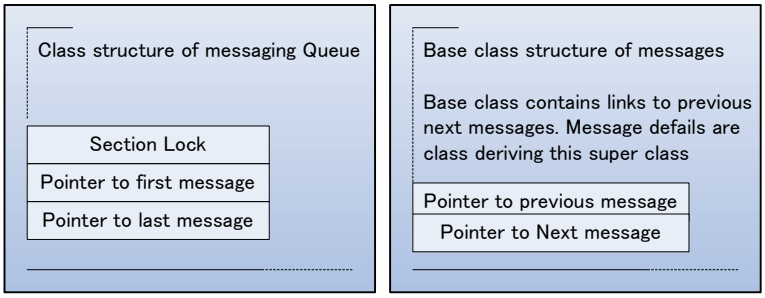
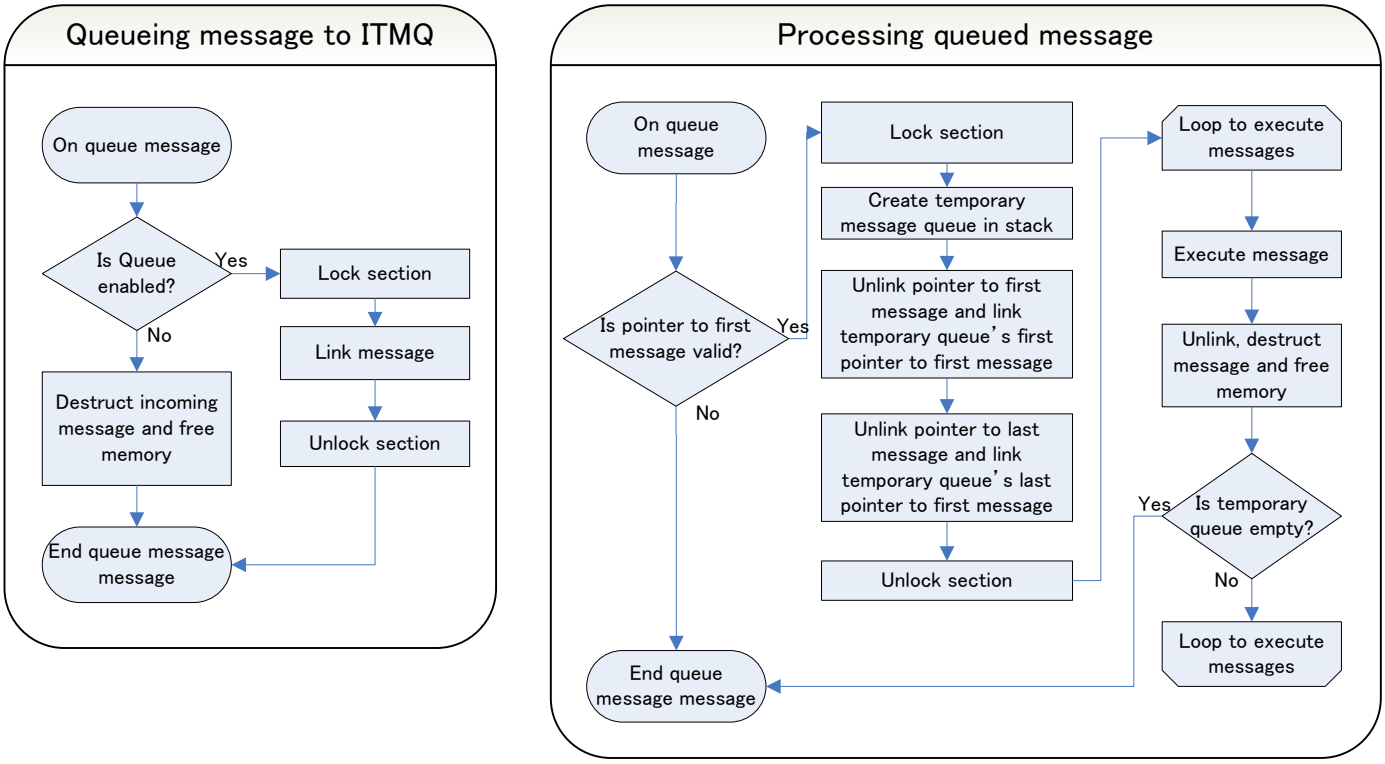
Processing Queued ITMQ messages

