

Universidad de Los Andes  
Facultad de Ingeniería  
Escuela de Sistemas

# Redes de Computadoras Introducción

Mérida - Venezuela  
Prof. Gilberto Díaz

# **Introducción a las Redes de Datos**

## **Compartiendo Recursos**

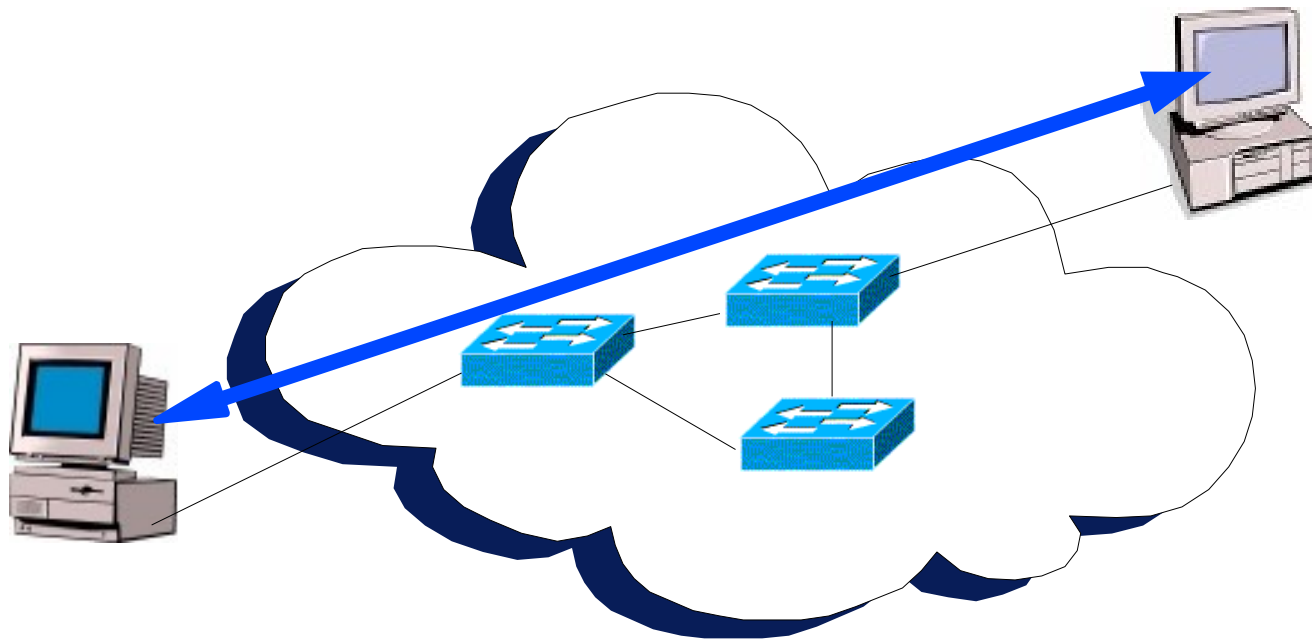
En la clase anterior vimos ciertas características de las redes conmutadas que las hacen más apropiadas para construir redes de computadores.

En el resto del curso nos enfocaremos a este tipo de redes.

# Introducción a las Redes de Datos

## Compartiendo Recursos

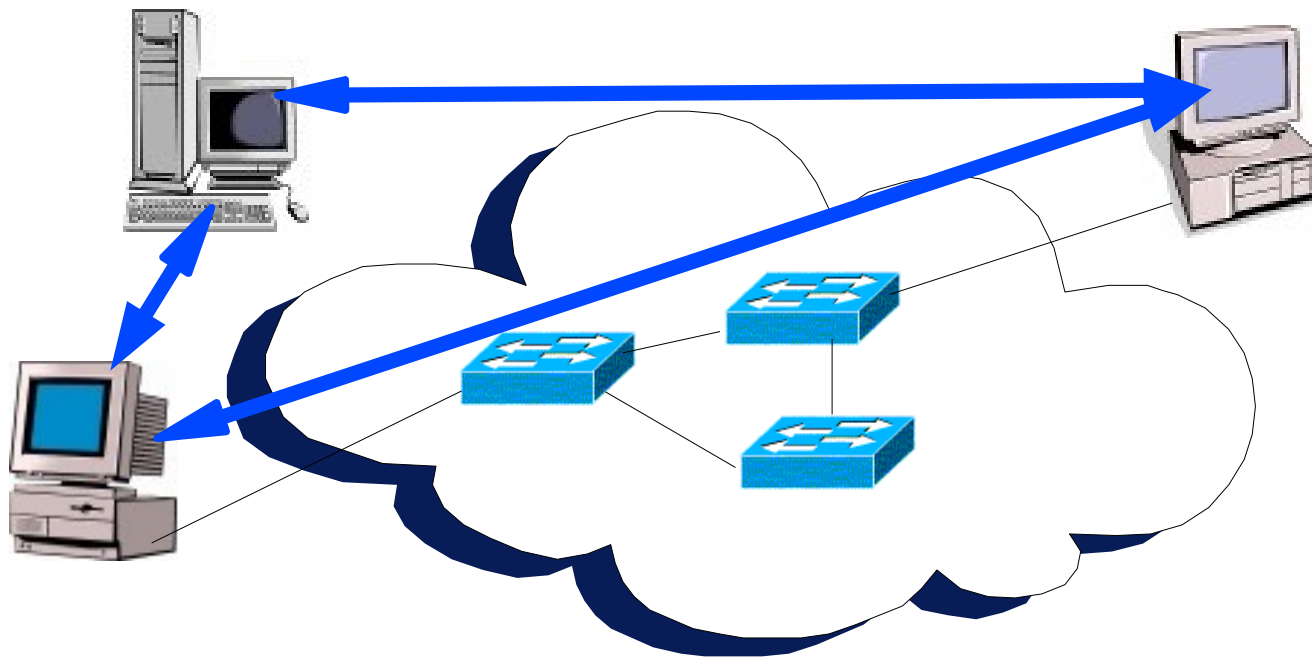
Hemos visto, de forma general, como en una red un par de hosts puede comunicarse



# Introducción a las Redes de Datos

## Compartiendo Recursos

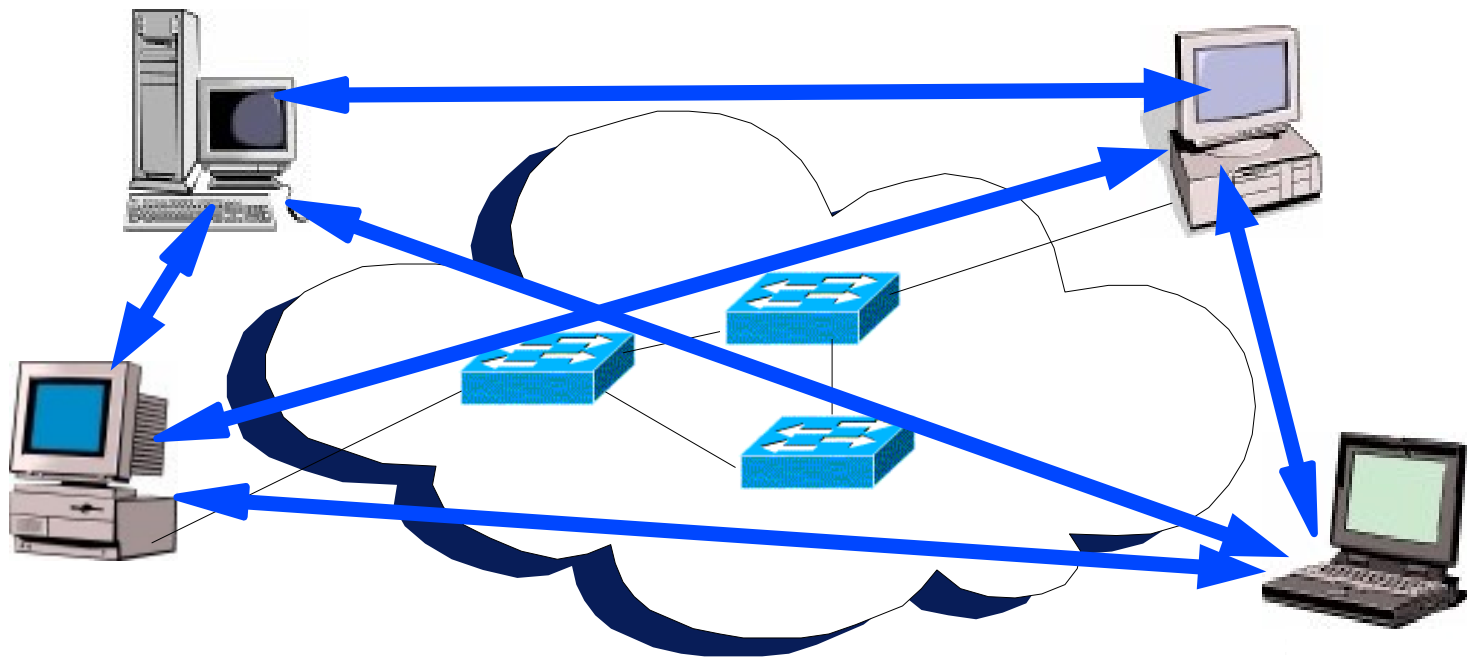
Pero nosotros queremos hacer más que eso.  
Queremos que que todos los pares de nodos se  
puedan comunicar



