lacnic 45

25 - 28 MAYO / CIUDAD DE PANAMÁ, PANAMÁ

Hacia la detección de ataques DoS en la NIC del servidor

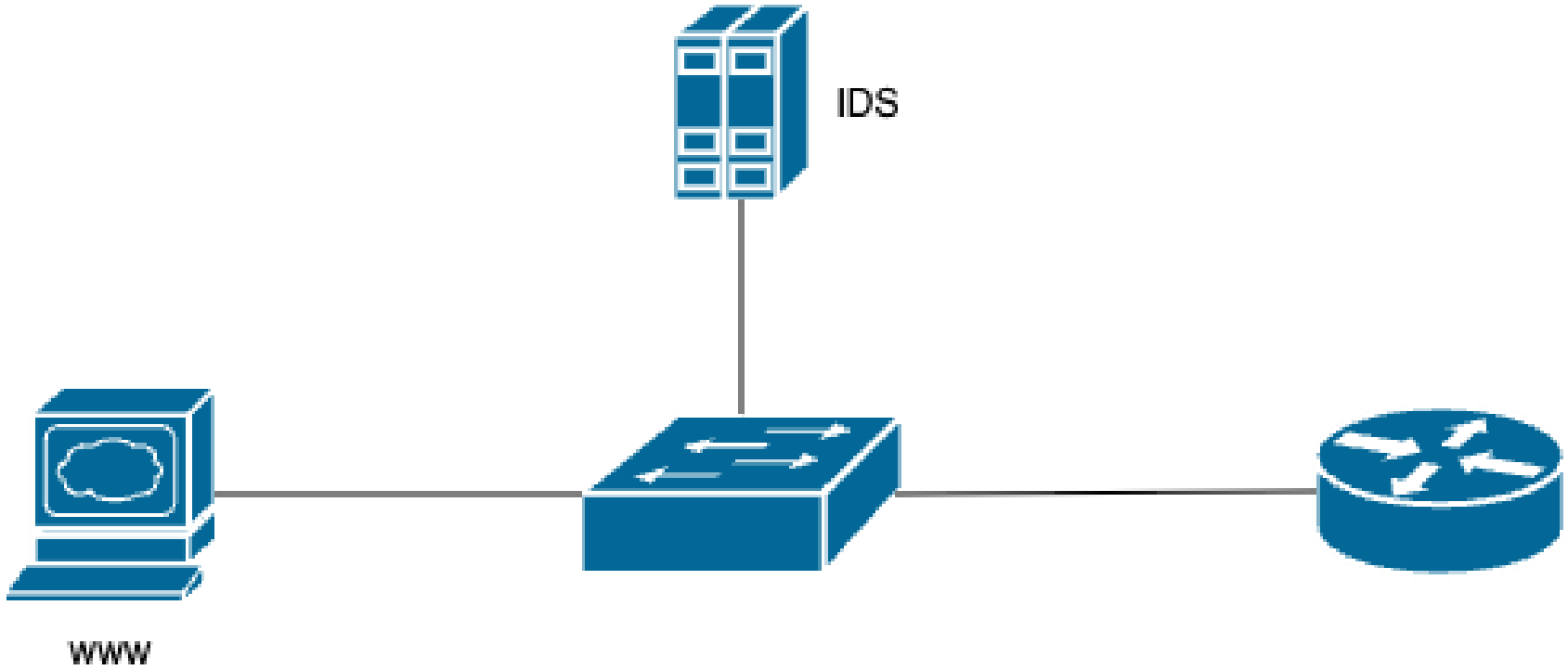
Adrián Lara, Universidad de Costa Rica, San José, Costa Rica

