Universidad Nacional del Litoral

Proyecto Final de Carrera

Diseño de un sistema para la visualización en tiempo real de tráfico en redes informáticas.

Pineda Leandro

dirijido por
Ing. Gabriel Filippa

Repositorio git del documento (solo versión borrador):
https://github.com/leandropineda/anteproyecto

29 de mayo de 2016

Justificación

Las redes informáticas son esenciales para las operaciones diarias de cualquier empresa o institución pública. A medida que estas entidades crecen, también crece el volumen de datos que generan, demandando así una infraestructura de red de mayor porte. Debido a esto, mantener una buena performance en el funcionamiento general de un sistema que se encuentra constantemente en crecimiento se convierte en una tarea desafiante.

Algunas de las herramientas disponibles para monitoreo de redes brindan reportes basados en estadísticas. La empresa Cisco® introduce una característica en algunos modelos de routers y switches llamada NetFlow, la cual permite recolectar información sobre el tráfico de red que atraviesa las distintas interfaces de los dispositivos. Sin embargo, estar limitado exclusivamente a routers y switches significa una desventaja pues en algunas ocasiones involucra una inversión en infraestructura considerable. Otros fabricantes introducen características similares pero con nombres diferentes: Traffic Flow de MikroTik, NetStream de HP, entre otros. Otra alternativa para monitorear una interfaz de red es el analizador de paquetes *tcpdump* (disponible en sistemas GNU/Linux), la cual permite observar una descripción en texto plano del contenido de los paquetes que pasan por una interfaz de red.

Una metodología diferente de monitoreo de red se relaciona con el BigData y consiste en almacenar todos los eventos que ocurren en la red y luego generar informes utilizando procesos *batch* sobre el conjunto de datos. Esto no solo es un proceso lento sino que requiere almacenar grandes volúmenes de datos, y si bien brinda información que es de gran utilidad es sobre eventos que ocurrieron en el pasado.

Como parte fundamental de la infraestructura de cualquier tipo de empresa, la tarea de administración se ve facilitada si podemos contar con la información más actualizada posible sobre la evolución del tráfico de red. Es por esto que el monitoreo en tiempo real es un herramienta de gran utilidad pues permite contar con información importante, al instante. El proceso de toma de decisiones a la hora de invertir en infraestructura se ve facilitado con la información que puede extraerse observando la utilización de la infraestructura. Por ejemplo, una empresa podría decidir si invertir en una conexión a Internet de mayor velocidad o bloquear el acceso a algún servicio que el administrador detecta, causa gran demanda de ancho de banda. El monitoreo en tiempo real puede hacer también que el proceso de detección de problemas en una red sea rápida y sencilla.

Se propone entonces diseñar un sistema capaz de mostrar información acerca del tráfico que está atravesando una interfaz de red utilizando el sistema de procesamiento en tiempo real Apache Storm y las tecnologías asociadas al mismo. Analizando la salida de texto plano de la herramienta *tcpdump* el sistema producirá en tiempo real informes acerca de los parámetros relevantes del tráfico de red, y mostrará informes gráficos según diferentes filtros como tráfico entrante y saliente, consumo de ancho de banda por protocolo, puerto o IP entre otros. La empresa Twitter utiliza Storm para procesar grandes volúmenes de *tweets* y determinar *trends*[1]. El uso de esta tecnología hace que el sistema sea escalable y pueda adaptarse a redes que están en crecimiento constante. Otras compañías que hacen uso de Storm para proveer servicios son Yahoo!, Spotify, Yelp y Groupon por mencionar algunas importante.

Sin duda alguna, la cantidad de dispositivos conectados a Internet crece día a día y con ellos lo hace la demanda de la infraestructura de red necesaria para brindar un servicio de calidad. Contar con información del uso de las redes es fundamental para hacer estimaciones para inversiones a futuro, y permite determinar si los recursos disponibles están siendo utilizados de manera eficiente.