# Introducción a las Redes de Datos

## Compartiendo Recursos

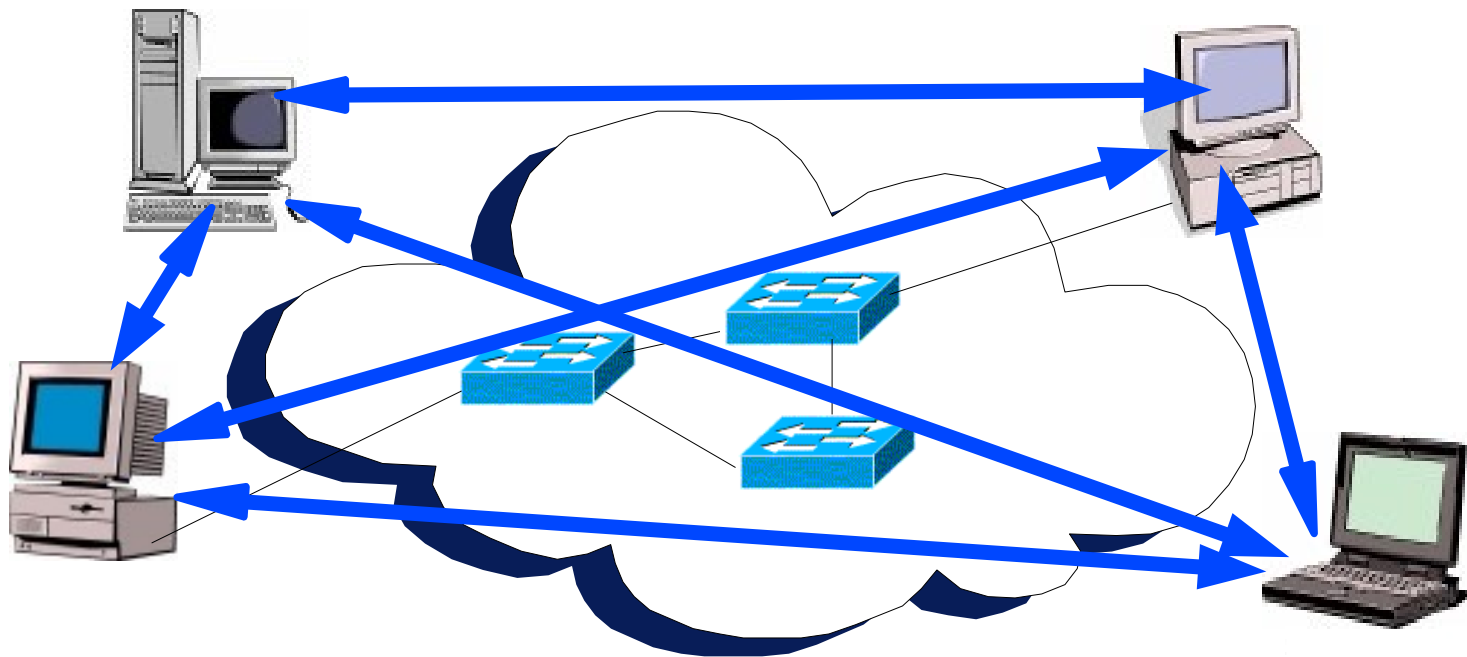
Pero nosotros queremos hacer más que eso. Queremos que que todos los pares de nodos se puedan comunicar



# Introducción a las Redes de Datos

## Compartiendo Recursos

Para entender como todos los nodos pueden compartir una red necesitamos introducir el concepto de **multiplexación**



# Introducción a las Redes de Datos

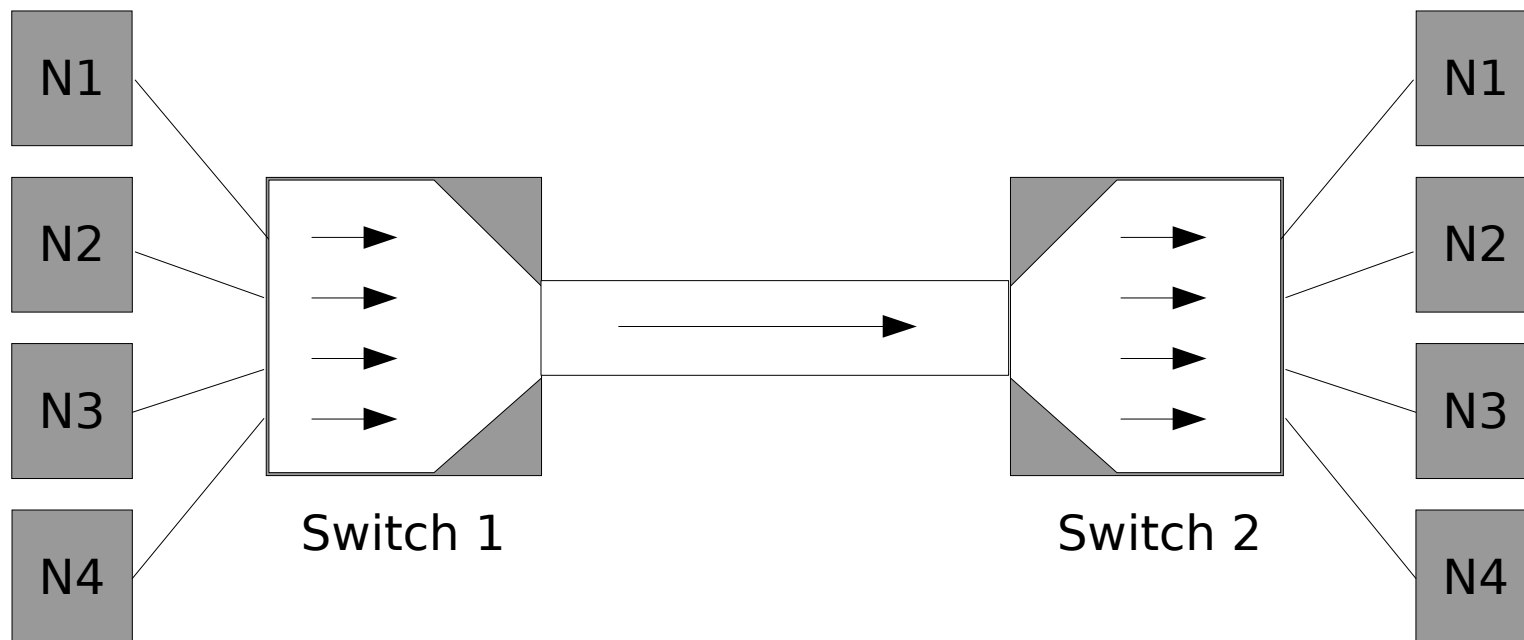
## Compartiendo Recursos

**Multiplexación:** es la combinación de dos o más canales de información en un sólo medio de transmisión usando un dispositivo llamado multiplexor. El proceso inverso se conoce como demultiplexación.

