# PAPER1 Report

z5089812, Xuan Tong

## 1.Summary

This thesis presents Antikernel, a novel operating system architecture. Antikernel system is highly modular and consists of many independent hardware state machines connected by a packet-switched network-on-chip (NoC), instead of a bus, it connectes the CPU to rest of the SoC with a network-like structure, give every node an 'IP address'. Antikernel is a decentralized architecture with no system calls, all OS functionality is accessed through message passing directly to the relevant service. Antikernel is composed of both hardware and software components, and designed to be fundamentally more secure than the state of the art.

Antikernel is said to be 'kernel on hardware', it mentions that having any software running in ring 0/privileged mode violates least required privilege principle, and almost no application will need all the services provided by modern kernel. Besides that, its design makes sure that there is no longer any authority that has access to everything, which is good for security.

Antikernel Network Architecture, an Antikernel-based system consists of a series of nodes organized in a quadtree5 and connected by a packet-switched NoC with 16-bit addressing. The network structure is indeed a pair of parallel networks. Serve for RPC and DMA. Data plane and control plane separated. It also provides reliable datagram, make sure instructions are arrive in FIFO order. It has guarantee to minimum QoS, so real time OS is possible. Lastly, each router is guaranteeing to have at least one fifth of the bandwidth, traffic between different nodes won't affect each other.

Memory management is achieved by using FIFO queue + ownership array to track free memory and the ownership of that memory, MMU provides translation, user application can dereference any address it wants to, and the access will be checked on physical address layer to see if it has corresponding ownership. It provides the ability for the individual node in the network to authenticate messages based on their origins, we can do access control on physical level in the individual devices.

SARATAGO CPU is a high-performance dual-issue in-order barrel processor using a modified version of the MIPS- 1 instruction set with an 8-stage pipeline11 and a parameterizable number of hardware threads. In addition to allocating one NoC address per hardware thread, the CPU has a dedicated management address used for out-of-band control functionality. Thread scheduling is based on linked-list.

Name Server, resolve 8 characters hostname to 16-bit addresses, mainly used for board portability.

Antikernel OS is good at security, due to it separates userspace applications from each other, also separates userspace application from OS features. It achieves damage control. Also, the current codes have been partially tested to be correct and safe.

## 2. Pros

The author successfully builds a prototype for both software and hardware, which is a huge project, this system is also tested and verified to some extent. And its modular structure is highly amenable to piecewise formal verification for future works.

The Antikernel system concludes that there is no 'all-powerful' application, and all the 'simple' application can benefit from this, since they can basically avoid some unnecessary overhead. Thus, there is no longer exists any single authority that has access to everything, which makes the system less susceptible to lose its control of 'privilege'.

It provides Name Server, which massively enhance the portability of programs running on Antikernel system. Names being registered by a random NoC node at run time, however, are not inherently trusted. In order to prevent malicious name registrations, the name server requires a cryptographic signature to be presented and validated before the name can be registered.

## 3. Cons

It is incompatible with existing operating system, and all userspace application have to be rewritten in order to run on Antikernel system.

From my perspective, I think Antikernel system only supports *single address space*, since it uses the mechanism is FIFO queue + ownership array to track free memory and the ownership of that memory, which makes the compiling work extremely hard, and the memory management becomes inflexible. Virtual address space becomes a scarce resource. Besides that, shared memory seems very hard in this case.

The RPC protocol will function over links with arbitrary latency (and thus register stages may be added at any point on a long link to improve timing), however a round-trip delay of more than one packet time will reduce throughput since the transmitter must block until an ACK arrives from the next-hop router before it can send the next packet.

## 4. Criticisms

It mentions that it allows userspace application to pick and choose the OS features they need, but didn't actually specify how it works, and for userland application, how to pick the features it needs.

For chap5.2, Excution Unit, EXEC is not well explained, I have no idea what's that is after reading the paragraph.

It didn’t do any benchmark to show how much the reduction of overhead for userland application contribute. Can’t get a rough idea of the performance of Antikernel syste. Besides that, it claims that part of system has been verified, but we can’t get a clue of how it is verified, what method it used. Lastly, it says to be tested to assure the correctness, however we don’t how is it tested, what methods they’ve used.

Merely MMU is used for address translation, the FIFO queue can only record available memory, the ownership array is only capable of authority checking, then the MMU would be extremely large, and responsible for address translation for all processes, it will become inefficient, and error-prone. However, the paper does not mention the flaws, make the memory management looks simple.

This paper mentions that each thread is hardware thread, and is NoC addressed, I spend some time to understand what is hardware, and still can’t find out how it is addressed. I think author should add more background to this part.

What is the router, how it is realized, this paper use the concept of router, but didn’t specify how it is implemented.

In chap6 security model, mentioned that Antikernel is designed to ensure that following are not possible given that an attacker has gained unprivileged code execution within the context of a userspace application or service, I think they’d better give out more detailed specification, and a concrete example would be better.

The trusted computing base is very large, `We have performed fairly extensive verification on the current prototype system using a mix of simulation, hardware-in-loop (HiL) testing on our test cluster, and formal methods.` is not convincing, and they should give more details about this.

NoC is already exists technique, but this paper spends a lot of efforts on this.

The cache is virtual addressed, if it transform to multiple address space design, this should be taken care of, it has potential problem for homonym and synonyms.