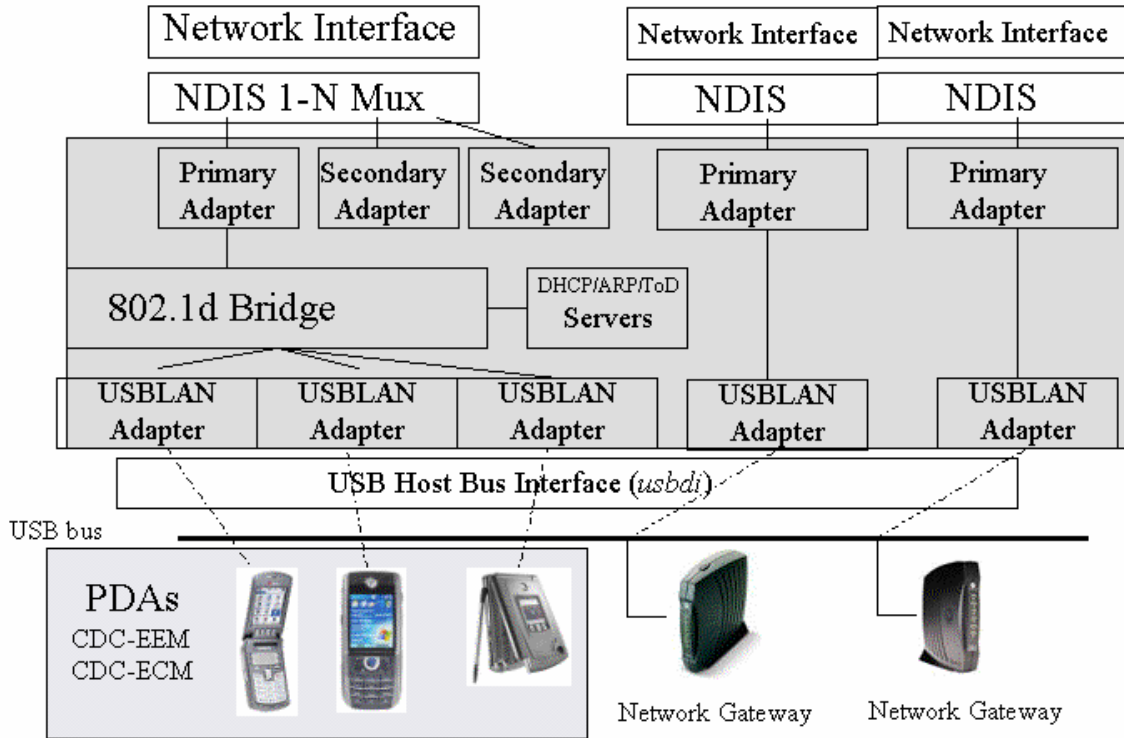
A USB bridge device acts as if it were a network interface card (NIC) inside the PC. It must have a MAC address that is globally unique and typically the USB enumeration process (via CDC-ECM) hands this MAC to the driver so that the PC can reply to ARP requests<sup>5</sup>. Otherwise, the bridge simply hands network frames from its external connection through its USB connection to the PC with no interaction.

A router device, besides interacting with multiple Ethernet LANs at the MAC level, may add the ability to selectively manipulate higher level (than MAC) routing (i.e. IP addressing) portions of the data frames but does not typically deal with session or port addresses. A router will typically have an IP address assigned by an upstream agent and in turn may either issue itself and its downstream PC an address via DHCP or pass through DHCP requests/answers.

In both cases the networking data to/from the PC must be handed up the networking stack from the USB subsystem to the higher-level operating system functions without any other interaction.

---

<sup>5</sup> ARP - Address Resolution Protocol – see [http://en.wikipedia.org/wiki/Address\\_Resolution\\_Protocol](http://en.wikipedia.org/wiki/Address_Resolution_Protocol)



### Belcarra USBLAN Windows™ Driver

Copyright© 2006 BTC

Figure 3-1

In Figure 3-1 the virtual LAN is implemented in the Windows USB driver system and enables traffic between the smart devices and the PC. Each smart device is able to send and receive IEEE 802.3 Ethernet data frames to/from any other device or the host via the IEEE 802.1d virtual bridge. The NDIS virtual multiplexer (MUX) multiplexes these frames across all USB network adapters so that only a single network interface is presented to the higher levels of the Windows operating system for all USBLAN aware peripherals.

