Comunicación de Datos 2

Implementación del protocolo IGMP en KivaNS

Clarisa Gallardo

Mauro Sebastián Melzi

Néstor de Dios

Índice

[Descripción del protocolo IGMPv2 3](#_Toc240632237)

[El simulador KivaNS 4](#_Toc240632238)

[Extensiones al simulador KivaNS 6](#_Toc240632239)

[Implementación del protocolo IGMP 7](#_Toc240632240)

[Próxima etapa 9](#_Toc240632241)

# Introducción

En el presente trabajo se realizó la implementación del protocolo IGMPv2 en el simulador KivaNS.

Para comenzar, se describirá el protocolo implementado tal como aparece en la norma RFC2236. La misma se utilizó como fuente para el desarrollo del trabajo.

Luego, se presentará el simulador KivaNS versión 1.1, exponiendo sus características principales. Las mismas incluyen el núcleo, la API de programación y la interfaz gráfica.

Seguidamente, se muestran las extensiones realizadas al simulador para que sea posible realizar una simulación más rica, punto que es de vital importancia para poder apreciar el funcionamiento del protocolo implementado.

A continuación se expone la parte más importante de este trabajo, o sea, la implementación del protocolo IGMP. La misma consistió de la creación de un nuevo tipo de modulo especializado para la interpretación de este protocolo y un nuevo tipo de mensaje que es el que se intercambian entre las instancias de este nuevo modulo en distintos equipos.

Por último, se detallan las modificaciones realizadas sobre la interfaz gráfica del simulador para adaptarlo a las nuevas características agregadas al núcleo.

Adicionalmente, se provee de una batería de ejemplos que demuestran el funcionamiento del protocolo.

# Descripción del protocolo IGMPv2

Los routers multicast utilizan el protocolo IGMP para determinar que grupos tienen miembros en cada una de sus redes. Ellos, mantienen una lista de membrecías a grupos de difusión para cada una de sus redes y una timer para cada una de éstas. Dichas membrecías significan la presencia de al menos un miembro de un grupo de difusión en una de sus redes en particular y no la lista de todos los miembros en esa red.

En cada una de sus redes, un router multicast puede asumir uno de los siguientes roles: Querier o Non-Querier. Normalmente hay solo un router cumpliendo el rol de Querier por cada red.

Todos los routers multicast comienzan como Querier en todas sus redes cuando se inician. Si recibe un “Query message” de un router con una dirección IP más baja, debe dejar de ser Querier y convertirse en Non-Querier para esa red. Si estando en el rol de Non-Querier, no recibe un “Query message” de otro router durante un período de [Other Querier Present Interval], asumirá el rol de Querier.

Los routers multicast periódicamente (cada [Query Interval]) enviarán un query general en cada una de sus redes para las cuales cumple el rol de Querier, para solicitar informes de membrecía a grupos de difusión. Cuando se inicia, el router debe enviar la cantidad de [Startup Query Count] de quieries generales, cada una a un tiempo de [Startup Query Interval] de distancia, para determinar de forma rápida y confiable la información sobre las membrecías. Un query general es dirigido al grupo que incluye todos los sistemas (224.0.0.1), posee el campo de dirección de grupo como 0.0.0.0 y un tiempo máximo de respuesta de [Query Response Interval].

Cuando un host recibe uno de los queries generales, prepara un timer para cada uno de los grupos a los que está suscripto (excepto para el grupo de todos los sistemas) en la interfaz por donde recibió el query. Cada uno de esos timers es configurado con un valor aleatorio diferente, utilizando el máximo nivel de granularidad del reloj disponible en ese host, dentro del rango (0, Max Response Time], con el “Max Response Time” especificado en el query.

Cuando un host recibe un query específico para n grupo, configura un timer con un valor aleatorio similar al utilizado para los queries generales para el grupo sobre el cual fue consultado, si es miembro de éste en la interfaz por la cual recibió el query. Si ya existe un timer para ese grupo, debe ser reiniciado solo si el nuevo valor del timer es menor que el que posee actualmente. Cuando el timer expira, el host envía un “versión 2 Membership Report” para el grupo, con el tiempo de vida con el valor 1. Si el host recibe un reporte de otro host mientras el timer aun está corriendo, detiene el timer para ese grupo específico y no envía el reporte, con el fin de eliminar los reportes duplicados.

Cuando un router recibe un reporte, agrega el grupo reportado a la lista de membrecías de grupos de difusión para la interfaz por la cual recibió el reporte y configura un timer para esa membresía de [Group Membership Interval]. Reportes repetidos para ese grupo reiniciarán el timer. Si, por el contrario, no se reciben reportes para un grupo en particular antes de que el timer expire, el router asumirá que el grupo no posee más miembros y que no necesita continuar reenviando mensajes multicast remotos para ese grupo de difusión en dicha red.

