



## PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito parcial para  
obtener el Título de INGENIERO DE SISTEMAS

# DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO DE REDES ORIENTADO A LA RECOLECCIÓN MASIVA DE DATOS.

Por

Br. Jesús Alberto Gómez Pérez

Tutor: Dr. Andrés Arcia-Moret

Septiembre 2015

# **Diseño e implementación de un sistema de monitoreo de redes orientado a la recolección masiva de datos.**

Br. Jesús Alberto Gómez Pérez

Proyecto de Grado — Sistemas Computacionales, 14 páginas

**Resumen:** En el presente proyecto se plantea el desarrollo de un sistema de monitoreo distribuido de enlaces críticos de redes a través de un servicio web centralizado. Este servicio además pretende hacer énfasis en la visualización de los datos recolectados a partir de pruebas periódicas. El sistema está planteado como una herramienta para facilitar el entendimiento del funcionamiento de la red y ofrecer una solución centralizada, de muy bajo costo y con componentes de hardware que puedan estar desatendidos.

**Palabras clave:** Monitoreo de redes, Calidad de servicio, Big Data, Benchmarking, Cloud

# Índice

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Antecedentes . . . . .	1
1.2	Planteamiento del Problema . . . . .	3
1.3	Justificacion . . . . .	4
1.4	Objetivos . . . . .	4
1.4.1	Objetivos Generales . . . . .	4
1.4.2	Objetivos Específicos . . . . .	4
1.5	Metodología . . . . .	5
1.6	Alcanze . . . . .	6
1.7	Estructura del Documento . . . . .	6
<b>2</b>	<b>Marco Teórico</b>	<b>7</b>
2.1	Monitoreo de Redes . . . . .	9
2.2	Principio Fin-a-Fin . . . . .	9
2.3	Métricas de calidad de un enlace . . . . .	9
2.3.1	Tiempo de ida y vuelta (RTT) . . . . .	9
2.3.2	Throughput . . . . .	9
2.3.3	Jitter . . . . .	9
2.3.4	Actividad del enlace . . . . .	9
2.3.5	Perdida de paquetes . . . . .	9
2.4	Big Data . . . . .	9
2.5	Computacion en la nube . . . . .	9
2.5.1	IaaS . . . . .	9

2.5.2	PaaS . . . . .	9
2.5.3	SaaS . . . . .	9
2.6	Herramientas usadas para el desarrollo de la aplicacion web . . . . .	9
2.6.1	Django . . . . .	9
2.6.2	Celery . . . . .	9
2.6.3	Bases de Datos . . . . .	9
2.6.4	Modelo Vista Controlador (MVC) . . . . .	9
2.6.5	Caching . . . . .	9
2.6.6	Dropbox API . . . . .	9
2.6.7	Json . . . . .	9
2.6.8	Diseño web adaptable . . . . .	9
2.6.9	Highcharts . . . . .	9
2.6.10	Google Maps . . . . .	9
2.6.11	freegeoip . . . . .	9
2.6.12	Ajax . . . . .	9
2.7	Herramientas usadas para el desarrollo del monitor de red . . . . .	9
2.7.1	APScheduler . . . . .	9
2.7.2	Ping . . . . .	9
2.7.3	Requests . . . . .	9
2.7.4	Traceroute . . . . .	9
2.8	Estado del Arte . . . . .	9
<b>3</b>	<b>Monitor de Red "Tentacle"</b>	<b>10</b>
3.1	Arquitectura . . . . .	10
3.2	Componentes . . . . .	10
3.3	Casos de uso . . . . .	10
<b>4</b>	<b>Servicio Web "Optopus Monitor"</b>	<b>11</b>
4.1	Diseño del sistema . . . . .	12
4.1.1	Arquitectura . . . . .	12
4.1.2	Estructura de la base de datos . . . . .	12
4.1.3	Flujo de navegacion . . . . .	12

4.1.4	Diseño de Pantallas . . . . .	12
4.2	Componentes . . . . .	12
4.2.1	Recolector de datos . . . . .	12
4.2.2	Generacion de visualizaciones . . . . .	12
4.3	Casos de Uso . . . . .	12
4.4	Pruebas de Rendimiento . . . . .	12
5	Conclusiones y Recomendaciones	13
	Bibliografía	14

# Capítulo 1

## Introducción

### 1.1 Antecedentes

Existen dos estrategias de monitoreo de redes: monitoreo activo o benchmarking que consiste en generar tráfico para realizar medidas y comprobar la respuesta de la red y monitoreo pasivo que consiste en escanear el tráfico de la red en ciertos puntos estratégicos para censar el tráfico en la red. El benchmarking tiene la desventaja de tener que inyectar tráfico lo cual puede entorpecer el funcionamiento normal de la red ejemplos de herramientas de benchmarking son ping, iperf y traceroute.