Juan Felipe Botero, Universidad de Antioquia, Medellín, Colombia

Agenda

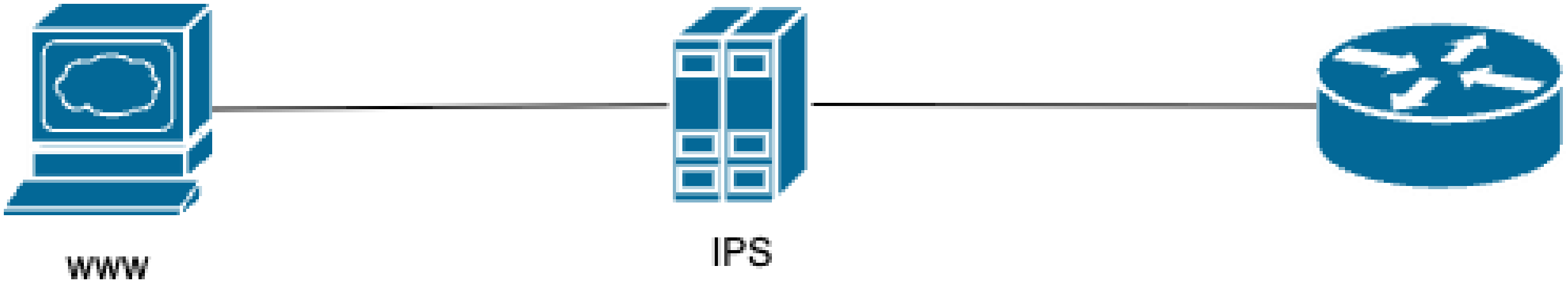
- Dificultades a la hora de detectar intrusos en la red
  - Duplicación de tráfico
  - Cuellos de botella
- Alternativas “in-network” como solución a estos problemas
- Smart NIC
  - Funcionalidades y ventajas
- Investigación actual: modelos de aprendizaje automático incrustados en una Smart NIC

Dificultades a la hora de detectar intrusos en la red



Escenario 1: Duplicación de tráfico

Dificultades a la hora de detectar intrusos en la red



Escenario 2: Cuello de botella

Planos de datos programables

*Históricamente, los ASICs de red eran cerrados (funciones fijas). Hoy, podemos programar el comportamiento del hardware mediante lenguajes como **P4**.*

Plano de Control: Cerebro de la red (rutas, políticas, actualización de modelos ML). Es lento pero inteligente.

Plano de Datos: Músculo de la red (parseo, tablas match-action, descarte de paquetes). Opera a "line rate".

Switches Programables

Ej: Intel Tofino. Ideales para telemetría a escala masiva y enrutamiento en el núcleo (Core).

SmartNICs

Ideales para granularidad por host, offloading de la CPU del servidor y micro-segmentación (Edge).

lacnic

Alternativa con In-Network Computing

La solución estudiada propone un cambio de paradigma: **moverse al plano de datos.**

Eliminación de Espejo: La SmartNIC procesa el tráfico localmente sin necesidad de duplicarlo a un IDS externo.

Inferencia a "Line Rate": Ejecución de modelos ML sencillos directamente en el hardware de red.

Programabilidad P4: Definición del pipeline de procesamiento mediante lenguajes de alto nivel para redes.

"Descargamos la lógica de detección de la CPU a la NIC, permitiendo que el servidor ignore el tráfico malicioso antes de que toque el sistema operativo."

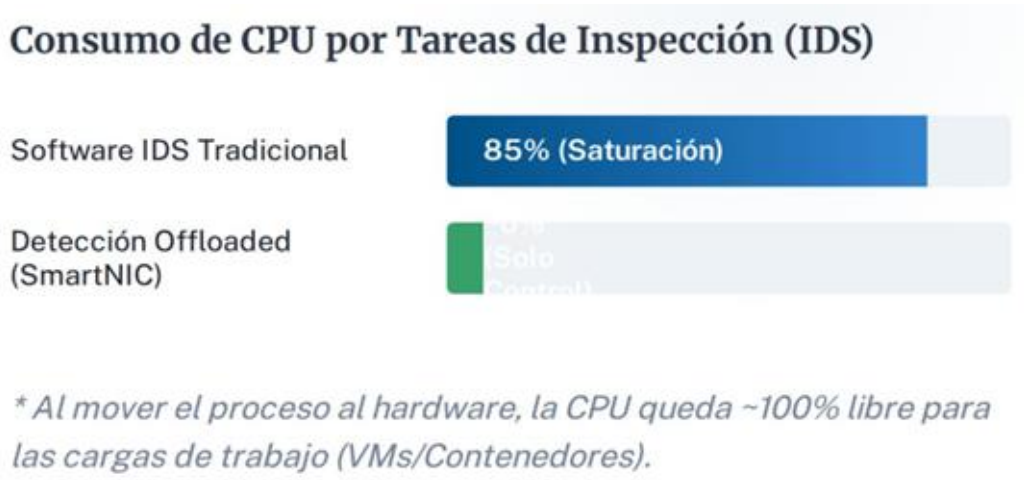
Offloading: Plano de datos Vs. CPU

CPU (Control Plane)

Gestiona políticas globales, actualización de modelos de ML y gestión de excepciones. Libera ciclos para las aplicaciones críticas (VMs/Contenedores).

