|  |  |  |
| --- | --- | --- |
| 4 | **COMUNICACIONES II** |  |
| Ingeniería en Informática - Licenciatura en Informática Programador Universitario | Mg. Ing. Hugo Ortega Esp. Ing. Luis Ortíz |
| **Trabajo práctico N°4** | Fecha:12/09/2025 |
| Tema: | **Conmutación de circuitos y paquetes** | |

1. Explique el punto débil del siguiente razonamiento: la conmutación de paquetes requiere que a cada paquete se le agregue bits de control y de dirección, lo que provoca un coste adicional en esta técnica. En conmutación de circuitos se establece un circuito transparente, no siendo necesario el uso de bits suplementarios. Entonces :

“No existe por tanto coste adicional en la técnica de conmutación de circuitos, por lo que la utilización de la línea es más eficiente que en la conmutación de paquetes.”

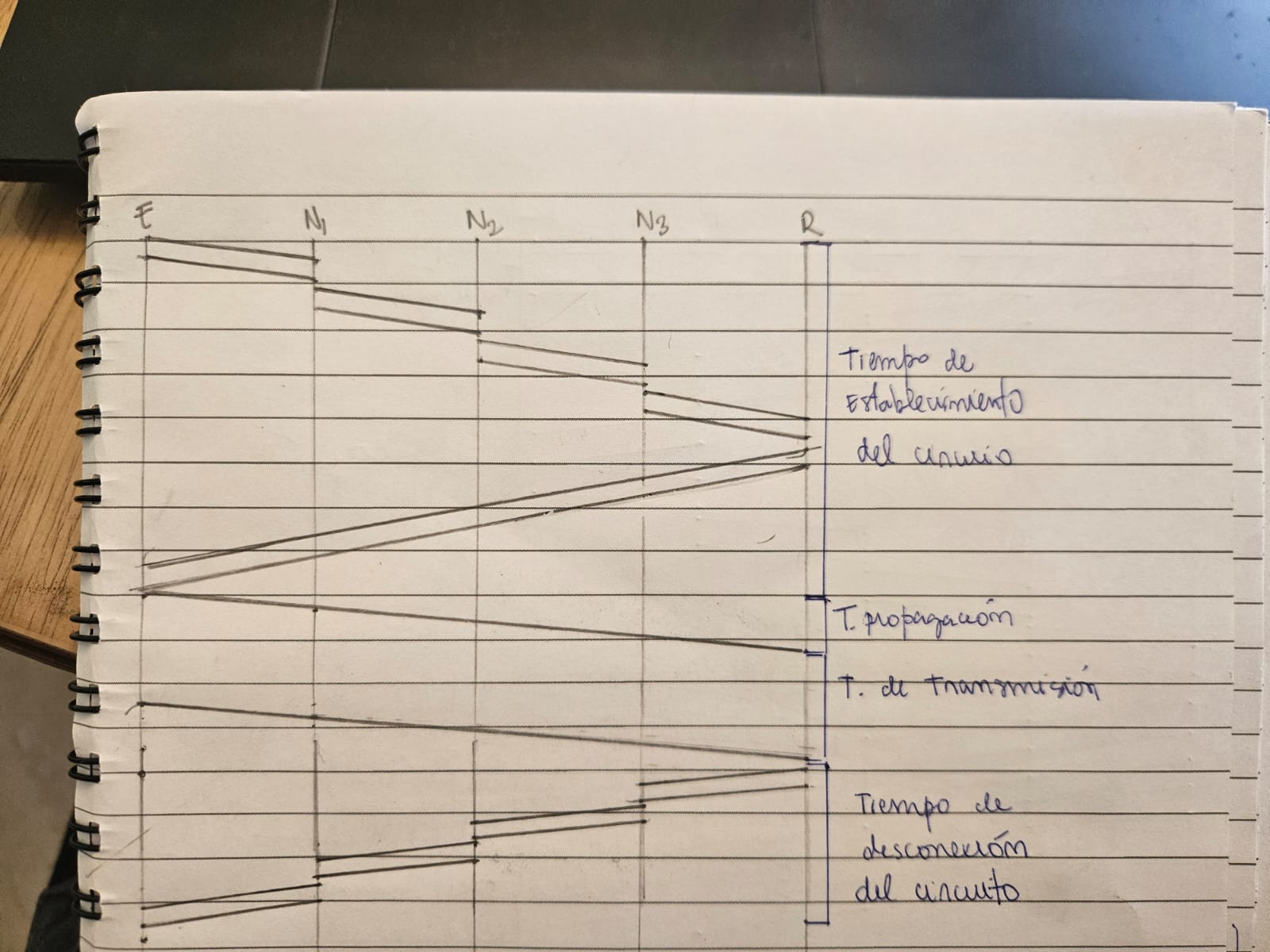
Este razonamiento sólo está considerando para analizar la eficiencia del canal el overhead de datos generados por ambas técnicas. Sin embargo, si analizamos la eficiencia como la maximización del uso de los recursos de una red, la conmutación de circuitos resulta ineficiente ya que reserva los recursos para una comunicación dedicada entre dos nodos. Además, esa reserva implica que durante todo el tiempo que dura la comunicación, los recursos de la red (ancho de banda, canales de transmisión, buffers intermedios, etc.) quedan bloqueados para los dos extremos, aunque no se estén enviando datos en ese momento. Esto genera tiempos muertos en los que la línea permanece ociosa pero igualmente ocupada, lo cual reduce la eficiencia global de la red cuando hay muchos usuarios.

