

Myrinet 2000 Serial Link

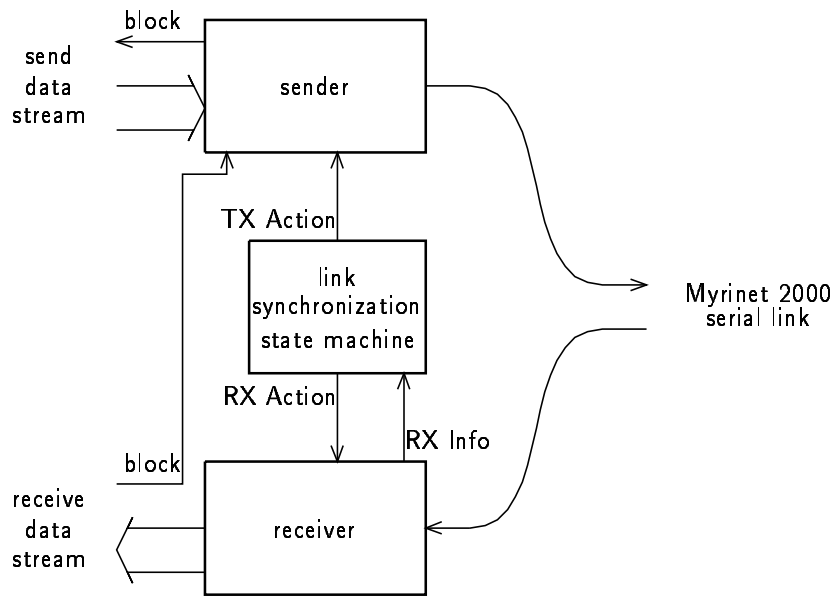
Myrinet 2000 uses the industry-standard 8b/10b encoding (IEEE 802.3 §36.2.4) to transfer data across serial links. This code uses 256 data code-groups for encoding data bytes, and 12 special code-groups for encoding control symbols.

All Myrinet control symbols, except for GAP (K29.7), consist of two code-groups: COMMA (K28.5), followed by a data code-group (see table below). The GAP is used as a packet delimiter: a packet of odd length is terminated by a single GAP, a packet of even length by two GAPs. Therefore, the code-group streams on Myrinet serial links always carry code-group pairs. COMMA code-group is always the first code-group of a pair, and is used for even/odd alignment during link synchronization (page 3). All Myrinet control symbols (except for GAP) can appear either within or between packets.

The following table specifies the mapping of Myrinet control symbols (ANSI/VITA 26-1998 §6.1) to code-groups:

SYMBOL	CODE-GROUPS	DESCRIPTION
LOST	K28.5 D5.1	The LOST symbol signals to the port on the other end of the link that this port's receiver has lost synchronization.
SYNC	K28.5 D5.6	When a port is in sync and the port on the other end is not (the local receiver is receiving LOST symbols), the SYNC symbol is emitted to enable the device on the other end of the link to regain synchronization.
GAP	K29.7	Data packets are encoded as streams of data code-groups; a packet of odd length is terminated by a single GAP, a packet of even length by two GAPs. Arbitrary number of redundant GAP symbols may be sent between packets.
STOP	K28.5 D4.1	The STOP symbol is used for flow-control, and it requests that the port on the other end of the link stop sending data symbols.
GO	K28.5 D4.6	The GO symbol is used for flow-control, and it notifies the port on the other end of the link that it may resume sending data symbols.
BEAT	K28.5 D10.4	The BEAT symbol is used for optional link-continuity monitoring. If this feature is enabled, a BEAT symbol should be emitted every $10\mu s \pm 10\%$; absence of BEATS should be reported if a BEAT has not been received for $25\mu s$.
IDLE	K28.5 D21.4	The IDLE symbol serves two purposes: 1) if a data source is slower than the Myrinet 2000 serial link, an IDLE symbol is emitted within data packets when there is no data to be sent, and 2) a BEAT or an IDLE has to be emitted at least every $25\mu s \pm 10\%$, to facilitate synchronization between two ports of a link that operate at slightly different frequencies. Myrinet 2000 serial links emit a code-group every 4ns, and the required clock-reference tolerance is $\pm 100\text{ppm}$. The maximum difference between any sender-receiver pair is therefore 200ppm, <i>i.e.</i> , a sender and the attached receiver can accumulate up to 2 code-groups worth of clock skew for every 10,000 code-groups emitted ($40\mu s$). If an elastic FIFO is used for synchronization in the receiver, it is allowed to drop or duplicate any BEAT or IDLE symbol in order to compensate for the clock skew.
ILGL	K28.5 D16.1	The ILGL symbol is not used in the normal course of operation, except by repeaters. For example, if a Myrinet 2000 repeater connects a Myrinet 2000 SAN port and a Myrinet 2000 serial port, it emits the ILGL symbol when an undefined control character is received on the SAN port.