SmartNIC (Data Plane)

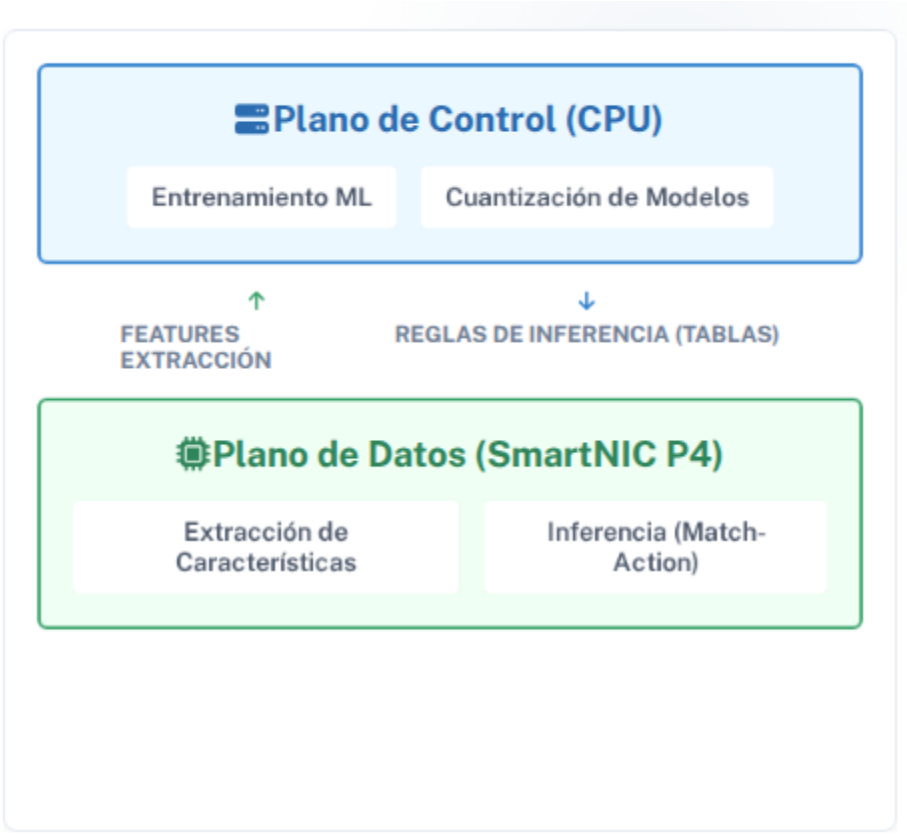
Ejecuta el filtrado de paquetes, extracción de características en tiempo real (Real-time stats) y la inferencia del modelo ML en cada paquete.



ML incrustado: División de roles

La integración de Machine Learning en hardware de red requiere separar estrictamente las responsabilidades:

- **Solo Inferencia:** El plano de datos evalúa paquetes a gran velocidad, *no entrena*.
- **Traducción a P4:** Modelos sencillos (ej. Árboles de Decisión) se convierten en *Tablas Match-Action*.
- **Cuantización:** Se reduce la precisión de los modelos para que quepan en la memoria limitada de la NIC.
- **Recolección de Features:** La NIC extrae características críticas y las envía a la CPU para entrenar o re-entrenar el modelo.



Lógica de detección en P4

```
control Ingress(...) {
  apply {
    // Extracción de entropía y tasa de paquetes
    stats_metadata.pkt_rate = counter_read(flow_id);
    if (stats_metadata.pkt_rate > THRESHOLD) {
      // Inferencia de modelo ML simple
      mark_as_dos(); drop();
    }
  }
}
```

La SmartNIC actúa como sonda, calculando métricas que el modelo necesita

Un árbol de decisión se linealiza en tablas de red. Cada rama se convierte en una entrada de tabla con rangos de valores.