# Introducción a las Redes de Datos

## Compartiendo Recursos

### Multiplexación

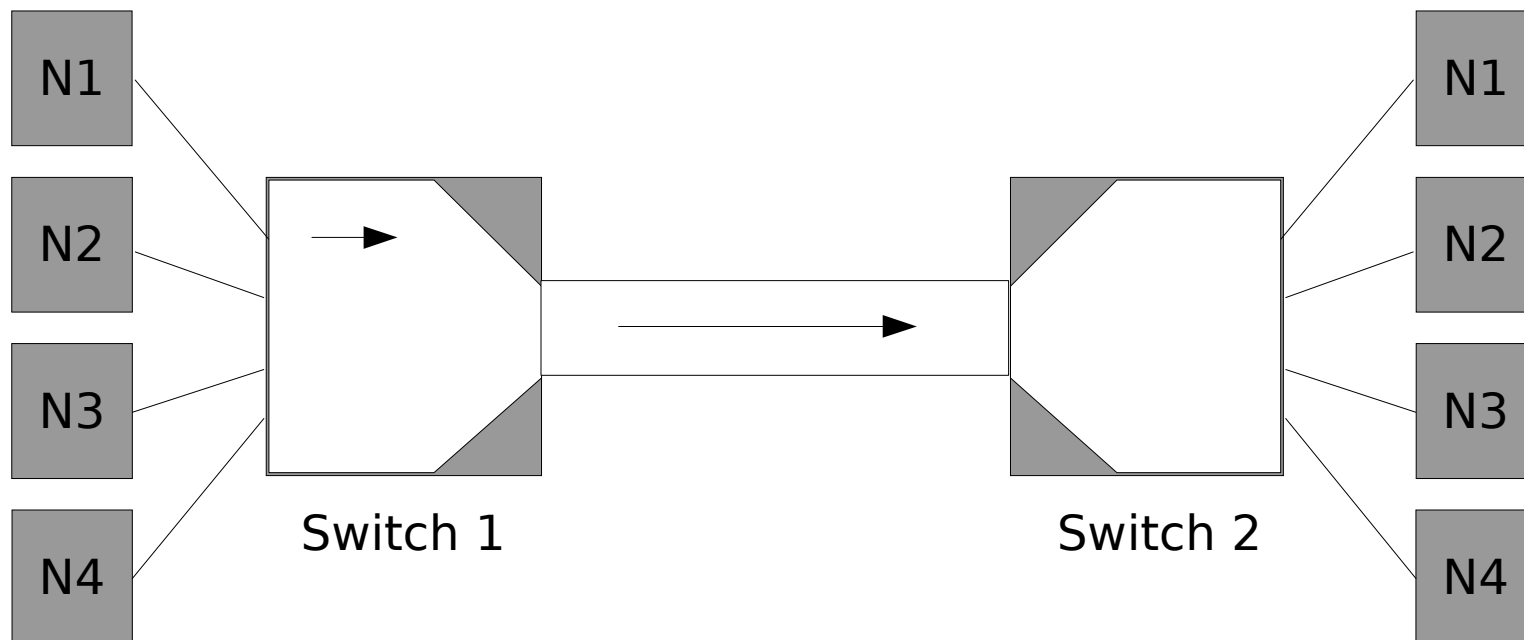




# Introducción a las Redes de Datos

## Compartiendo Recursos

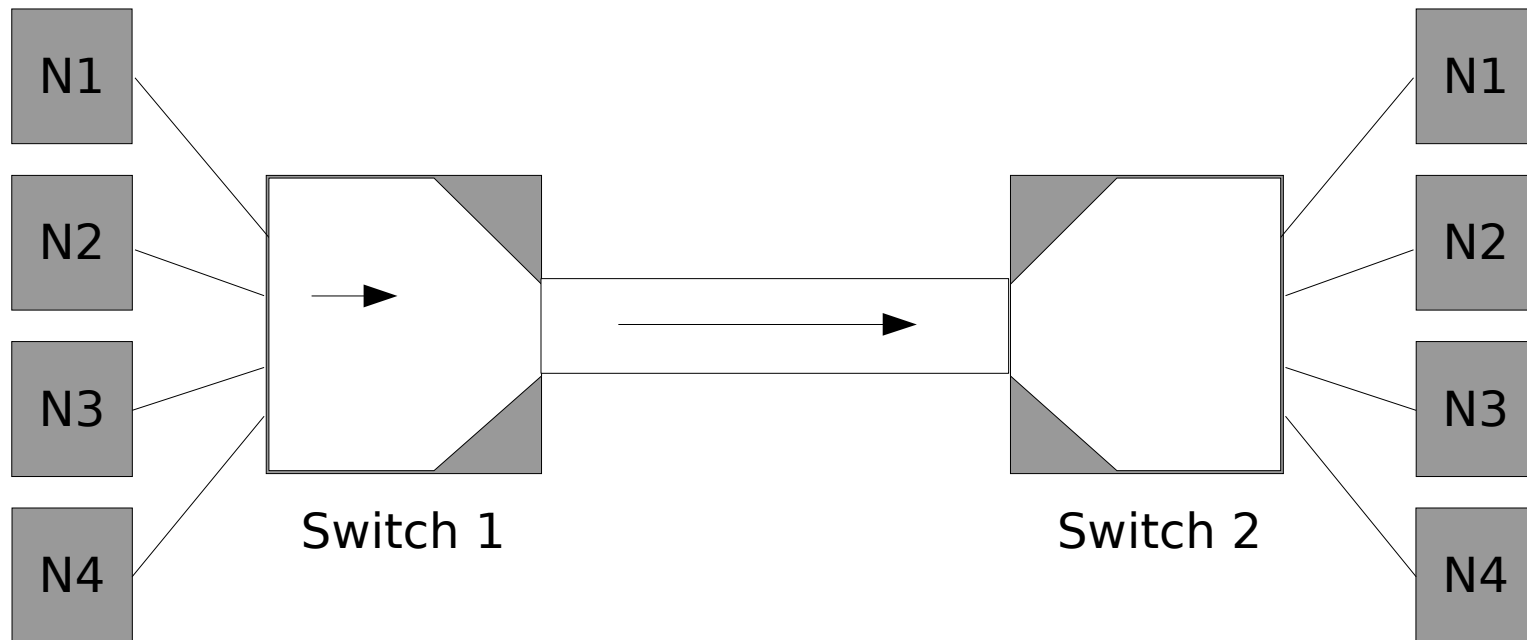
**Synchronous time-division (STDM)** Una de las estrategias utilizadas es asignar un tiempo de transmisión a cada nodo



# Introducción a las Redes de Datos

## Compartiendo Recursos

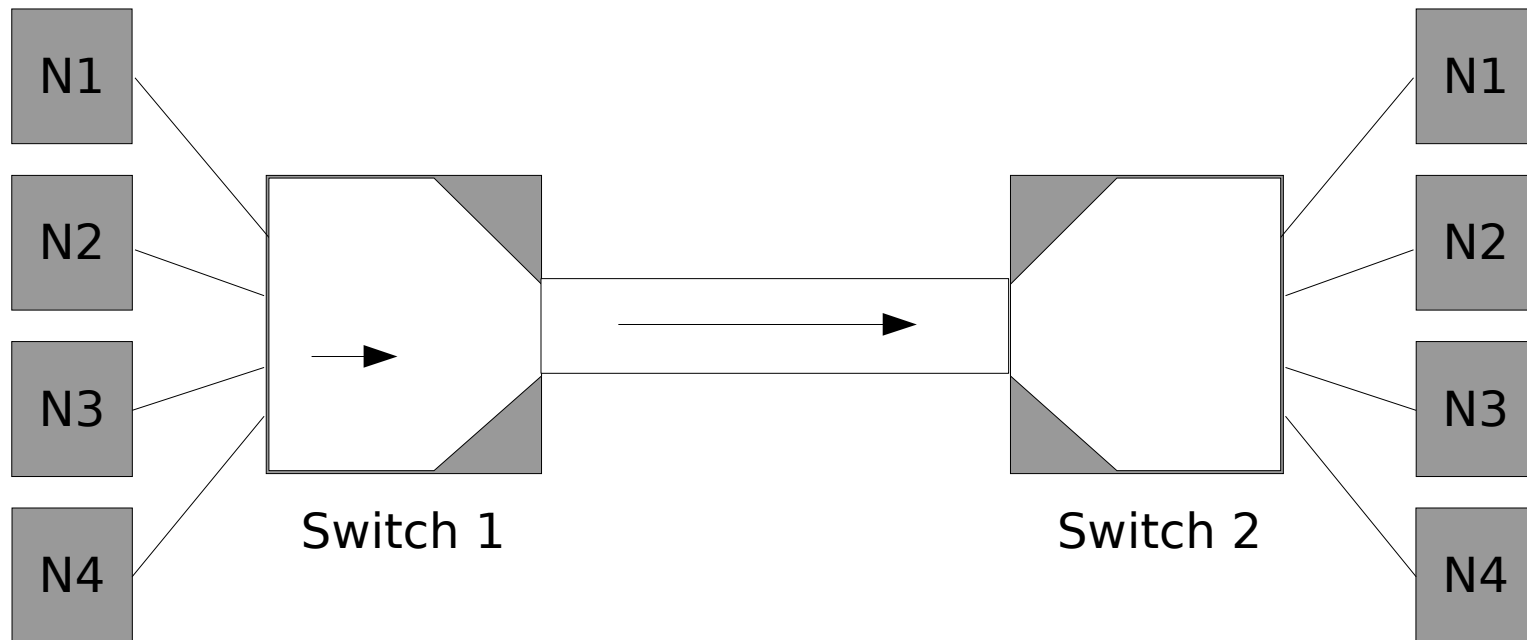
**Synchronous time-division (STDM)** Cada nodo tiene un lapso de tiempo para transmitir



