



Embedded XINU – Network Analysis

Student: Jason Cowdy Mentor: Dennis Brylow

Department of Mathematics, Statistics, and Computer Science



Background

Embedded XINU^[1] is an ongoing research project in the areas of operating systems and embedded systems. The latest release of XINU 2.0 contains support for various network protocols such as ARP, TCP, UDP. The goal of this project is to test and improve the stability of XINU's new network features and create useful documentation for the next version of XINU.

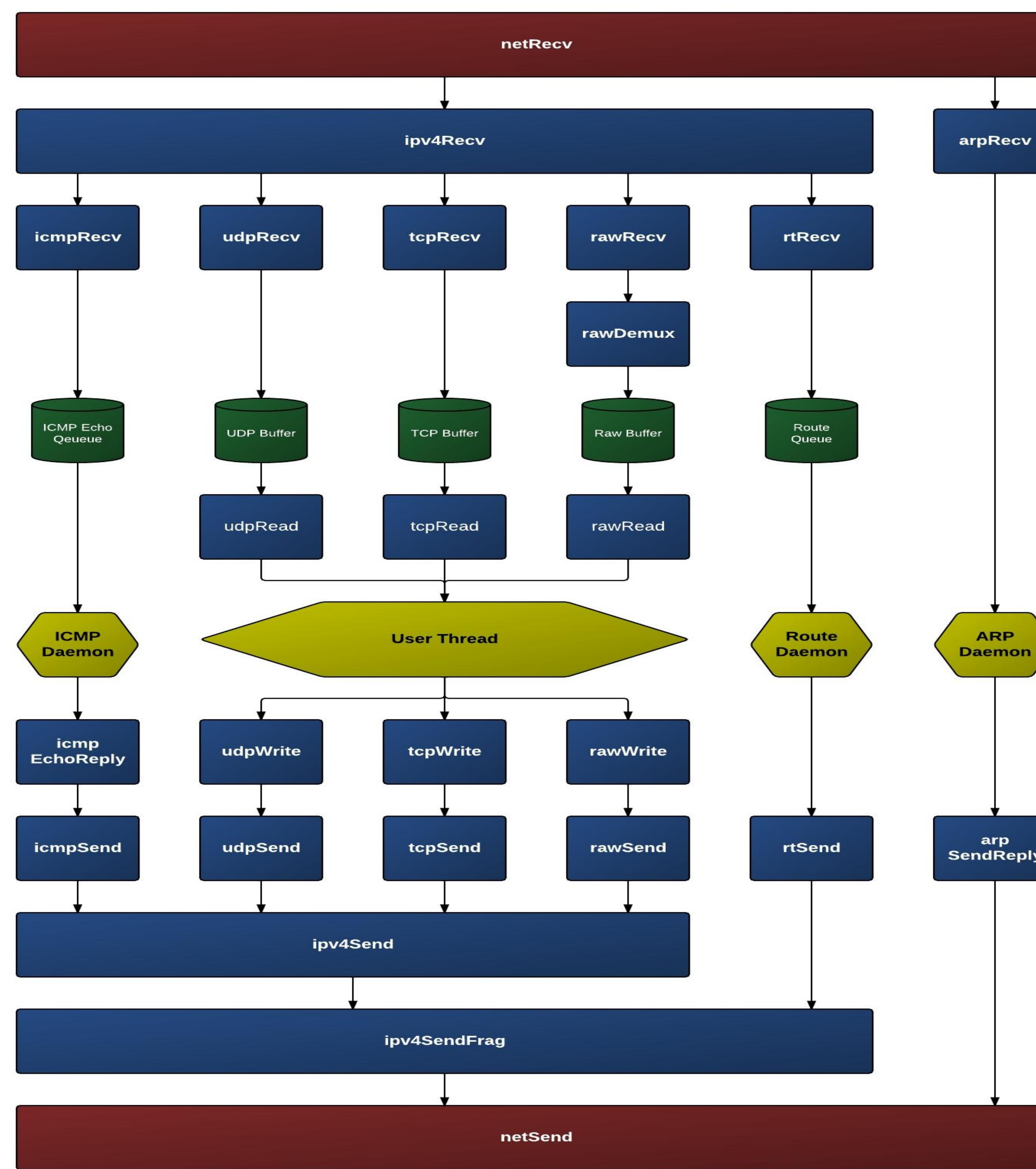
Current Methods

XINU is currently tested using a combination of both internal and external testing. Nightly builds are checked out and tested on each platform through the XEST^[2] automated testing suite. External machines and manual testing are used in some more complex cases that are hard to replicate on a single device. Little documentation exists on how to implement network features.