Finalmente, contar con esta información es una gran ayuda para mejorar la seguridad de las

[1] Trend refiere a un tópico identificado por *HashTags* que resulta sumamente popular en un momento dado.

redes. Muchos de los ataques a redes informáticas son realizados por usuarios fuera de la red que no tienen acceso físico a la misma, pero muchos de los ataques más peligrosos son llevados a cabo por usuarios internos. Tener este tipo de información sobre el tráfico de la red puede ser una herramienta de enorme valor para identificar estos sucesos.

Objetivos

Generales

Diseñar un sistema que permita mostrar en tiempo real información acerca del tráfico de red que atraviesa una interfaz de red.

Específicos

- Identificar un conjunto de tecnologías y herramientas adecuadas para el diseño del sistema.
- Construir una herramienta de software que permita visualizar como varía el tráfico de red a través de tiempo.
- Diseñar un sistema independiente de un dispositivo de hardware.
- Describir conceptualmente la arquitectura del sistema.

Alcances

Funcionales

- El sistema permitirá mostrar tiempo real el tráfico de red según criterios de filtrado dados.
- El sistema proveerá al usuario una interfaz para la visualización del tráfico de red.
- El sistema ofrecerá estadísticas generales de la red.
- Los informes mostrarán información de la Capa de Transporte y la Capa de Red[2].[1][2][3]

No Funcionales

- El sistema será escalable. Las tecnologías seleccionadas permiten configurar un cluster de nodos de procesamiento en caso que el volumen de datos a procesar aumente.
- Se proveerá un manual de instalación y configuración.

Exclusiones

El proyecto no contempla la instalación del sistema en un ámbito de producción. El sistema a desarrollar solo analizará el tráfico que atraviesa una única interfaz de red, y no contempla el caso de conexiones múltiples con balance de carga. No se implementará ningún mecanismo de detección ataques o de patrones de comportamiento sospechoso.

[2]Modelo TCP/IP

Supuestos

Los datos necesarios para realizar pruebas pueden ser generados arbitrariamente. El término *tiempo real* refiere al denominado *tiempo real blando*, es decir, no es necesario asegurar la ejecución de ciertas tareas o mostrar informes en el mismo instante en el que se generan los datos. La generación de los informes tendrá entonces demoras de cientos de milisegundos.

Criterios de Aceptación

Se considera que el proyecto está aceptado cuando cumple en un 90 % los requisitos funcionales, con un nivel mínimo de aceptación del 75 %. Los prototipos deben tener implementadas todas las funcionalidades planificadas.

Metodología

La metodología de desarrollo del proyecto será incremental e iterativa. Incremental porque varias componentes del sistema se desarrollarán en momentos diferentes y serán integradas cuando sean completadas. Iterativa pues se invertirán esfuerzos en revisar constantemente partes del sistema, tanto para mejorar la calidad externa como interna del software[4].

Con esta metodología se puede dividir el trabajo en incrementos que son revisados constantemente a medida que el proyecto se ejecuta. Luego de una investigación preliminar acerca de las tecnologías disponibles y sus características, correspondiente a la primera fase de la ejecución del proyecto, se tiene el punto de partida para comenzar el desarrollo. Dado que es necesario que el sistema cumpla con los requisitos de *tiempo real*, el incremento donde se implementan las funcionalidades esenciales de captura de paquetes es revisado constantemente para mejorar su calidad mientras se desarrollan los incrementos de informes y visualización (ver Plan de Tareas). Una vez todos los incrementos son terminados, serán integrados y se llevarán a cabo las pruebas de integración correspondientes.

Plan de Tareas

El proyecto se divide en 4 incrementos. A continuación se da una breve descripción de los mismos:

Incremento 1: Diseño La primer parte del proyecto consiste en determinar que conjunto de tecnologías serán utilizadas, y elaborar una descripción a alto nivel de las diferentes componentes del sistema y cómo se relacionan.