# Introducción a las Redes de Datos

## Compartiendo Recursos

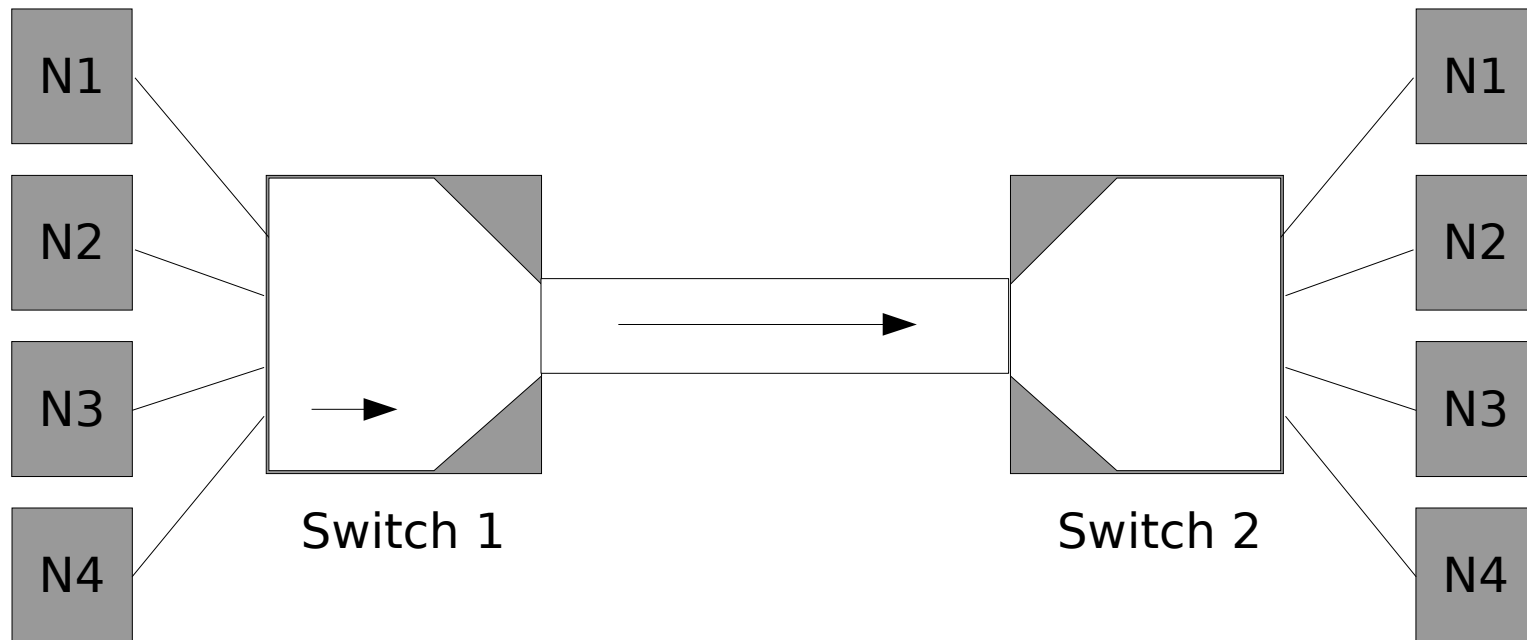
**Synchronous time-division (STDM)** Y luego pasa al siguiente nodo



# Introducción a las Redes de Datos

## Compartiendo Recursos

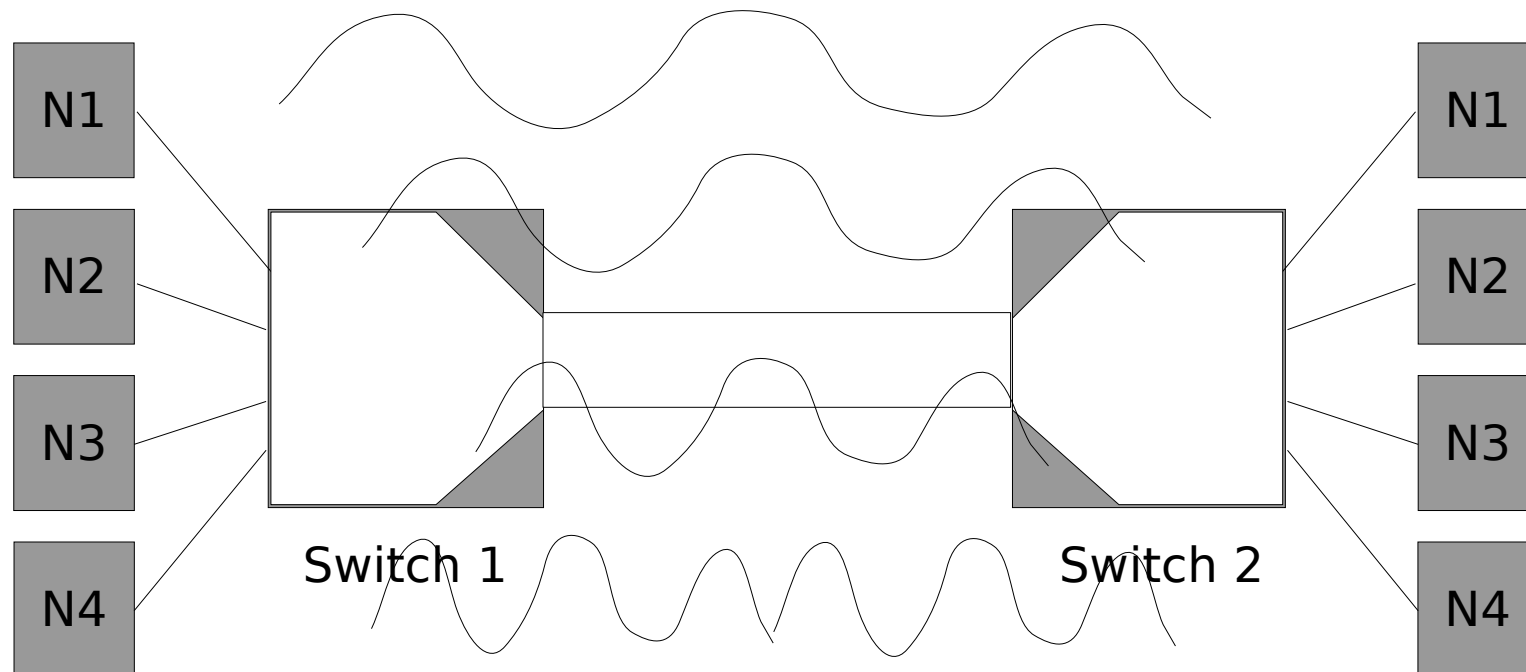
**Synchronous time-division (STDM)** El ciclo se repite siguiendo una política *round robin*



# Introducción a las Redes de Datos

## Compartiendo Recursos

**Frecuency-division multiplexing (FDM)** Cada flujo de datos es transmitido a una frecuencia diferente



# Introducción a las Redes de Datos

## Compartiendo Recursos

**Multiplexación** Sin embargo estos mecanismos tienen desventajas

Si uno de los flujos (un par de hosts) no tiene nada que transmitir, el tiempo que le es asignado se pierde pues el canal permanece desocupado.

El resto de los pares de nodos que tienen que transmitir deben esperar su turno.