Proposed Solutions

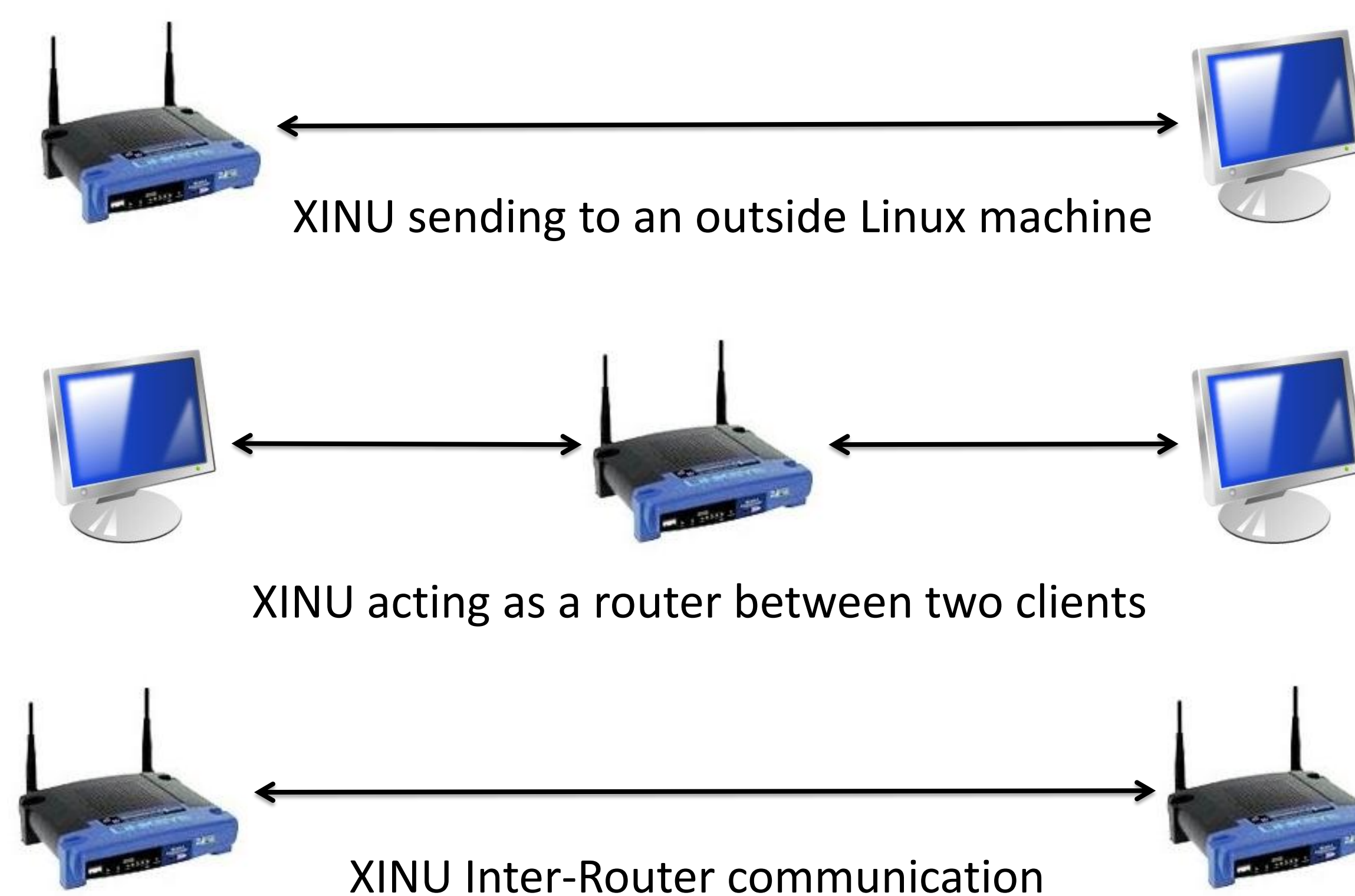
- Determine key network processes that require further testing
- Test each module of the stack in isolation
- Test modules in combination with each other
- Complete automated tests for nightly builds
- Create documentation for new features
- Identify and fix bugs for the next XINU release

