NAME

smc – SMC 8003/8013/8216/8416 Ethernet device driver

SYNOPSIS

/dev/smc

AVAILABILITY

x86

DESCRIPTION

The SMC 8003/8013/8216/8416 Ethernet driver is a multi-threaded, loadable, clonable, STREAMS hardware driver supporting the connectionless Data Link Provider Interface, **dlpi**(7P), over an SMC 80*X*3/8216/8416 Ethernet controller. Multiple SMC controllers installed within the system are supported by the driver. The **smc** driver provides basic support for the SMC hardware. Functions include chip initialization, frame transmit and receive, multicast and "promiscuous" support, and error recovery and reporting.

APPLICATION PROGRAMMING INTERFACE

The cloning character-special device **/dev/smc** is used to access all SMC controllers installed within the system.

The **smc** driver is a “style 2” Data Link Service provider. All **M_PROTO** and **M_PCPROTO** type messages are interpreted as DLPI primitives. Valid DLPI primitives are defined in **<sys/dlpi.h>**. Refer to **dlpi**(7P) for more information.

An explicit **DL_ATTACH_REQ** message by the user is required to associate the opened stream with a particular device (**ppa**). The **ppa** ID is interpreted as an unsigned long and indicates the corresponding device instance (unit) number. The unit numbers are assigned sequentially to each board found. The search order is determined by the order defined in the **smc.conf** file. An error (**DL_ERROR_ACK**) is returned by the driver if the **ppa** field value does not correspond to a valid device instance number for this system. The device is initialized on first attach and de-initialized (stopped) on last detach.

The values returned by the driver in the **DL_INFO_ACK** primitive in response to the **DL_INFO_REQ** from the user are as follows:

- The maximum SDU is **1500** (**ETHERMTU** - defined in **<sys/ethernet.h>**).
- The minimum SDU is **0**.
- The **dlsap** address length is **8**.
- The MAC type is **DL_ETHER** or **DL_CSMACD**.
- The **sap** length value is –**2**, meaning the physical address component is followed immediately by a 2-byte **sap** component within the **DLSAP** address.
- The service mode is **DL_CLDLS**.
- No optional quality of service (QOS) support is included at present, so the QOS fields are **0**.
- The provider style is **DL_STYLE2**.
- The version is **DL_VERSION_2**.
- The broadcast address value is Ethernet/IEEE broadcast address (**FF:FF:FF:FF:FF:FF**).

Once in the **DL_ATTACHED** state, the user must send a **DL_BIND_REQ** to associate a particular SAP (Service Access Pointer) with the stream. The **smc** driver interprets the **sap** field within the **DL_BIND_REQ** as an Ethernet "type." Therefore valid values for the **sap** field are in the **[0-0xFFFF]** range. Only one Ethernet "type" can be bound to the stream at any time.

In addition to Ethernet V2 service, an "802.3 mode" is provided by the driver and works as follows. **sap** values in the range **[0-1500]** are treated as equivalent and represent a desire by the user for "802.3 mode". If the value of the **sap** field of the **DL_BIND_REQ** is within this range, then the driver computes the length, not including the initial **M_PROTO** message block, of all subsequent **DL_UNITDATA_REQ** messages and transmits 802.3 frames having this value in the MAC frame header length field. All frames received from the media having a "type" field in this range are assumed to be 802.3 frames and are routed up all open streams that are bound to any **sap** value within this range. If more than one stream is in "802.3 mode," then the frame will be duplicated and routed up multiple streams as **DL_UNITDATA_IND** messages.

The **smc** driver **DLSAP** address format consists of the 6-byte physical (Ethernet) address component followed immediately by the 2-byte **sap** (type) component producing an 8-byte **DLSAP** address. Applications should *not* hardcode to this particular implementation-specific **DLSAP** address format but use information returned by the **DL_INFO_ACK** primitive to compose and decompose **DLSAP** addresses. The **sap** length, full **DLSAP** length, and **sap**/physical ordering are included within the **DL_INFO_ACK**. The physical address length can be computed by subtracting the **sap** length from the full **DLSAP** address length or by issuing the **DL_PHYS_ADDR_REQ** to obtain the current physical address associated with the stream.

Once in the **DL_BOUND** state, the user may transmit frames on the Ethernet by sending **DL_UNITDATA_REQ** messages to the **smc** driver. The **smc** driver will route received Ethernet frames up all those open and bound streams having a **sap** which matches the Ethernet type as **DL_UNITDATA_IND** messages. Received Ethernet frames are duplicated and routed up multiple open streams if necessary. The **DLSAP** address contained within the **DL_UNITDATA_REQ** and **DL_UNITDATA_IND** messages consists of both the **sap** (type) and physical (Ethernet) components.

smc Primitives

In addition to the mandatory connectionless DLPI message set, the driver additionally supports the following primitives.

The **DL_ENABMULTI_REQ** and **DL_DISABMULTI_REQ** primitives enable/disable reception of individual multicast group addresses. A set of multicast addresses may be iteratively created and modified on a per-stream basis using these primitives. These primitives are accepted by the driver in any state following **DL_ATTACHED**.

The **DL_PROMISCON_REQ** and **DL_PROMISCOFF_REQ** primitives with the **DL_PROMISC_PHYS** flag set in the **dl_level** field enables/disables reception of all ("promiscuous mode") frames on the media including frames generated by the local host.

When used with the **DL_PROMISC_SAP** flag set, this enables/disables reception of all **sap** (Ethernet type) values. When used with the **DL_PROMISC_MULTI** flag set, this enables/disables reception of all multicast group addresses. The effect of each is always on a per-stream basis and independent of the other **sap** and physical level configurations on this stream or other streams.

The **DL_PHYS_ADDR_REQ** primitive returns the 6-octet Ethernet address currently associated (attached) to the stream in the **DL_PHYS_ADDR_ACK** primitive. This primitive is valid only in states following a successful **DL_ATTACH_REQ**.

The **DL_SET_PHYS_ADDR_REQ** primitive changes the 6-octet Ethernet address currently associated (attached) to this stream. The credentials of the process that originally opened this stream must be superuser or **EPERM** is returned in the **DL_ERROR_ACK**. This primitive is destructive in that it affects all other current and future streams attached to this device. An **M_ERROR** message is sent up all other streams attached to this device when this primitive on this stream is successful. Once changed, all streams subsequently opened and attached to this device will obtain this new physical address. Once changed, the physical address will remain in effect until this primitive is used to change the physical address again or the system is rebooted, whichever comes first.

CONFIGURATION

The **smc.conf** file supports the following options:

intr Specifies the IRQ (Interrupt Request) level for the board.

reg Specifies the shared RAM the board is jumpered for.

It is important to ensure that there are no conflicts for the board's I/O port, shared RAM, or IRQ level.

FILES

/dev/smc	character special device
/kernel/drv/smc.conf	**smc** configuration file.

SEE ALSO

streamio(7I), **dlpi**(7P)