Sistema de comunicaciones seguras con segmentación virtual de dominios

Avance de proyecto

Alberto Daniel Lange

Dirección: Juan Ignacio Vaccarezza
Codirección: Santiago Pérez Ghiglia

Ingeniería en Telecomunicaciones
Instituto Balseiro

26 de febrero de 2025

Contexto

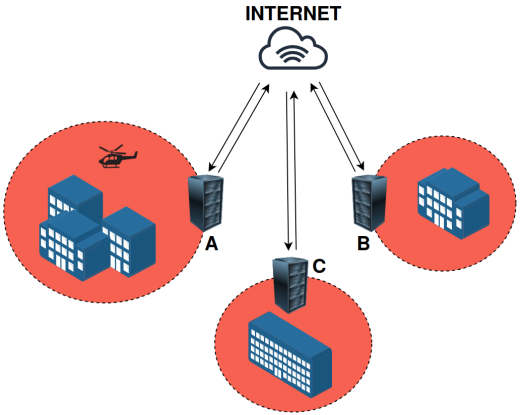

Figura 1: Esquema simplificado de operación del sistema.

Progreso alcanzado anteriormente

- Definición y validación de arquitectura lógica en laboratorio virtual.

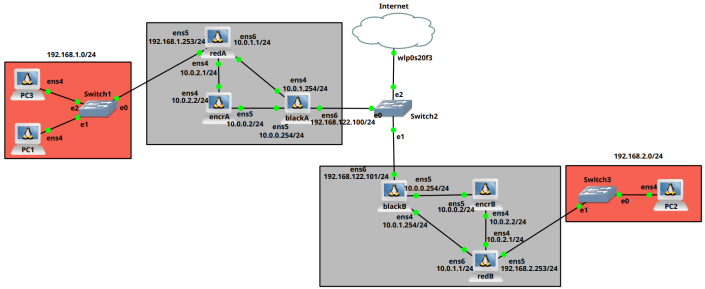

Figura 2: Simulación de la propuesta de solución en GNS3.

Pendientes

- Validación del encriptador en ambiente virtualizado.
- Implementación sobre hardware.

seL4

- Microkernel de código abierto.
- Hipervisor tipo 1.
- Aislamiento entre componentes garantizado.

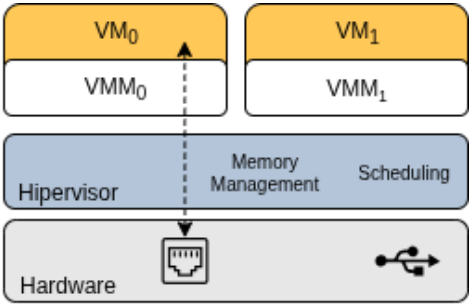

Figura 3: Esquema de entorno virtualizado usando seL4 como hipervisor.

CAmkES

- **C**omponent **A**rchitecture for **m**icro**k**ernel-based **E**mbedded **S**ystems.
- Framework de desarrollo para seL4.
- Comunicación mediante interfaces bien definidas.

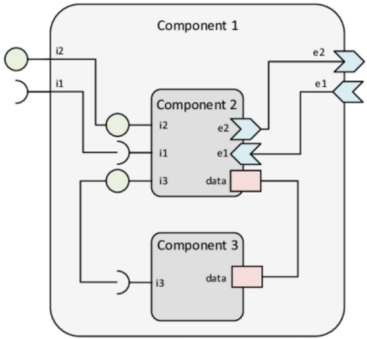

Figura 4: Ejemplo de componentes e interfaces en CAmkES.

Modelos ejemplo en CAmkES

minimal_64

- Implementación mínima de una VM para arquitectura x86_64.
- Punto de partida para el desarrollo.

zmq_samples

- Ejemplo de comunicación entre tres VMs usando ZeroMQ.
- Modelo base para la implementación del encriptador.

Estrategia de modelos en entornos virtualizados

¿Qué?

- Realizar modelos que validen progresivamente los componentes desarrollados.

¿Para qué?

- Ligar problemas concretos a cada modelo y resolverlos de forma independiente.
- Implementar un encriptador funcional en un entorno virtualizado como paso previo a trabajar con hardware.