Incremento 2: *Core* o Núcleo En esta etapa se implementa la funcionalidad que permite capturar la información de entrada, visualizarla en texto plano y prepararla para procesamientos posteriores.

Incremento 3: Filtrado Se implementa el módulo que procesa el tráfico de la red y se implementan las funcionalidades de filtrado junto con los filtros predeterminados.

Incremento 4: Visualización En este incremento se integran las funcionalidades de captura y filtrado. Además se implementa la interfaz de usuario en la plataforma determinada en la etapa de diseño.

Al finalizar la primera iteración de cada incremento se obtiene una herramienta de software con las funcionalidades descriptas y calidad de producto final, con excepción de la etapa de diseño donde se obtendrá un informe en soporte escrito o digital.

Plan de Tareas

La duración total del proyecto es de 488 horas, con una dedicación de 20 horas semanales. A continuación se define el plan de tareas.

1. **Diseño** (84hs)
   1.1. Estudio comparativo de las tecnologías disponibles. (24hs)
   1.2. Diseño conceptual del sistema. (40hs)
   1.3. Diseño de interfaz de usuario y filtros predeterminados. (20hs)
2. ***Core* o Núcleo** (96hs)
   2.1. Instalación y configuración de servidor de aplicación. (32hs)
   2.2. Implementación de funcionalidad de captura de tráfico de red. (52hs)
   2.3. Documento de instalación. (12hs)
3. **Filtrado y visualización** (168hs)
   3.1. Implementación de funcionalidad de filtrado en capa de red. (52hs)
   3.2. Implementación de funcionalidad de filtrado en capa de transporte. (52hs)
   3.3. Implementación de filtros predeterminados. (24hs)
   3.4. Implementación de interfaz de usuario. (40hs)
4. **Integración** (140hs)
   4.1. Integración de funcionalidades e interfaz de usuario. (60hs)
   4.2. Pruebas de integración. (40hs)
   4.3. Elaboración de informe. (40hs)

Informes de avance

Se presentarán 4 informes de avance en las fechas de finalización de cada etapa, detalladas en el Cuadro 1. A continuación se detalla que información será incluida en cada informe:

Informe de avance 1 Contendrá los resultados obtenidos en los estudios comparativos de las tecnologías y justificará la elección de las mismas. Además se incluirá una descripción general de la arquitectura del sistema.

Etapa	Inicio	Finalización
Diseño	15/08/2016	16/09/2016
Core o Núcleo	19/09/2016	21/10/2016
Filtrado y visualización	24/10/2016	16/12/2016
Integración	09/01/2017	24/02/2017

Cuadro 1: Fechas estimativas de inicio y fin de actividades.

Informe de avance 2 Contendrá información sobre el desempeño del núcleo del sistema. Se proveerá la guía de instalación y configuración de la plataforma. Además tendrá información sobre cambios realizados en los entregables anteriores.

Informe de avance 3 Contendrá una descripción de los mecanismos de filtrado utilizados y los resultados obtenidos. Además tendrá información sobre cambios realizados en los entregables anteriores.

Informe de avance 4 Contendrá una información sobre los avances en el desarrollo de la interfaz de usuario. Se describirán las pruebas de integración del sistema. Además tendrá información sobre cambios realizados en los entregables anteriores.

Entregable	Fecha de entrega
Informe de avance 1	09/09/2016
Informe de avance 2	07/10/2016
Informe de avance 3	09/12/2016
Informe de avance 4	10/02/2017

Cuadro 2: Fechas de entrega de informes de avance.

Diagrama de Gantt

A continuación se muestra el diagrama de Gantt del proyecto.