Además, esa reserva implica que durante todo el tiempo que dura la comunicación, los recursos de la red (ancho de banda, canales de transmisión, buffers intermedios, etc.) quedan bloqueados para los dos extremos, aunque no se estén enviando datos en ese momento. Esto genera tiempos muertos en los que la línea permanece ociosa pero igualmente ocupada, lo cual reduce la eficiencia global de la red cuando hay muchos usuarios.

1. Se quiere transmitir un fichero de 10 MBytes por una red que admite conmutación circuitos. El fichero debe atravesar dos conmutadores intermedios de circuitos . ¿Cuál es el tiempo total de intercambio de datos en Conmutación de circuitos?: Los tiempos de establecimiento y liberación del circuito son de 1 segundo. La capacidad de los enlaces es de 100 Mbps y el tiempo de procesamiento es despreciable. Cada uno de los enlaces mide 1 km, y la velocidad de propagación de la luz en los enlaces es de 250.000 km/seg

Grafique los eventos temporales propios del establecimiento de una comunicación:

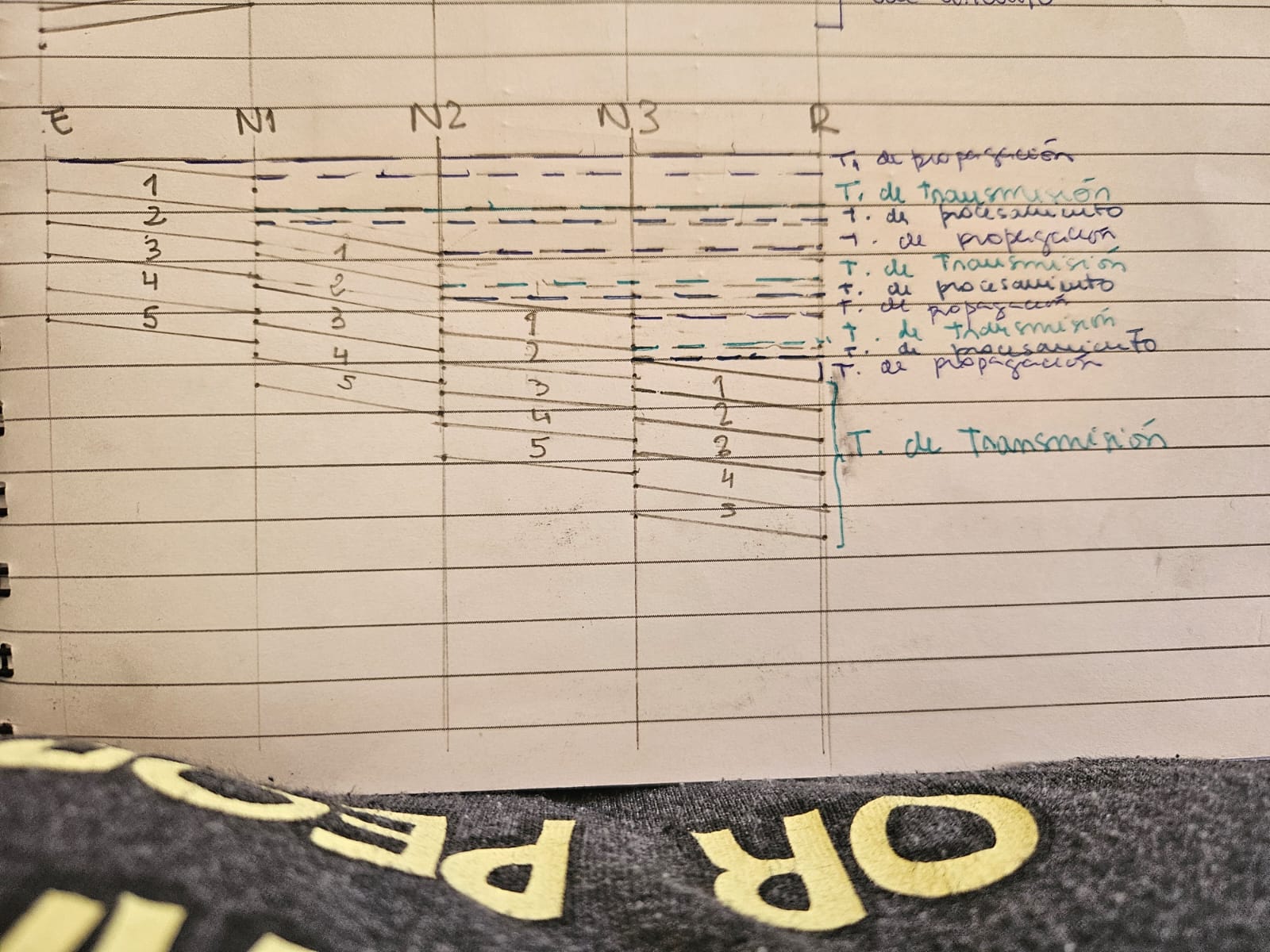
* 1. En una red de conmutación de circuitos.



