# Alta Disponibilidad de Cortafuegos en Linux

Pablo Neira Ayuso
<pablo@netfilter.org> Proyecto Netfilter
<pneira@us.es> Universidad de Sevilla

# Esbozo de la presentación

- Introducción: Cortafuegos y Alta Disponibilidad
- Escenario Problemático: Cortafuegos con estados y Alta Disponibilidad
- Conntrackd: Replicación de estados de las conexiones
- Conclusiones y Trabajos futuros
- Tema Sorpresa...

# Introducción: Cortafuegos

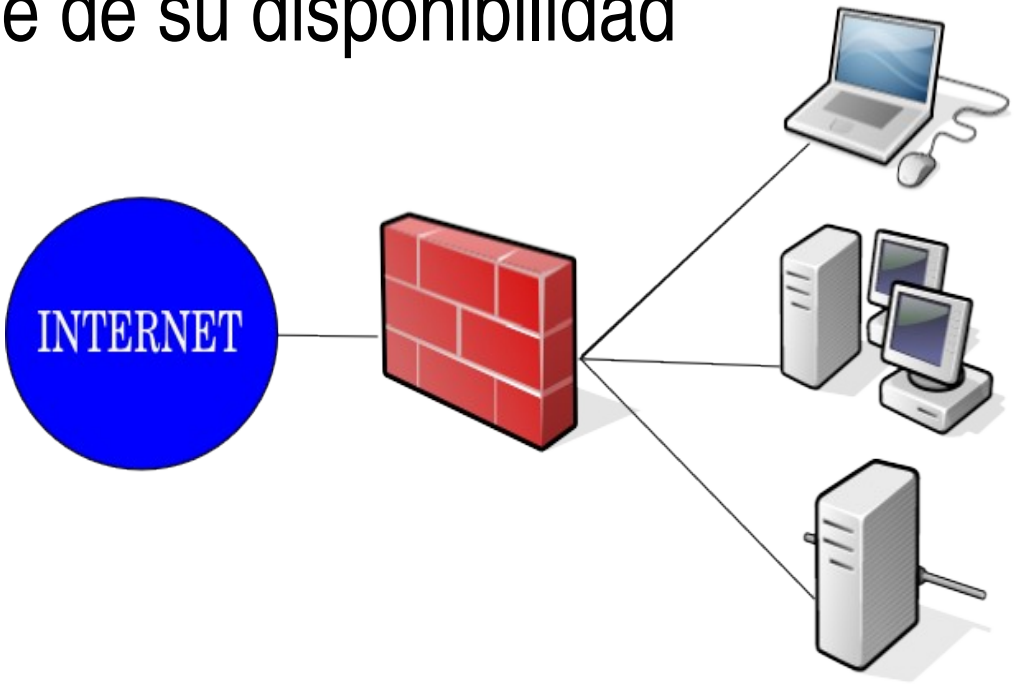
- ¿Qué es un cortafuegos **(Firewall**)? Dispositivo que permite implementar políticas de filtrado sobre el tráfico que circula por la red: Paradigma de seguridad perimetral
- La 1ª generación de cortafuegos permite implementar la política de filtrado en base a las cabeceras de niveles OSI 2,3 y 4: Cortafuegos **sin** estados (**Stateless Firewalls)**
- Dicha 1ª generación presenta problemas relacionados con dicha visión estática: DOS attacks (como TCP reset)

# Introducción: Cortafuegos (2)

- Para solucionar los problemas... la 2ª Generación de cortafuegos permiten implementar una política de filtrado más “inteligente” que se basa en el **estado** de la conexión: Se mantiene una tabla con el estado de las conexión.
- En Linux, 4 tipos de estados:
  - NEW: primer paquete visto de una conexión
  - ESTABLISHED: se han visto paquetes en ambas direcciones
  - INVALID: paquete malformado, fuera de secuencia...
  - RELATED: relacionado con otra conexión (ICMP unreachable)