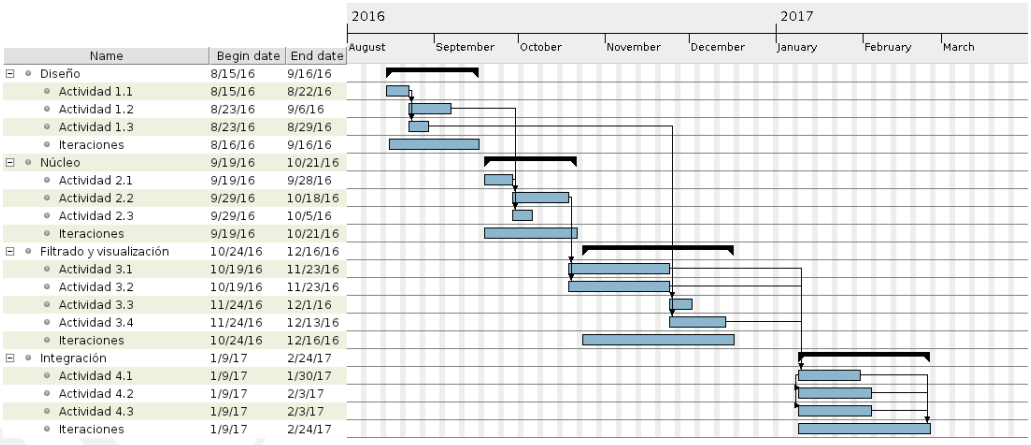

Figura 1: Diagrama de Gantt del proyecto.

Riegos

En esta sección se enumeran los riesgos identificados, indicadores y estrategia a adoptar según corresponda. Para realizar el análisis cualitativo de los riesgos se asigna una probabilidad de ocurrencia y un impacto a cada riesgo. Luego se priorizan según su *severidad*. Los riesgos identificados son:

1. **No se pueden satisfacer las restricciones de performance:** la característica principal del sistema es la posibilidad de procesar tráfico en tiempo real. Puede darse un escenario donde las tecnologías disponibles no permitan alcanzar este requisito.

   Indicador: se observan demoras o un progreso lento en las etapas de Núcleo y/o Integración.
   Probabilidad de ocurrencia: Media.
   Impacto: Muy alto.
   Estrategia a adoptar: Mitigar (probabilidad). Se revisan continuamente los entregables de la etapa de Diseño con el fin de mejorar su calidad.

2. **No se dispone del hardware necesario:** el servidor de aplicaciones necesario para el desarrollo de la aplicación no se encuentra disponible

   Indicador: la etapa de Núcleo no puede comenzar.
   Probabilidad de ocurrencia: Media.
   Impacto: Alto.
   Estrategia a adoptar: Mitigar (impacto). Se utilizará la infraestructura provista por la facultad.

3. **Los módulos del sistema no pueden ser integrados correctamente:** aunque cada uno de los módulos cumpla con los requisitos funcionales del sistema, puede ocurrir que el rendimiento se vea degradado producto de los retardos que puede introducir las interfaces de comunicación entre los módulos.

   Indicador: se observan demoras o un progreso lento en la etapa de Integración.
   Probabilidad de ocurrencia: Baja.
   Impacto: Muy alto.
   Estrategia a adoptar: Mitigar (probabilidad). Se revisan continuamente los entregables de la etapa de Integración con el fin de mejorar su calidad.

4. **Las tecnologías seleccionadas no pueden ser integradas:** se utilizarán un conjunto de tecnologías que deben coexistir para realizar el procesamiento de los datos. Es posible que se encuentren incompatibilidades entre alguna de ellas y no sea posible utilizarlas de forma conjunta.

   Indicador: se observan demoras o un progreso lento en la etapa de Diseño.
   Probabilidad de ocurrencia: Baja.
   Impacto: Alto.
   Estrategia a adoptar: Mitigar (impacto).
   Se realiza un estudio comparativo de varias tecnologías para disponer de alternativas a las determinadas inicialmente.

