David Choffnes

Supporting Parallel Applications on Clusters of Workstations: The Intelligent Network Interface Approach
M.-C. Rosu, K. Schwan, R. Fujimoto (Ga. Tech)

One-liner: The authors present a loadable kernel module combined with a user library interface to a virtualized NI to provide high performance communication specifically tailored towards NIs with coprocessors.

Although the goal of this paper is similar to U-Net's goal, the authors take a different tack for addressing network performance by 1) targeting the same class of applications, but using only commodity hardware and 2) trying to do as much as possible to take advantage of the coprocessor on the NI. The "Intelligent Network Interface" (INI) approach distinguishes itself from other related work by making the NI have an "active" role in the system, meaning that it allows the coprocessor to do memory operations and communication-related processing. In addition, the INI allows the user to manage how and when to pin pages to memory and supports multiple NIs, something not supported in U-Net. I was happy to see that the authors seemed to invest more time into addressing protection in the INI approach. Finally, the authors show that in many cases, their approach leads to improved performance.

My major problem with this paper is that the graphs don't show me anything that convinces me why this is better than U-Net. In fact, no direct comparison between the approaches in comparable environments is attempted. In this paper, the authors seem to rely on the NI's coprocessor, but in the U-Net paper, the authors said that it was a limiting factor. I'm not sure why it's any better here, unless the passage of time led to improved implementation of the same coprocessor. If I'm wearing my exokernel hat, I also have to say that I'm not comfortable with the LKM approach; it isn't strictly necessary to have a component loaded into the kernel (if the kernel is designed properly).

The impact of this work today is hard to say. It seems like the authors were thinking about the impact of their work on a future generation of NI hardware that contained more processing power and a faster I/O bus. A lot of the arguments for using an "intelligent" NI make perfect sense, much like offloading graphics processing to a GPU makes sense. Beyond the improved memory management and support for multiple NIs, I'm not sure what the major contribution of this paper is.