El monitoreo pasivo tiene la ventaja de que nos puede dar una muy buena idea del uso de la red sin embargo para mayor efectividad debe realizarse en nodos intermedios a los que muchas veces no tenemos acceso, ejemplos de herramientas de monitoreo pasivo son tcpdump y wireshark; en ambos casos hay que resaltar que es difícil tener una imagen completa de la realidad de la red.

Uno de los trabajos más importantes en el área de monitoreo de redes es el Protocolo Simple de Administración de Red o SNMP (del inglés Simple Network Management Protocol) este emergió como una de las primeras soluciones al problema de manejo de redes y se ha convertido en la solución más ampliamente aceptada debido a su diseño modular e independiente de productos o redes específicas. SNMP consiste de (1) un administrador de red, (2) una serie de dispositivos remotos monitoreados (3) bases de información de administración (MIBs) en estos dispositivos, (4) agentes remotos que

reportan la información de las MIBs al administrador de red y toman acciones si les indica y (5) un protocolo de comunicación entre los dispositivos [1].

SNMP no solo ofrece al administrador de red reportes sobre cada uno de los dispositivos administrados sino que también permite tomar acción sobre ellos proactivamente antes de que ocurran problemas o de forma reactiva para solucionar problemas cuando ocurren de forma inesperada.

La red de TVWS (TV White Spaces o Espacios blancos en el espectro radioeléctrico) de Malawi fue implementada en el marco de un proyecto para llevar Internet a áreas rurales en países en vías de desarrollo utilizando soluciones de bajo costo a través de los espacios en blanco en el espectro radioeléctrico específicamente en la banda UHF (siglas del inglés Ultra High Frequency, 'frecuencia ultra alta') [2].

Muchas veces esta red debe dejarse desatendida durante largos periodos de tiempo ya que sus nodos son de difícil acceso o es muy costoso tener a profesionales dedicados que se encarguen de su mantenimiento, este escenario hace evidente la necesidad de una solución de monitoreo de redes a distancia y de mínimo mantenimiento. Además es atractivo para el centro de monitoreo de la red poder añadir, modificar o eliminar nodos de interés de manera sencilla a la interfaz de monitoreo.

Para intentar solucionar este problema y recolectar información sobre la red de Malawi se implementó un sistema en dos partes: un monitor de red instalado en la estación base (BS) en Malawi y un servicio web remoto, el monitor en la estación base se conecta a cada uno de los nodos de la red y determina el tiempo de ida y vuelta (RTT por sus siglas en inglés) de forma automatizada en ciertos intervalos de tiempo, guarda los resultados en archivos y los coloca en una carpeta que se sincroniza a través de un servicio en la nube de tipo PaaS (Plataforma como servicio) con el servidor web, que a su vez escanea la carpeta compartida y actualiza su base de datos que puede usarse para generar gráficas de RTT promedio y determinar tiempos de actividad continuos y porcentaje de disponibilidad de servicio [3].

A pesar de que este sistema recolecta información útil y presenta gráficas muy sencillas de entender, su programación no permite agregar nuevos nodos, esto trae como consecuencia que debe ser modificado manualmente cuando la red se expande, dando lugar a la necesidad de que el servicio web se pueda expandir para dar servicio

a múltiples monitores remotos simultáneamente.

Por estos motivos proponemos la construcción de un sistema de monitoreo a gran escala que llamaremos Octopus Monitor, para hacerlo totalmente configurable y robusto además de agregar una interfaz de configuración vía web que permita manejar usuarios, agregar monitores remotos, agregar nodos y modificar los parámetros de las pruebas, todo esto apoyándonos en un sistema de archivos compartidos a través de la nube.