5. **El sistema no puede ser probado en un entorno productivo:** la implementación del sistema en un entorno productivo implica un riesgo para los administradores de las redes. Por esto, es posible que no se disponga de un escenario real para las pruebas finales de integración.

Probabilidad de ocurrencia: Baja.
Impacto: Bajo.
Estrategia a adoptar: Aceptar (activamente).
Se generarán los datos necesarios para realizar pruebas de integración.

Análisis Riesgos

En esta sección se muestra el análisis realizado de los riesgos del proyecto, luego se definen las estrategias a adoptar y los riesgos ordenados por importancia según su severidad.

	1	2	3	4	5
1					
2					
3					
4					
5					

Cuadro 3: Matriz probabilidad/impacto.

Severidad	Estrategia
menor que 4	Aceptar
5 a 15	Mitigar
16 a 25	Evitar

Cuadro 4: Estrategia a adoptar según severidad.

Riesgo	Imp.	% ocur.	Sev.
No se pueden satisfacer las restricciones de performance.	5	3	15
No se dispone del hardware necesario.	4	3	12
Los módulos del sistema no pueden ser integrados correctamente.	5	2	10
Las tecnologías seleccionadas no pueden ser integradas.	4	2	8
El sistema no puede ser probado en un entorno productivo.	2	2	4

Cuadro 5: Lista de riesgos ordenados por severidad.

Presupuesto

A continuación se detalla el presupuesto necesario para el desarrollo del proyecto. El valor de la infraestructura de hardware necesaria es aproximado y comprende una estación de trabajo para el desarrollo del sistema y un servidor dedicado con 3 interfaces de red. Las especificaciones técnicas se determinarán en la etapa de Diseño del proyecto.

Infraestructura			
Estación de trabajo			$16000.
Servidor de desarrollo			$10000.
Servicios			
Conexión a internet			$4000
Recursos humanos	**Costo por hora**	**Horas**	
Diseñador	$40	84	$3360
Desarrollador	$40	364	$14560
Tester	$40	40	$1600
Costo total			$**49520**.

Bibliografía

[1] J. Postel, *Internet Protocol*, RFC 791 (INTERNET STANDARD), Updated by RFCs 1349, 2474, 6864, Internet Engineering Task Force, sep. de 1981. dirección: `http://www.ietf.org/rfc/rfc791.txt`.

[2] ——, *Transmission Control Protocol*, RFC 793 (INTERNET STANDARD), Updated by RFCs 1122, 3168, 6093, 6528, Internet Engineering Task Force, sep. de 1981. dirección: `http://www.ietf.org/rfc/rfc793.txt`.

[3] T. Socolofsky y C. Kale, *TCP/IP tutorial*, RFC 1180 (Informational), Internet Engineering Task Force, ene. de 1991. dirección: `http://www.ietf.org/rfc/rfc1180.txt`.

[4] ISO/IEC, *ISO/IEC 9126. Software engineering – Product quality*. ISO/IEC, 2001.

[5] A. Tanenbaum, *Computer Networks*, 4th. Prentice Hall Professional Technical Reference, 2002, ISBN: 0130661023.

[6] W. Stallings, *Data and Computer Communications (5th Ed.)* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997, ISBN: 0-02-415425-3.

[7] Q. Anderson, *Storm Real-Time Processing Cookbook*. Packt Publishing, 2013, ISBN: 1782164421, 9781782164425.

[8] P. Hunt, M. Konar, F. P. Junqueira y B. Reed, "Zookeeper: wait-free coordination for internet-scale systems", en *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, ép. USENIXATC'10, Boston, MA: USENIX Association, 2010, págs. 11-11.

[9] F. Junqueira y B. Reed, *ZooKeeper: Distributed Process Coordination*. O'Reilly Media, 2013, ISBN: 9781449361280.

[10] J. Leibiusky, G. Eisbruch y D. Simonassi, *Getting Started with Storm*. O'Reilly Media, Inc., 2012, ISBN: 1449324010, 9781449324018.