Tiempo total de la comunicación = (tiempo de establecimiento + tiempo de desconexión del circuito) + tiempo de propagación de datos + tiempo de transmisión de datos

Tiempo total = 1 s + 4 km / 250 000 km/s + 80 Mb/100 Mbps = 1 s + 0.000016 s + 0.8 s = 1.800016 s

* 1. En una red de conmutación paquetes con datagramas.



Por simplicidad del gráfico se consideraron 5 paquetes.

Tiempo total = N°de saltos\*tiempo de propagación + N°nodos intermedios\*tiempo de transmisión + N°nodos intermedios\*tiempo de procesamiento + cantidad de paquetes\*tiempo de transmisión.

En este caso, se considera el tiempo de procesamiento despreciable.

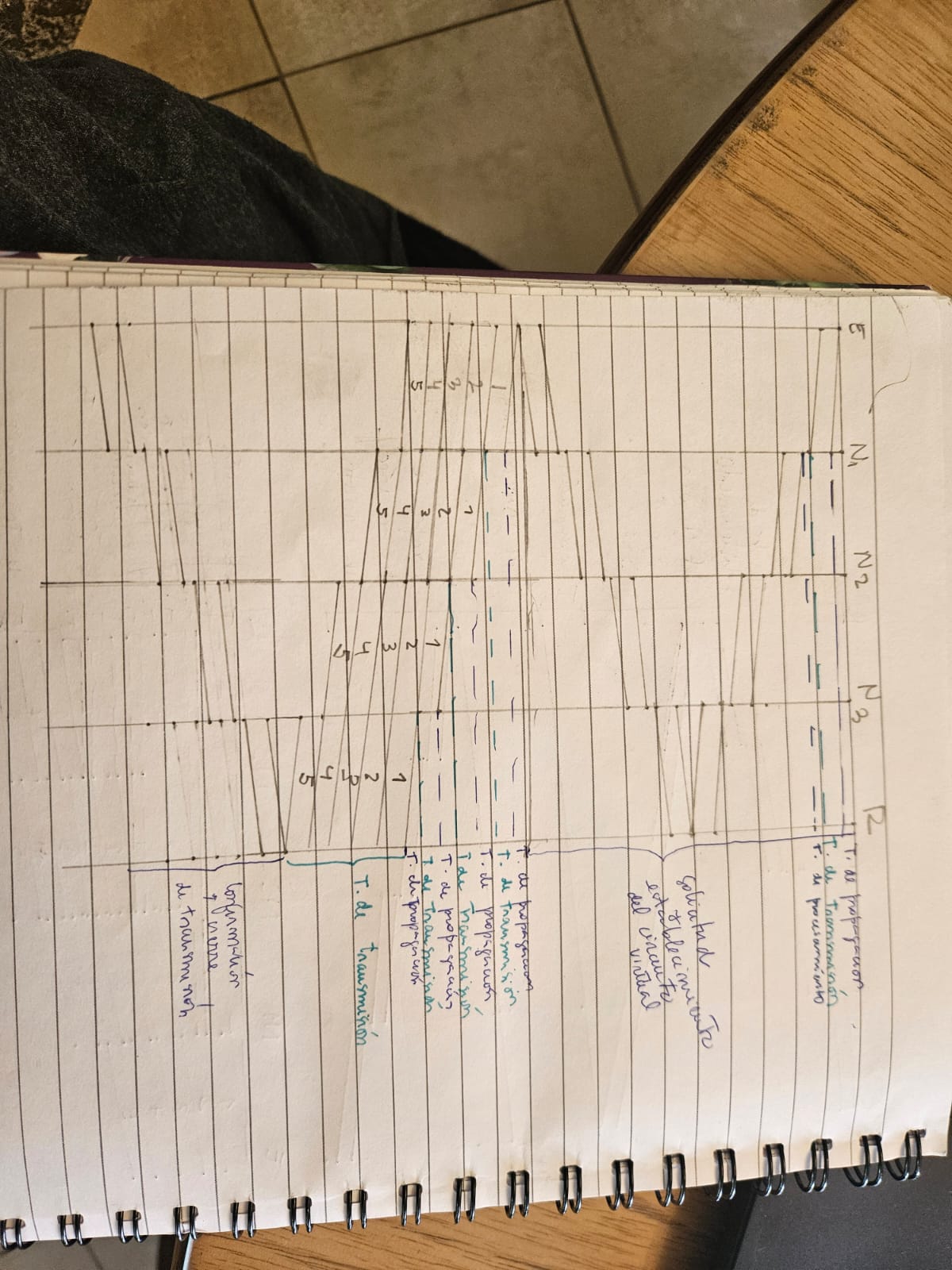
Además, consideraremos que cada 2,2 Mb de paquete se transmiten 2 Mb de datos efectivos.