# **Introducción a las Redes de Datos**

## **Compartiendo Recursos**

### **Multiplexación**

Las dos estrategias tienen limitaciones en situaciones donde el número máximo de flujos es fijo y el tiempo máximo es conocido.

No tiene sentido práctico variar el quantum (STDM) o agregar nuevas frecuencias (FDM)

# **Introducción a las Redes de Datos**

## **Compartiendo Recursos**

### **Multiplexación estática**

El tipo de multiplexación que se utiliza en la mayoría de los casos es similar a STDM en el hecho de compartir el canal utilizando el tiempo como criterio de compartimiento.



# **Introducción a las Redes de Datos**

## **Compartiendo Recursos**

### **Multiplexación estática**

La diferencia es que los flujos son transmitidos por demanda. Cada par de hosts que necesiten transmitir lo hacen en lugar de esperar un tiempo determinado.

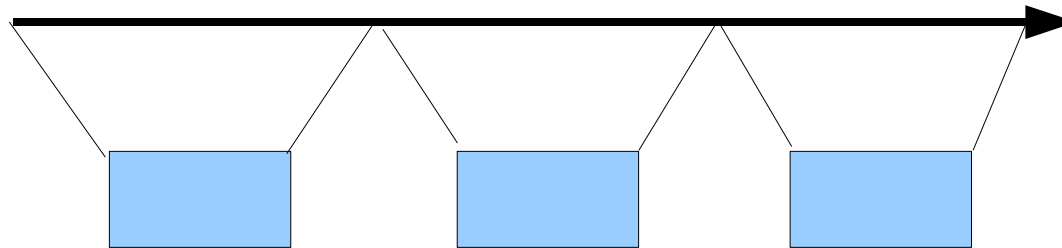
Esto evita que el uso del canal se desperdicie con tiempo ocioso.

# Introducción a las Redes de Datos

## Compartiendo Recursos

### Multiplexación estática

Para evitar que un par de hosts monopolicen el enlace se divide cada flujo en **paquetes**



Cada paquete tiene un tamaño máximo

Esto puede hacer que cada host utilice más de un paquete para enviar un mensaje grande

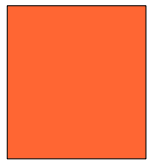
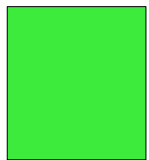
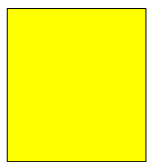
# Introducción a las Redes de Datos

## Compartiendo Recursos

### Multiplexación estática

Si más de un par de nodos tiene paquetes para enviar

nodos

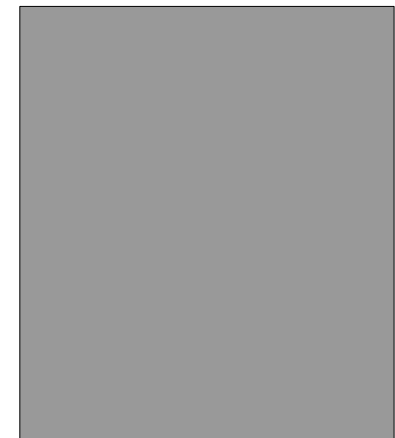


Switch A



Enlace

Switch B

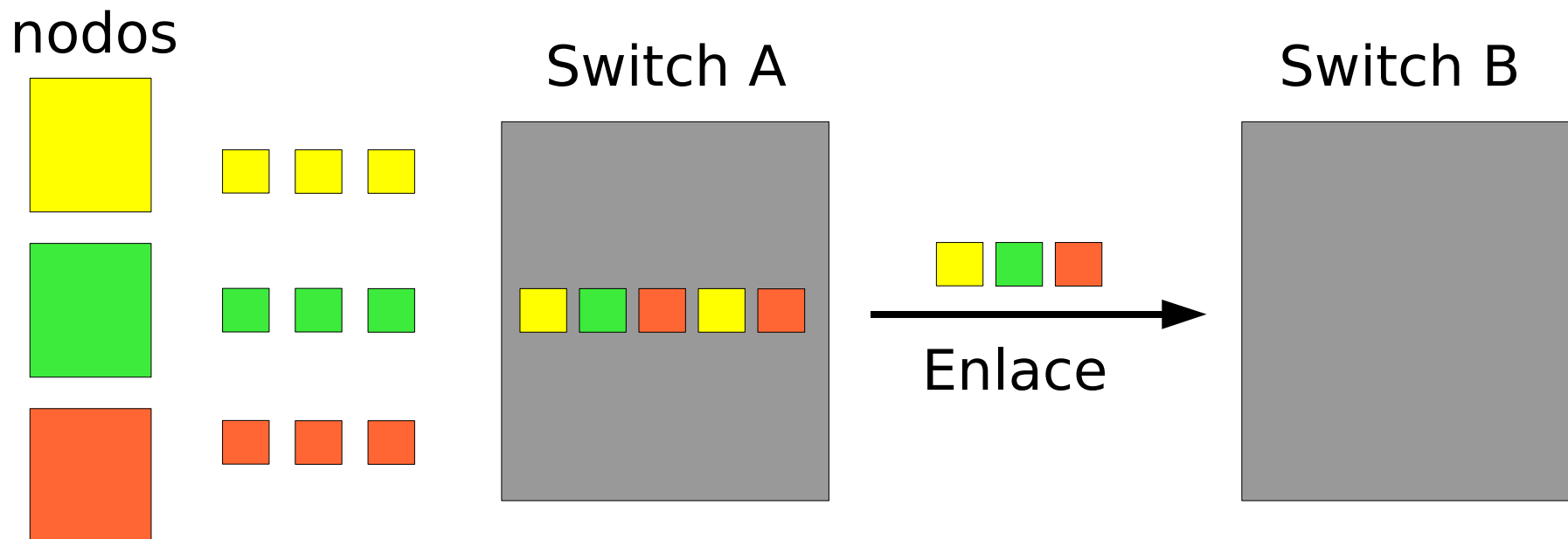


# Introducción a las Redes de Datos

## Compartiendo Recursos

### Multiplexación estática

Entonces el switch envía los paquetes de todos de forma intercalada.



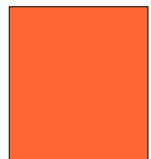
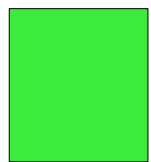
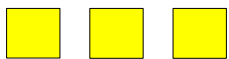
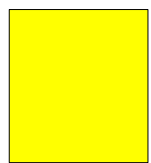
# Introducción a las Redes de Datos

## Compartiendo Recursos

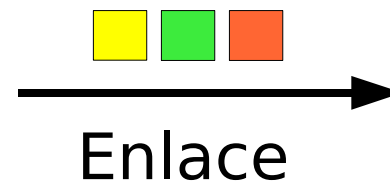
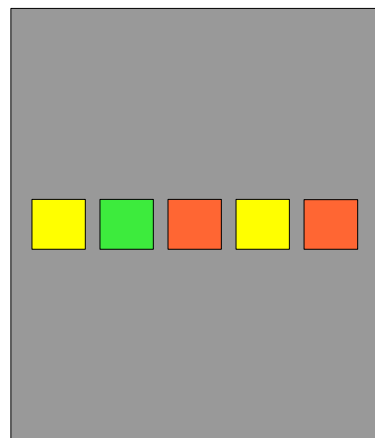
### Multiplexación estática

Existen varias estrategias para decidir cual de los paquetes enviar primero.

