Mach 3.0 IPC Internals

Richard P. Draves

Outline

- Review
- Modules
- Control flow
- Data structures

Mach IPC Abstractions

Message

- Fixed-length header
- Variable amount of typed data

Port

- Protected communication channel
- Accessed via capabilities (port rights)

Port set

- Receive from multiple ports

Messages

Message header:

- Destination, reply ports
- Message size, id

Message body:

- Can hold port rights
- Can hold out-of-line memory

Ports

Protected message queues.

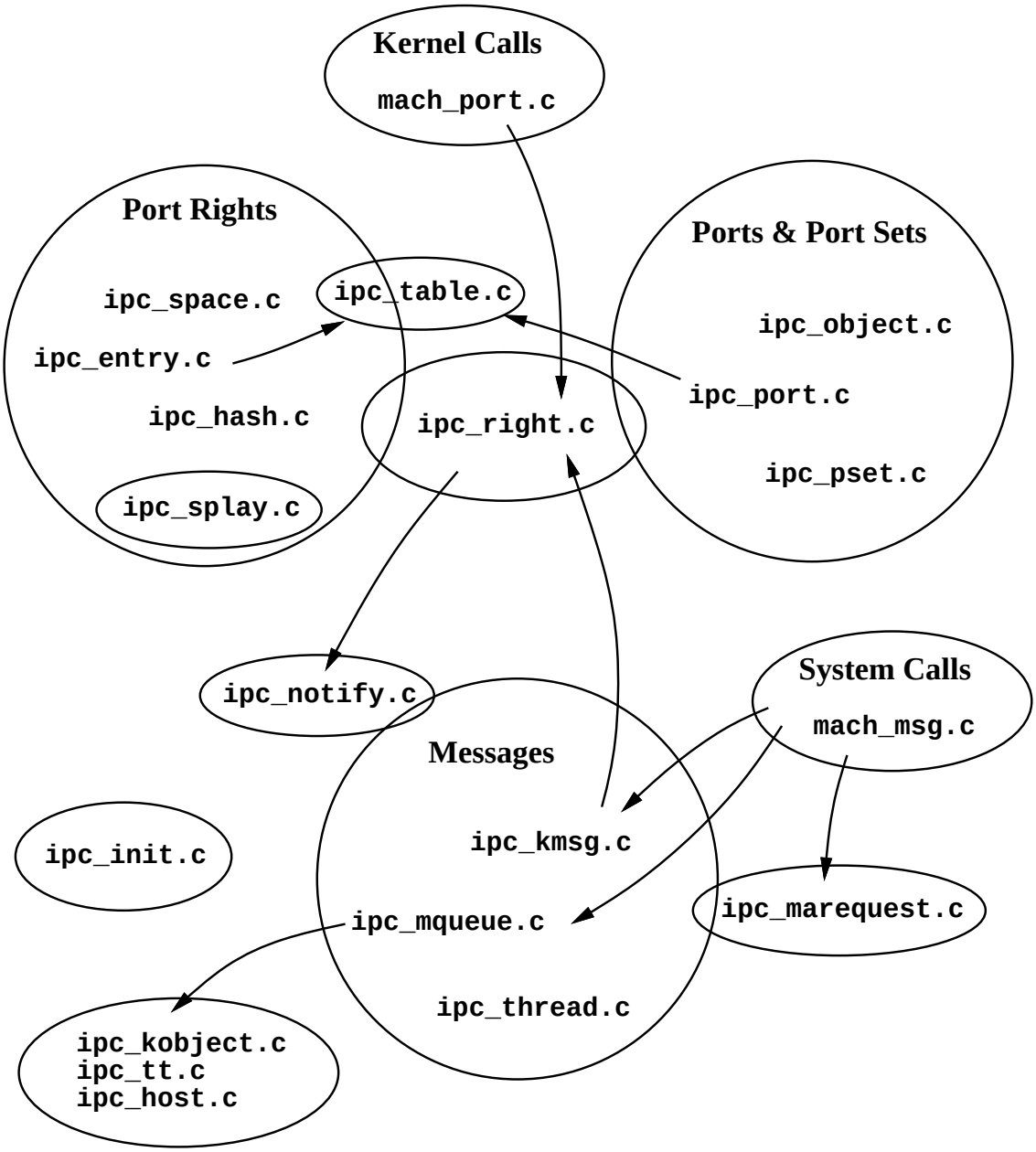
Tasks have a port name space.