Cuando un host desea unirse a un grupo multicast, enviará un reporte de membrecía no solicitado para el grupo de difusión al que se suscribe de manera inmediata. Para minimizar las posibilidades de que este reporte inicial se pierda o dañe, es recompendable que se repita el envío una o dos veces, con una diferencia de tiempo de [Unsolicited Report Interval].

Cuando un host desea dejar un grupo de difusión, si fue el último en responder a un Query con un “Membership Report” para ese grupo, debe enviar un “Leave Group Message” al grupo de todos los routers multicast (224.0.0.2). Si no fue el último en responder el Query, no debe enviarlo, debido a que debe haber otro miembro del grupo de difusción en la subred. Eta es una optimización para reducir el tráfico.

Cuando un Querier recibe un mensaje de “Leave Group” para un grupo de difusión que tiene miembros en la interfaz por la cual lo recibió, envía la cantidad de [Last Member Query Count] Queries específicos cada [Last Member Query Interval] para el grupo que se está dejando. Estos Queries poseen el tiempo máximo de respuesta como [Last Member Query Interval]. Si el router no recibe reportes después de que el tiempo de respuesta del último Query expira, el router asume que el grupo no posee más miembros en la subred. Las transiciones de Querier a Non-Querier son ignoradas durante este período, el mismo router envía estos Queries específicos.

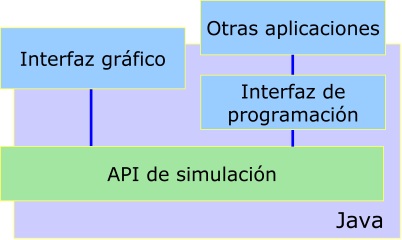
Los routers multicast que cumplen el rol de Non-Querier, deben ignorar los mensajes de “Leave Group”. Cuando reciben un Query específico para un grupo, si poseen un timer para la membrecía de ese grupo en esa subred que es mayor que [Last Member Query Count] multiplicado por el tiempo de respuesta del mensaje, deben configurar el timer al nuevo valor.

# El simulador KivaNS

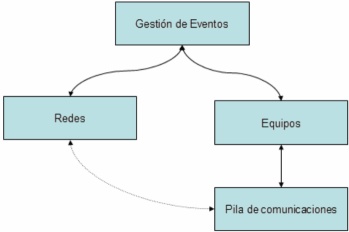
KivaNS (Kiva Network Simulator) es una aplicación gratuita y de código abierto basada en Java para especificar esquemas de redes de datos y simular el enrutamiento de paquetes a través de esas redes. Está orientado principalmente a simular el comportamiento del protocolo IP y especialmente el tratamiento de los datagramas y el enrutamiento de los mismos por una red. Para ello KivaNS también considera el funcionamiento de protocolos auxiliares como ARP e ICMP, y emula el funcionamiento básico de tecnologías de enlace como Ethernet.

El objetivo principal del entorno es ayudar a diseñar y comprender el funcionamiento de redes de datos, y en especial el enrutamiento de paquetes en la arquitectura TCP/IP, sin necesidad de una infraestructura real y de herramientas de análisis de tráfico. KivaNS también es capaz de simular distintos tipos de errores en el funcionamiento de las redes, como la pérdida de paquetes o fallos en tablas de enrutamiento.

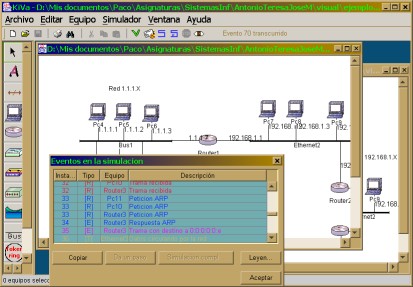
Se compone de dos partes, enteramente implementadas con Java. La primera es una API (Application Programming Interface) que ofrece un motor de simulación de redes a otras aplicaciones, y la segunda es una completa interfaz gráfica que hace uso del API de simulación. Dado que todo el entorno esta realizado con Java, funciona en múltiples sistemas operativos, como pueden ser GNU/Linux o Microsoft Windows.

  
Partes de KivaNS

El API está formado por cuatro bloques: una gestión de eventos discretos, los objetos que representan las redes de datos, los objetos que representan los equipos (finales o de interconexión), y una pila de comunicaciones. Este API es modular y extensible, de forma que se puedan incorporar fácilmente nuevos tipos de redes y de equipos.

  
Bloques que componen el API

En la versión actual, la interfaz de usuario está implementada con un conjunto de clases que se deben descargar y ejecutar en el equipo del usuario. Esta interfaz permite especificar las topologías de redes de datos mediante un editor gráfico, configurar mediante diálogos el direccionamiento y enrutamiento en los equipos de la red, y acceder a las características que ofrece el API de simulación de una forma sencilla, sin necesidad de programar.