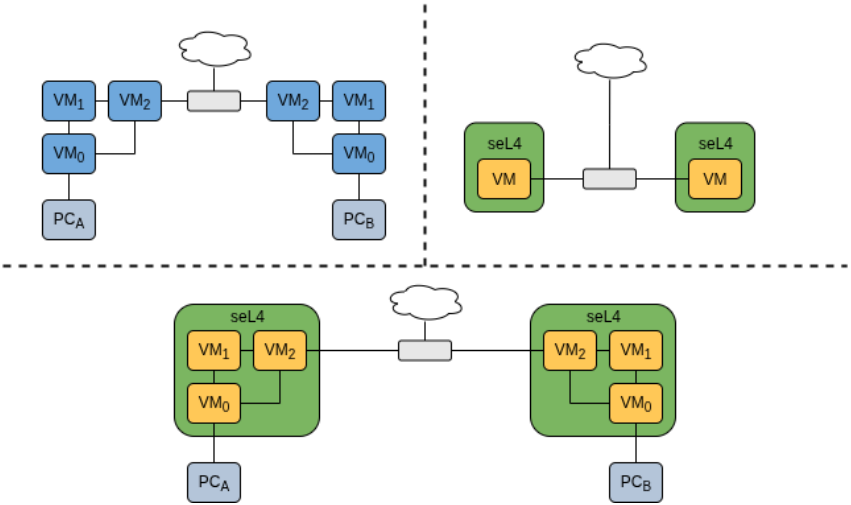

Figura 5: Modelos en entornos virtualizados.

Modelo virtualizado

- Validar la arquitectura lógica del encriptador.
- Funcionalidad *split-tunneling*.

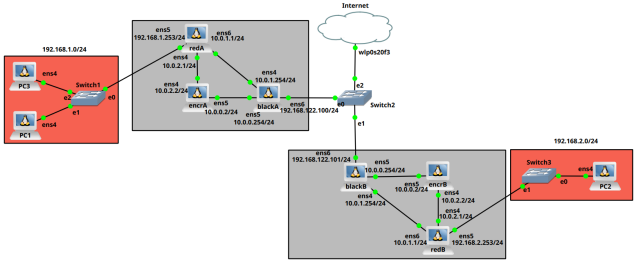

Figura 6: Simulación de la arquitectura lógica en GNS3.

Modelo emulado

- Validación del passthrough de hardware y la compatibilidad del kernel Linux 4.9 con seL4.

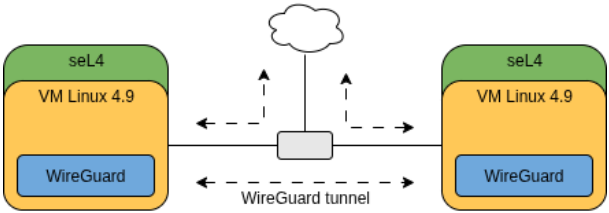

Figura 7: Esquema del modelo emulado.

Modelo emulado: passthrough de hardware

CAmkES actúa de interfaz para la configuración del passthrough de dispositivos PCI a una VM a través de los Base Address Registers (BARs) e interrupciones.

Ejemplo de configuración en CAmkES:

```
vm0.vm_ioports = [start, end];
vm0.pci_devices = [bus, device, function, memory];
vm0.vm_irqs = [source, dest];
```

Modelo emulado: gestión de memoria VMs

`simple_untypedN_pool`: prealocación de memoria para las VMs.
`heap_size`: tamaño del heap del VMM.
`guest_ram_mb`: tamaño de la RAM asignada a cada VM.

Modelo de hardware

- Solución completa como paso previo a la implementación en hardware.

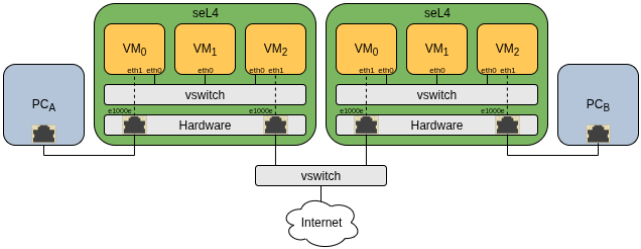

Figura 8: Esquema del modelo de hardware.

Modelo de hardware: comunicación entre VMs

¿Cómo funciona?

Virtio-Net en VM

- El host implementa un dispositivo PCI virtual (virtio-net).
- La VM encuentra este dispositivo y carga el driver correspondiente.
- En la transmisión, la VM escribe los datos a enviar en un buffer asociado a una cola del hipervisor. El VMM notifica al host que hay datos listos en la cola (virtqueue).

Virtio-Net en host

- El host lee el paquete en la virtqueue y lo copia a un buffer dentro del espacio de memoria de la VM destino.
- El host inyecta una interrupción en la VM destino para notificarle que hay datos disponibles.

Modelo de hardware: integración

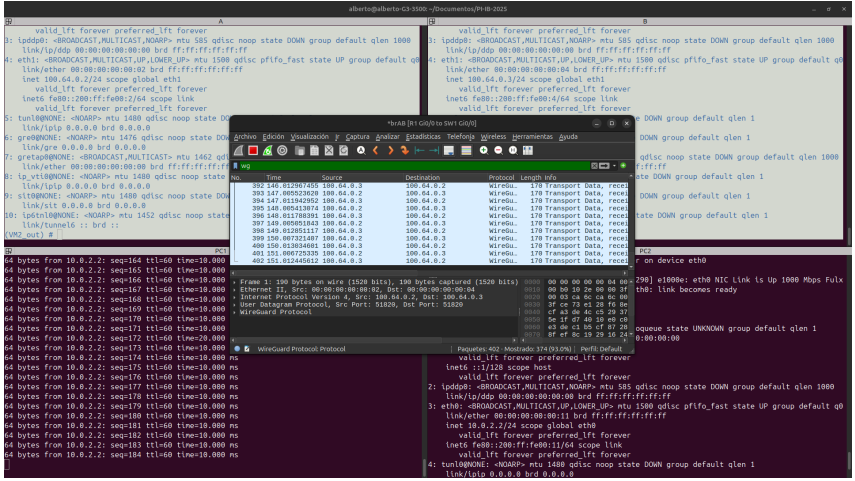

Figura 9: Comunicación entre PCs en nodos distintos.