Luego, como se deben transmitir 80 Mb, se enviarán 40 paquetes de 2,2 Mb cada uno.

Tiempo total = 4\*1 km/250000 km/s + 3\*2,2 Mb/100 Mbps+ 40\*2,2 Mb/100 Mbps

Tiempo total = 0.946016 s

* 1. En una red de conmutación paquetes con circuito virtual.



Tiempo total = (tiempo de establecimiento + tiempo de desconexión del circuito) + N°de saltos\*tiempo de propagación + N°de nodos intermedios\*tiempo de transmisión + cantidad de paquetes\*tiempo de transmisión.

Tiempo total = 3\*(N°de saltos\*tiempo de propagación + N°saltos\*tiempo de transmisión+ N°nodos intermedios\*tiempo de procesamiento) + N°de saltos\*tiempo de propagación + N°de nodos intermedios\*tiempo de transmisión + cantidad de paquetes\*tiempo de transmisión.

En este caso, se considera el tiempo de procesamiento despreciable.

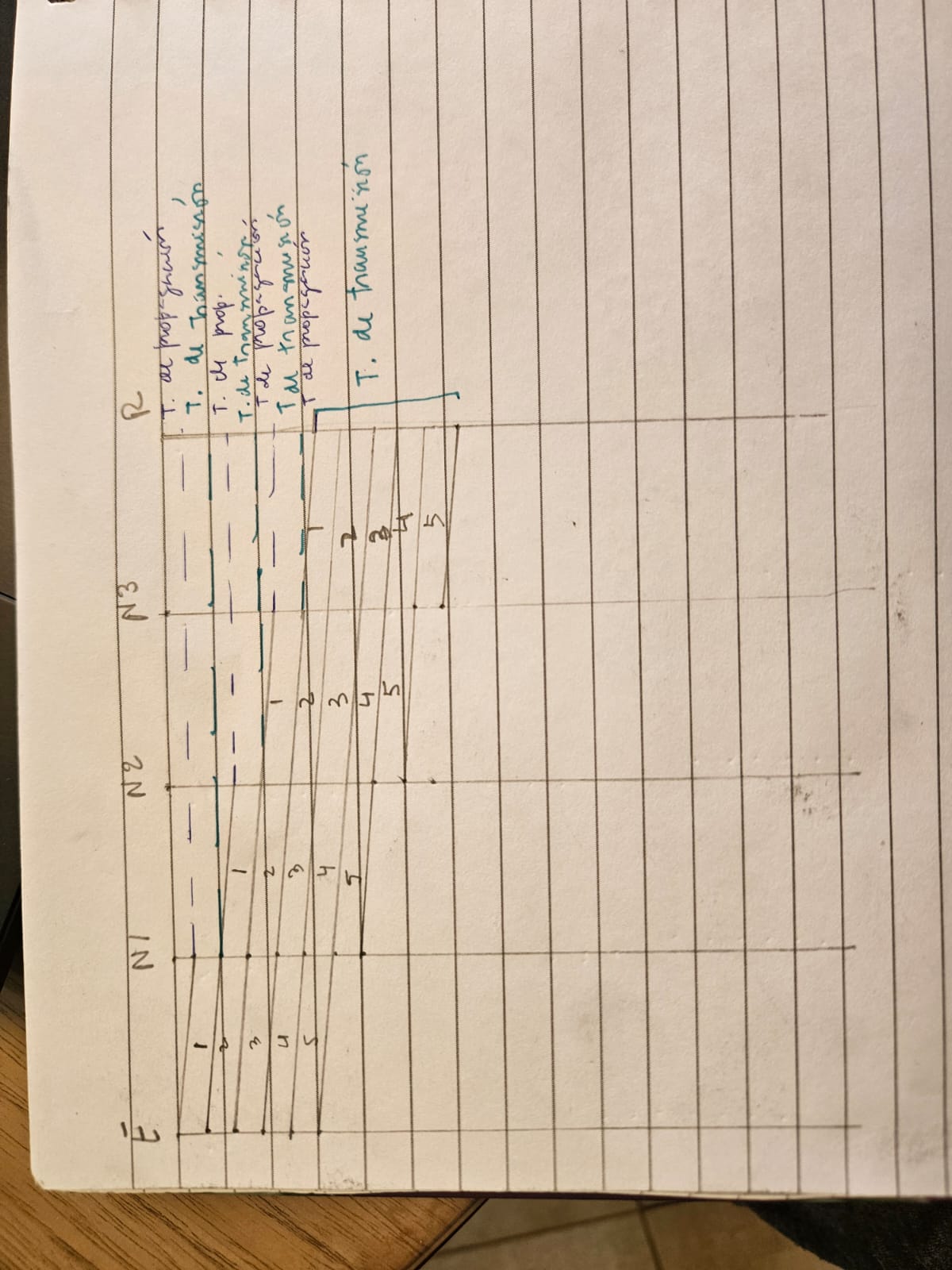
Además, consideraremos que cada 2,2 Mb de paquete se transmiten 2 Mb de datos efectivos.

Luego, como se deben transmitir 80 Mb, se enviarán 40 paquetes de 2,2 Mb cada uno.

Tiempo total = 3\*( 4\*1 km/250000 km/s + 4\*Mb/100 Mbps) + 4\*1 km/250000 km/s + 3\*2,2 Mb/100 Mbps) + 40\*2,2 Mb/100 Mbps

Tiempo total = 1.210064 s

