The MSU UDP Implementation of PktWay (MsgWay)

IETF PktWay (MsgWay) WG

June 24, 1996

Thomas P. McMahon

Overview

- Targeted PktWay Issues
- Overview of the MSU PktWay API
- General PktWay Design
- UDP Implementation Design
- Input and Output processes
- Reusability Issues and Future Work

PktWay Issues We Target

- High Performance, 0 copy (wherever possible), low latency
- Efficient memory usage and management (for possible ports to embedded systems)
- Demonstration of feasibility and usefulness of PktWay
- Accommodation of efficient layering of MPI, RDP, etc. on top of PktWay
- The UDP implementation is only a starting point, existing as a proof of principle -- high performance implementations will follow later

MSU PW Send API

`PW_Post_send`
(dest, PT, TE, *buf, len, is_L3, &request)

`PW_Persistent_send`
(dest, PT, TE, *buf, len, is_L3, &request)

`PW_Multi_send`
(dest, PT, TE, **bufs, num_bufs, len, is_L3, &request)

`PW_Get_info`	(info, request)
`PW_Wait`	(request, status)
`PW_Cancel`	(request)
`PW_Test`	(request)

MSU PW Receive API

`PW_Post_recv`
(src, PT, TE, *buf, len, &request)

`PW_Persistent_recv`
(src, PT, TE, *buf, len, &request)

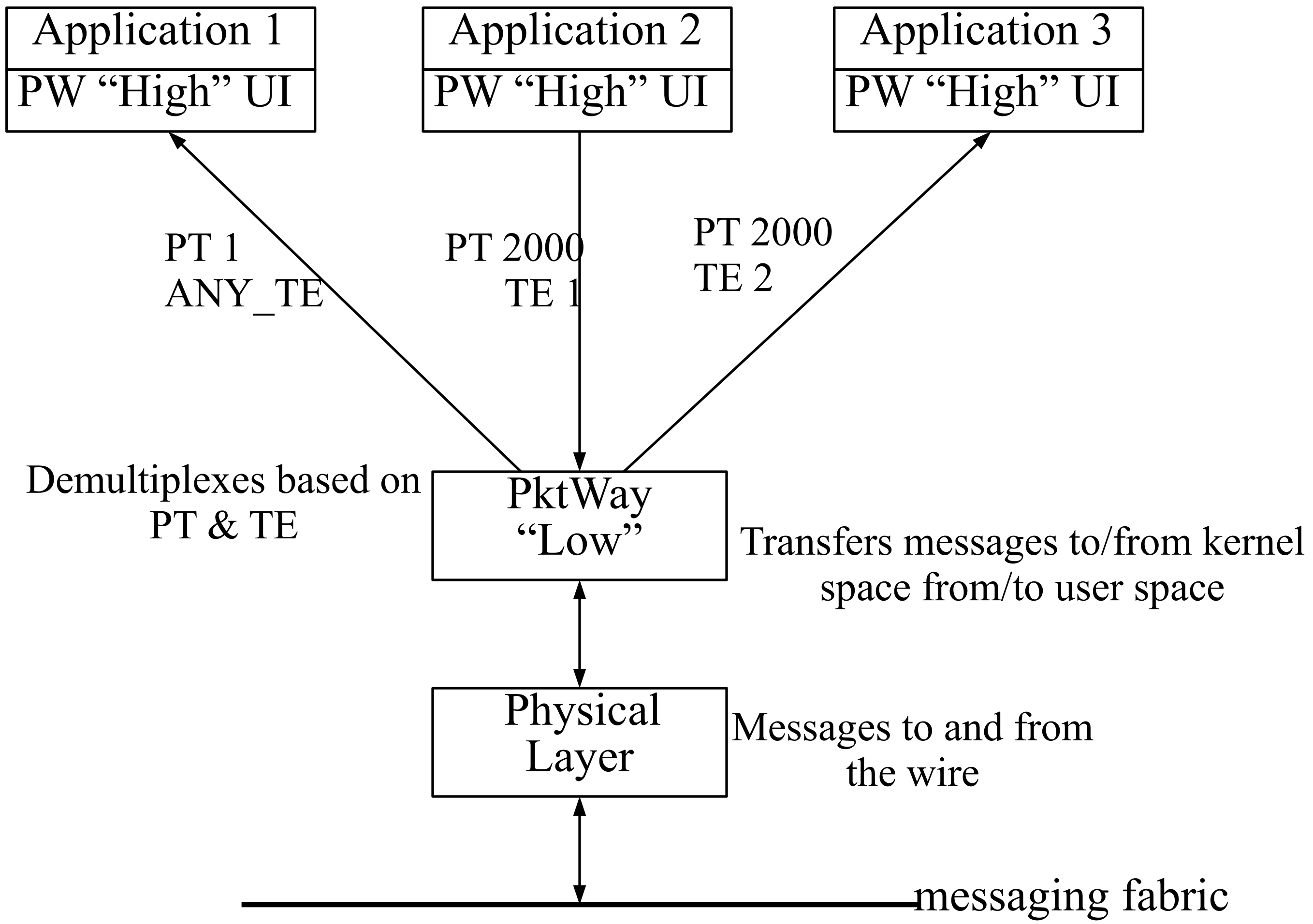
`PW_Multi_recv`
(src, PT, TE, **bufs, num_bufs, len, &request)