  
Interfaz gráfica de KivaNS

# Extensiones al simulador KivaNS

Debido a que el simulador KivaNS está desarrollado para mostrar cómo se enruta un datagrama IP, fue necesario realizar una extensión del mismo que posea las características que se enumeran a continuación, con el fin de dar una mayor flexibilidad a la simulación.

Las características incorporadas en el simulador extendido son:

* Capacidad para programar eventos. En el simulador original, se podía simular el envío de un paquete IP, el cual comenzaba a ser enviado cuando comenzaba la simulación y una vez que el paquete llegaba a su destino dicha simulación terminaba. En esta nueva versión del simulador, es posible decidir en qué paso de la simulación se desea iniciar el envío del paquete, además de otras nuevas posibles acciones a simular:
  + Encender un equipo. Permite simular que tanto un router como un ordenador se encienda en un paso dado de la simulación.
  + Apagar un equipo. Permite simular que tanto un router como un ordenador se apague en un paso dado de la simulación.
  + Unirse a un grupo. Permite simular que un ordenador se una a un grupo de difusión en un paso dado de la simulación.
  + Dejar un grupo. Permite simular que un ordenador deje un grupo de difusión en un paso dado de la simulación.

Estas nuevas acciones a simular permiten una simulación mucho más rica ya que proveen la capacidad de, por ejemplo, simular que un equipo se caiga en un determinado momento o que se una a la red un equipo en algún momento.

En este simulador, cada paso de la simulación representa un avance en el tiempo de 0,1 segundos (1 décima de segundo).

* Routers Multicast. Se agregó a la lista de equipos del simulador este nuevo tipo de router que permite mantener listas de membresías a grupos de difusión para las subredes conectadas a dicho router.
* Módulo IGMP. Se creó un módulo específico que encapsula la funcionalidad definida en el protocolo IGMP. En la sección “Implementación del protocolo IGMP” se detallará como fue implementado el mismo mediante la creación de las clases necesarias.
* Mensaje IGMP. Este nuevo tipo de mensaje encapsula los mensajes correspondientes al protocolo IGMP.

# Implementación del protocolo IGMP

Para realizar la implementación del protocolo IGMP, fue necesaria la creación de las siguientes clases:

* “ModuloIGMP”. Subclase que extiende de la clase “Nivel” (que es la clase base para todos los módulos de KivaNS). La misma contiene la funcionalidad común que deben tener tanto los routers como los ordenadores que implementan el protocolo IGMP. Engloba las funciones de encolar y procesar los mensajes de salida y procesar la cola de entrada. También define las constantes del protocolo (como el tiempo entre queries, etc). Posee 2 subclases:
  + “ModuloIGMPOrdenador”. Posee toda la lógica que el protocolo establece para los ordenadores. Esto incluye:
    - Mantener la lista de los grupos a los cuales pertenece en cada red y si fue el último equipo en la misma en reportar pertenencia a ese grupo (esto último es necesario para saber si cuando quiere dejar el grupo envía o no el mensaje de “leave group”).
    - Procesar los mensajes de entrada:
      * “Membership Query”. Configura los temporizadores para enviar respuestas por los grupos a los que pertenece.
      * “Membership Report”. Si recibe un reporte de otro equipo debe “marcar” que no fue el último en reportar ese grupo en esa subred.
    - Unirse o dejar un grupo. Preparar el mensaje de salida correspondiente según la acción (“Membership Report” o “Leave Group”, según el caso y si corresponde).
    - Determinar si un paquete IP está destinado al ordenador. Comprobando si la dirección de destino del paquete es una dirección de un grupo al cual está suscripto.
  + “ModuloIGMPRouter”. Posee toda la lógica que el protocolo establece para los ruters multicast. Esto incluye:
    - Mantener la lista de grupos que poseen miembros en cada una de sus subredes, con todos los timers necesarios.
    - Procesar los mensajes de entrada:
      * “Membership Query”. Para detectar si existe otro router multicast en la subred.
      * “Membership Report”. Para detectar cuando un nuevo grupo debe ser agregado a la lista de grupos de una subred.
      * “Leave Group”. Para saber si se debe quitar un grupo de la lista de grupos de una subred (previa consulta por mas miembros de ese grupo en la subred).
    - Procesar todos los timers necesarios para el protocolo.
      * Para enviar queries de membresia, tanto generales como específicos.
      * Para dar de baja grupos sin miembros.
      * Para saber si debe pasar a ser querier por no detectar actividad de otro querier.
* “MensajeGMP”. Subclase que extiende de la clase “Buffer” (que es la clase base para todos los paquetes de KivaNS). La misma contiene la toda la información que debe ser intercambiada para el funcionamiento del protocolo. La estructura de un paquete es la siguiente:

|  |  |  |
| --- | --- | --- |
| 0 | 8 | 16 |
| Tipo | T. Max. De Resp. | Checksum |
| Direccion Clase D | | |

* “RouterMultiCast”. Subclase que extiende de la clase “Router” (que es la clase que implementa la funcionalidad de los routers en KivaNS). La misma agrega a la clase base el modulo IGMP para routers, que trata todos los aspectos de la implementación del protocolo IGMP. De esta forma, se puede incluir en una simulación un router normal o uno multicast.