- Receive rights
- Send rights
- Send-once rights
- Dead names
- Port sets

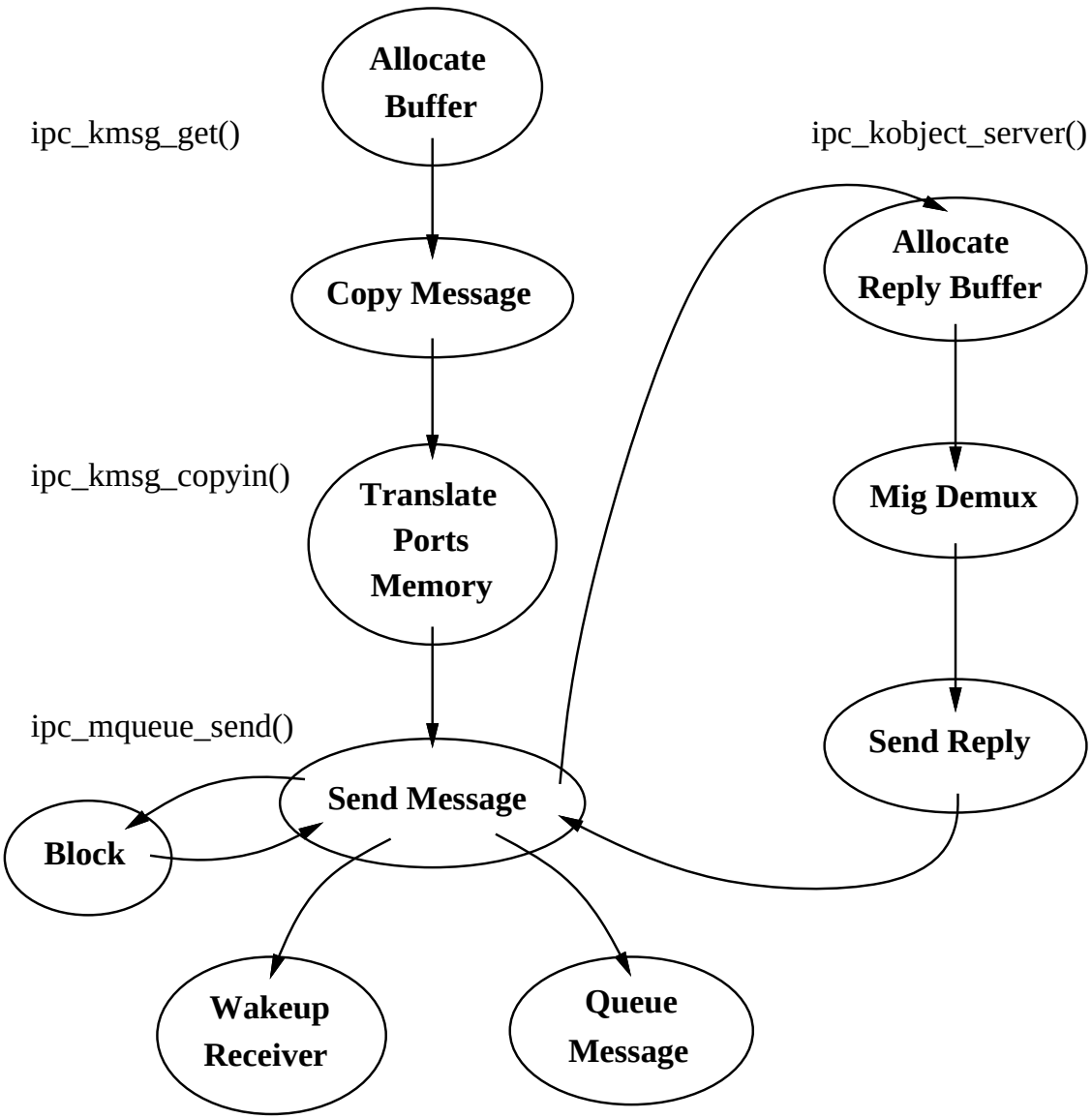
Port Sets

- Allow a server to receive messages from multiple ports
- Port may not belong to two port sets
- Can't receive from a port in a port set
- Ports and port sets share name space

