4.1. Stack Memory Requirements

4.1.1. Stack Heap Size

The stack uses a heap to allocate run-time objects such as packets, timers and endpoints. The size of the heap can be configured by the application using the ZbInit `struct ZbInitTblSizesT` parameters, specifically **heapPtr** and **heapSz**. The heap memory is allocated from the M4 core so any memory, such as packets, passed between the M0 and M4 can be read from and written to by either core. The default heap size for a Full Function Device (FFD) is 32 KiB, and 8 KiB for a Reduced Function Device (RFD). The FFD requires more memory so it can handle routing packets and the overhead that entails. If the application requires a lot of endpoints, which the stack needs to service through the ZDO layer, these will use up additional heap memory and should be factored into the size of the buffer provided to ZbInit.

4.1.2. Stack Table Sizes

The available configurable table sizes are for the:

- Network Neighbor Table (`nwkNeighborTblSz`)
- Network Routing Table (`nwkRouteTblSz`)
- Network Address Map Table (`nwkAddrMapTblSz`)
- Network Broadcast Transaction Table (`nwkBttSz`)
- Network Route Request Table (`nwkRReqSz`),
- APS Link Key Table (`apsPeerLinkKeyTblSz`)

`nwkBttSz` and `nwkRReqSz` should be set to zero (`0`) to ensure their respective default values are used.

For the remaining configuration parameters, the following table describes the impact on memory usage for each additional entry.

Description	Each Entry (bytes)
`nwkNeighborTblSz`	128

Description	Each Entry (bytes)
`nwkRouteTblSz`	28
`nwkAddrMapTblSz`	24
`apsPeerLinkKeyTblSz`	48

Table 1. Memory Usage Impact

4.1.3. Example Device Types

For the Coordinator device type, the number of devices essentially defines the maximum size of the network, if operating as a Centralized Trust Center. The limit is a result of the need to maintain individual unique link keys for each device in the network.

For both the Coordinator and Router device types, the number of neighbors or children the device can support is dependent on the size of the Network Neighbor Table (NNT). The NNT must be large enough to maintain the other Routers it is in direct contact with in order to route packets through the network, and also large enough to maintain any potential End-Device children which want to join to it.

Due to the fact Router-capable devices have applications that typically only send messages to the Coordinator and a few other devices in the network, they only need a small number of APS Link Key Table entries. If your application expects to communicate directly with a larger number of nodes, then the APS Link Key Table size should be

increased. Additionally, the Network Address Map Table, which translates an extended address to a network short address, should have an equal size to the number of required nodes because the application may reference other nodes in the network by their globally unique extended address (EUI-64).
In the case of the End-Device type, the neighbor table is used during network discovery to find potential parents to attempt to join to. The routing table size can be set to zero (`0`) since it only "routes" packets through its parent device. An end-device application typically only sends to the Coordinator, so it needs only a small Network Address Map and APS Link Key Table. Adjust this depending on your application requirements.
The following table provides a summary of example device configurations, with typical table sizes and memory usage.

Description	nwkNeighborTblSz	nwkRouteTblSz	nwkAddrMapTblSz	apsPeerLinkKeyTblSz	Heap kB	Total Memory (kB)*
Coordinator, 30 Devices, 30 Neighbors / Children	30	30	30	30	32	54
Coordinator, 100 Devices, 40 Neighbors / Children	40	30	100	100	32	60
Coordinator, 200 Devices, 40 Neighbors / Children	40	30	200	200	32	67
Coordinator, 1000 Devices, 40 Neighbors / Children	40	30	1000	1000	32	123
Router, 30 Neighbors / Children	30	30	4	4	32	52
Router, 100 Neighbors / Children	100	30	4	4	32	61
End-Device, 8 Neighbors	8	0	2	2	8	14.5

Table 2. Device Configurations and Memory Usage

* Total Memory includes some items with constant overhead, which are not listed in this table.