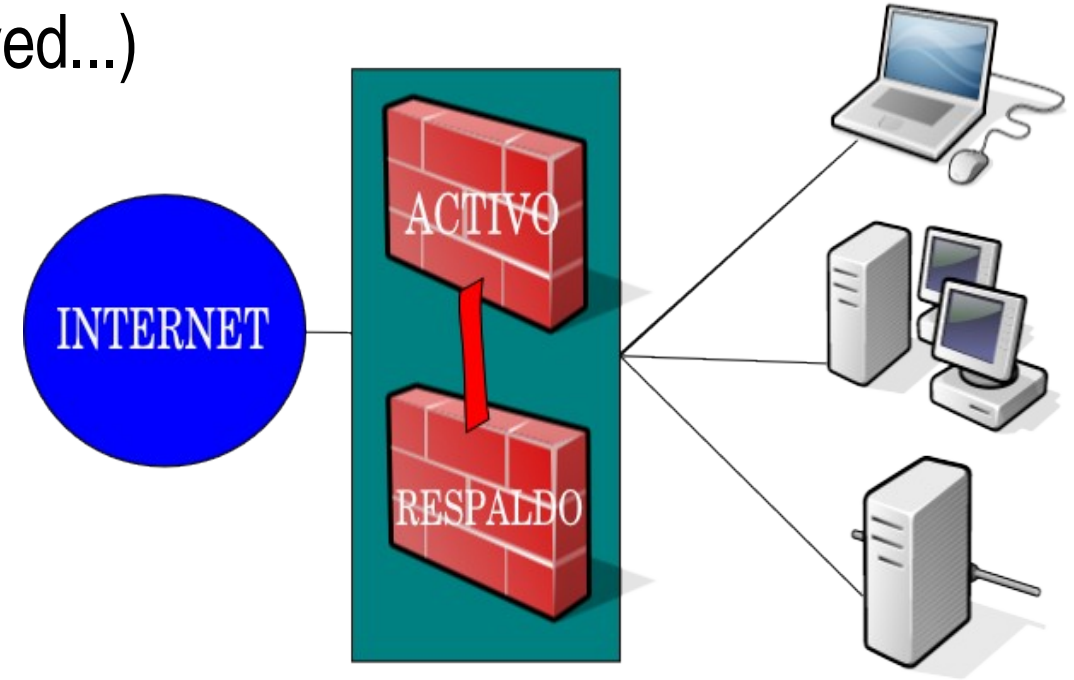
# Introducción: Alta disponibilidad

- Cortafuegos introducen un punto de fallo único (Single Point of Failure, SPOF) en el esquema de red: La disponibilidad de los servicios ofrecidos depende directamente de su disponibilidad



# Introducción: Alta disponibilidad (2)

- Solución: Replicación de puntos únicos de fallo
  - Solución económica con respecto a las tolerantes a fallos
  - Automatización de la recuperación en fallo (heartbeat, keepalived...)



# Escenario problemático

iptables -P FORWARD DROP

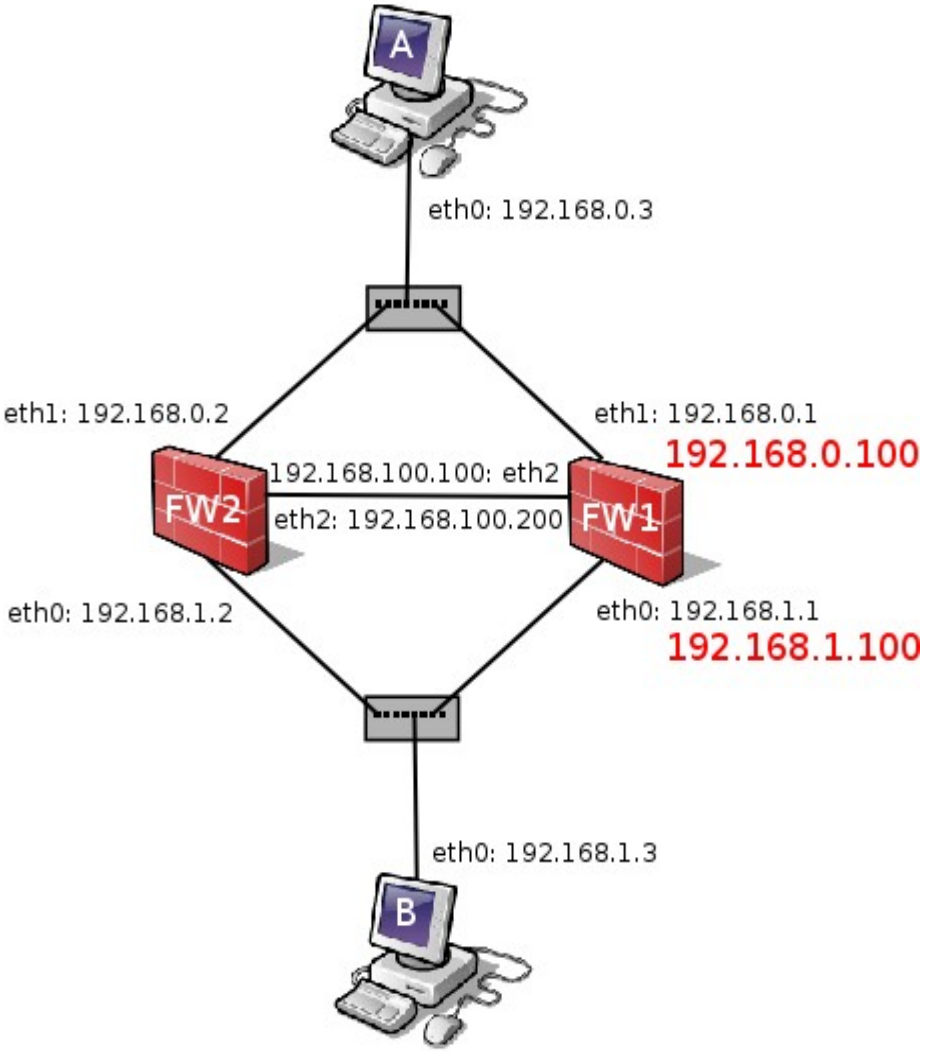
iptables -A FORWARD -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -i eth1 -p tcp --syn -m state --state NEW -j ACCEPT