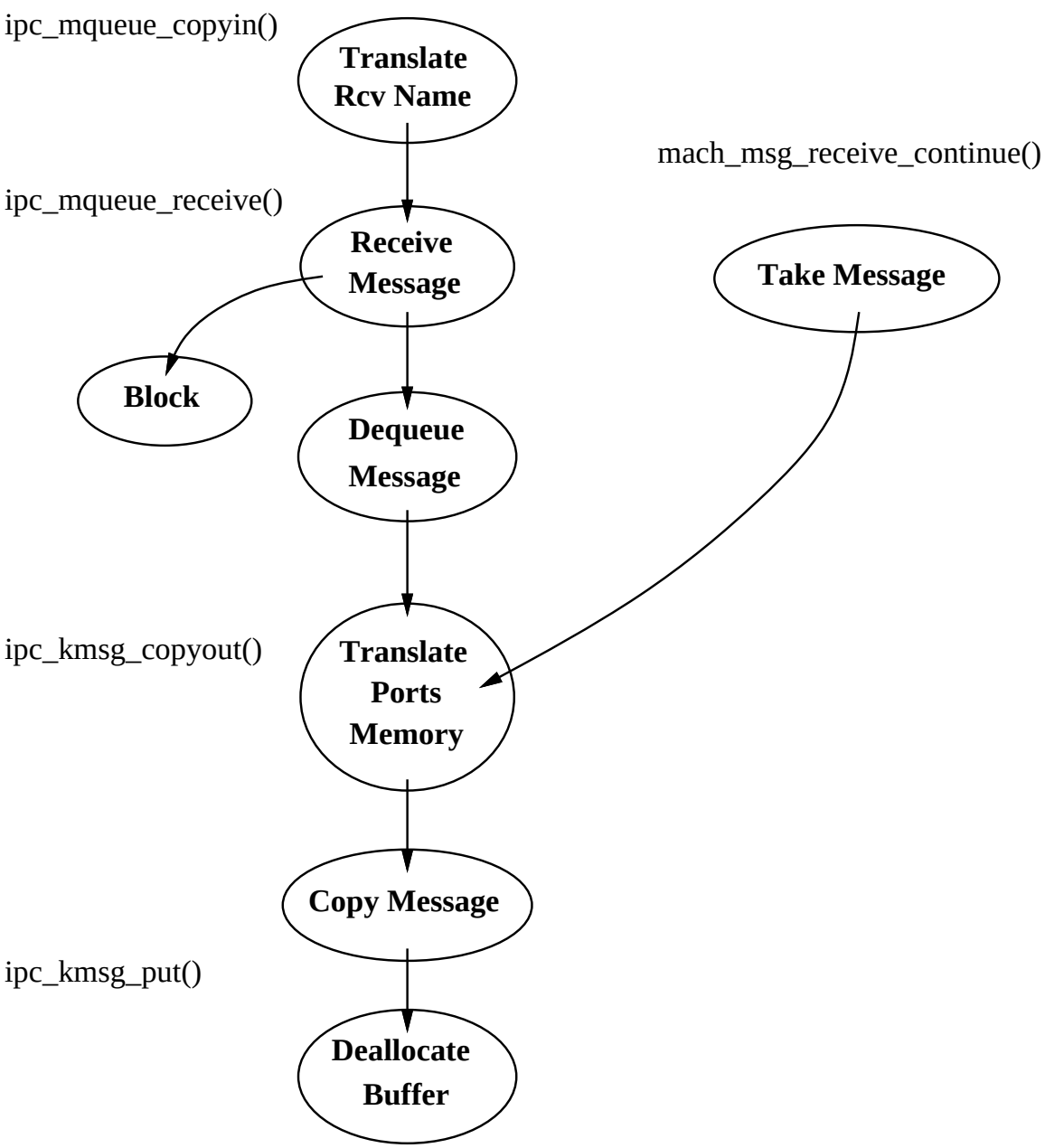
Modules


Kernel Calls
mach_port.c
Port Rights
ipc_space.c
ipc_table.c
ipc_entry.c
ipc_hash.c
ipc_splay.c
ipc_right.c
Ports & Port Sets
ipc_object.c
ipc_port.c
ipc_pset.c
ipc_notify.c
System Calls
mach_msg.c
Messages
ipc_init.c
ipc_kmsg.c
ipc_mqueue.c
ipc_marequest.c
ipc_thread.c
ipc_kobject.c
ipc_tt.c
ipc_host.c

