

BLIP 2.0 Porting Guide

From TinyOS Wiki

This guide is intended to help application developers who have written code for BLIP-1 (up to 2.1.1) and wish to move to blip-2.0 for one reason or another. For applications which have restricted themselves to the user-facing interfaces (primarily UDP), this should be very quick. Applications which have implemented more complicated changes such as routing protocols will need more work since this version of blip substantially re-writes the internals.

These notes refer to the source of an application; in addition the IPBaseStation/ip-driver step for running an Edge Router have been superseded by the PppRouter/pppd applications; users should consult the Blip 2.0 Tutorial for details on using this.

Simple Application Guide

Applications using only basic IPv6 networking will need to make only a few changes. UDPEcho is an example of such an application.

1. The include path has changed: applications should `#include <lib6lowpan/ip.h>`, instead of `#include <ip.h>`
2. The main IP component is no longer called IPDispatchC: applications should wire to IPStackC for SplitControl. Making a new `UdpClientC()` is still the right way to allocate a UDP socket.
3. The struct `ip_metadata` in event `UDP.recvfrom` has been renamed `struct ip6_metadata`. The `sender` field has been changed to `ieee154_addr_t`.
4. BLIP now uses the The TinyOS printf Library instead of `PrintfUART`; consult that documentation for details on how to see the `printf` output.

Other pieces such as `NWProgC` and `UDPShellC` have been updated and should work identically.

Addressing

One major change throughout blip-2.0 is that it fully supports 64-bit link-layer addresses, and uses them by default. This means that your mote will respond to pings on its link-local interface using an address derived from it's IEEE EUI-64 unique hardware id, if available. It is still possible to also use `TOS_NODE_ID` to specify a short address; to do this, `BLIP_DERIVE_SHORTADDRESSES` should be `#define'd` at compile time. This must be combined with either a DHCP address assignment or a static `IN6_PREFIX`: see the discussion in the Tutorial for more details of static vs. DHCPv6 addressing mode.

Advanced Applications

For applications constructing more complicated packets or using the IP interface to send/receive other protocols, blip has made more changes. The Blip 2.0 Internals page [will] have more details on using these interfaces, but at a high level the important changes are:

1. Internally, nearly all blip code has been revised to process `struct ip6_packet` structures; these consist of an IPv6 header, and an `struct ip_iovec` containing the packet data. This is exposed in both the IP interface and UDP through the new `UDP.sendtov` API which allows application to construct packets using scatter-gather. Utilities in `lib6lowpan/iovec.h` are available for dealing with these buffer chains.
2. `IPStackC` now exposes the `ForwardingTable` and `ForwardingTableEvents` interface to allow applications to inspect and receive notifications about routing table state.
3. Blip now supports multiple interfaces on the bottom -- the routing table contains an `ifindex` field which determines which interface a packet will go out on. This is convenient for many reasons.

4. Extension headers are now fully supported and compressed in the 6lowpan layer. The IPPacketC component contains some utility functions for inspecting and mutating IPv6 extension headers.
5. ICMPv6 support is located in `tos/lib/net/blip/icmp` and allows applications to receive ICMPv6 messages based on their "type" field.
6. A DHCPv6 client and relay agent are present in `tos/lib/net/blip/dhcp` and available for address assignment. This is used in lieu of 6lowpan-nd at the present.

Retrieved from "http://tinyos.stanford.edu/tinyos-wiki/index.php?title=BLIP_2.0_Porting_Guide&oldid=5443"

- This page was last modified on 14 June 2011, at 15:51.
- This page has been accessed 6,442 times.