Mientras que el mayor valor de Malawinet Monitor es la visualización de grandes volúmenes de datos que se pueden obtener a partir de pruebas de bajo impacto de tráfico (ping, traceroute), se han realizado otros trabajos en el área de monitoreo de redes como Bowlmap; este es un sistema de monitoreo de redes a través de la visualización de mediciones para el Laboratorio Abierto Inalámbrico de Berlín (BOWL, por sus siglas en inglés). Este sistema tiene una alta flexibilidad ya que permite realizar cambios en sus pruebas existentes así como agregar pruebas totalmente nuevas y a su vez generar las visualizaciones necesarias para el análisis de dicha información, además tiene la ventaja de solo transmitir la información necesaria para cada actualización lo que acelera las peticiones y permite la visualización de data en tiempo real [4].

## 1.2 Planteamiento del Problema

Las redes de computadoras se componen de un conjunto de nodos interconectados sujetos a numerosos factores que escapan de nuestro control y sobre los que muchas veces no tenemos conocimiento, en otras palabras la red puede llegar a ser impredecible y no ofrece garantías sobre el servicio que ofrece; algunas aplicaciones dependen de una alta disponibilidad y estabilidad por lo que es esencial para un administrador de red tener información del estado de la red, para diagnosticar, solucionar problemas y asegurar la calidad de servicio.

Existen muchos otros ejemplos en los que es importante tener datos del estado de la red como en redes de bajo costo en las que pueden ocurrir largas interrupciones de servicio o para un cliente de un servicio de alojamiento web que desea saber si su sitio web está disponible y que tan rápido responde; plataformas como Pingdom [5] o



UptimeRobot [6] permiten monitorear distintos servicios en Internet y generan alertas cuando encuentran problemas, sin embargo no son gratuitas y no permiten la inclusión de nuevos tipos de pruebas o la visualización masiva de datos históricos.

Mantener estas mediciones con las herramientas existentes se vuelve una tarea compleja mientras crece el número de nodos a monitorear (es decir, las fuentes de información) y la cantidad de datos aumenta a través del tiempo, sumado a esto solo podemos capturar información a partir de los nodos externos de la red, por lo que en la mayoría de los casos no es posible tener una imagen completa de los enlaces a monitorear.

A pesar de que Malawinet Network Monitor podría ofrecer estadísticas de tiempo de ida y vuelta (RTT) y disponibilidad de un enlace solo a partir de las trazas capturadas con Ping, se desea además implementar un marco de trabajo que permita agregar nuevas pruebas automatizadas que ayuden a obtener una imagen más completa de la red.

## **1.3 Justificacion**

## **1.4 Objetivos**

### **1.4.1 Objetivos Generales**

Construir un servicio web de monitoreo de redes de bajo costo para países en vías de desarrollo con almacenamiento de datos en la nube de tipo PaaS que sea de fácil instalación y permita configurar múltiples monitores remotos para ajustarse a cambios en las características de las redes a monitorear y la carencia de personal in sitio.

### **1.4.2 Objetivos Específicos**

- Desarrollar un servicio de monitoreo de bajo costo para países en vías de desarrollo que de servicio a múltiples monitores de red remotos, presente visualizaciones gráficas a partir de los datos recogidos y ofrezca un marco de trabajo para agregar nuestros tipos de pruebas a los monitores de red existentes.

- Desarrollar un cliente monitor para desplegar en nodos desatendidos con dispositivos recolectores de muestra de bajo costo (ej. Raspberry PI, Alix boards, APU) para observar el comportamiento de los enlaces a través de aplicaciones de monitoreo sencillas y de consola.
- Utilizar de sistemas de bajo costo y alta disponibilidad en la nube para almacenamiento y transferencia de datos.
- Integrar los distintos subsistemas que conforman el servicio de monitoreo.
- Desarrollar un modulo de calculo asíncrono de gráficas que permita mejorar los tiempos de interacción del usuario final con el sistema utilizando técnicas para agilizar cómputo como caching, prefetching, threads, etc.

## 1.5 Metodología

En este trabajo se seguirá una metodología en espiral; el modelo en espiral es un modelo del ciclo de vida del software donde el esfuerzo del desarrollo es iterativo. Cada ciclo de la espiral representa una fase del desarrollo de software, cada uno de los ciclos consiste de los siguientes pasos:

1. Determinar o fijar los objetivos. En este paso se definen los objetivos específicos para posteriormente identificar las limitaciones del proceso y del sistema de software, además se diseña una planificación detallada de gestión y se identifican los riesgos.
2. Análisis del riesgo. En este paso se efectúa un análisis detallado para cada uno de los riesgos identificados del proyecto, se definen los pasos a seguir para reducir los riesgos y luego del análisis de estos riesgos se planean estrategias alternativas.
3. Desarrollar, verificar y validar. En este tercer paso, después del análisis de riesgo, se eligen un paradigma para el desarrollo del sistema de software y se lo desarrolla.
4. Planificar. En este último paso es donde el proyecto se revisa y se toma la decisión si se debe continuar con un ciclo posterior al de la espiral. Si se decide continuar, se desarrollan los planes para la siguiente fase del proyecto.