If a smart device lacks a hard-coded MAC address, the included Reverse ARP server (RARP<sup>6</sup>) will answer RARP requests from the smart device with a “locally administered” MAC. In such cases the virtual bridge will not allow the MAC to be seen outside the local address pool.

<sup>6</sup> RARP – Reverse Address Resolution Production and Discovery



The embedded DHCP<sup>7</sup> server hands out addresses and other necessary IP information (gateway, name server addresses, etc.) and the Time of Day server synchronizes the downstream devices with the Windows date and time.

Once IP addresses have been assigned, client PDAs can appear in the PC's "Network Neighbourhood" and share files and abilities with the PC such as printing and mass storage. Likewise the PC can appear in the PDA's equivalent to network neighbourhood and share its devices and facilities with the PDA.

If more than one PDA is enabled on the network, each can appear as if it were another PC on the LAN, also sharing storage, facilities or peripherals.

The host driver on the PC provides a single Ethernet protocol entry into the host Windows operating system network stack, and emulates an Ethernet bridge (802.1d) to connect to all of the USBLAN clients on the USB bus.

In implementation, the Windows USBLAN network class is compatible with both CDC-ECM and CDC-EEM. The Belcarra USBOTG network interface function driver for smart peripherals is compatible with the USBLAN class driver.

In case the ECM framing is required, a full speed connection is limited to 1000 frames per second (heartbeat=1ms) of minimal size frames by Windows' inability to queue up more than one frame at a time. Where a full 1500 byte frame is transmitted the time to transmit takes more than a single heartbeat so maximum number of Ethernet frames per second is roughly 50% of maximum USB frames/sec. or 500x1500 byte frames or roughly 750kBps with typical 2ms latency. This wastes approx ½ USB frame every second frame.

For EEM framing Belcarra implements a 4k buffer pool and, at the cost of a slight rise in latency due to the buffering, delivers maximum throughput by ensuring that all USB frames transmit with full payloads except the last one. This gives close to the theoretical maximum of 900kBps with 3ms latency.

## **4 Minimal Configuration Required**

Belcarra Technologies' USBLAN driver includes built-in RARP and ARP daemons which automatically sense IP assignments from the peripheral or willingness of the peripheral to accept assignment, all at the link layer and not involving any USB protocol extensions. For the purpose of allowing the multiple concurrent connections of related smart peripherals, USBLAN also has a built-in DHCP server available, activated by suitable INF or Registry entries. Coupled with the built-in bridge, this allows all such peripherals to receive a locally administered IP address within the same block and participate in IP peer-to-peer conversations.

### **4.1 Smart Device Configuration**

For smart (non-infrastructure) devices, the USBLAN driver will respect a globally unique MAC address supplied by the device, or hand it a locally administered MAC address as necessary. The driver acts as a

---

<sup>7</sup> Dynamic Host Configuration Protocol. See <http://en.wikipedia.org/wiki/DHCP> for details.

bridge between all the attached non-infrastructure devices and allows any valid network frames to be sent to them.

By default, CDC-EEM devices are assumed to be smart/non-infrastructure devices if they do not support the RARPD protocol.

Smart devices that do not have an in-built MAC address at manufacture can reply to the RARPD request with an error code that signifies they are not infrastructure devices, and then in turn can send their own RARPD request to the USB host. A MAC address, useful for the duration of the connection and with the “locally administered” bit set, is returned.

Once a MAC address is established for each peripheral, valid Ethernet frames may be transmitted via the EEM framing. The bridge portion of the driver handles whether the data should traverse farther up the driver stack or not based upon the MAC address in the Ethernet frames rather than requiring any device information other than the end points in the USB frame.