A port consists of a sender, a receiver, and a synchronization state machine. The synchronization state machine monitors the input code-group stream (through RX Info), and controls the sender (TX Action) and the receiver (RX Action) according to the state diagram on page 3.



When the port is *in sync* (page 3), the sender forwards packets from the send data stream to the serial port, and the receiver forwards packets from the serial port to the receive data stream.

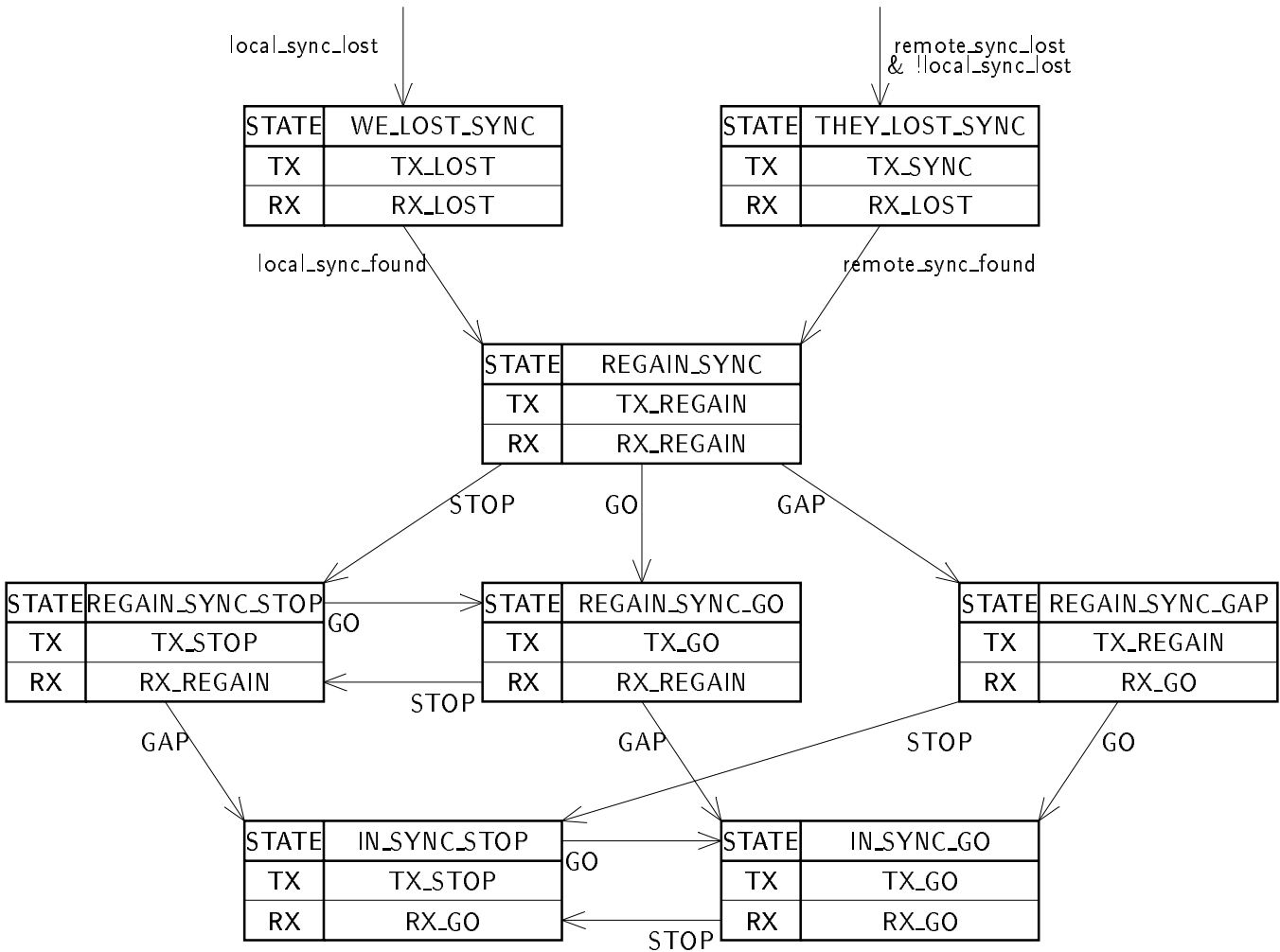
When the port is not in sync, all incoming packets are dropped.

