define a network topology at compile time and to embed the topology in the source code in a straightforward fashion. It also enables the use of external tools to generate topologies automatically and it facilitates simple initialization. Endpoints can also be created by requesting the next available endpoint which allows for topology flexibility.

It is up to the implementation to ensure that the resulting endpoint reference is unique in the system. MCAPI allows a reference to endpoints to be obtained by a node other than the node whose node_id is associated with the endpoint, using the mcapi_endpoint_get() function, and this endpoint reference can be used as a destination for messages or to specify the opposite end of a connection. Endpoint references created by requesting the next available endpoint must be passed by the creating node to other nodes to facilitate communication. Only the creating node is allowed to receive from an endpoint.

There are three types of endpoints, message, packet channel and scalar channel. The endpoint type defines certain aspects of the endpoint's behavior. The type can for example be used to manage avoidance of messages being sent to connected endpoints. The type is set with mcapi_endpoint_set_attribute(). An endpoint is always created as a message type, which is the default type. Message type endpoints can have multiple outstanding endpoint get endpoint requests from other nodes, whereas channel type endpoints can have only one outstanding get.

Endpoints have a set of attributes. These attributes may be related to Quality of Service (QoS), buffers, timeouts, etc. Currently MCAPI defines a basic set of attribute numbers and their associated structure or data type, found in the mcapi.h file. Implementations may request a range of vendor specific attribute numbers from the Multicore Association to add additional attributes (numbers and data types). MCAPI does define API functions to get and set the attributes of endpoints. Attributes can only be set on node local endpoints but be gotten from local and remote endpoints. Furthermore, it is an error to attempt a connection between endpoints whose attributes are set in an incompatible way (for now, whether attributes are compatible or not is implementation defined). It is also an error to attempt to change the attributes of endpoints that are connected.

3.4 MCAPI Channels

Channels provide point-to-point FIFO connections between a pair of endpoints. MCAPI channels are unidirectional. There are two types of channels: packet channels and scalar channels. There is no separate channel object. Instead, channels can be referenced through the endpoints to which they are bound. Since channels are point-to-point connections between a pair of endpoints, either endpoint can be used to refer to the channel.

3.4.1 MCAPI Communications

Applications in an MCAPI topology communicate with one another by sending data between communication endpoints. Three modes of communication are supported: connectionless messages, connection-oriented packet channels, or connection-oriented scalar channels. Each communication action in all three methods (for example, a message send, a packet channel send, or a scalar channel send) requires the specification of a pair of endpoints – a send endpoint and a receive endpoint, explicitly for messages and implicitly for channels using a handle. Messages and packet channels provide both blocking and non-blocking send and receive functions.The MCAPI runtime system implements each of these logical communication actions over some physical medium that is supported by MCAPI (such as on-chip interconnect, bus, shared memory, or Ethernet) by automatically establishing a “link” to enable communication with the other node in the network, and takes care of routing traffic over the appropriate link.

Messages and packet channels communicate using data buffers, while scalar channels transfer scalar values such as four-byte words.

From an application's perspective, a data buffer to be communicated is a byte string from 0 up to some maximum length determined by the amount of memory in the system, whose internal structure is