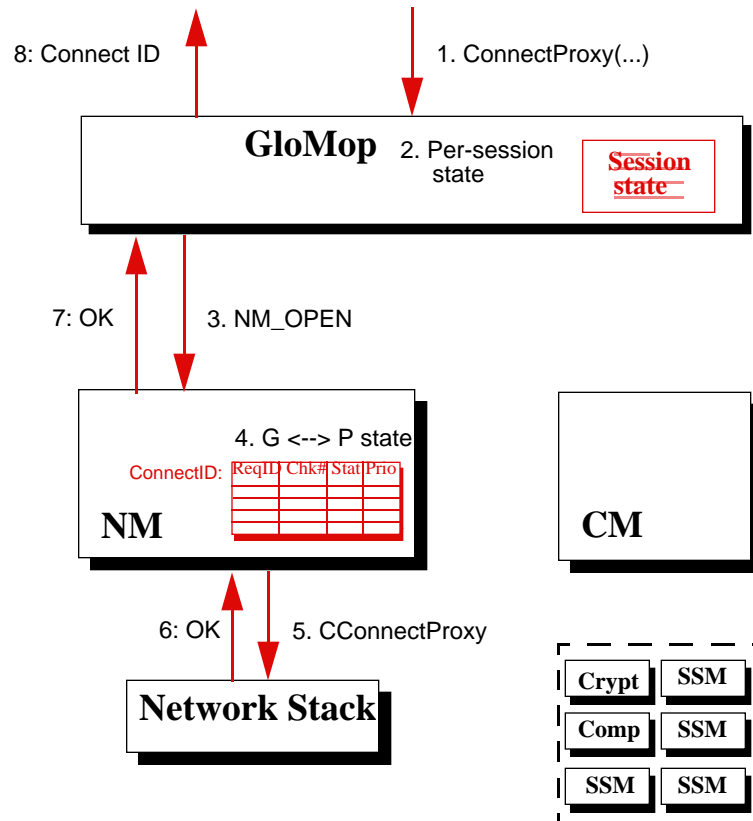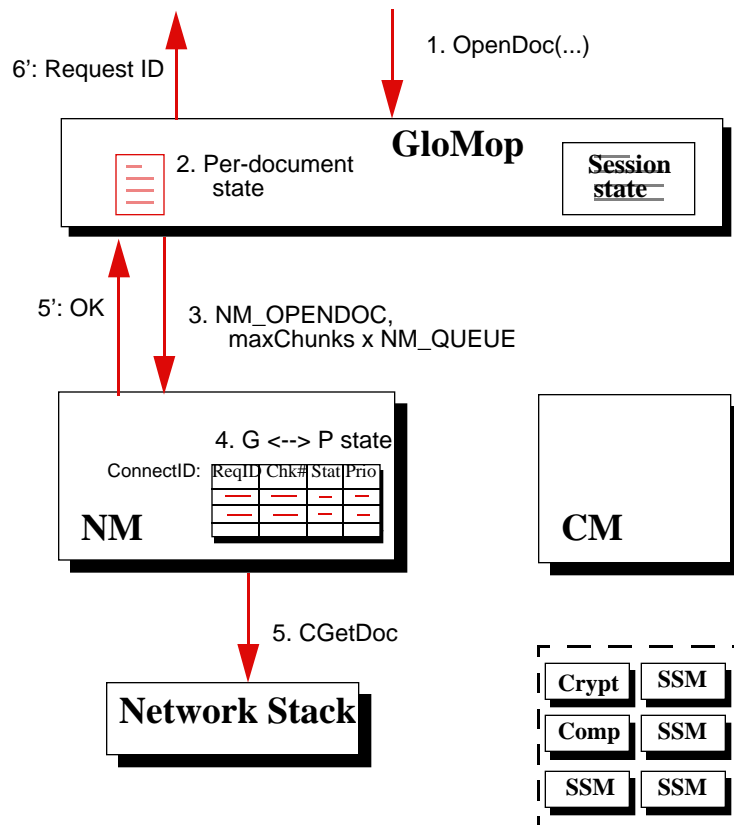## 1.0 Example scenario