iptables -A FORWARD -i eth1 -p tcp -m state --state ESTABLISHED -j ACCEPT

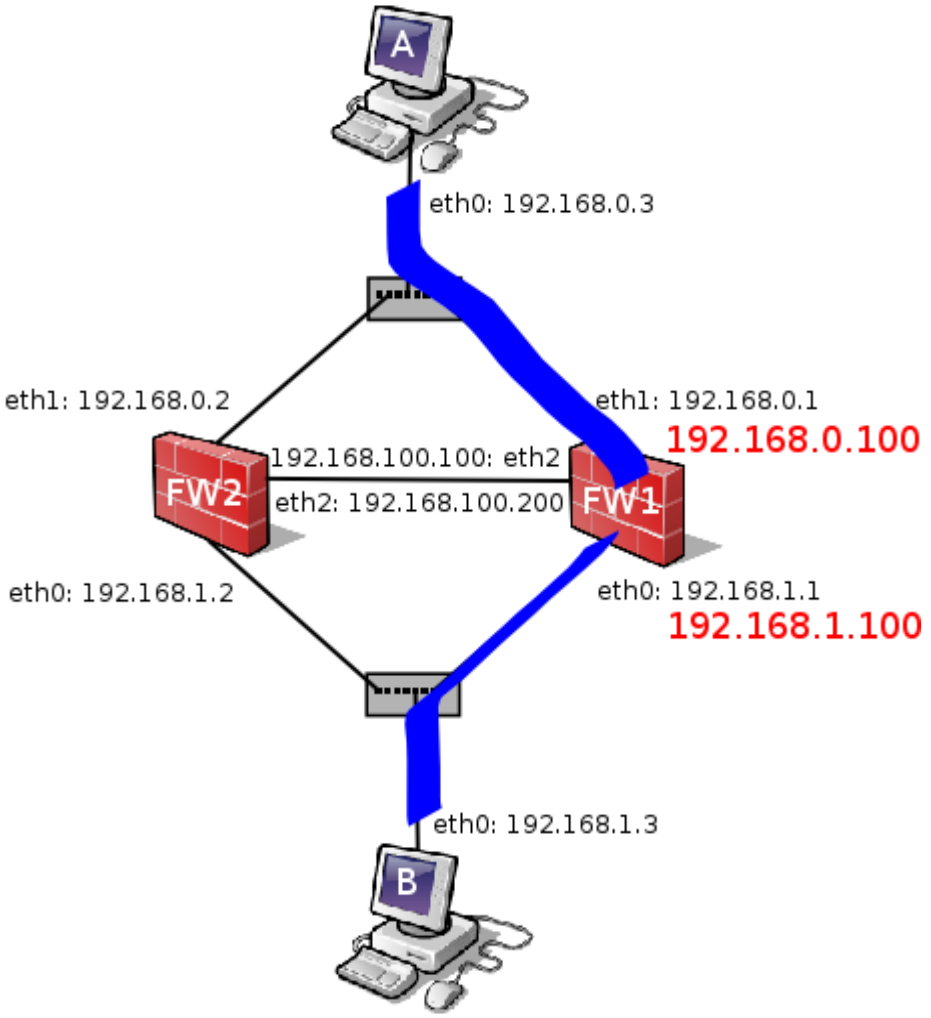
iptables -I FORWARD -j LOG

iptables -I POSTROUTING -t nat -s 192.168.0.3 -j SNAT --to 192.168.1.100



# Escenario problemático (2)

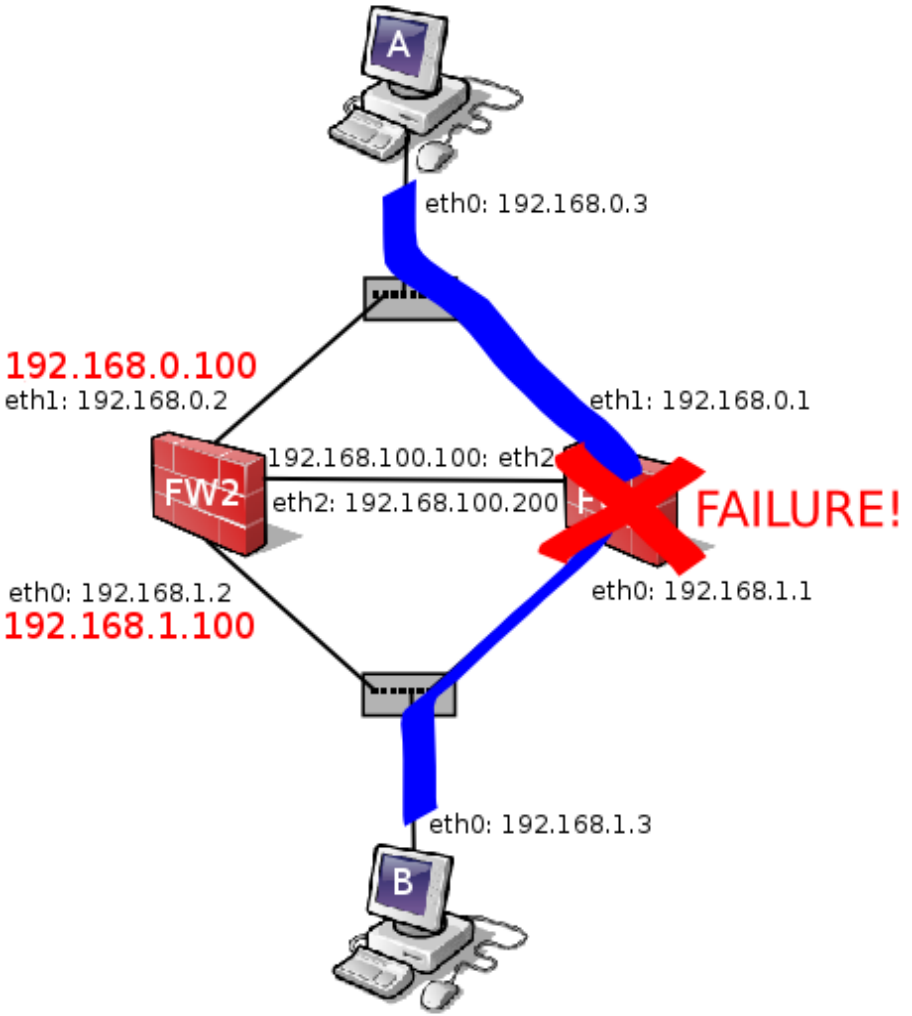
Paso 1) Estación de trabajo A establece una conexión SSH con B: ESTABLISHED



# Escenario problemático (3)

Paso 1) Estación de trabajo  A establece una conexión SSH con B: ESTABLISHED

Paso 2) Se produce un fallo en el cortafuegos activo (FW1): Respaldo recupera la IP virtual