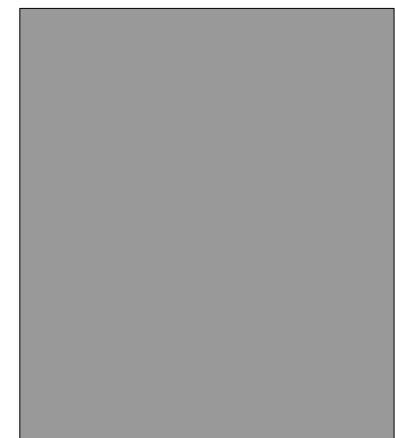
nodos



Switch A



Switch B

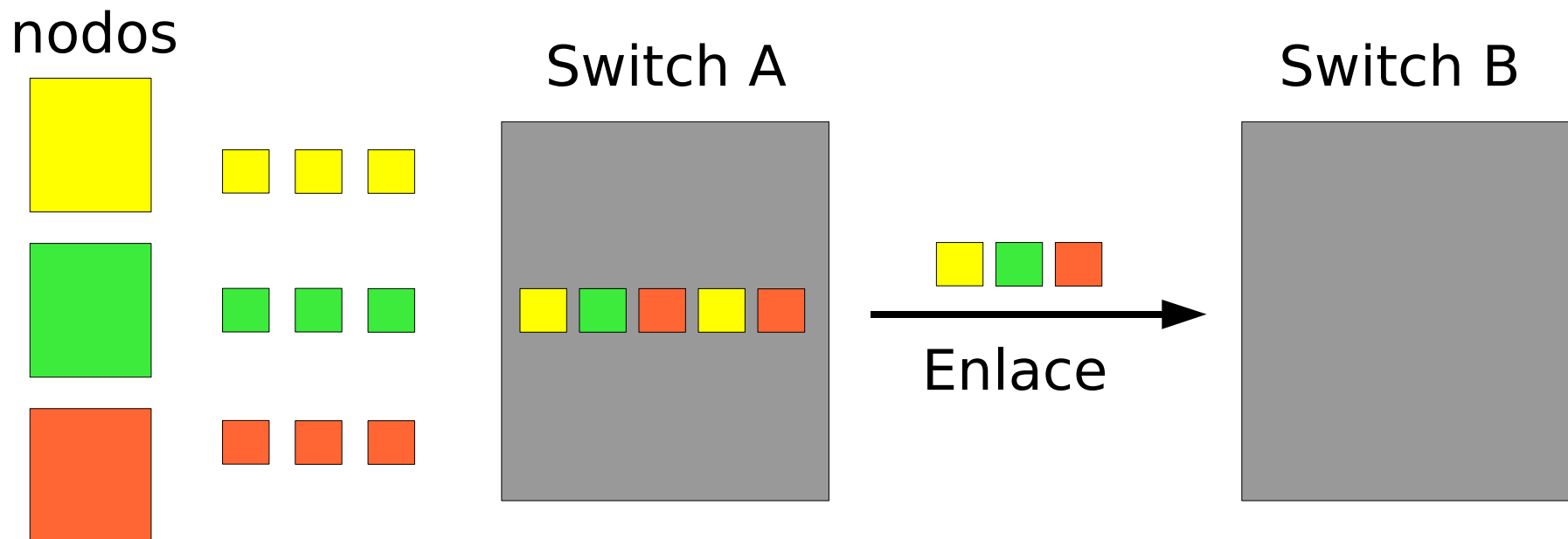


# Introducción a las Redes de Datos

## Compartiendo Recursos

### Multiplexación estática

FIFO: el primero que entra el primero que sale

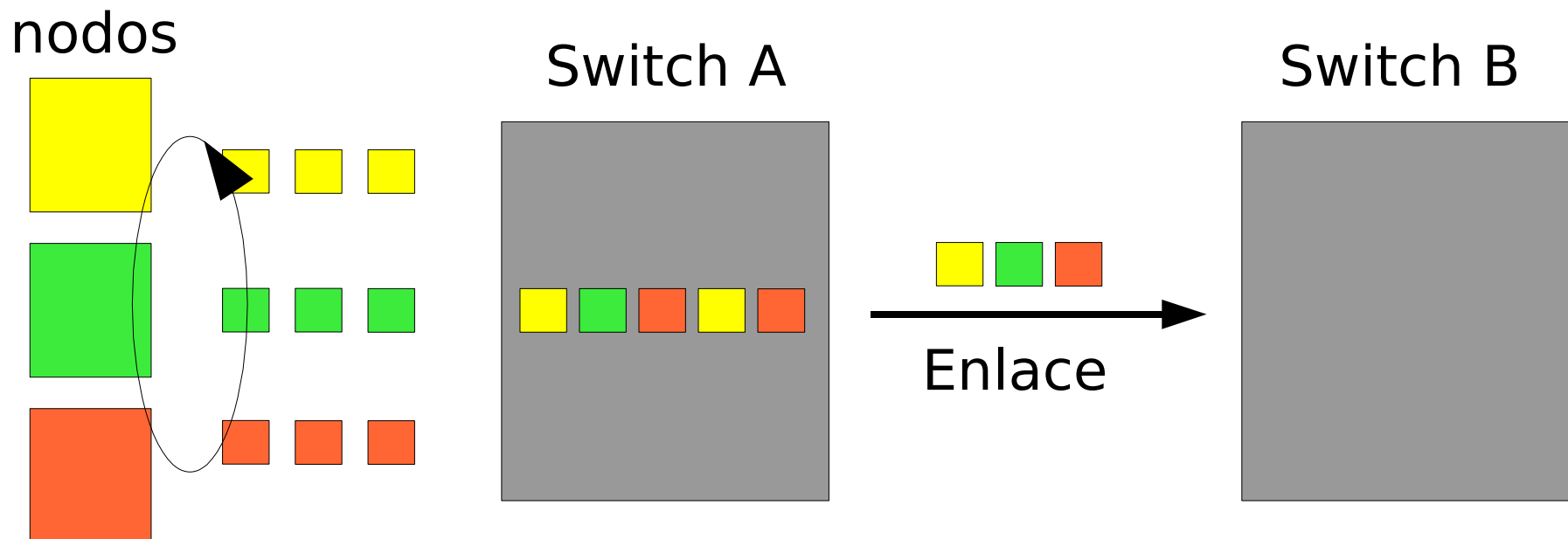


# Introducción a las Redes de Datos

## Compartiendo Recursos

### Multiplexación estática

Round-robin: enviar uno de cada uno y luego repetir el ciclo



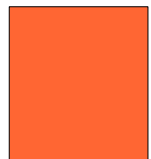
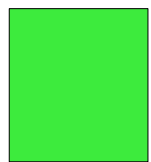
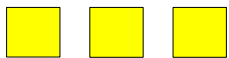
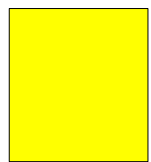
# Introducción a las Redes de Datos

## Compartiendo Recursos

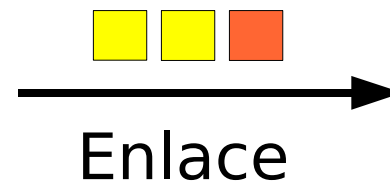
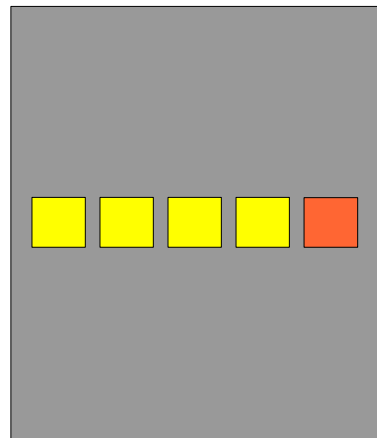
### Multiplexación estática

QoS: Podemos darle prioridad a cierto tipo de paquetes (Calidad de Servicio)

