Linux Phonet protocol family

Introduction

Phonet is a packet protocol used by Nokia cellular modems for both IPC and RPC. With the Linux Phonet socket family, Linux host processes can receive and send messages from/to the modem, or any other external device attached to the modem. The modem takes care of routing.

Phonet packets can be exchanged through various hardware connections depending on the device, such as:

- USB with the CDC Phonet interface,
- infrared,
- Bluetooth,
- an RS232 serial port (with a dedicated "FBUS" line discipline),
- the SSI bus with some TI OMAP processors.

Packets format

Phonet packets have a common header as follows:

```
  struct phonethdr {
    uint8_t  pn_media;  /* Media type (link-layer identifier) */
    uint8_t  pn_rdev;   /* Receiver device ID */
    uint8_t  pn_sdev;   /* Sender device ID */
    uint8_t  pn_res;    /* Resource ID or function */
    uint16_t pn_length; /* Big-endian message byte length (minus 6) */
    uint8_t  pn_robj;   /* Receiver object ID */
    uint8_t  pn_sobj;   /* Sender object ID */
  };
```

On Linux, the link-layer header includes the pn_media byte (see below). The next 7 bytes are part of the network-layer header.

The device ID is split: the 6 higher-order bits constitute the device address, while the 2 lower-order bits are used for multiplexing, as are the 8-bit object identifiers. As such, Phonet can be considered as a network layer with 6 bits of address space and 10 bits for transport protocol (much like port numbers in IP world).

The modem always has address number zero. All other device have a their own 6-bit address.

Link layer

Phonet links are always point-to-point links. The link layer header consists of a single Phonet media type byte. It uniquely identifies the link through which the packet is transmitted, from the modem's perspective. Each Phonet network device shall prepend and set the media type byte as appropriate. For convenience, a common phonet_header_ops link-layer header operations structure is provided. It sets the media type according to the network device hardware address.

Linux Phonet network interfaces support a dedicated link layer packets type (ETH_P_PHONET) which is out of the Ethernet type range. They can only send and receive Phonet packets.

The virtual TUN tunnel device driver can also be used for Phonet. This requires IFF_TUN mode, _without_ the IFF_NO_PI flag. In this case, there is no link-layer header, so there is no Phonet media type byte.

Note that Phonet interfaces are not allowed to re-order packets, so only the (default) Linux FIFO qdisc should be used with them.

Network layer

The Phonet socket address family maps the Phonet packet header:

```
  struct sockaddr_pn {
    sa_family_t spn_family;    /* AF_PHONET */
    uint8_t     spn_obj;       /* Object ID */
    uint8_t     spn_dev;       /* Device ID */
    uint8_t     spn_resource;  /* Resource or function */
    uint8_t     spn_zero[...]; /* Padding */
  };
```

The resource field is only used when sending and receiving;
It is ignored by bind() and getsockname().

Low-level datagram protocol

Applications can send Phonet messages using the Phonet datagram socket protocol from the PF_PHONET family. Each socket is bound to one of the 2^10 object IDs available, and can send and receive packets with any other peer.

```
  struct sockaddr_pn addr = { .spn_family = AF_PHONET, };
  ssize_t len;
  socklen_t addrlen = sizeof(addr);
  int fd;

  fd = socket(PF_PHONET, SOCK_DGRAM, 0);
  bind(fd, (struct sockaddr *)&addr, sizeof(addr));
  /* ... */

  sendto(fd, msg, msglen, 0, (struct sockaddr *)&addr, sizeof(addr));
  len = recvfrom(fd, buf, sizeof(buf), 0,
                 (struct sockaddr *)&addr, &addrlen);
```

This protocol follows the SOCK_DGRAM connection-less semantics. However, connect() and getpeername() are not supported, as they did not seem useful with Phonet usages (could be added easily).

Phonet Pipe protocol

The Phonet Pipe protocol is a simple sequenced packets protocol with end-to-end congestion control. It uses the passive listening socket paradigm. The listening socket is bound to an unique free object ID. Each listening socket can handle up to 255 simultaneous connections, one per accept()'d socket.

```
  int lfd, cfd;

  lfd = socket(PF_PHONET, SOCK_SEQPACKET, PN_PROTO_PIPE);
  listen (lfd, INT_MAX);

  /* ... */
  cfd = accept(lfd, NULL, NULL);
  for (;;)
  {
    char buf[...];
    ssize_t len = read(cfd, buf, sizeof(buf));

    /* ... */

    write(cfd, msg, msglen);
  }
```

Connections are established between two endpoints by a "third party" application. This means that both endpoints are passive; so connect() is not possible.

WARNING:
When polling a connected pipe socket for writability, there is an intrinsic race condition whereby writability might be lost between the polling and the writing system calls. In this case, the socket will block until write becomes possible again, unless non-blocking mode is enabled.

The pipe protocol provides two socket options at the SOL_PNPIPE level:

PNPIPE_ENCAP accepts one integer value (int) of:

PNPIPE_ENCAP_NONE: The socket operates normally (default).

PNPIPE_ENCAP_IP: The socket is used as a backend for a virtual IP interface. This requires CAP_NET_ADMIN capability. GPRS data support on Nokia modems can use this. Note that the socket cannot be reliably poll()'d or read() from while in this mode.

PNPIPE_IFINDEX is a read-only integer value. It contains the interface index of the network interface created by PNPIPE_ENCAP, or zero if encapsulation is off.

Authors

Linux Phonet was initially written by Sakari Ailus.
Other contributors include Mikä Liljeberg, Andras Domokos,
Carlos Chinea and Rémi Denis-Courmont.