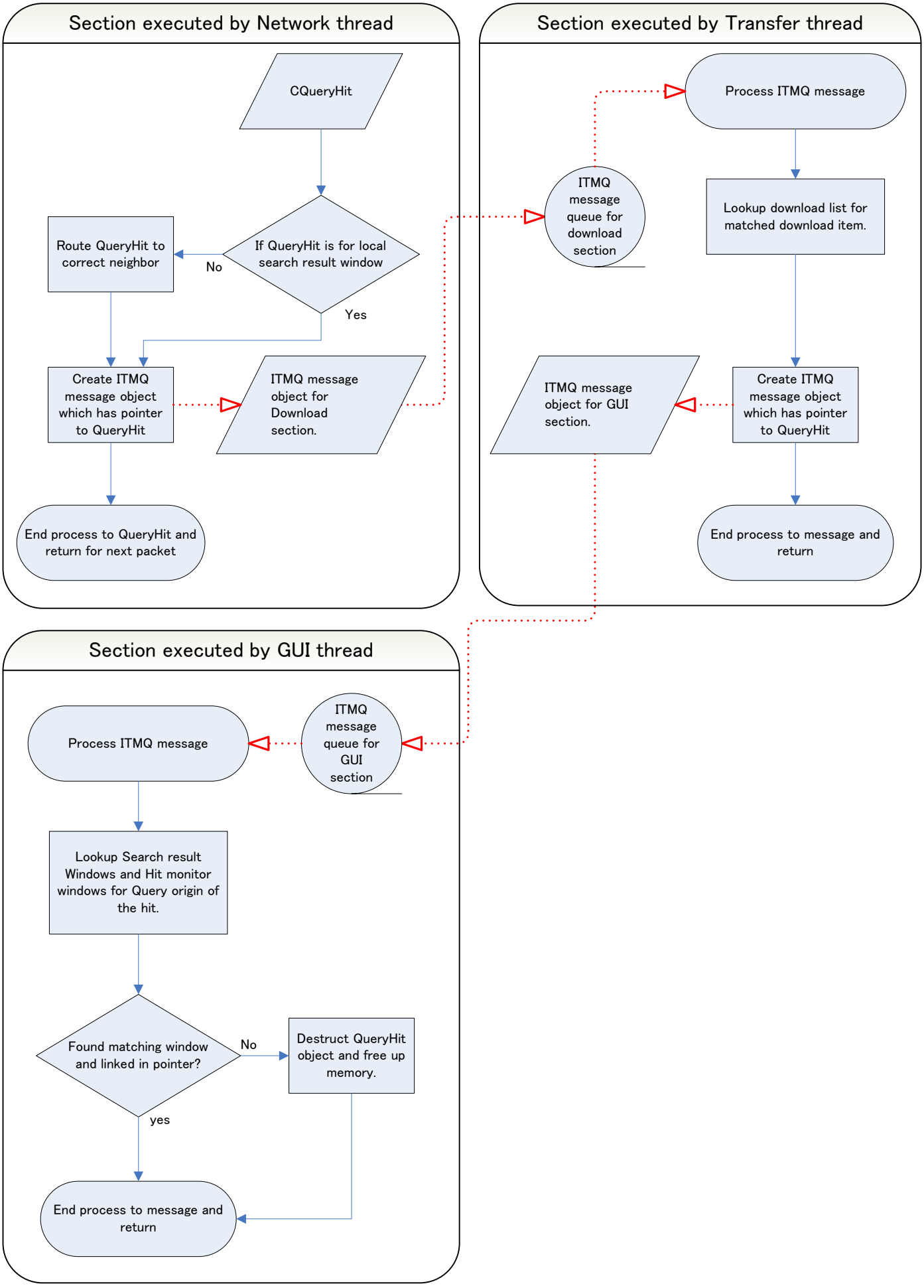




Aim of ITMQ is to reduce amount of multilocking in code. In real life situation, there are possibility of lock two or more sections needed to be locked at same time. This is not impossible, but there are situations that the lock can not be acquired. In ShareazaPlus, based on Shareaza, originally had a lot of code required multilocking in some timings.

One example is when received QueryHit, search result, from network. Adding download source from QueryHit required to lock Transfer section. But generally all the QueryHits are received and decoded in Network section which is on different section locking. Neighbour connections, packet buffers in are in network section and can not unlock while they are processed, because of processing in FIFO manner and efficiency of execution. So When any QueryHit packet comes in, need to lock transfer section in order to add them to download as download sources. However there are some situations that transfer thread, which locks transfer section, needs to lock network section. They are normally search sources and PUSH/CallBack connect requests. when both events happened at same time, unfortunately one of them has to be rejected with lock timeout, or remove time out which cause DeadLock because each thread try to lock each other's main sections.

One solution to this situation is to use messaging technology. Send message to other thread's message queue and let the receiving thread process what the sender thread want.


Section executed by Network thread
CQueryHit
If QueryHit is for local search result window
No
Route QueryHit to correct neighbor
Yes
Create ITMQ message object which has pointer to QueryHit
ITMQ message object for Download section.
End process to QueryHit and return for next packet
Section executed by Transfer thread
Process ITMQ message
ITMQ message queue for download section
Lookup download list for matched download item.
ITMQ message object for GUI section.
Create ITMQ message object which has pointer to QueryHit
End process to message and return
Section executed by GUI thread
Process ITMQ message
ITMQ message queue for GUI section
Lookup Search result Windows and Hit monitor windows for Query origin of the hit.
Found matching window and linked in pointer?
No
Destruct QueryHit object and free up memory.
yes
End process to message and return

File exsistence check in SearchResult windows