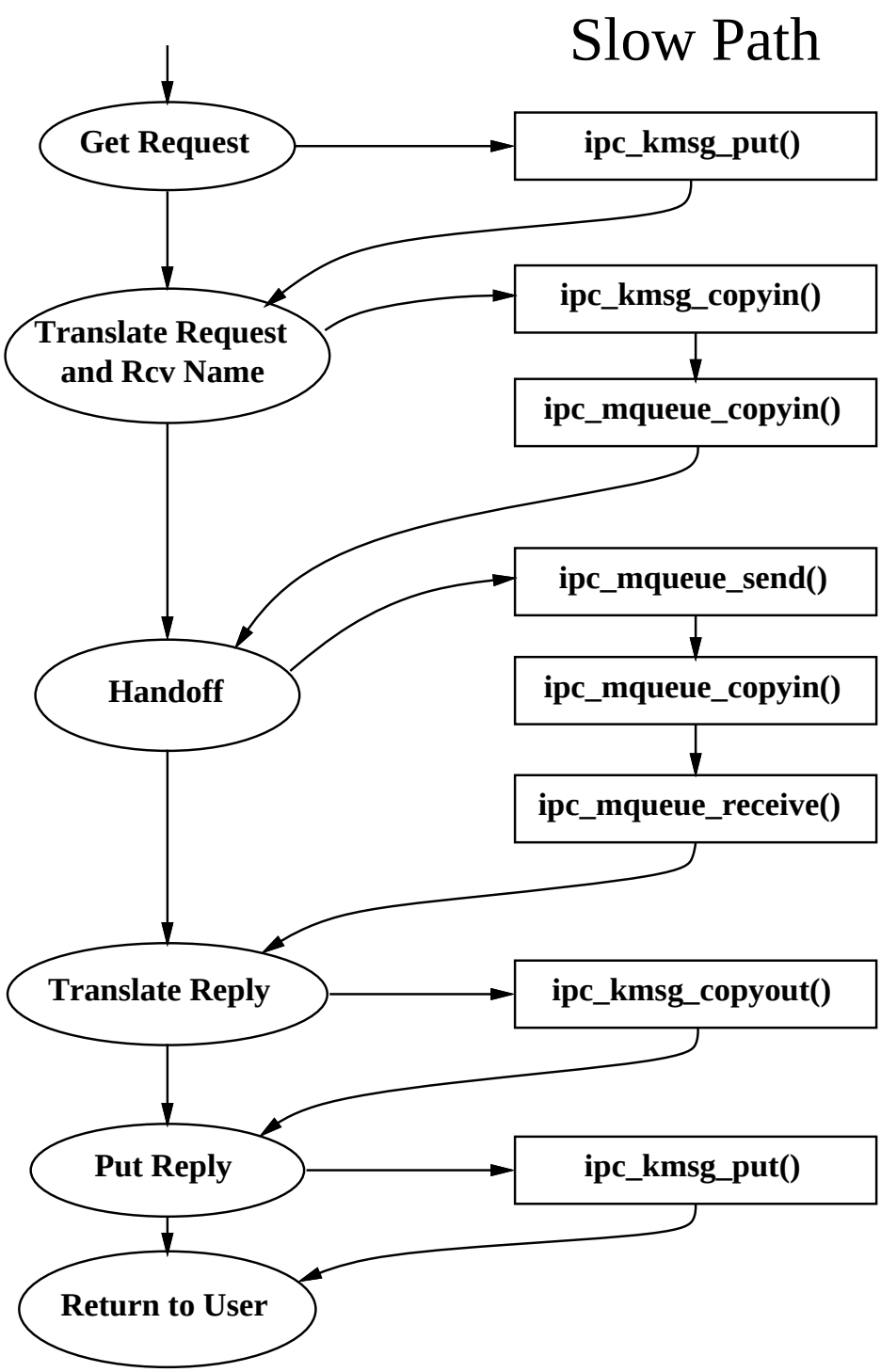
Sending Messages



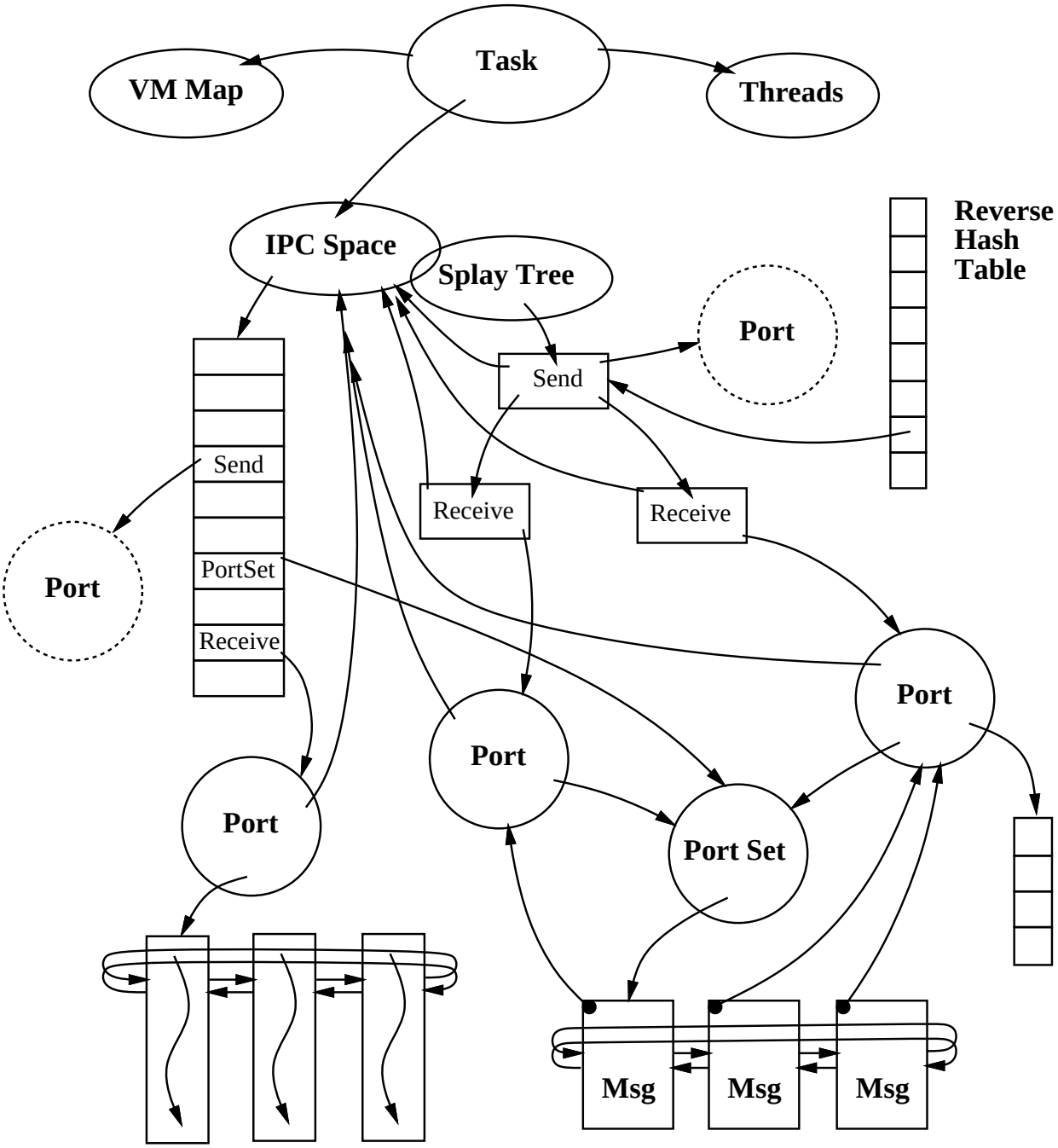
Receiving Messages



Combined Send/Receive



Data Structures



Enabling Interface Features

Interface and implementation designed together.

- Send-once rights:
  Reverse lookup not as important

- Dead names:
  Make reuse of names feasible
  Lazy port deallocation

Important Operations

- Given space and name, find port
- Given space and port, find name