Se realizarán cuatro ciclos, el primero corresponde a la realización de un monitor remoto básico que realice mediciones de RTT de la red con almacenamiento en la nube y visualizaciones de los datos obtenidos.

El segundo ciclo consiste en permitir el monitoreo de una cantidad arbitraria de monitores remotos permitiendo a múltiples usuarios manejar sus monitores remotos desde el servicio web y observar las visualizaciones.

El tercer ciclo corresponde en diseñar e integrar una prueba con otras herramientas (traceroute, iperf, etc) para conseguir puntos comunes y generar un enfoque de integración sencillo de los wrappers futuros a las aplicaciones. (ej. Lidar con aplicaciones que requieren enfoque cliente solo [ping] o cliente-servidor [iperf]).

El cuarto ciclo consiste en hacer análisis del rendimiento del sistema y hacer las optimizaciones necesarias para ofrecer una calidad de servicio apropiada, determinar costos, limitaciones y requisitos mínimos para implementar en países en vías de desarrollo.

## 1.6 Alcanze

## 1.7 Estructura del Documento

# Capítulo 2

## Marco Teórico

En este capítulo se presentan los conceptos necesarios para la comprensión de este documento y se describen las herramientas de software y formatos que se emplearán para el desarrollo del sistema propuesto.



## 2.1 Monitoreo de Redes

## 2.2 Principio Fin-a-Fin

## 2.3 Métricas de calidad de un enlace

### 2.3.1 Tiempo de ida y vuelta (RTT)

### 2.3.2 Throughput

### 2.3.3 Jitter

### 2.3.4 Actividad del enlace

### 2.3.5 Perdida de paquetes

## 2.4 Big Data

## 2.5 Computacion en la nube

### 2.5.1 IaaS

### 2.5.2 PaaS

### 2.5.3 SaaS

## 2.6 Herramientas usadas para el desarrollo de la aplicacion web

### 2.6.1 Django

### 2.6.2 Celery

### 2.6.3 Bases de Datos

### 2.6.4 Modelo Vista Controlador (MVC)

### 2.6.5 Caching

## Capítulo 3

# Monitor de Red "Tentacle"

### 3.1 Arquitectura

### 3.2 Componentes

### 3.3 Casos de uso





## Capítulo 4

# Servicio Web "Optopus Monitor"

### 4.1 Diseño del sistema

#### 4.1.1 Arquitectura

#### 4.1.2 Estructura de la base de datos

#### 4.1.3 Flujo de navegacion

#### 4.1.4 Diseño de Pantallas

### 4.2 Componentes

#### 4.2.1 Recolector de datos

#### 4.2.2 Generacion de visualizaciones

Cálculo de Mapas de Calor de RTT

Cálculo de horas activas

Cálculo de días activos

Cálculo de periodos de actividad continua

### 4.3 Casos de Uso

### 4.4 Pruebas de Rendimiento

## Capítulo 5

### Conclusiones y Recomendaciones

# Bibliografía

- [1] J. F. Kurose and K. W. Ross, *Computer Networking. A Top-Down Approach*. Pearson, 2013.
- [2] M. Zennaro, E. Pietrosevoli, J. Mlatho, M. Thodi, and C. Mikeka, “An assessment study on white spaces in malawi using affordable tools,” in *Global Humanitarian Technology Conference (GHTC), 2013 IEEE*, (San Jose, CA), pp. 265 – 269, IEEE, 2013.
- [3] J. Gomez, M. Porcar, M. Hernandez, and E. Velasquez, “Malawinet network monitor,” tech. rep., Universidad de los Andes, 2015.
- [4] B. Vahl, T. Luque, F.H.and Huhn, and C. Sengul, “Network monitoring and debugging through measurement visualization,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium*, (San Francisco, CA), pp. 1 – 6, IEEE, 2012.
- [5] S. Cloud, “Pingdom - website monitoring.” [Página web en línea] Disponible en <https://www.pingdom.com/>, 2015.
- [6] U. R. B. A.S., “Uptime robot.” [Página web en línea]. Disponible en : <https://uptimerobot.com/>, 2015.