`PW_Wait`	(request, &status)
`PW_Cancel`	(request)
`PW_Test`	(request)

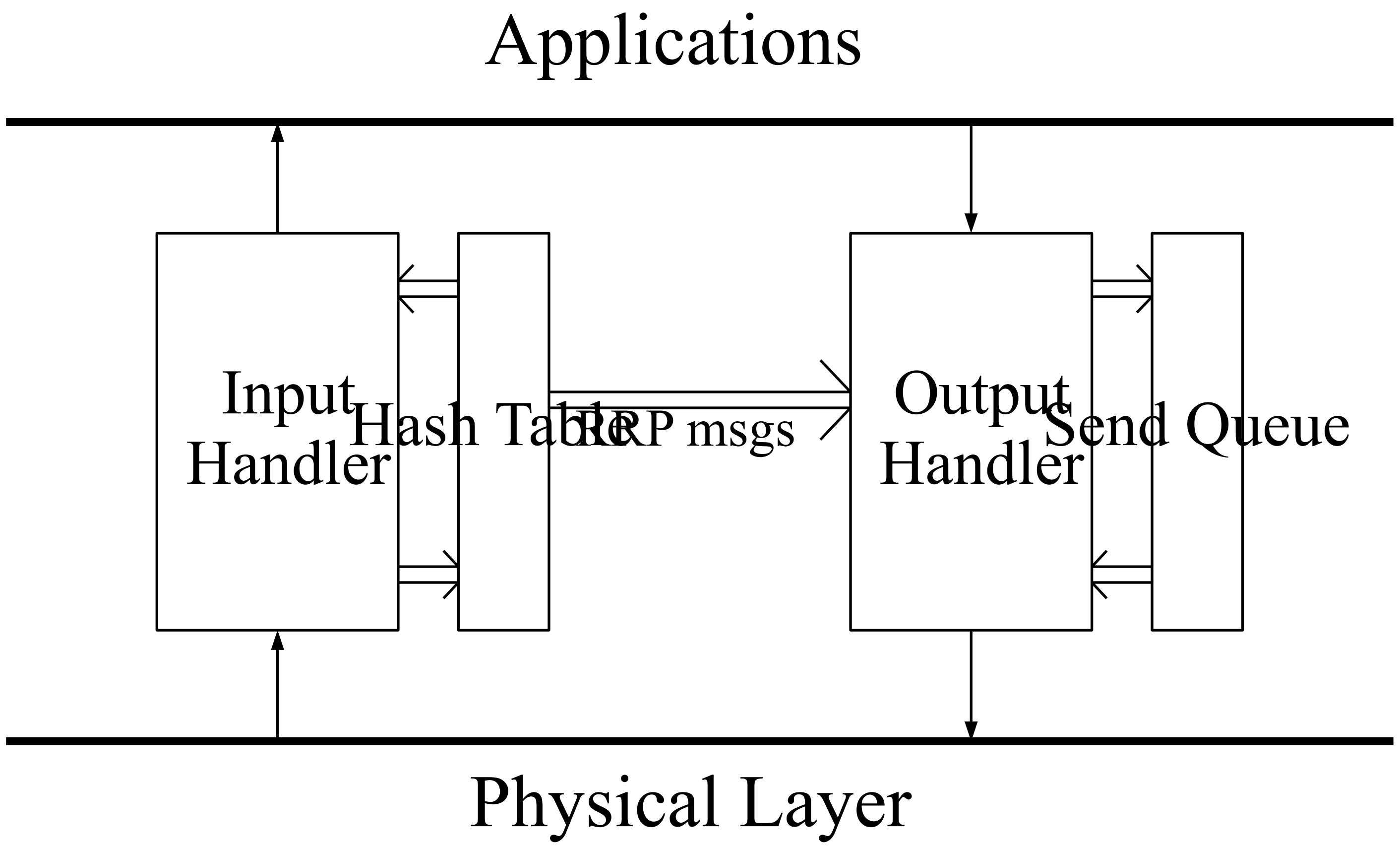
Overview of Design

- PktWay User Interface (PW "High")
- PktWay "Low"
  - Output Handler
    - handles send requests of all applications
