7 Use Cases

7.1 Example Usage of Static Naming for Initialization

MCAPI's static tuple based naming mechanism makes it straightforward to implement a simple initialization scheme, including third party set up of static connections. This example describes how a packet channel can be created and used using the static tuple based naming scheme.

```
#define SENDER_DOMAIN 0
#define SENDER_NODE 0
#define SEND_PORT_ID 17

#define RECEIVER_DOMAIN 0
#define RECEIVER_NODE 1
#define RECV_PORT_ID 37
```

Sender Process:

```
mcapi_endpoint_t send_endpoint = mcapi_endpoint_create(SEND_PORT_ID,
&status);
```

Receiver Process:

```
mcapi_endpoint_t receive_endpoint = mcapi_endpoint_create(RECV_PORT_ID,
&status);
```

The connection can now be established by the receiver, the sender, or a third party process. We will use the example of a third party process.

Third party process:

```
mcapi_endpoint_t send_endpoint = mcapi_endpoint_get(SENDER_DOMAIN,
SENDER_NODE, SEND_PORT_ID, &status);
mcapi_endpoint_t receive_endpoint = mcapi_endpoint_get(RECEIVER_DOMAIN,
RECEIVER_NODE,
                                                 RECV_PORT_ID, &status);
mcapi _pktchan_connect_i(send_endpoint, receive_endpoint, …)
```

Thus, the performance impact of a global name-service lookup is entirely in the mcapi_endpoint_get() call.

7.2 Example Initialization (Discovery and Bootstrapping) of Dynamic Endpoints

This example describes how MCAPI performs discovery and bootstrapping when the static naming based on tuples is not being used, rather the dynamic endpoint scheme is used.