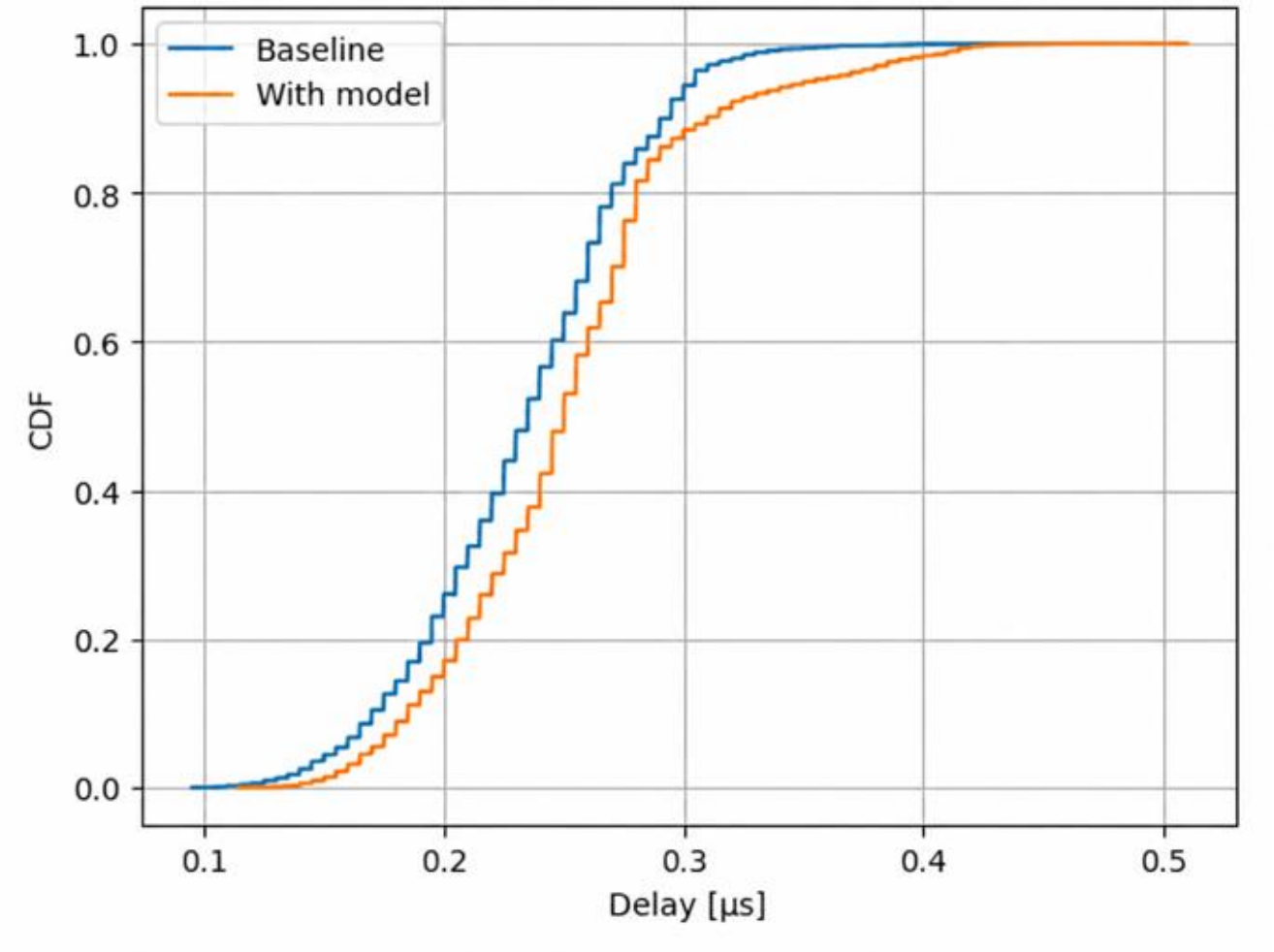
Inferencia In-Line:
La NIC busca la combinación de features en la tabla. Si hay coincidencia (match), se ejecuta la acción de mitigación (drop) instantáneamente.

```
// Mapeo de un árbol de decisión a Tabla P4
table ddos_classification_tree {
  key = {
    meta.features.pkt_rate : range;
    meta.features.avg_size : range;
  }
  actions = {
    drop_packet; allow_packet;
  }
}

apply {
  // Inferencia: Si (rate > 10k AND size < 64) -> Drop
  ddos_classification_tree.apply();
}
```