    - handles incoming and outgoing RRP messages
  - Input Handler
    - handles receive requests of all applications
    - matches incoming PW messages with receive requests

General PktWay Design



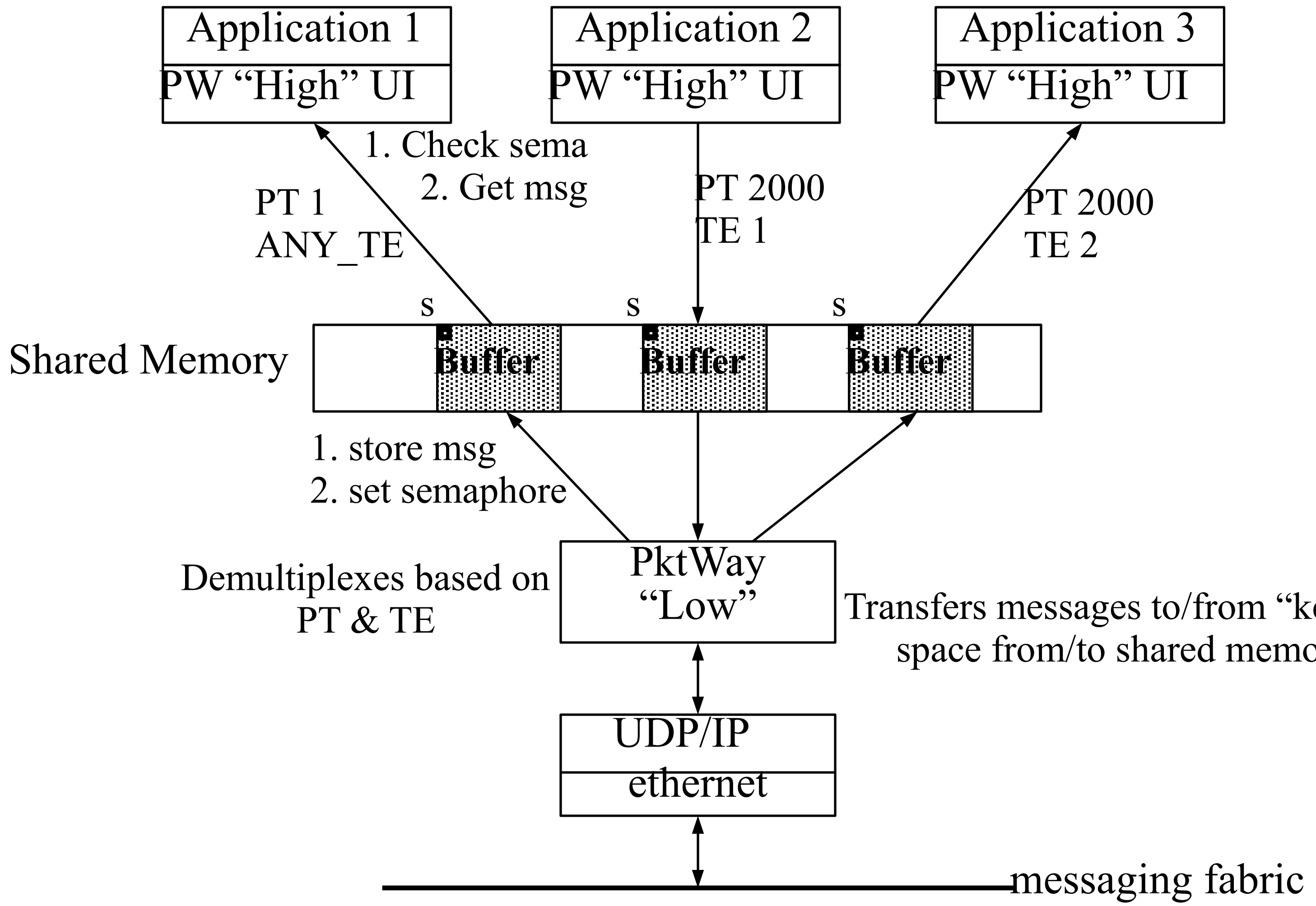
The Structure of PW “Low”