XINU Network Stack



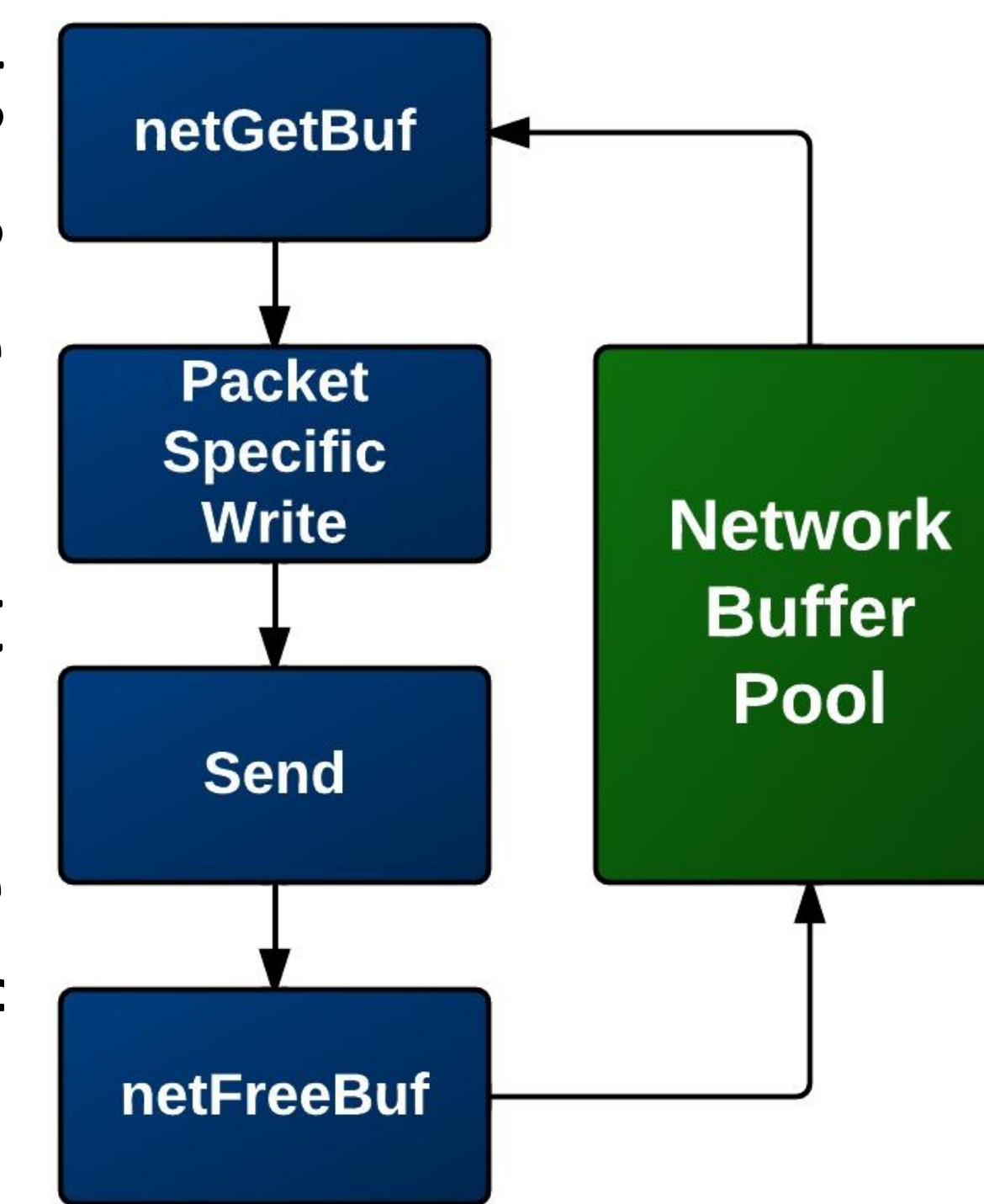
Incoming packets trigger the netRecv thread which calls a series of successive receive functions which process the packet at each level of the stack. Sending packets works in a similar fashion in which the user thread calls a specific write command which in turn calls its parent function until netSend is called.

Test Configurations



Memory Optimization

A global buffer pool of packets exists for storing outgoing packets. A process obtains a buffer from the pool, calls its parent function and then returns the packet once it has been sent. Through testing we have optimized the tradeoff between memory and network performance.



Conclusions

- Network buffer pool allocations can be smaller, leading to a significant savings in memory.
- XINU is able to both send and receive at the maximum transmission rates supported by its hardware.
- Numerous network bugs have been identified and squashed.

Future Work

- Expand automated testing to all supported platforms.
- Perform network tests on new platforms
- Implement more complex protocols based on current features

Resources

- [1] Dennis Brylow, An experimental laboratory environment for teaching embedded operating systems, SIGCSE 2008
- [2] Matthew H. Netkow, Dennis Brylow, Xest: an automated framework for regression testing of embedded software, WESE 2010
- [3] Marquette University ,Embedded XINU – <http://xinu.ms.cs.mu.edu>