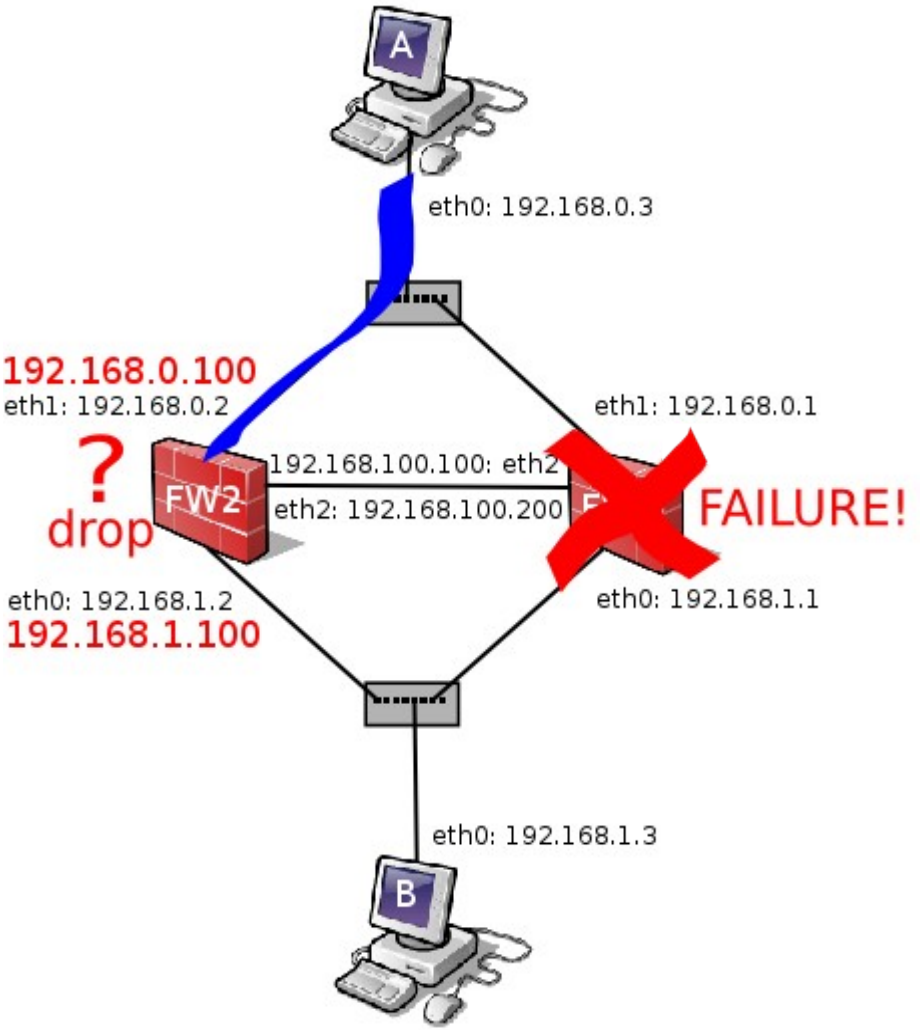
# Escenario problemático (4)

Paso 1) Estación de trabajo  A establece una conexión SSH con B: tráfico permitido

Paso 2) Se produce un fallo en el cortafuegos activo (FW1): Respaldo recupera la IP virtual

Paso 3) El nuevo nodo activo aplica la política de filtrado definada, al ser el primer paquete que ve asociado a la conexión considera que está en estado **NEW**: Conexión queda interrumpida