Modelo de hardware: integración

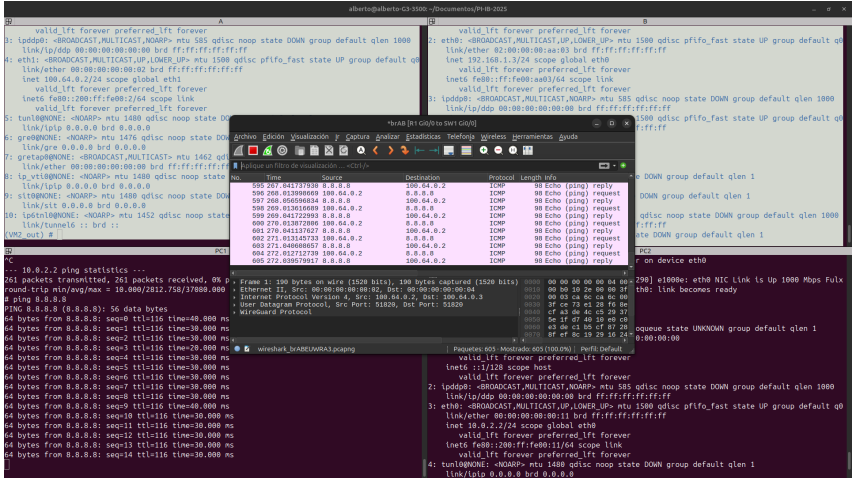

Figura 10: Ping de un nodo a internet.

Implementación en hardware

Figura 11: SuperMicro SYS-E300-9D.

Características:

- CPU: Intel Xeon D-2123IT (4C/8T)
- RAM: 32 GB DDR4
- 4x NIC Intel i210 1GbE
  2x NIC Intel X553 10GbE
- Soporte para virtualización (VT-x, VT-d)

Implementación en hardware: desafíos

Figura 12: SuperMicro SYS-E300-9D.

Desafíos:

- ✓ Redirección de consola.
- × Passthrough de controlador Ethernet.
- × Throughput entre VMs.

Implementación en hardware: throughput entre VMs

- La implementación actual requiere de 7 mapeos de memoria por paquete, lo que limita drásticamente el throughput.

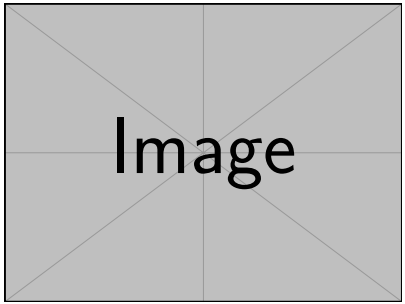

Figura 13: Throughput medido entre VMs.

Implementación en hardware: throughput entre VMs

Existen alternativas para mejorar el rendimiento:

- Translation Vspace.
- Optimización de copias en memoria.

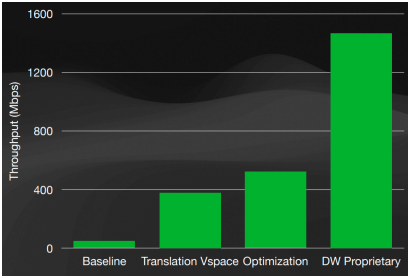

Figura 14: Optimizaciones logradas sobre el throughput entre VMs.
Fuente: seL4 Summit 2023.

Implementación en hardware: throughput entre VMs

Existen alternativas para mejorar el rendimiento:

- Translation Vspace.
- Optimización de copias en memoria.

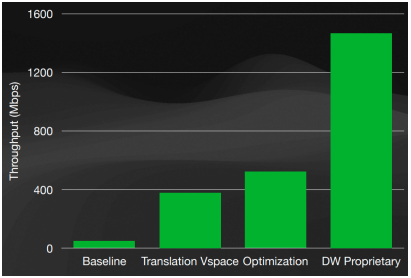

Figura 15: Optimizaciones logradas sobre el throughput entre VMs.
Fuente: seL4 Summit 2023.

Resumen

- Introducción a los modelos de trabajo utilizados.
- Validación del encriptador en un entorno virtualizado.
- Avances en la implementación sobre hardware.

Trabajo a futuro

- Resolver desafíos pendientes en la implementación de hardware.
- Documentar y presentar la solución final.

¡Muchas gracias!
¿Preguntas?