lacnic

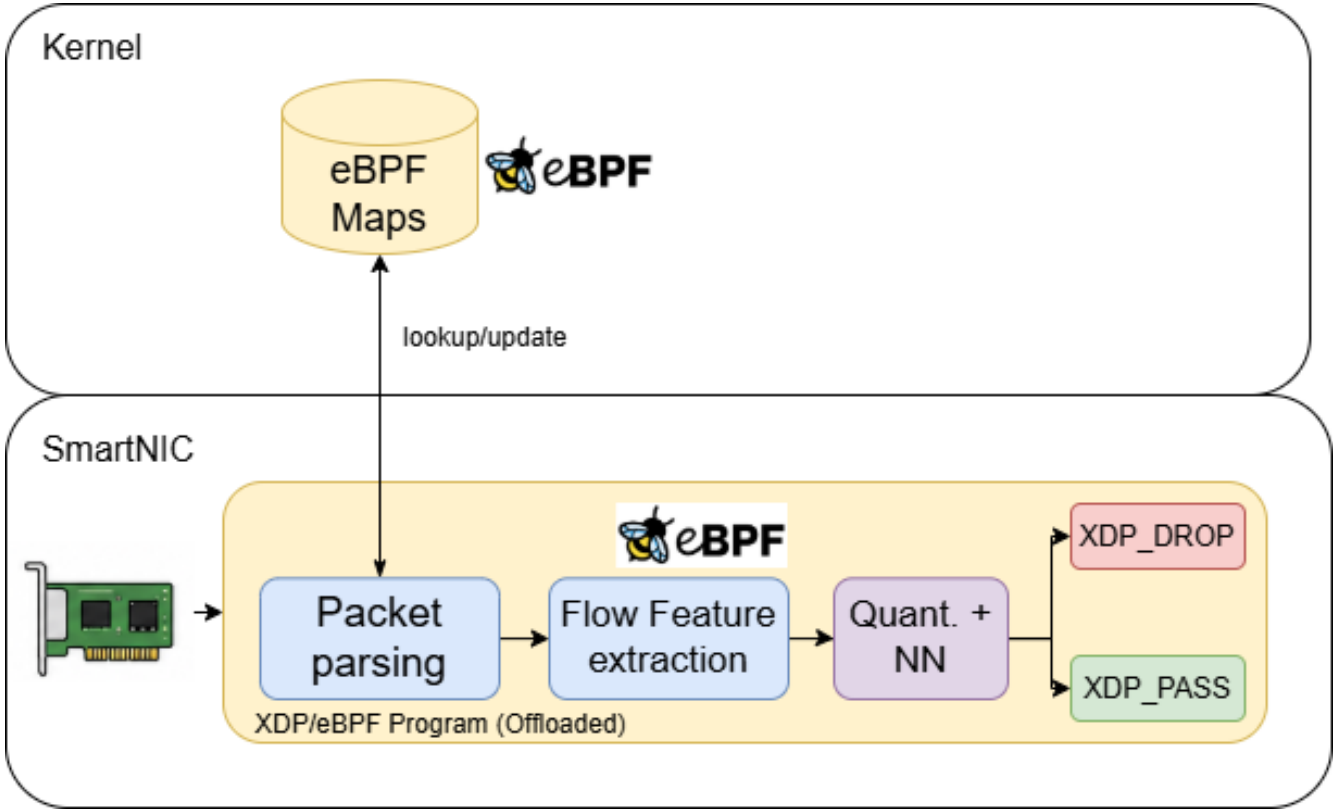
Delay de detección en P4



Distribución acumulada del delay medido en la SmartNIC Netronome.