* 1. En una red de conmutación paquetes con circuito virtual Permanente



Tiempo total = N°de saltos\*tiempo de propagación + N°nodos intermedios\*tiempo de transmisión + cantidad de paquetes\*tiempo de transmisión.

Además, consideraremos que cada 2,2 Mb de paquete se transmiten 2 Mb de datos efectivos.

Luego, como se deben transmitir 80 Mb, se enviarán 40 paquetes de 2,2 Mb cada uno.

Tiempo total = 4\*1 km/250000 km/s + 3\*2.2 Mb/100 Mbps + 40 \* 2.2 Mb/100 Mbps

Tiempo total = 0.946016 s

Nota: Suponga en todos los casos que para llegar del emisor E al receptor R los paquetes pasan por los nodos 1,2 y 3 secuencialmente. Incluye retardos de transmisión, propagación y procesamiento.

1. Dada una red de conmutación de paquetes en la cual se utilizan 2 Bytes para información de control sin incluir la referente a direccionamiento y 10 Bytes para datos. Si un identificador de CV ocupa 1 Byte y una dirección de host ocupa 4 Bytes. Suponga que los paquetes de establecimiento pueden entrar en un paquete normal, las velocidades, retardos y caminos en ambos casos son iguales para todos los paquetes.
   1. ¿Cuál es la longitud mínima de un mensaje para que fuese conveniente que esta red use circuitos virtuales?