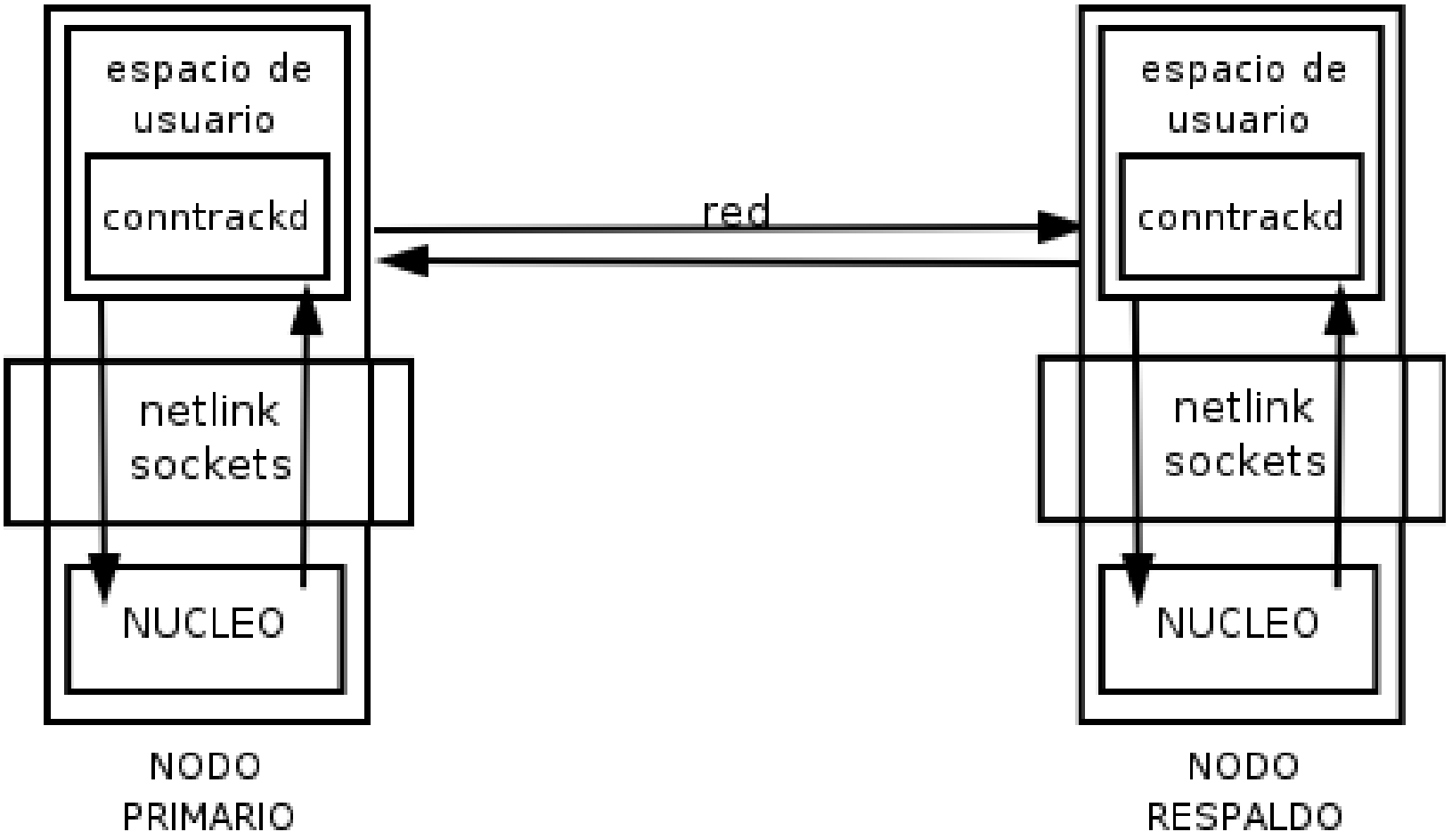
¿Solución? Nodo Réplica necesita saber cuál es el estado de las conexiones que el activo está procesando



# Conntrackd: Replicación Estados

- Demonio de espacio de usuario que realiza la replicación del estado de las conexiones entre nodos escrito en C: extensión de la soluciones de Alta Disponibilidad existentes, requiere núcleo de linux>= 2.6.18
- Soporta escenarios Primario/Backup
- Soporta escenarios Activo/Activo: Balanceo de carga
- Protocolo basado en Netlink sobre Multicast
- Configurable
- Fácilmente extensible: Obtención de estádisticas

# Conntrackd: Replicación Estados



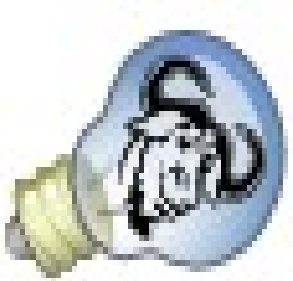
# Conclusiones

- Conntrackd es software libre (GPL)
- Actualmente en desarrollo, última versión disponible 0.9.1: Administradores de Sistemas están reportando la implementación de cortafuegos en Alta Disponibilidad sobre Linux **con éxito** (netfilter-failover@lists.netfilter.org)
- Disponible en: http://people.netfilter.org/pablo/conntrackd/

# Trabajos Futuros

- Soporte full TCP window tracking: Parches para el kernel actualmente bajo discusión
- Mejorar interacción con soluciones de Alta Disponibilidad
- Integración con CLUSTERIP (balanceo de carga basado en funciones de hash)
- Documentación: Página de Manual, Configuración
- Paquete Debian: Parche recientemente recibido
- Mejora del funcionamiento en modo estadísticas


- Objetivo: Motivar a los estudiantes universitarios a que desarrollen software libre porque...
  - es bueno para ellos: pueden crecer técnicamente
  - es bueno para la sociedad: generan software libre al alcance de todos
- Resultados: 93 proyectos inscritos, no sólo hay cantidad, también hay calidad.
- Planet: http://concurso-softwarelibre.us.es/planet/
- Os invitamos a la fase final en Mayo 2006 en Sevilla



Concurso Universitario de Software Libre

# Gracias, ¿preguntas?

- Agradecimientos: SOLFA, Rediris, Carmen López (SIC Universidad de Sevilla), a nuestros patrocinadores... a Rafa, Ana y Arcturus. Mamá, Papá os quiero.