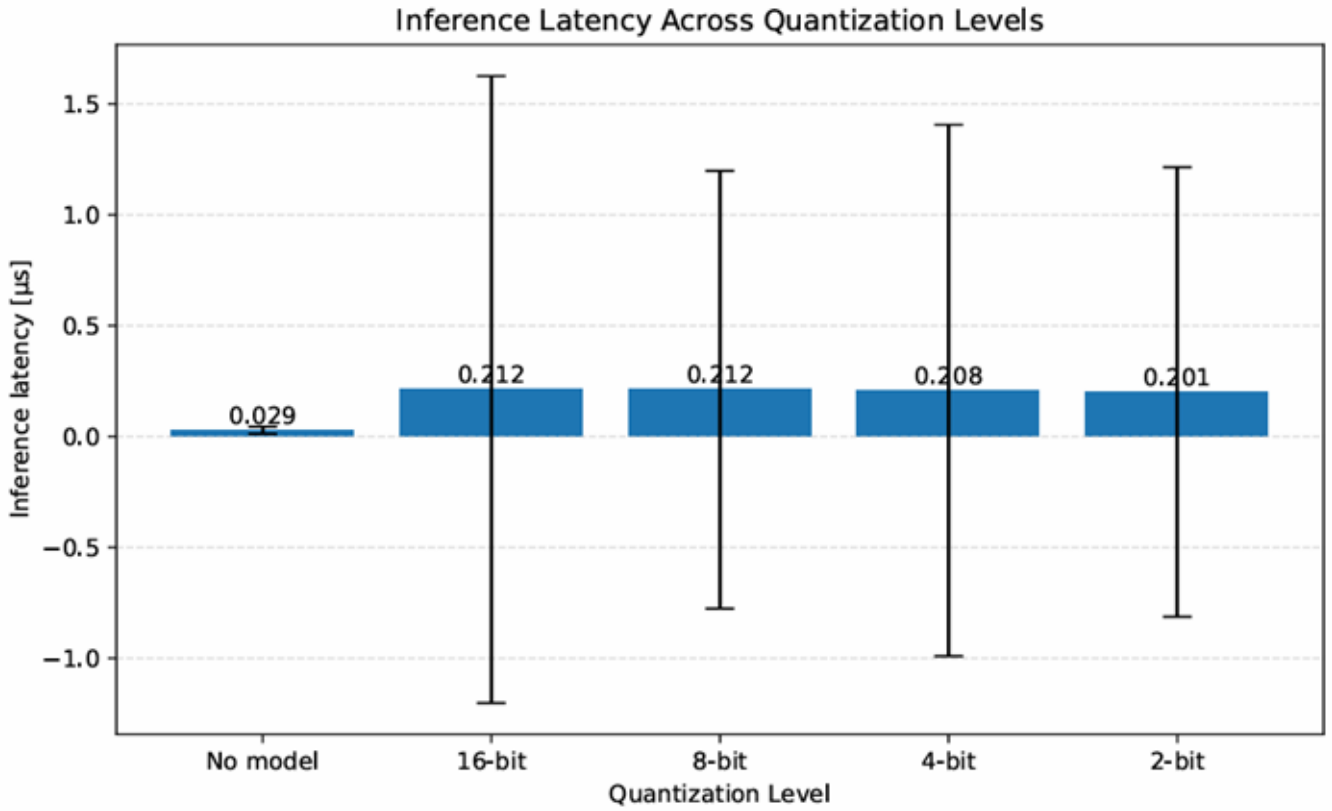
- Baseline (sin modelo) vs sistema con modelo ML.
- El modelo introduce un incremento leve en la latencia, observable por el desplazamiento de la curva hacia la derecha.
- Sin embargo, ambas distribuciones mantienen un comportamiento similar y baja dispersión.
- Esto indica que la inferencia puede ejecutarse con overhead reducido dentro del plano de datos.

Otros casos: ML en el plano de datos con eBPF



- CASO: Evaluación de técnicas de cuantización de modelos de ML para la detección de DoS/DDoS en el plano de datos.
- Cuantización de 16-bit, 8-bit, 4-bit y 2-bit.
- Uso de XDP + eBPF.

Otros casos: ML en el plano de datos con eBPF



- Latencia de inferencia para los distintos niveles de cuantización utilizando la mediana y el percentil P95.
- Todos los modelos cuantizados presentan medianas similares cercanas a 0.2 µs, indicando que el overhead introducido por la inferencia neuronal es bajo independientemente del nivel de cuantización.

Conclusiones

- Procesos como detección de ataques se pueden realizar a "line-rate" dentro de la tarjeta de red de un servidor o incluso dentro de un Switch programable.
- El retraso y la sobrecarga de reenviar la información a un IDS se puede evitar con la detección directa en el plano de datos.
- El retraso que añade la implementación de los modelos en el plano de datos programable es pequeño (casi descartable).
- La programabilidad de redes añade funcionalidades no solo de deteción de ataques sino de gestión de QoS, ingeniería de tráfico, telemetría en banda, entre otras.

Preguntas

lacnic 45

25 - 28 MAYO / CIUDAD DE PANAMÁ, PANAMÁ

¡Gracias!

Hacia la detección de ataques DoS en la NIC del servidor

Adrián Lara, Universidad de Costa Rica, San José, Costa Rica

Juan Felipe Botero, Universidad de Antioquia, Medellín, Colombia

Anexo

Otros casos: ML en el plano de datos con eBPF

Level	Accuracy	Δ Acc	Precision	Δ Prec	Recall	Δ Rec	F1	Δ F1
32-bit	0.996	—	0.996	—	0.996	—	0.996	—
16-bit	0.997	0.001	0.995	-0.001	0.998	0.003	0.997	0.001
8-bit	0.935	-0.061	0.902	-0.094	0.978	-0.018	0.938	-0.058
4-bit	0.674	-0.322	0.712	-0.284	0.688	-0.308	0.664	-0.332
2-bit	0.568	-0.428	0.544	-0.452	0.824	-0.172	0.635	-0.361

Summary of metrics by quantization level