Analicemos el tamaño del paquete para cada técnica:

Paquete de CV = 2 bytes de control + 10 Bytes de datos + 1 byte de id de CV = 13 bytes

Paquete Datagrama = 2 bytes de control + 10 bytes de datos + 8 bytes de direcciones = 20 bytes.

Entonces, cada paquete de CV ahorra 7 B respecto al datagrama.

El costo adicional de circuito virtual es la establecimiento, confirmación y finalización del circuito. Para ello envía un paquete como lo hace datagrama.

El circuito virtual necesita:

* Establecimiento: un paquete con direcciones (20 B).
* Confirmación: un paquete con Id de CV (13 B).
* Finalización: otro paquete con Id de CV (13 B).

Establecimiento/confirmación/finalización CV = 20 bytes + 13 byte + 13 bytes. = 46 bytes

Por cada paquete de circuito virtual me ahorro 7 bytes de datagrama. Necesito averiguar cuantos paquetes necesito enviar para compensar los 46 bytes extras para el establecimiento del circuito.

Luego, 46 bytes / 7 bytes ≈ 6,22 -> redondeamos a 7

La mayor cantidad de datos que yo puedo transmitir en 6 paquetes es 60 bytes. Luego, a partir de 61 bytes de mensaje es conveniente que la red use circuito virtual.

* 1. ¿Qué sucedería en el caso anterior si el tamaño de la carga útil fuese 10 veces mayor?

Analicemos el tamaño del paquete en este caso donde la carga útil es igual a 100 B:

Paquete de CV = 2 bytes de control + 100 Bytes de datos + 1 byte de id de CV = 103 bytes

Paquete Datagrama = 2 bytes de control + 100 bytes de datos + 8 bytes de direcciones = 110 bytes.

Entonces, cada paquete de CV ahorra 7 B respecto al datagrama al igual que antes.

Analicemos el costo adicional de circuito virtual en el establecimiento, confirmación y finalización del circuito.

El circuito virtual necesita:

* Establecimiento: un paquete con direcciones (110 B).
* Confirmación: un paquete con Id de CV (103 B).
* Finalización: otro paquete con Id de CV (103 B).

Establecimiento/confirmación/finalización CV = 110 bytes + 103 byte + 103 bytes. = 316 bytes

Por cada paquete de circuito virtual me ahorro 7 bytes de datagrama. Necesito averiguar cuantos paquetes necesito enviar para compensar los 316 bytes extras para el establecimiento del circuito.

Luego, 316 bytes / 7 bytes ≈ 45.14 -> redondeamos a 46

La mayor cantidad de datos que yo puedo transmitir en 45 paquetes es 4500 bytes. Luego, a partir de 4501 bytes de mensaje es conveniente que la red use circuito virtual.

* 1. ¿Y si se redujera a la mitad?

Analicemos el tamaño del paquete en este caso donde la carga útil es igual a 5 B:

Paquete de CV = 2 bytes de control + 5 Bytes de datos + 1 byte de id de CV = 8 bytes

Paquete Datagrama = 2 bytes de control + 5 bytes de datos + 8 bytes de direcciones = 15 bytes.

Entonces, cada paquete de CV ahorra 7 B respecto al datagrama al igual que antes.

Analicemos el costo adicional de circuito virtual en el establecimiento, confirmación y finalización del circuito.

El circuito virtual necesita:

* Establecimiento: un paquete con direcciones (15 B).
* Confirmación: un paquete con Id de CV (8 B).
* Finalización: otro paquete con Id de CV (8 B).

Establecimiento/confirmación/finalización CV = 15 bytes + 8 bytes + 8 bytes. = 31 bytes

Por cada paquete de circuito virtual me ahorro 7 bytes de datagrama. Necesito averiguar cuantos paquetes necesito enviar para compensar los 31 bytes extras para el establecimiento del circuito.

Luego, 31 bytes / 7 bytes ≈ 4.42 -> redondeamos a 5

La mayor cantidad de datos que yo puedo transmitir en 4 paquetes es 20 bytes. Luego, a partir de 21 bytes de mensaje es conveniente que la red use circuito virtual.

1. Se definen los siguientes parámetros para una red de conmutación: N = Número de saltos entre dos sistemas dados

L = Longitud del mensaje en bits

B = Velocidad de transmisión en bits por segundos de todos los canales P = Tamaño fijo del paquete en bits

H = Bits de redundancia o suplementarios (cabecera) por paquete S = Tiempo de establecimiento de la llamada en segundos

D = Retardo de propagación por saltos en segundos Dados los siguientes casos

* 1. N=4 L=3.200 B=9.600 P=1.024 H=16 S=0.2 D=0.001
  2. N=6 L=4.200 B=54.600 P=1.024 H=16 S=0.12 D=0.002
  3. N=3 L=6.200 B=18.600 P=1.024 H=16 S=0.1 D=0.001
     + Realice un gráfico donde se observen todos los tiempos intervinientes en la comunicación (transmisión, propagación y procesamiento)
     + Suponga que no se hace uso de confirmaciones.

Los gráficos son iguales a los del punto 2. Solo cambiar la cantidad de nodos intermedios.

* + - Calcule el tiempo total que toma la comunicación para Conmutación de Circuitos, Circuitos virtuales y Datagrama.

Para conmutación de circuitos:

Suponemos que tiempo de establecimiento = tiempo de desconexión

Tiempo total de la comunicación = (tiempo de establecimiento + tiempo de desconexión del circuito) + tiempo de propagación de datos + tiempo de transmisión de datos

Tiempo total = 2 \* S + N\*D + L/B = 2\*0.2 s + 4\*0.001s + 3200 b/9600 bps = 0.737s

Para datagrama:

Tiempo total = N°de saltos\*tiempo de propagación + N°nodos intermedios\*tiempo de transmisión + N°nodos intermedios\*tiempo de procesamiento + cantidad de paquetes\*tiempo de transmisión.

Supongo despreciable el tiempo de procesamiento:

Tiempo total = N\*D + (N-1)\*P/B + (entero)(L/(P-H))\*(P/B) = 4\*0.001 s + 3\*1024b/9600bps +(entero)(3200 b/(1024-16))\*(1024 b/9600 bps) = 0.751 s

Para circuito virtual:

Tiempo total = 3\*(N°de saltos\*tiempo de propagación + N°saltos\*tiempo de transmisión+ N°nodos intermedios\*tiempo de procesamiento) + N°de saltos\*tiempo de propagación + N°de nodos intermedios\*tiempo de transmisión + cantidad de paquetes\*tiempo de transmisión.

Supongo despreciable el tiempo de procesamiento:

Tiempo total = 3\*( N\*D + N\*P/B) + N\*D + (N-1)\*P/B + (entero)(L/(P-H))\*(P/B).

Tiempo total = 3\*(4\*0.001 s + 4\*1024 b/9600 bps) + 4\*0.001 s + 3\*(1024 b/9600 bps) + (entero)(3200 b/ (1024b-16b))\* (1024 b/9600 bps) = 2.043 s

b)

N=6 L=4.200 B=54.600 P=1.024 H=16 S=0.12 D=0.002

Para conmutación de circuitos:

Suponemos que tiempo de establecimiento = tiempo de desconexión

Tiempo total de la comunicación = (tiempo de establecimiento + tiempo de desconexión del circuito) + tiempo de propagación de datos + tiempo de transmisión de datos

Tiempo total = 2 \* S + N\*D + L/B = 2\*0.12 s + 6\*0.002s + 4200 b/54600 bps = 0.329s

Para datagrama:

Tiempo total = N°de saltos\*tiempo de propagación + N°nodos intermedios\*tiempo de transmisión + N°nodos intermedios\*tiempo de procesamiento + cantidad de paquetes\*tiempo de transmisión.

Supongo despreciable el tiempo de procesamiento:

Tiempo total = N\*D + (N-1)\*P/B + (entero)(L/(P-H))\*(P/B) = 6\*0.002 s + 5\*1024b/54600bps (entero)(4200 /(1024-16))\*(1024 b/54600 bps) = 0.199 s

Para circuito virtual:

Tiempo total = 3\*(N°de saltos\*tiempo de propagación + N°saltos\*tiempo de transmisión+ N°nodos intermedios\*tiempo de procesamiento) + N°de saltos\*tiempo de propagación + N°de nodos intermedios\*tiempo de transmisión + cantidad de paquetes\*tiempo de transmisión.

Supongo despreciable el tiempo de procesamiento:

Tiempo total = 3\*( N\*D + N\*P/B) + N\*D + (N-1)\*P/B + (entero)(L/(P-H))\*(P/B).

