- If the sender specifies a destination address for a connectionless message, that does not currently exist within the MCAPI communication topology, MCAPI does *NOT* treat this as an invalid send request (i.e. it's not the sender's fault that the destination doesn't exist). Implementations may however treat this as an error condition, and if so has to document the specifics. Instead MCAPI creates the message and then sends it. The return value for the send operation will indicate success since the message was successfully created and processed by MCAPI. MCAPI cannot return an error since the connectivity failure could be occurring elsewhere in the MCAPI communication topology.

3.5.2 Data Routing

Once a send of some data has been requested, MCAPI then determines what node the data should be sent to. The endpoint address indicates the destination node to which the packet should be sent. The data is then either handed off to the destination endpoint directly if it is on the same node as the sender, or passed to a link for off-node transmission to the destination node.

One problem that can arise during the routing phase of packet delivery is that no working link to the specified destination node can be found. In this case, the packet is discarded. Implementations may optionally report this is an error condition.

3.5.3 Data Consumption

When the data reaches the destination endpoint, it is added to an endpoint receive queue. The data typically remains in the endpoint's receive queue until it is received by the application that owns the endpoint. Queued data items are consumed by the application in a FIFO manner for channels, and FIFO order per priority level for messages. For messages, the user application must supply a buffer into which the MCAPI runtime system fills in the data. For packet channels, on the other hand, the MCAPI runtime system supplies the buffer containing the data to the user. For messages, the application can use its data buffer after it is filled in by MCAPI in any way it chooses. Specifically, the application can re-supply the data buffer to MCAPI to receive the next message. Data buffers supplied by MCAPI during packet channel receives can also be used by the application in any way it chooses. MCAPI reuses a data buffer it has supplied only after the user application has specifically freed the buffer using the `mcapi_pktchan_release()` function.

If an application terminates access to the before all data in the receive queue is consumed, all unconsumed data items are considered undeliverable and are discarded.

It is very important that MCAPI applications be engineered to consume their incoming data items at a rate that prevents them from accumulating in large numbers in any endpoint receive queue.

For the packet and scalar channels interfaces, a FIFO model is used. FIFOs have limited storage, and when the storage is used up, an implementation can choose to block until more storage is available or return an error code.

Non-blocking send operations will still return in a timely fashion, but data transfer will not occur if the FIFO is full. The sender will block upon calling `service routines to check for completion of a non-blocking operation`, and will continue to block until a receive operation is performed or a timeout occurs.

The user can use the `mcapi _endpoint_get/set attribute()` APIs to query or to control the amount of receive buffering that is provided for a given endpoint. Implementations may define a static receive buffer size according to the amount of buffering provided in drivers and/or hardware.
On the other hand, an implementation could be allowed to dynamically allocate storage until the system runs out of memory.

3.5.4 Waiting for non-blocking operations