The integrated 802.3 bridge in the driver supports a virtual shared network for all attached non-infrastructure devices. All such devices (whether MAC is locally or globally unique) may send network (Ethernet) frames to each other via the USB host. Smart devices with a globally unique MAC may send frames via the bridge to the host OS and through it to other connected network peripherals, including those connected as infrastructure devices on the USB bus.

The Belcarra USBLAN class driver also implements a DHCP server. This server coordinates IP address assignment across all attached non-infrastructure (smart) devices and the host (Windows) network interface. This DHCP server configures itself to avoid locally known address blocks, typically using one of the non-routable blocks in the 192.168.0.0/16 block. Settings in the Windows registry show the choice and may be changed if necessary.

The USBLAN driver also implements a Time-of-day server that can provide the smart devices with information at enumeration to synchronize the device with the Windows server time and will act as a limited-function Network Time Daemon (NTP4 - RFC-2030)

## **4.2 Infrastructure Device Configuration**

Infrastructure devices are capable of acting as a network (802.3) bridge<sup>8</sup>. By design, the USBLAN driver will only send network frames to an infrastructure device which:

- are directed to the MAC address of that infrastructure device
- are framed with a globally unique MAC address with an official (IEEE) OUI (as opposed to a locally administered MAC such as the driver’s RARPD might give out if necessary)

Infrastructure devices can inform the host of its globally unique MAC address by means of a RARP reply to a request issued by the built-in RARP client in the USBLAN driver.

---

<sup>8</sup> The concept of “network bridge” is discussed in detail at [http://en.wikipedia.org/wiki/Network\\_bridge](http://en.wikipedia.org/wiki/Network_bridge)



---

USBLAN also supports the CDC/MDLM protocol because some older devices could not conform to the requirements of the CDC-ECM protocol.

The USBLAN driver creates a separate network interface to Windows (NDIS) for each infrastructure device that is plugged in. These devices do not participate in the USBLAN bridge, and the integrated DHCP server will not process DHCP requests. Infrastructure devices may use either CDC-ECM or CDC-EEM. By default, CDC-ECM devices are assumed to be infrastructure devices even if they do not support the RARPD protocol.

## 5 Conclusions

The Belcarra Technologies USBLAN Windows driver provides a stable, efficient and easily implemented third-generation USB networking system.

Whether you are implementing an infrastructure device or smart device, the Belcarra Technologies USBLAN driver will support your product in the Windows desktop environment.

The use of Belcarra's USBLAN technology enables fully bridged networking between Ethernet and Internet aware smart peripherals in a controlled and physically secure USB environment with minimal or no user intervention when using standards-based USB peripheral drivers such as Belcarra's USBOTG. Today's network aware smart peripherals can make unlimited use of the broad capabilities Belcarra's USBLAN brings to the desktop USB peripheral scene.

Companies creating new smart peripherals of all kinds who want to include networking functionality to and through the desktop PC via USB can easily and quickly integrate the Belcarra solution into their product, with the results of:

- Reduced implementation time
- Reduced end-user support costs
- Increased end-user satisfaction through increased functionality.

And with the inclusion of the latest CDC-EEM sub-class in Belcarra's new USBLAN version, these benefits are added:

- Reduced USB hardware complexity (only bulk in and bulk out are required)
- Increased network throughput (900+kBps vs. 750kBps on a full-speed link)

Note: A reduced-function version of ECM and EEM enabled USBLAN is also available as a SRPM for Linux desktop systems.

### **About Belcarra Technologies Corp.**

Belcarra Technologies is the leader in high performance network over USB solutions for the Windows™ desktop as well as Linux and Apple MAC.OS/X systems.

Belcarra's USBLAN Windows driver system is available under license for OEM inclusion with all manner of smart and infrastructure USB networking products.



---

Belcarra's USBOTG solution for smart USB peripherals includes those based upon Intel, AMD and Freescale System-On-Chip as well as many others.

For further information please contact [info@belcarra.com](mailto:info@belcarra.com) or see our web site at [www.belcarra.com](http://www.belcarra.com) or phone: +1-604-617-3649