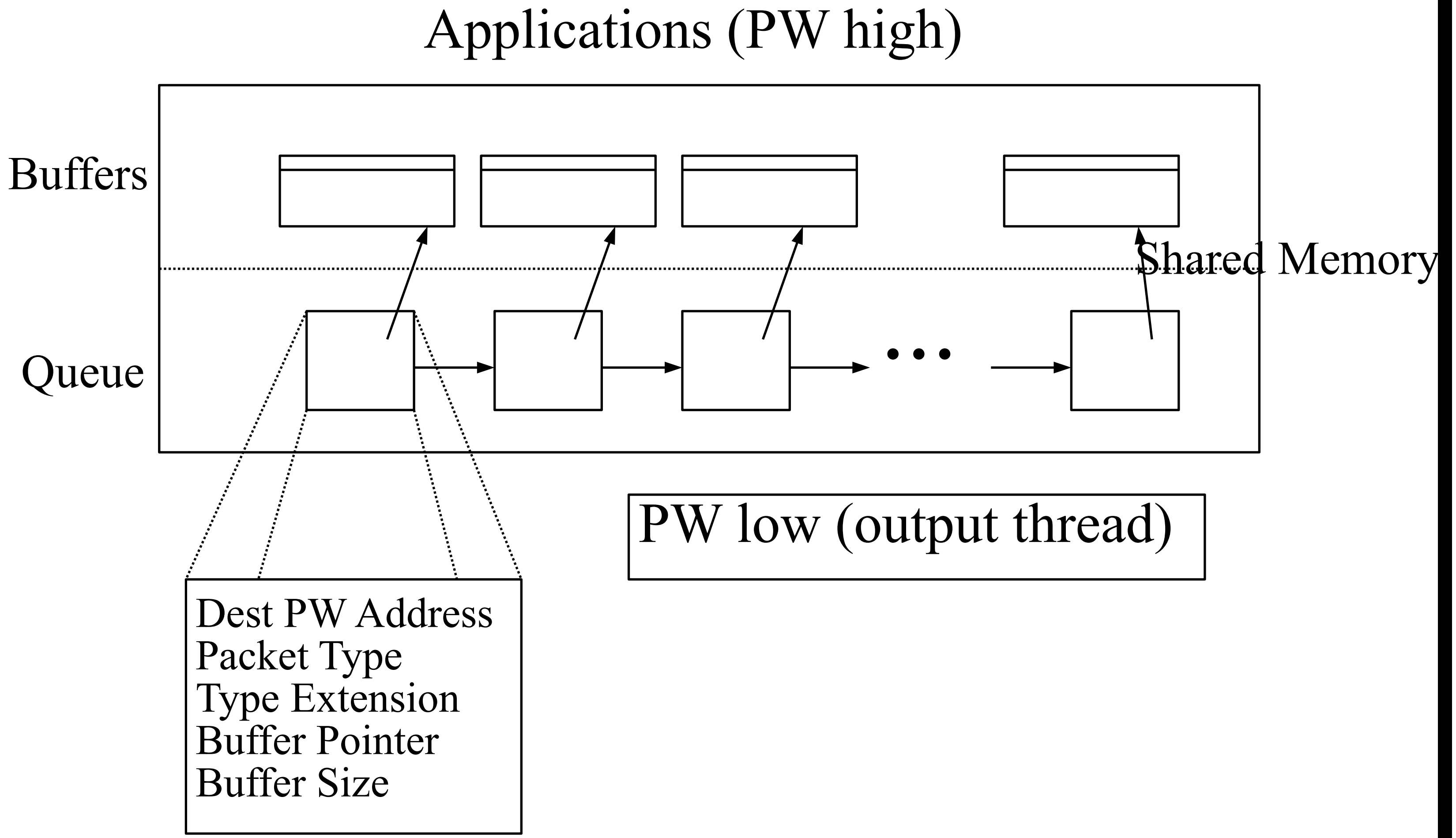
UDP Modifications to General Design

- Shared Memory Region
  - common communication region between multiple applications and single PW “Low”
  - holds memory buffers for messages
  - holds send and receive requests
- Parts of PW “Low” functionality moved into the User Level PW code
  - inserting user send and receive requests into PW “Low” request management data structures
  - moving data to/from user memory