- Allocate name
- Deallocate name
- Destroy space
- Destroy port

IPC Space

Represents a task's port name space.

Hybrid data structure:

- Table — 16 byte entries
- Splay tree — 32 byte entries

Table grown dynamically to minimize space.

Tables

- 8 bits — generation number
  1 bit — collision
  1 bit — compatibility
  1 bit — msg-accepted request
  5 bits — type
  16 bits — send right references

- object pointer

- next-free index or dead-name request index

- reverse-hash index

	type	object	next	hash
0	free		7	3
1	recv	80174cf4		
2	recv	80178480		
3	send	80178400		5
4	recv	80178510		
5	send	801784d0		6
6	send	8013cb54		
7	free		0	

Splay Trees

Self-adjusting binary search trees.

- Space efficient: two pointers per entry.
- Amortized logarithmic time.
- Recursion not needed.

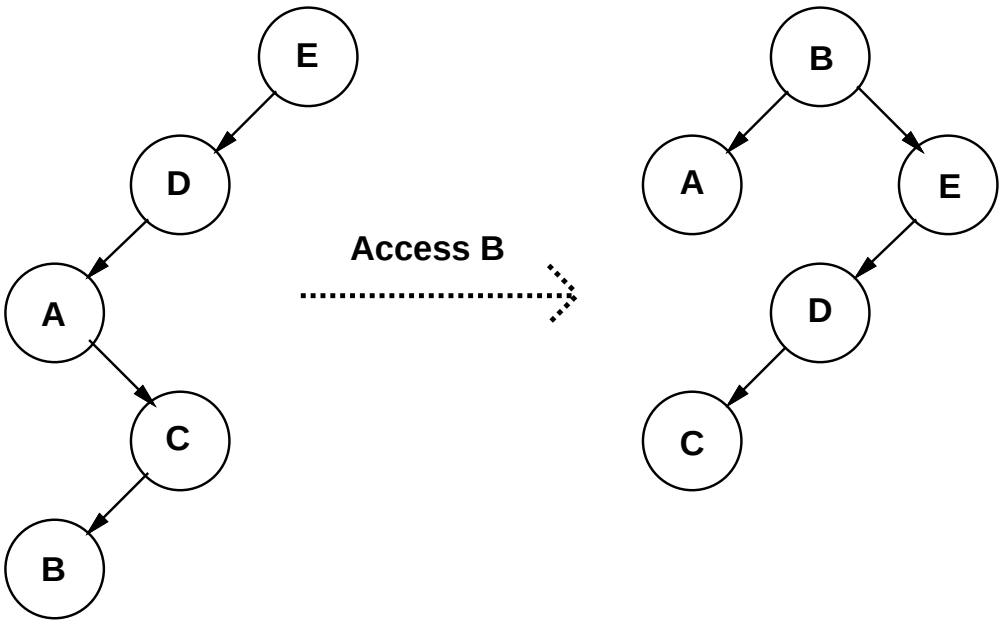
Important operations:

- Lookup, insert, delete.
- Split, join, traversal.

