



INSTITUTO TECNOLÓGICO DE COSTA RICA

REDES DE COMPUTADORAS

QR-Net

Alejandro Rojas

Luis I. Cubero

*Profesor:
Kevin Moraga*

Mayo 2019

1. Introducción

La libertad de expresión y el anonimato en la telecomunicación, son dos de los principales obstáculos que enfrentamos actualmente, a pesar que hoy 2019 nos encontramos en un momento privilegiado en la historia, hay países donde aún se siguen teniendo problemas de libertad de expresión. Desde el poder comunicarse debido a que las TELCOS locales no les interesa invertir en pueblos donde tiene pérdidas, lugares donde el gobierno calla a sus habitantes y hasta lugares donde la privacidad es una cuestión de vida o muerte. Países como Corea del Norte, donde el gobierno no permite acceso general a internet debido a sus políticas aislacionista, usando en cambio una red interna nacional llamada Kwangmyong [1]; o en Estados Unidos, donde organizaciones como la NSA monitorean constantemente las comunicaciones de los habitantes, con la excusa de protegerlos de amenazas externas [2]. El objetivo del proyecto es conocer distintas soluciones que nos permitan acercarnos más a la libertad de expresión, y el acceso a la información de manera anónima y segura, permitiendo que podamos crear redes independientes y privadas que se pueden comunicar a grandes distancias de forma creativa.

2. Ambiente de desarrollo

Herramientas utilizadas en este proyecto:

- Linux 64bits
- Python 3.6
- PIP3 9.0.1
- Scapy 2.4.2
- Libdnet
- Python-Dumbnet
- Mesh-Networking
- OpenCV
- Qrcode

3. Estructuras de datos usadas y funciones

3.1. QRNET

- IRCProgram es la clase encargada de iniciar y recibir el IRC.
- RouterProgram es la clase encargada de recibir y enrutar paquetes (por ahora solo envía a un link).

- NodeProgram es la clase que levanta el nodo (cliente) para recibir paquetes.

3.2. QR

- DispositivoLuzAdaptador es la clase encargada de la creación y lectura de códigos QR, esta clase usa funciones para desarmar el archivo o el paquete si es mayor a 128 bytes.
- Chunk file es el encargo de dividir el archivo en pequeños chunks.
- Decode es el encargado de leer el código QR, si son muchos códigos puede volver a unir los bytes y recrear el archivo.

3.3. Links

- VirtualLink es la clase encargada de crear conexiones entre nodos, capaz de enviar y recibir paquetes.
- UDPLink es la clase encargada de enviar un broadcast a todos los dispositivos conectados.
- IRCLink es la clase que conecta el IRC channel con los nodos, puede enviar y recibir mensajes con el irc server.

3.4. Nodes

- Node es la clase encargada de representar una computadora.

- Los nodos pueden procesar tráfico entrante, darle el paquete al programa correspondiente y procesarlo para enviar una respuesta.

3.5. Programs

- BaseProgram es la clase encargada de representar un programa corriendo dentro de un nodo.
- La clase contiene 5 hijos, entre ellos los programas encargados de routear, imprimir o almacenar los paquetes.

3.6. Filters

- BaseFilter sirve como los filtros de iptables, aplicados a todos los paquetes que entran o salen.
- La clase contiene filtros predefinidos como por ejemplo evitar enviar los paquetes repetidos.

4. Instrucciones para ejecutar el programa

Para ejecutar el programa se necesita:

- Ejecutar en terminal: `sudo python3 QRNET.py`
- En el explorador acceder a <https://webchat.freenode.net/>
- Entrar al canal `##QRNETChat`

5. Actividades realizadas por el estudiante

Bitácora			
Estudiante	Descripción	Horas	Fecha
Alejandro	Instalación de ambiente y herramientas	3	10-5-19
Luis	Instalación de ambiente y herramientas	3	11-5-19
Luis	Creación de códigos QR y creación de la trama	5	11-5-19
Alejandro	Instalación del Mesh Network	4	11-5-19
Alejandro	Comunicación básica entre nodos en el Mesh	4	12-5-19
Alejandro	Instalación y comunicación con servidor IRC	3	19-5-19
Luis	Creación del escáner de códigos QR y rearmado de archivos	8	20-5-19
Luis	Unir código de red mesh con QR	7	20-5-19
Alenjandro	Unir código de red mesh con QR	7	20-5-19

6. Comentarios finales (estado del programa)

El estado final del proyecto permite comunicación punto a punto desde el chat IRC, a través de un nodo de ruteo, el cual puede enviar ya sea a un nodo de usuario local, ya sea por WiFi o Ethernet, o al dispositivo de transmisión. El dispositivo luz adaptador genera una imagen animada con códigos QR, con un tamaño máximo de 128 bytes, la cual es leída por el dispositivo opuesto y decodificada para transmitir el mensaje hacia otro nodo de ruteo, el cual realiza la transmisión a los destinatarios.

Encontramos las siguientes limitaciones en el estado del programa, la ventana que muestra el qr se debe cerrar manualmente (nos dimos cuenta muy tarde de la limitación de la librería), el anonimato no pudo ser desarrollado, nntp no pudo ser desarrollado, el RFC no pudo ser realizado ya que el protocolo no está definido como tal ya que se está usando un protocolo de comunicación predefinido por la librería.

7. Conclusiones y recomendaciones del proyecto

Es posible crear redes privadas con diversas funciones con facilidad, demostrando que no estamos atados a un solo medio de comunicación ni a una sola forma de transmisión. La implementación de un proyecto como este a gran escala es posible. El anonimato se puede encontrar fácilmente con este proyecto, un correcto uso de paquetes y encriptación habilita esta opción.

Queda demostrado que la comunicación a distancia a través de luz es posible, los QRs son un ejemplo de ello y las señales con luz infraroja es otro gran ejemplo en la vida cotidiana. Al igual que las VPNs, este tipo de redes pueden ser utilizadas como medios alternativos para el acceso a la red de redes, el límite para este tipo es el tiempo y la creatividad de los desarrolladores.

Las recomendaciones para este proyecto es dedicarle mucho más tiempo para cumplir con los requerimientos especificados, con dos personas con tiempo limitado (trabajan) el proyecto llega a complicarse. Las especificaciones están abiertas a dudas e inquietudes que deberían manejarse con alguna metodología ágil de desarrollo, divide y vencerás.

Referencias

- [1] Look at how bizarre north korea's 'internet' is. (2019).
<https://www.businessinsider.com/a-look-at-north-koreas-tightly-controlled-internet-services-2014-12>.
- [2] E G., MacAskill. Nsa files decoded: Edward snowden's surveillance revelations explained.
<https://www.theguardian.com/world/interactive/2013/nov/01/snowden-nsa-files-surveillance-revelations-decodedsection/1>, 2019.