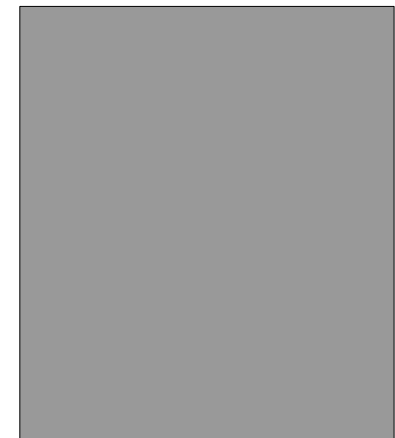
nodos



Switch A



Switch B



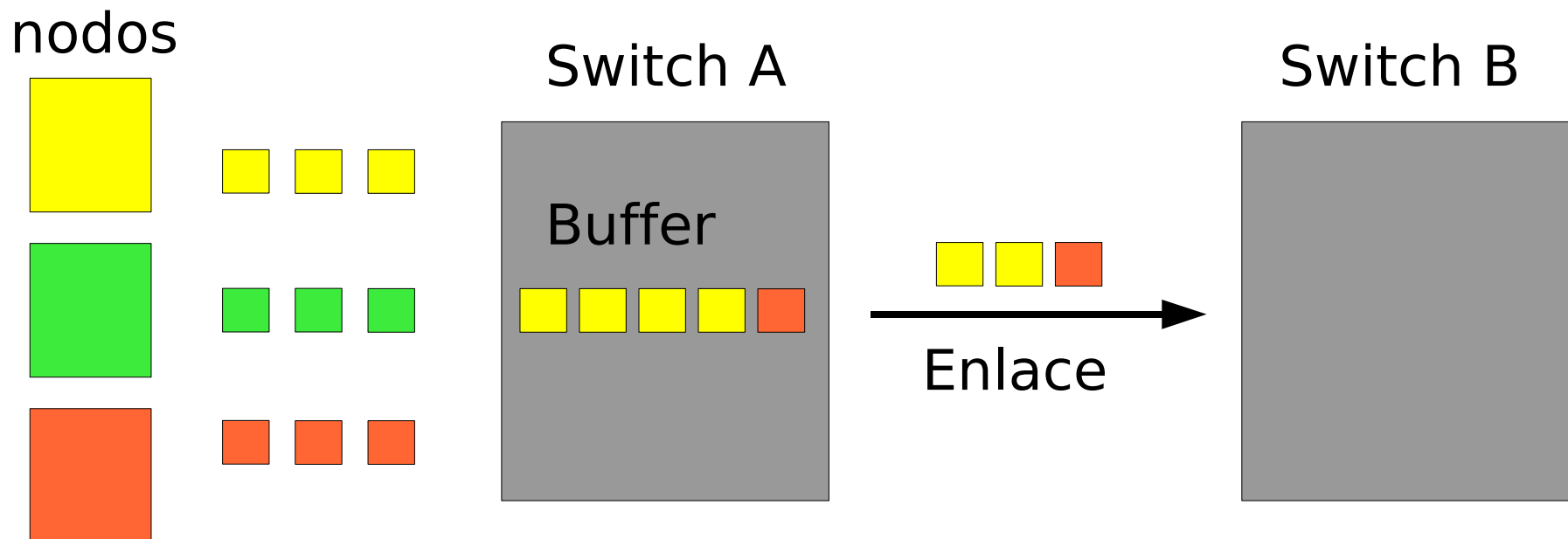


# Introducción a las Redes de Datos

## Compartiendo Recursos

### Multiplexación estática

Cuando los paquetes llegan muy rápido el switch debe comenzar a guardarlos

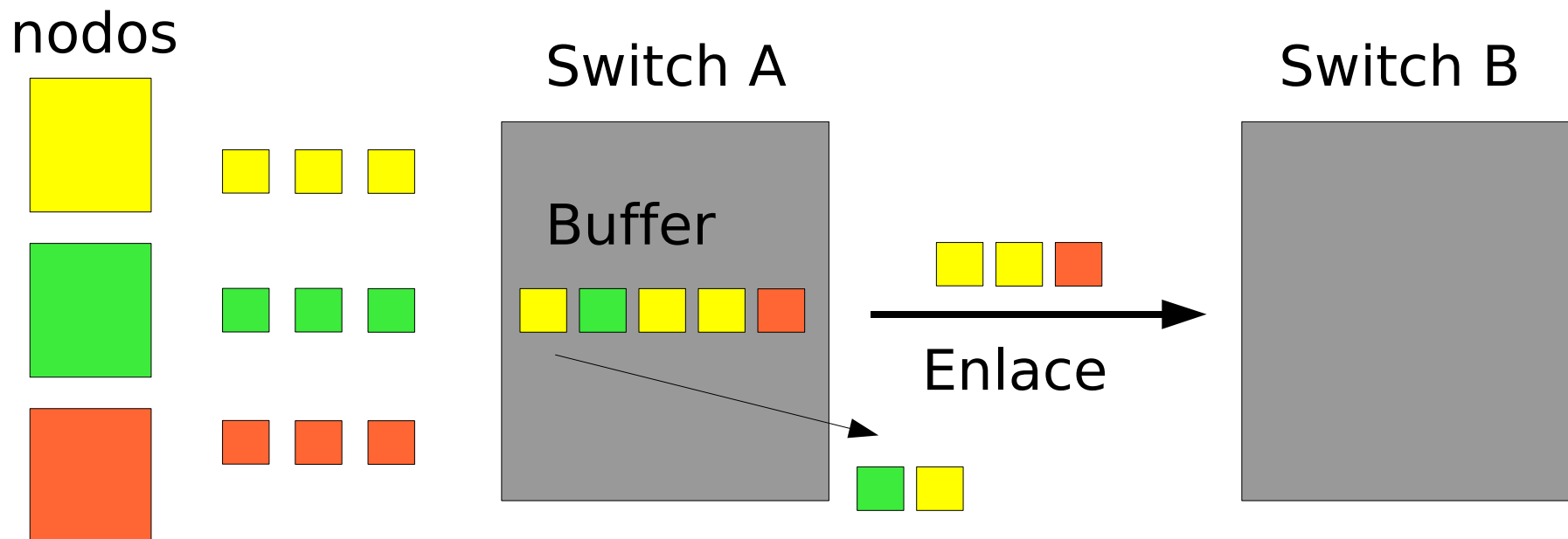


# Introducción a las Redes de Datos

## Compartiendo Recursos

### Multiplexación estática

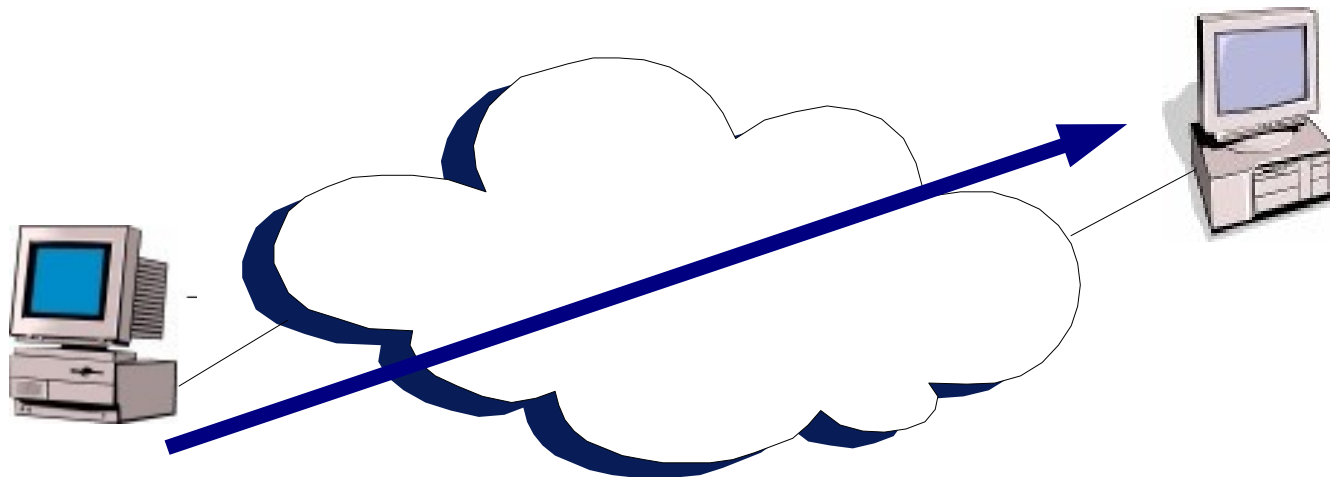
Cuando se llena el buffer el switch debe descartar paquetes