Both send and receive data streams support flow control. However, while the send data stream can be stopped on a byte-by-byte basis, blocking the receive data stream will cause a STOP Myrinet control symbol to be emitted on the serial link, and that will stop the flow of incoming data symbols only after a cable round-trip delay. When the receive data stream is unblocked, a GO Myrinet control symbol is emitted, and the data symbols will start arriving after a minimum of a cable round-trip delay.

For example, standard Myrinet 2000 fiber links are up to 200-meters long, and the round-trip delay in such a cable is approximately $2.6\mu\text{s}$. At 250MB/s data rate, this corresponds to approximately 650 bytes of data in flight. In other words, once a port emits a STOP symbol to the port on the other end of the link, it must be ready to receive up to 650 more bytes before the STOP takes effect. Myricom's products use approximately 1KB of buffer space for this purpose, for an extra safety margin. To avoid starving the receive data stream, there should be at least 650 bytes of data (plus margin) in the receive buffer when the GO symbol is emitted. The final consideration is that the hysteresis between the "high watermark" (when STOP is emitted) and the "low watermark" (when GO is emitted) should be large enough so that, even in the worst case, the STOP/GO symbols do not consume too large a fraction of the serial-link bandwidth.

The synchronization state machine controls the sender and the receiver according to the following state diagram. In each state, TX and RX specify the Action requested of sender and receiver, respectively.

Each state transition is marked with the condition that has to be satisfied for the transition to take place; if no exit conditions are satisfied, the state machine remains in its current state.



CONDITION	DEFINITION
local_sync_lost	A port loses sync upon reset, or if there are 8 or more errors in 892 consecutive code-groups. An error is an invalid code-group, an unused special code-group, or an unaligned COMMA.
local_sync_found	A port has found sync when it detects an aligned COMMA code-group during 16 consecutive code-group pairs.
remote_sync_lost	The remote port is considered to have lost sync when a LOST symbol has been received.
remote_sync_found	The remote port is considered to have found sync when no LOST symbols have been received during 16 consecutive code-group pairs.

The sender emits a continuous stream of 8b/10b code-groups on the serial link, depending on the Action requested by the synchronization state machine. In the table below, STOP/GO means either STOP or GO, depending on the state of the local receive buffer.

TX Action	DESCRIPTION
TX_LOST	Drop any data arriving from the send data stream. Emit LOST symbol.
TX_SYNC	Drop any data arriving from the send data stream. Emit SYNC symbol.
TX_REGAIN	Drop any data arriving from the send data stream. Emit arbitrary mix of IDLE and STOP/GO symbols.
TX_GO	Forward any data arriving from the send data stream. If there is no data to be sent, emit an arbitrary mix of IDLE and STOP/GO symbols within packets, GAP and STOP/GO symbols between packets. When the sender's action changes from TX_REGAIN to TX_GO, the sender will drop the packet currently arriving from the send data stream (if any), emit a GAP, and then proceed as specified above.
TX_STOP	Block the send data stream. Emit an arbitrary mix of IDLE and STOP/GO symbols if in a middle of a packet, GAP and STOP/GO symbols if between packets. When the sender's action changes from TX_REGAIN to TX_STOP, the sender will drop the packet currently arriving from the send data stream (if any), emit a GAP, and then proceed as specified above.

The receiver interprets the continuous stream of 8b/10b code-groups arriving on the serial link. Depending on the Action requested by the synchronization state machine, the packets arriving from the serial link are either forwarded to the receive data stream or dropped.

RX Action	DESCRIPTION
RX_LOST	Drop any data arriving on the serial link. When the receiver's action changes to RX_LOST, it terminates the packet currently forwarded to the receive data stream (if any).
RX_REGAIN	Drop any data arriving on the serial link.
RX_GO	Forward any data arriving on the serial link. See the discussion on the receive-data-stream flow control on page 2.