También se modificó la clase “Ordenador”, agregándose a la misma el soporte para el protocolo IGMP. Este soporte es dado por la funcionalidad encapsulada en la clase “ModuloIGMPOrdenador” descripta anteriormente.

Otra modificación importante fue la realizada sobre la clase “Red” (clase base para las redes implementadas, las cuales son: la red Ethernet y punto a punto), la cual permite realizar una transformación entre una dirección IP multicast a una dirección de red multicast (si la misma soporta multicasting).

Por último, se agregó una validación en la clase NivelIPv4 para que al evaluar si un paquete está dirigido al host o no, revise la lista de los grupos a los cuales el mismo pertenece y así determinar si debe ser procesado en el host o no (esta evaluación se realiza siempre y cuando la dirección de destino sea de clase D).

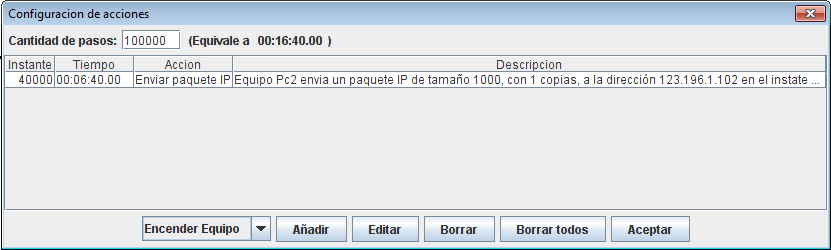
# Modificaciones en la interfaz gráfica

Las modificaciones realizadas en la interfaz gráfica apuntan principalmente a dos aspectos:

* Incluir las nuevas características del simulador extendido a todas las opciones con las que ya cuenta la herramienta
* Facilitar a los usuarios el nuevo tipo de router para que puedan incluirlo en sus simulaciones.

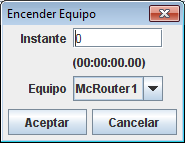
Con respecto al primer punto, se detallarán a continuación cada uno de los puntos agregados:

Se cambió el cuadro de configuración de envió de paquetes IP por un cuadro que permite configurar todos los tipos de acciones que pueden utilizarse en el nuevo simulador. En este mismo cuadro es posible configurar la cantidad de pasos que tendrá la simulación. Esta última característica no estaba disponible en la versión original de KivaNS, por lo cual podemos destacar que aporta una mayor flexibilidad a las simulaciones. En la siguiente figura se puede observar este nuevo cuadro.

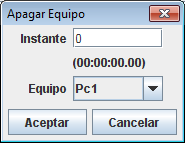


Desde este cuadro existe la posibilidad de agregar nuevas acciones, editar o borrar las que se agregaron anteriormente. Dichas acciones podrán ser configuradas en un nuevo cuadro de dialogo adaptado para permitir el ingreso de los datos propios de cada tipo de acción. Estos cuadros específicos son los siguientes:

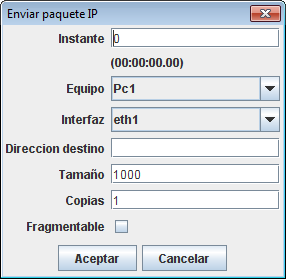
* Encender equipo: permite agregar a la simulación una acción de encender un equipo, configurando todos los datos necesarios para llevar a cabo la acción. La siguiente imagen ilustra este tipo de cuadro:



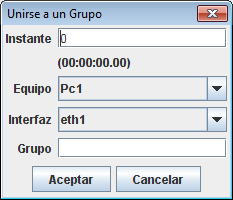
* Apagar equipo: permite agregar a la simulación una acción de apagar un equipo, configurando todos los datos necesarios para llevar a cabo la acción. La siguiente imagen ilustra este tipo de cuadro:



* Enviar paquete IP: permite agregar a la simulación una acción de enviar un paquete IP, configurando todos los datos necesarios para llevar a cabo la acción. La siguiente imagen ilustra este tipo de cuadro:



* Unirse a un grupo: permite agregar a la simulación una acción de unirse a un grupo, configurando todos los datos necesarios para llevar a cabo la acción. La siguiente imagen ilustra este tipo de cuadro:



* Dejar grupo: permite agregar a la simulación una acción de dejar un grupo, configurando todos los datos necesarios para llevar a cabo la acción. La siguiente imagen ilustra este tipo de cuadro:

