

Proyecto 3

Detección y Corrección de Errores

Para esta 3ra etapa del proyecto de redes incorporaremos la posibilidad de que dos *host* se comuniquen y el receptor pueda verificar si la información recibida es correcta.

Elementos a implementar

1. En las *tramas* (*frames*) antes había un campo extra que era 00000000 siempre. A partir de ahora, este campo va a contener el tamaño de la **información de verificación**. Esta información adicional vendrá después de los datos. El frame quedará de la siguiente forma:
 - Primeros 16 *bits*: dirección *MAC* destino. (igual que antes)
 - Próximos 16 *bits*: dirección *MAC* origen. (igual que antes)
 - Próximos 8 *bits*: tamaño de los datos (en *bytes*) de los datos. (igual que antes)
 - Próximos 8 *bits*: tamaño de los datos de verificación (en *bytes*) (**esto es nuevo**)
 - A continuación vienen los datos (tamaño especificado en el 3er campo). (igual que antes)
 - Después de los datos vienen los datos de verificación (tamaño especificado en el 4to campo) (**esto es nuevo**)

Detalles de los algoritmos de detección de errores

- A partir de ahora cada *host* agregará al final de cada *trama*, información que servirá al *host* de destino para verificar y comprobar si los datos llegaron correctamente.
- Existen muchas técnicas y algoritmos posibles de los cuales ustedes escogerán (o inventarán) una o varias y las implementarán en el sistema.
- Estas técnicas deberían ser capaces de detectar si hubo algún cambio en los datos que se transmitieron.
- Una técnica simple y bastante común que pueden implementar es crear una especie de *hash* de los datos y enviarlos a continuación de los datos. Y el receptor, lee los datos, aplica la misma función *hash* y comprueba el resultado.
- Por ejemplo, **el hash más simple posible** es una suma de cada uno de los valores. Supongamos que la *data* es AABBCDD (tamaño 4 *bytes* recuerden que se especifica en hexadecimal y un 1 *byte* tiene 2 dígitos hexadecimales):
 - 1er *byte* = AA (170 en decimal)
 - 2do *byte* = BB (187 en decimal)
 - 3er *byte* = CC (204 en decimal)
 - 4to *byte* = DD (221 en decimal)
 - suma total = 782
 - necesitas 2 *bytes* para transmitir eso y por lo tanto
 - el campo extra tendría valor 2 (00000010) lo cual indica que cuando se terminen de transmitir los datos, vienen 2 *bytes* más
 - al final de los datos transmites 2 *bytes* con valor 782 (00000011 00001110)

- el receptor lee los datos, lee el campo extra, suma los valores de los datos y le tiene que dar 782.
- Cada algoritmo que ustedes implementen tienen que quedar bien documentado en el directorio docs/ del proyecto.
- **En caso de que decidan implementar uno solo:**
 - Cada *host* transmite los frames y rellena esa información de verificación utilizando ese algoritmo.
 - Y cada *host* recibe y analiza los datos usando ese mismo algoritmo.
- **En caso de que decidan implementar más de un algoritmo:**
 - En el fichero *config.txt* de su proyecto tienen que incluir un elemento configurable con keyword: **error_detection** cuyo valor corresponda con alguna de las diferentes técnicas implementas por ustedes:
 - Por ejemplo supongamos ustedes implementan un algoritmo llamado *CRC*.
 - E implementan además otro algoritmo llamado *Hamming*.
 - Ustedes tienen que documentar ambos algoritmos en el directorio docs/
 - Además tienen que poner en el fichero *config.txt* una entrada que sea `error_detection = crc`.
 - En ese caso, los *hosts* a la hora de transmitir o recibir utilizarán el algoritmo *CRC*.
 - Y si yo voy al fichero *config.txt* y donde dice “crc” pongo “hamming” entonces los *hosts* a la hora de transmitir y recibir información utilizarán esta otra técnica.

Instrucciones

No hay ninguna instrucción nueva en este 3er proyecto.

Salida

- En el fichero *_data.txt* asociado a cada *host* se pondrá un **ERROR** al final de cada línea en los casos en los que los datos llegaron corruptos y ustedes no pudieron recuperarlos.
 - Si llegan los datos correctamente ustedes escriben en el fichero lo mismo que antes.
 - Si llegan datos corruptos, pero los algoritmos que ustedes implementan les permiten recuperar los datos, ustedes escriben lo mismo que antes en el fichero.
 - Si ustedes no son capaces de reconstruir la data original de ninguna forma, solo pueden saber que los datos están mal, entonces en la línea correspondiente ponen **ERROR** al final.

Objetivo del proyecto

Implementar una o varias técnicas de detección y corrección de errores en el sistema.

Hasta el Momento

Entregables

Archivo `steve_rodgers_311_natasha_romanov_312.zip` subido al EVEA que contenga:

- `README.txt` con información para compilar y ejecutar el programa
- archivo `script.txt`
- archivo `config.txt`
- directorio `docs/`
 - decisiones tomadas en la capa física
 - decisiones tomadas en la capa de enlace
 - algoritmos de detección y corrección de errores
 - cualquier otra cosa que nos ayude a entender mejor el proyecto
- directorio `output/`
 - un archivo `.txt` por cada *dispositivo*
 - un archivo `_data.txt` por cada *host*

Elementos Configurables

- [1] `signal_time` = <time in ms>
- [3] `error_detection` = <algorithm name>

Elementos Físicos

- [1] *Cable*
- [1] *Computadora / Host*
- [1] *Concentrador / Hub*
- [2] *Cable Duplex*
- [2] *Conmutador / Switch*

Conceptos

- [2] Dirección *MAC*
- [2] *Trama / Frame*
- [3] Detección y Corrección de *Errores*

Instrucciones

- [1] <time> `create host` <name>
- [1] <time> `create hub` <name> <number of ports>
- [1] <time> `connect` <port1> <port2>
- [1] <time> `disconnect` <port>
- [1] <time> `send` <host> <data>
- [2] <time> `create switch` <name> <number of ports>
- [2] <time> `mac` <host> <address>
- [2] <time> `send_frame` <host> <mac destination> <data>