Tiempo total = 3\*(6\*0.002 s + 6\*1024b/54600bps) + 6\*0.002 s + 5\*(1024b/54600bps) + (entero)(4200 b/ (1024b-16b))\* (1024b/54600bps) = 0.573 s

c)

N=3 L=6.200 B=18.600 P=1.024 H=16 S=0.1 D=0.001

Para conmutación de circuitos:

Suponemos que tiempo de establecimiento = tiempo de desconexión

Tiempo total de la comunicación = (tiempo de establecimiento + tiempo de desconexión del circuito) + tiempo de propagación de datos + tiempo de transmisión de datos

Tiempo total = 2 \* S + N\*D + L/B = 2\*0.1 s + 3\*0.001s + 6200 b/18600 bps = 0.536 s

Para datagrama:

Tiempo total = N°de saltos\*tiempo de propagación + N°nodos intermedios\*tiempo de transmisión + N°nodos intermedios\*tiempo de procesamiento + cantidad de paquetes\*tiempo de transmisión.

Supongo despreciable el tiempo de procesamiento:

Tiempo total = N\*D + (N-1)\*P/B + (entero)(L/(P-H))\*(P/B) = 3\*0.001 s + 2\*1024b/18600bps + (entero)(6200 /(1024-16))\*(1024 b/18600 bps) = 0.498 s

Para circuito virtual:

Tiempo total = 3\*(N°de saltos\*tiempo de propagación + N°saltos\*tiempo de transmisión+ N°nodos intermedios\*tiempo de procesamiento) + N°de saltos\*tiempo de propagación + N°de nodos intermedios\*tiempo de transmisión + cantidad de paquetes\*tiempo de transmisión.

Supongo despreciable el tiempo de procesamiento:

Tiempo total = 3\*( N\*D + N\*P/B) + N\*D + (N-1)\*P/B + (entero)(L/(P-H))\*(P/B).

Tiempo total = 3\*(3\*0.001 s + 3\*1024b/18600bps) + 3\*0.001 s + 2\*(1024b/18600bps) + (entero)(6200 b/ (1024b-16b))\* (1024b/18600bps) = 0.930 s

1. Otro esquema de encaminamiento adaptable es el conocido como aprendizaje hacia atrás. Todo paquete encaminado a través de la red contiene no sólo las direcciones destino y origen, sino también un contador de saltos que se incrementa con cada salto. Cada nodo construye una tabla de encaminamiento que especifica el nodo siguiente y el número de saltos para cada destino. ¿Cómo se usa la información contenida en el paquete para construir la tabla? ¿Cuáles son las ventajas y desventajas de esta técnica?

En **aprendizaje hacia atrás (backward learning)** cada nodo aprovecha la información de los paquetes que pasan por él:

1. **El paquete trae origen, destino y contador de saltos.**
2. Cuando un nodo recibe un paquete, **lee la dirección de origen y el contador de saltos**.
3. El nodo **actualiza su tabla de encaminamiento** para el **origen** (no para el destino) indicando:
   * **Próximo salto:** el nodo desde el cual acaba de recibir el paquete.
   * **Distancia:** el valor actual del contador de saltos.
4. De esta forma, el nodo aprende “hacia atrás”: sabe que para llegar al origen conviene reenviar por el enlace del que llegó ese paquete.
5. Con el paso del tiempo, a medida que llegan más paquetes desde distintos orígenes, cada nodo va **construyendo o corrigiendo** su tabla de encaminamiento.

Ventajas

* Simplicidad: no requiere un protocolo de enrutamiento complejo ni intercambio explícito de tablas. Los nodos aprenden pasivamente de los paquetes que circulan.
* Autoadaptación: si cambia la topología (caída de un enlace, nodo nuevo, etc.), las rutas se actualizan automáticamente cuando vuelven a circular paquetes.
* Escalabilidad básica: no necesita sincronización global ni cálculos costosos como Dijkstra en cada nodo.
* Menor sobrecarga de control: la red no necesita inundaciones de información de estado de enlace ni de vectores de distancia.

## Desventajas

* **Dependencia del tráfico:** un nodo solo aprende rutas hacia destinos desde los que recibe tráfico. Si nadie manda paquetes desde cierto origen, no se crea la entrada.
* **Convergencia lenta:** pueden pasar muchos paquetes antes de que la tabla esté completa o actualizada.
* **Rutas subóptimas:** al basarse solo en información observada, puede mantener caminos más largos o no descubrir el más corto si no circula tráfico que lo revele.
* **Inestabilidad:** los cambios en topología o en patrones de tráfico pueden producir oscilaciones o inconsistencias temporales.
* **No apto para redes grandes:** en redes extensas, la dependencia del azar del tráfico y la lentitud de convergencia lo hacen poco eficiente frente a algoritmos clásicos (link-state, distance-vector).