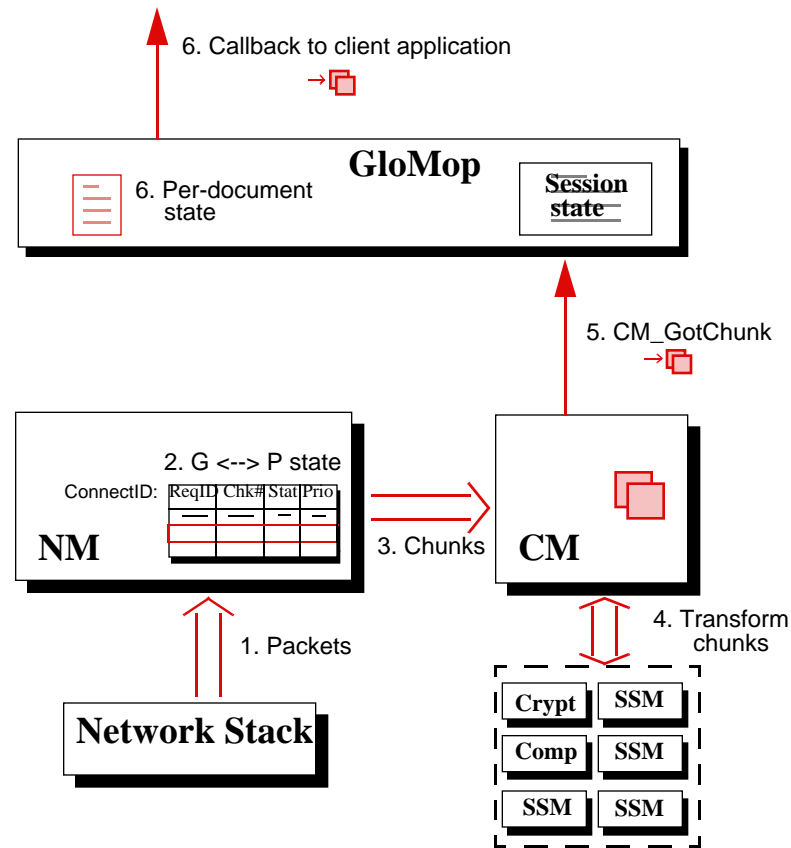### 1.1 Connecting to a Proxy

1. **ConnectProxy:** The client application issues a "ConnectProxy" request to GloMop. Parameters to this request include the proxy address.

2. **Session state:** The GloMop layer begins to store state information associated with this new session

3. **NM_OPEN:** The GloMop layer issues an NM_OPEN signal to the network management layer, asking it to forge a connection with the proxy.

4. **G <--> P state:** The NM layer initializes state information associated with this particular proxy connection.

5. The NM layer passes a well-formed and addressed connection initiation message to the network stack

6. The connection initiation succeeds. (Should the client application block on ConnectProxy until the connection is forged, or merely until the connection request is sent to the proxy?)

7. The success message is propagated back to GloMop.

8. The success message and new connection id are returned to the client application.

1.2 A Document Request

6': Request ID

1. OpenDoc(...)

GloMop

2. Per-document state

Session state

5': OK

3. NM_OPENDOC, maxChunks x NM_QUEUE

4. G <--> P state

ConnectID: | ReqID | Chk# | Stat | Prio |

NM

CM

5. CGetDoc

Network Stack

Crypt	**SSM**
Comp	**SSM**
SSM	**SSM**

1. OpenDoc: The client application issues an "OpenDoc" request to GloMop. Arguments include the document locator, QOS and encryption prefs, a callback routine, and the callback threshold and preload count.

2. Per-document state is initialized in the GloMop layer. This state includes QOS prefs, the callback threshold and preload count, callback routines for outstanding chunks, etc.

3. An NM_OPENDOC request is forwarded to the Network Manager, resulting in ...

4. ... the addition of state information to the per-proxy-connection state table.

5. The application thread returns from the network manager once the state information has been added to the table. Concurrently, the network manager thread forwards a CGetDoc request to the network stack, addressed to the proxy.

6. The application thread returns from the GloMop; a requestID for the document is returned.

1.3 Chunk arrival

6. Callback to client application

GloMop

6. Per-document state

Session state

5. CM_GotChunk

2. G <--> P state

ConnectID: | ReqID | Chk# | Stat | Prio |

NM

3. Chunks

CM

4. Transform chunks

1. Packets

Network Stack

Crypt **SSM**

Comp **SSM**

SSM **SSM**

1. The network manager receives packets from the network stack. If fragmentation of chunks has occured at the source, the packets are reassembled into a chunk, otherwise the chunk is simply extracted from a single packet.

2. The network manager's state table is modified to reflect the arrival of the new chunk.

3. The extracted chunk is forward to the chunk manager.

4. The generic transformation libraries and SSMs are used to transform the chunk into a native format or object.

5. The chunk manager passes a handle (or pointer, depending on the particular OS) to the chunk to the GloMop layer in a CM_GotChunk message.

6. The per-document state information associated with the new chunk's document is updated. If the threshold specified in the original OpenDoc call has been exceeded, then the GloMop layer notifies the client application of the new chunk via a callback. The handle (or pointer) to the new chunk is delivered in this callback.