Extra fields:

- Left and right child pointers
- Space and name

Splaying



Garbage Collection

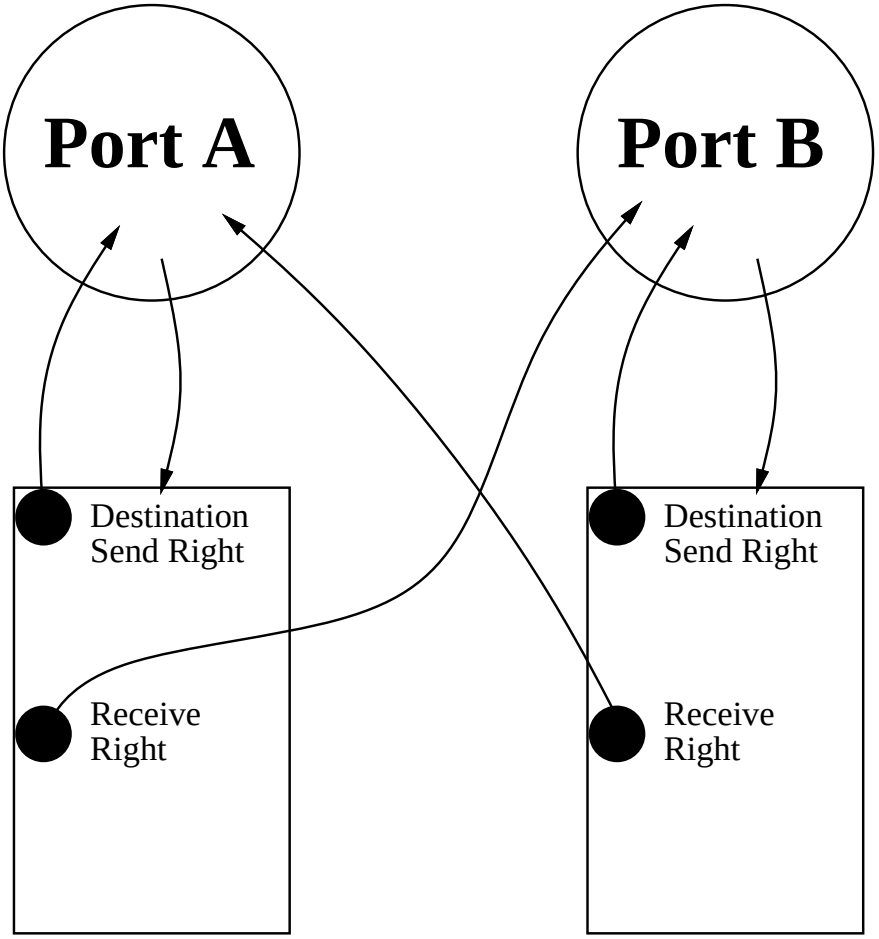
Reference counts

- Objects (ports and port sets)
- Spaces

Port rights, messages hold references

Alive/dead indication for atomicity

Message Circularity



Circularity Detection

Loops might develop when messages carrying receive rights are queued.

