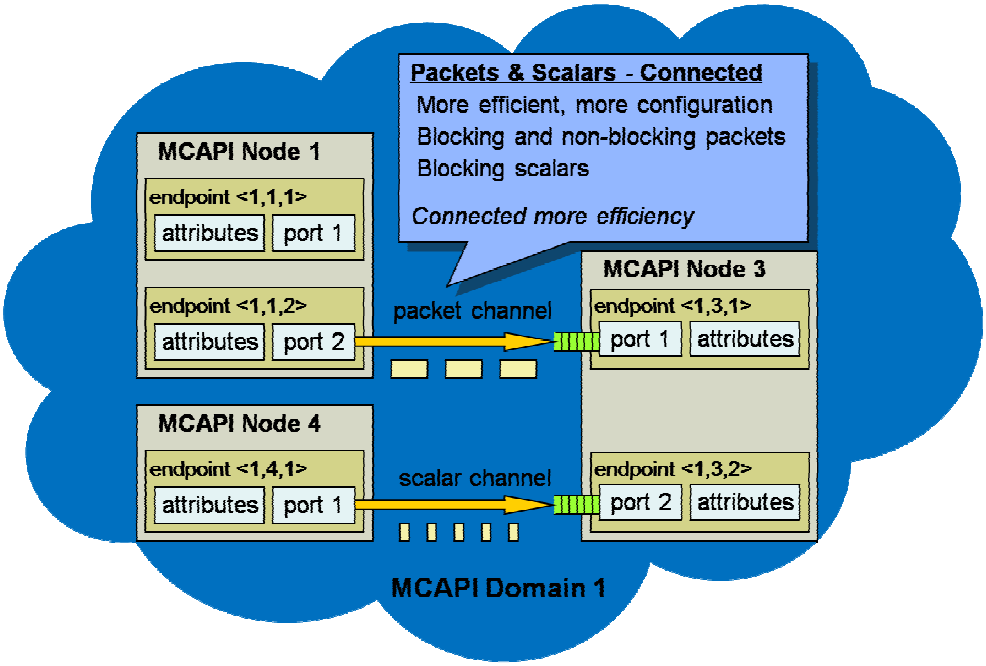

Figure 2 - Connected channels

3.9 MCAPI Zero Copy

Zero copy means that data is passed by reference instead of a physical copy. This method is useful when multiple cores have shared memory, as one core can operate on a buffer of data and simply pass a pointer to the buffer to the next core, and save the time and energy of moving the data physically. Zero copy requires specific buffer management and is intended to be orthogonal to other API operations, to avoid complexity (simplifying code migration) and performance penalties for non-zero copy operations. With the addition of the `mcapi_pktchan_release_test()` function (Version 1.1) MCAPI is capable of supporting Zero Copy, directly through the API functions, provided the availability of shared memory and support in the underlying implementation. The application is responsible for allocation of shared memory.

3.10 MCAPI Error Handling Philosophy

Error handling is a fundamental part of the specification, however, some accommodations have been made to allow for trading off completeness for efficiency of implementation. For example, a few API functions allow implementations to optionally handle errors[2]. Consistent and efficient coding styles also govern the design of the error handling. In general, function calls include an error code parameter used by the API function to indicate detailed status. In addition, the return value of several API functions indicates success or failure which enables efficient coding practice. A value of type, `mcapi_status_t`, indicates success or failure states of API calls. `MCAPI_NULL` is a valid return value for `mcapi_status_t` (can be used for implementation optimization) and implementations should state when this is the case.

[2] The `mcapi_sclchan_unit_send_*` and `mcapi_sclchan_uint_recv_*` routines define error arguments, but implementation are free to assume the routines always succeed. The message an packet channel send and receive functions can optionally report transmission errors.