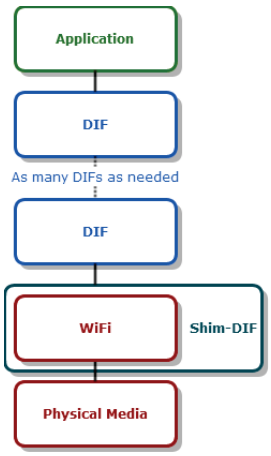
Abstract

The current version of the Internet is a jungle of protocols. Generic protocols interfere with each other, modern protocols overlap with their outdated versions, old issues such as multihoming, layer boundaries, ... were never solved. It is clear we need a new architecture. Following the architecture from John Day, we opt to use the Recursive InterNetwork Architecture (RINA).

With the Internet being born out of ARPANET, we instantly notice several issues. ARPANET was a proof of concept, it was never a research net. This means it was never suitable to be the foundation of the current Internet. Several research networks were constructed but were discontinued when ARPANET was moving towards a public network. Several protocols were added to ARPANET and finally it stepped out of its infancy and moved towards a public net. None of the issues were solved though. Protocols were added on top of this, issues that rose during the early staged were never addressed, but the expansion continued anyway. RINA is an alternative architecture that offers an entire new architecture instead of one more protocol meant to patch one specific problem. RINA starts from the basic rule: ´´Networking is Inter Process Communication (IPC) and IPC only". Every IPC has IPC processes that wish to communicate with each other. These IPC processes are bundled in a Distributed IPC Facility (DIF). These DIFs are stacked recursively on top of each other. Every DIF has the same mechanism, but different scopes and policies make these DIFs different.

RINA is a theoretical architecture, without a technical implementation its use cannot be proven. The technical implementation we will be cooperating with is: "Investigating RINA as Alternative to TCP/IP", IRATI. This project aims to provide an implementation of RINA at the lowest possible layer that is currently feasible. While the lowest layer is simply the physical layer, for testing purposes this would prove to be fairly useless as this layer is not directly available in operating systems. This means we have to place the project's code right on top of the link layer. The current existing code already provides a

working Shim-DIF for Ethernet. A Shim-DIF is a DIF which uses the functionality from the lower layer (e.g. Ethernet) and presents this as a DIF to the upper RINA DIF. Note that these Shim-DIFs are not fully functional DIFs, but are needed to implement the code as low as possible. Our goal is to construct a Shim-DIF over WiFi on Android, based on the Shim-DIF over Ethernet on Debian Wheezy.



In Linux kernels and Linux-like kernels we remark that WiFi frames are being reconstructed as Ethernet frames. After this reconstruction they are presented to the kernel. This means that WiFi frames are inaccessible for the user, even the admin (root). In the future this could change or people who currently write the code for the devices which handle this reconstruction could be willing to add implementation for RINA. For this reason we write a specification which stipulates how the Shim-DIF over WiFi should be constructed. This implementation is specific for the IRATI project.

Our implementation on Android is composed through different approaches and a plethora of iterations. After the construction of the base kernel we attempt to add the IRATI specific code to the kernel, which proves to be impossible without drastic changes to the code. Other issues arise here as well, such as the addition of system calls to this Android kernel. Further issues such a userspace implementation, package installation, C library imcopatibility, . . . all added to the difficulty of implementation. With the current timeframe, manpower, and knowledge the success of implementation on Android fell outside the scope of this Master's Thesis.