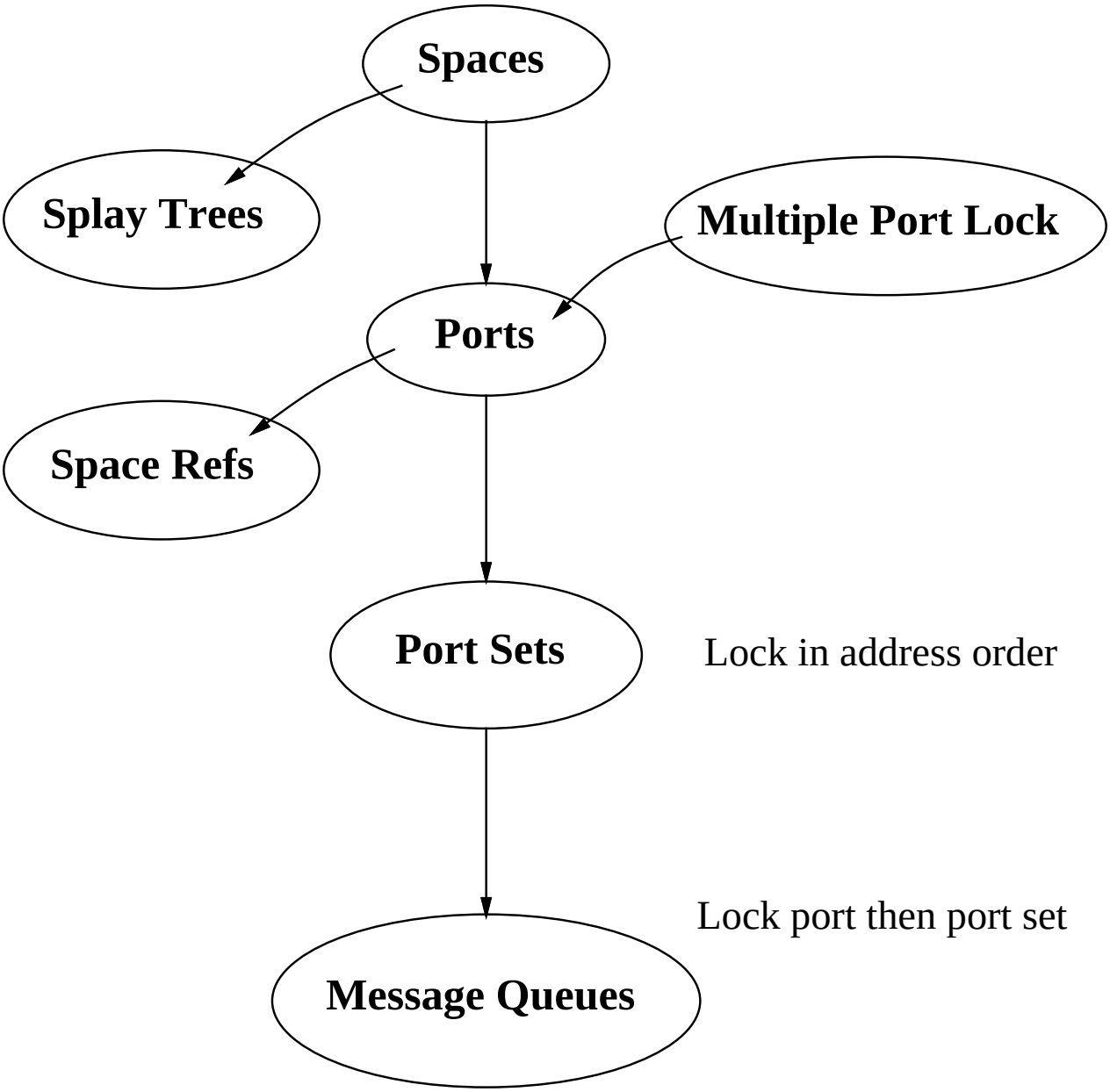
Solution:

- Ports in transit reference the port to which they are queued.
- Detect circularities in `ipc_kmsg_copyin`.
- Destroy messages in `ipc_mqueue_send`.
- Use global lock to lock multiple ports.

Locks

- Simple locks only
- No interrupt protection
- Read/write macros for spaces
- Memory allocation difficult
- Lookup functions
  Action functions

Lock Hierarchy



Message Buffers

- Unbounded size
- `kalloc`, `kfree`
- Per-processor cache
  - No locking
  - 256 byte buffers

Resource Exhaustion

No panics.

No quotas.

Problems during message receive.

Recursion Avoidance

Destroying a receive right destroys the queued messages.

Destroying a message destroys the rights it carries.

Solution:

- Recursive algorithm, except
- Curtail recursion with queue of messages.

Splay tree lookup, traverse