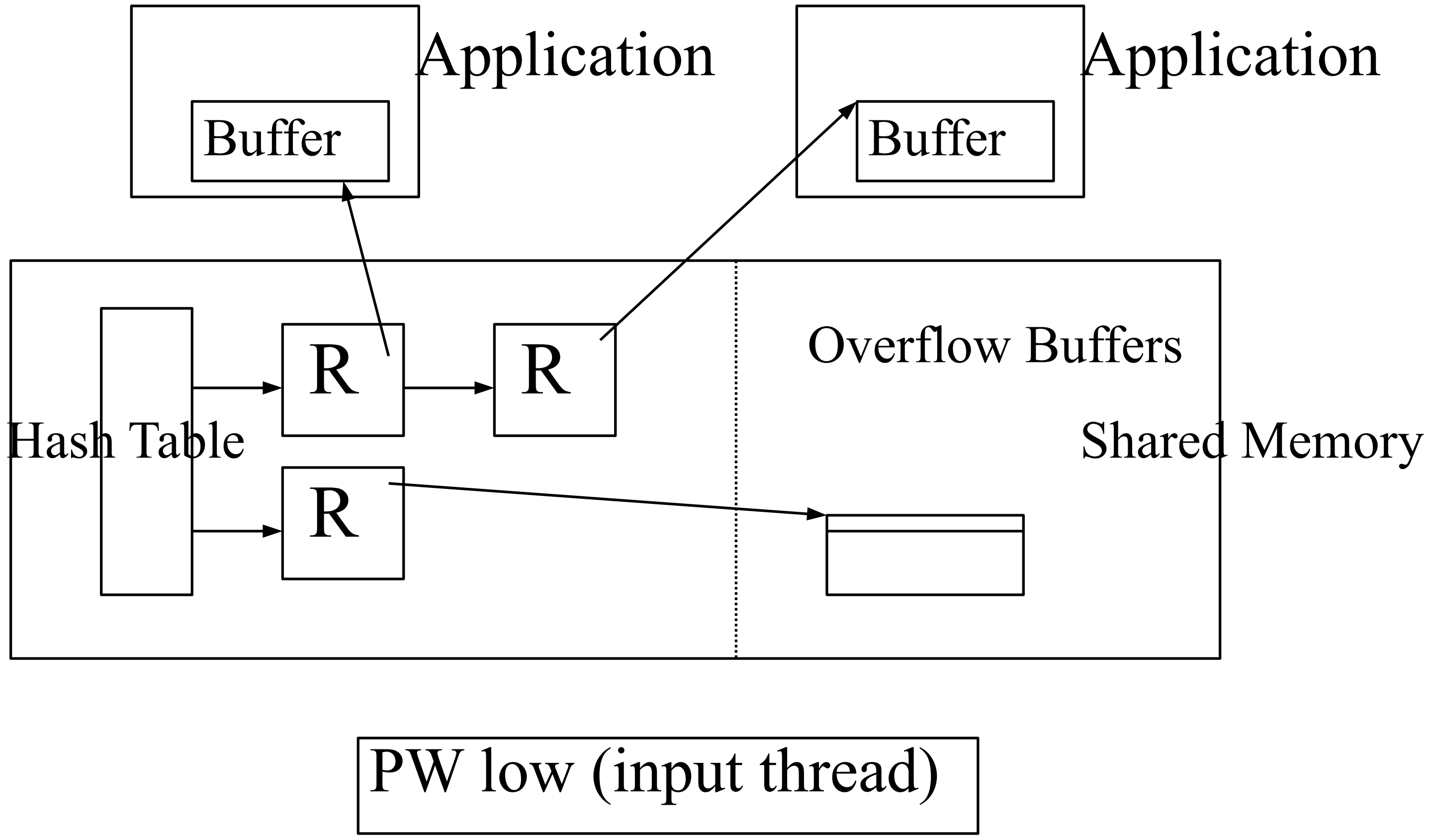
MSU UDP PktWay Design



UDP PW Send Queue



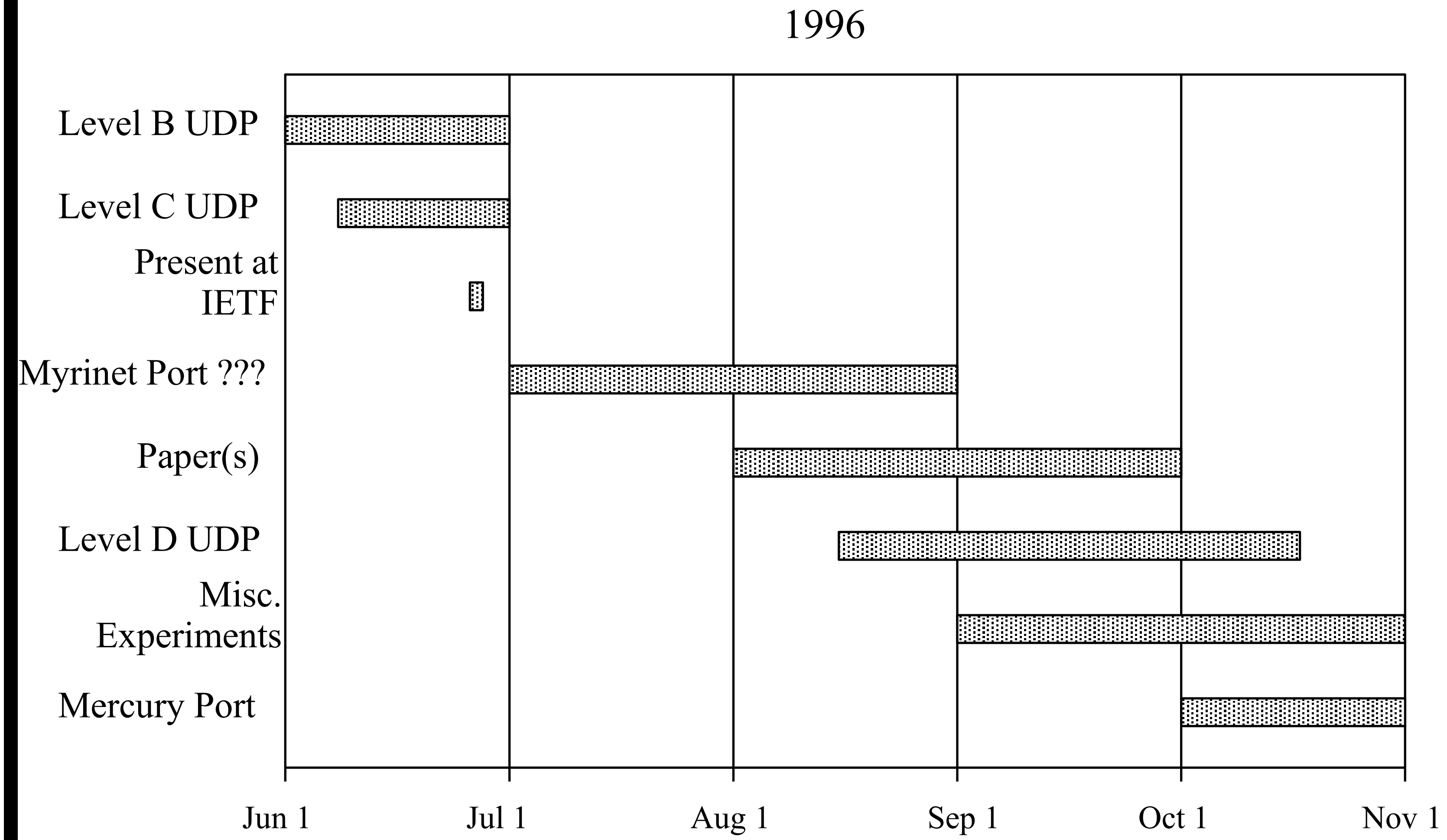
UDP PW Receive Hash Table



Reusable Code for Future Endeavors

- Most modules have been written so that they are not dependent on existence of shared memory:
  - Request management
  - Send Queue management
  - Hash Table management
- Thus, most of the PW “Low” code is reusable in future high performance PW implementations
- Some of the User API code can be reused (the code to transfer data to/from shared memory will be replaced)

Optimistic Time Chart


Level B UDP
Level C UDP
Present at IETF
Myrinet Port ???
Paper(s)
Level D UDP
Misc. Experiments
Mercury Port
Jun 1
Jul 1
Aug 1
Sep 1
Oct 1
Nov 1