

Solución al ensamble de paquetes IP

Para resolver el ensamblado de los diferentes fragmentos de paquetes IP se tomaron las siguientes consideraciones:

- Se creyó conveniente que quienes poseyeran la lógica de como ensamblar paquetes sean los mismos paquetes, dejándole al ensamblador solo la responsabilidad de administrar los paquetes a ensamblar.
- También son los paquetes quienes saben responder si están completos o no. Se supuso que el ensamble sólo se lleva a cabo con otro paquete que no haya sido previamente ensamblado, esto simplifica el algoritmo de ensamble y es suficiente para el uso que se le da en el TP al ensamblado de paquetes. Que esta condición se cumpla lo garantiza el ensamblador.
- El ensamblador no filtra los paquetes repetidos, es tarea del detector filtrar si un mismo paquete ya fue procesado para no generar alertas repetidas.

Gracias a estas consideraciones se logro reducir el algoritmo para el ensamble de paquetes a tres casos:

- El paquete `this` o el paquete con el cual se va a ensamblar es el último fragmento (fragmento con offset mayor).
- El paquete con el que se va a ensamblar esta adentro de `this`. Esto ocurre cuando `this` ya fue ensamblado previamente con otro paquete y quedaron bytes faltantes entre los paquetes ensamblados.
- Los paquetes están separados pero ninguno es el último.

Jerarquía de clases y polimorfismo

Una de las jerarquías de clase implementadas es la de las reglas. Esto se realizó para que a el detector no le importara que tipo de regla estaba aplicando cuando se le pedía procesar un paquete, si no que al guardar referencias a la clase madre ya sabía que podían responder al mensaje aplicar sin importar que tipo de regla fuera en realidad.

Esquema de diseño

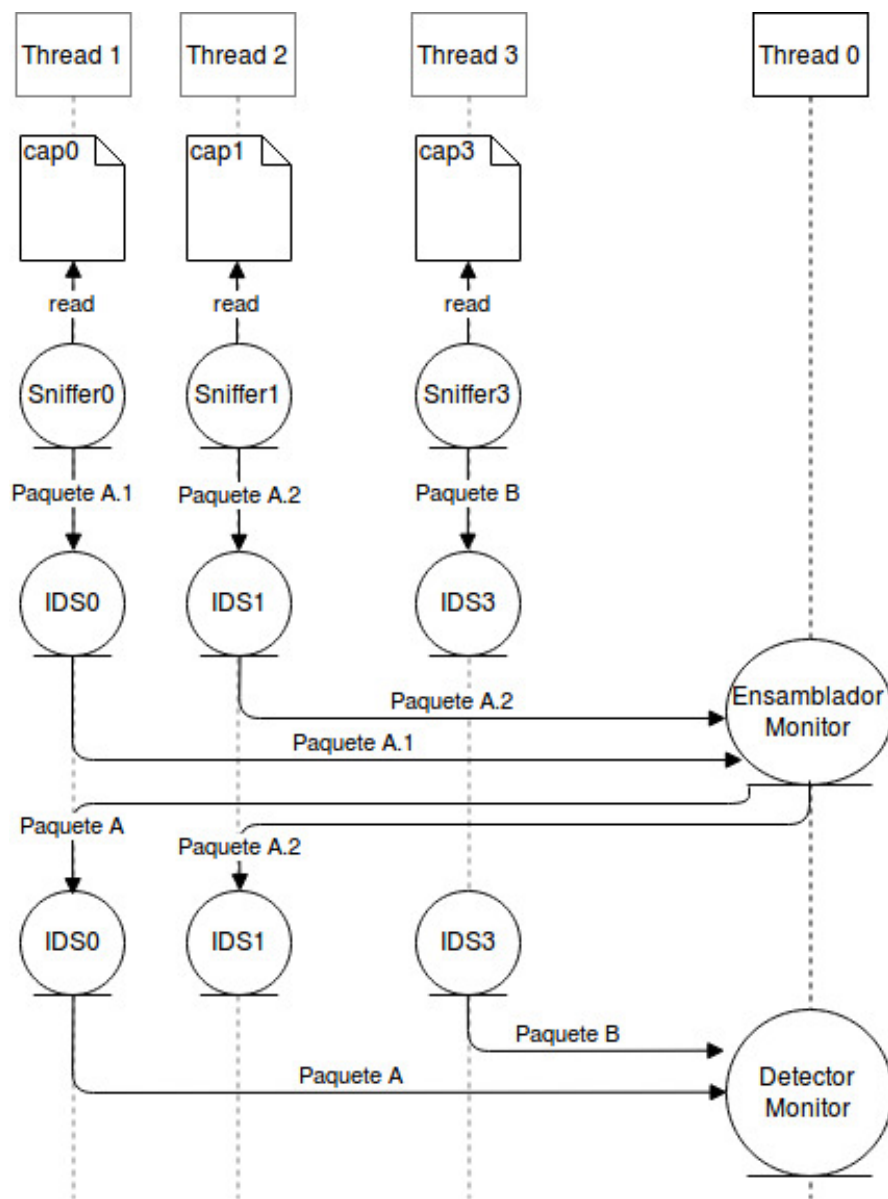


Figura 1: Diagrama del ensamble y detección de paquetes.

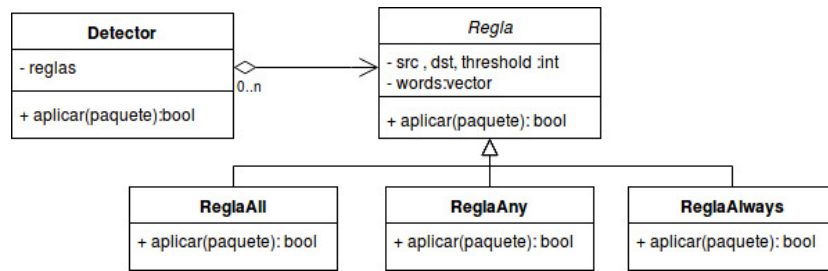


Figura 2: Jerarquía clases reglas