# Introducción a las Redes de Datos

## Funcionalidad

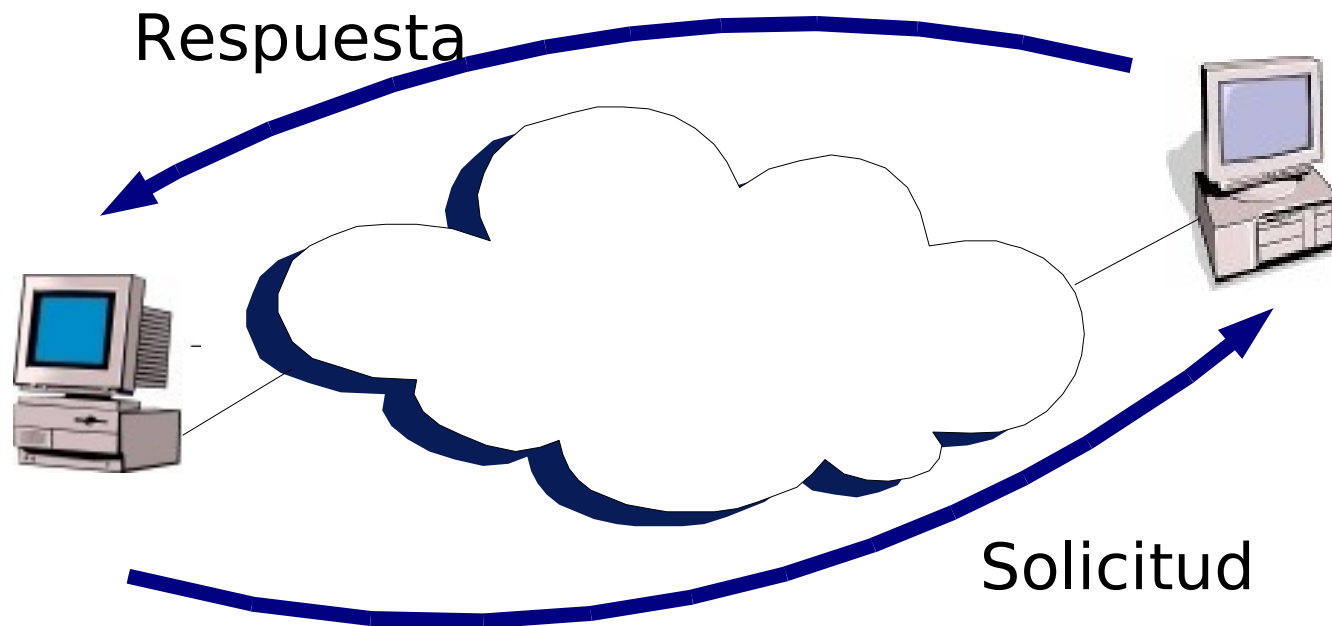
Desde el punto de vista del usuario, una red debe proporcionar servicios más que transmitir paquetes de forma eficiente



# Introducción a las Redes de Datos

## Funcionalidad

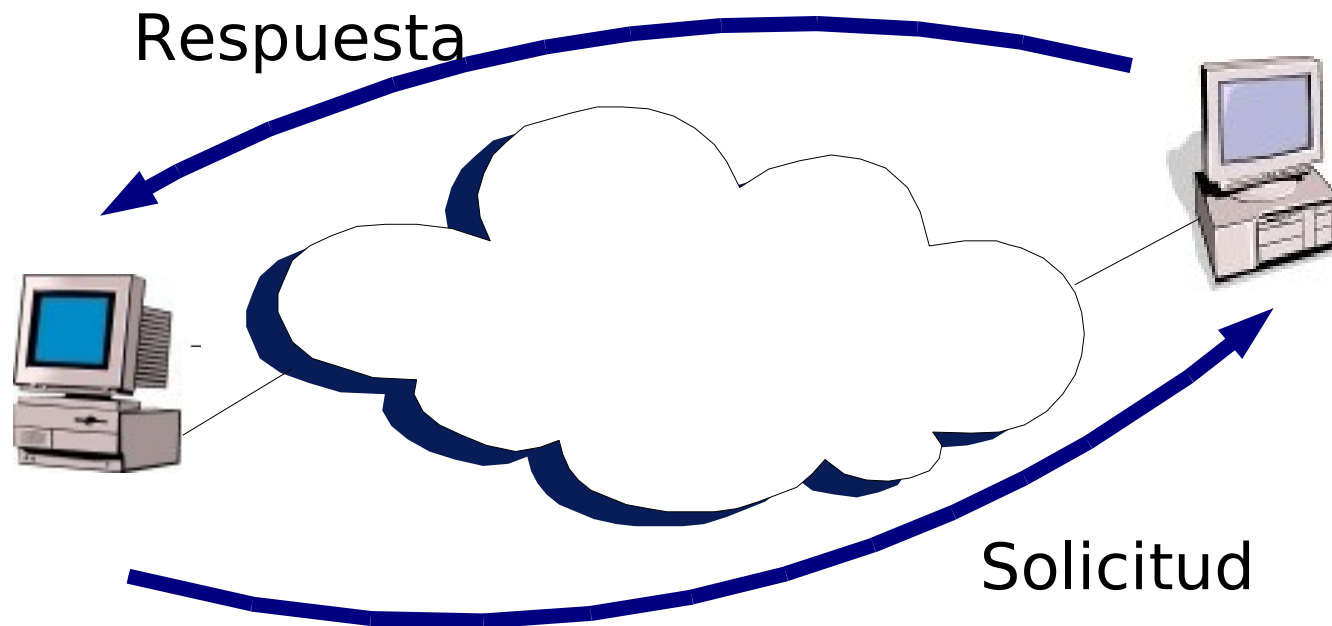
Otro requisito importante que se debe considerar es que cada aplicación que corre en los hosts debe poder comunicarse de una manera apropiada



# Introducción a las Redes de Datos

## Funcionalidad

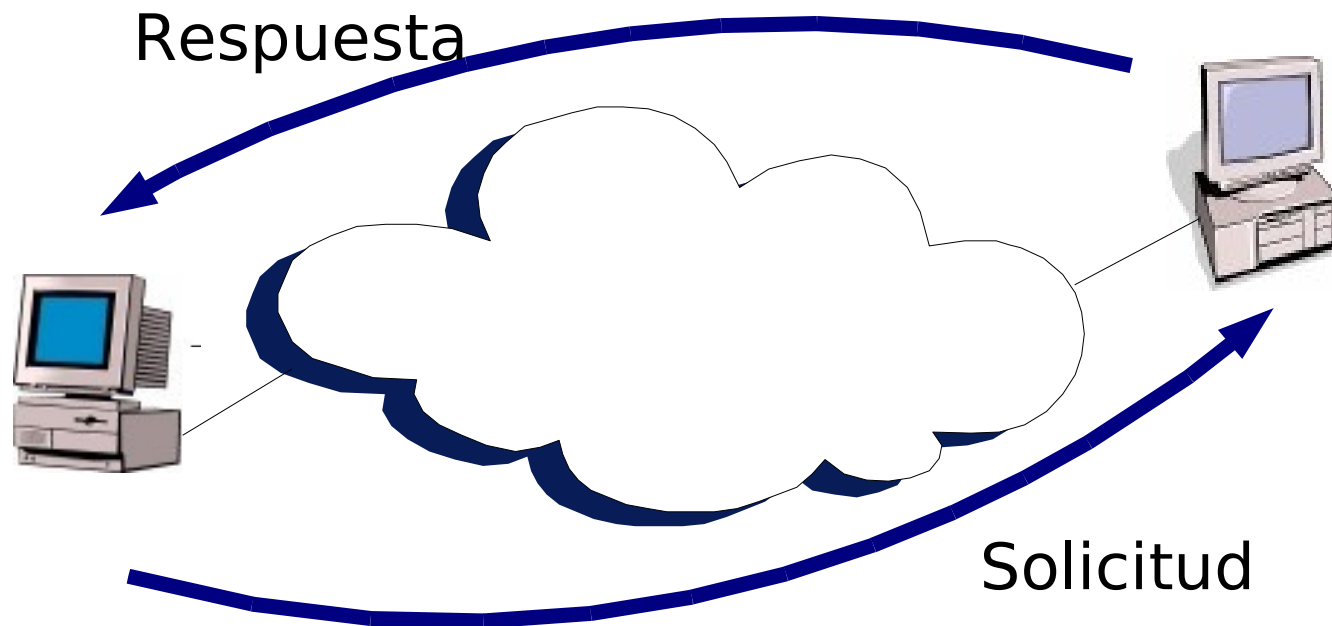
Cuando dos aplicaciones se comunican se deben realizar muchos pasos complicados



# Introducción a las Redes de Datos

## Funcionalidad

En lugar de que cada aplicación realice todos los pasos, es más eficiente implementar los servicios comunes una sólo vez y que luego cada diseñador los utilice



# Introducción a las Redes de Datos

## Prestaciones

Una red debe tener un alto rendimiento a la hora de prestar servicios y más importante aun, una buena relación costo – rendimiento

El rendimiento se mide generalmente con dos criterios

- **Ancho de Banda (Band width)**
- **Latencia o demora (Latency or delay)**

# Introducción a las Redes de Datos

## Prestaciones

### Ancho de Banda

Es la medida de cuanta información puede fluir desde un lugar a otro en una cantidad de tiempo definido.

Generalmente se mide en bits por segundos

- Kbps: ADSL, modems, cable modems, enlaces seriales, etc
- Mbps: Redes ethernet 10, 100, Token Ring, redes inalámbricas 802.11abg
- Gbps: Redes ethernet 1, 10

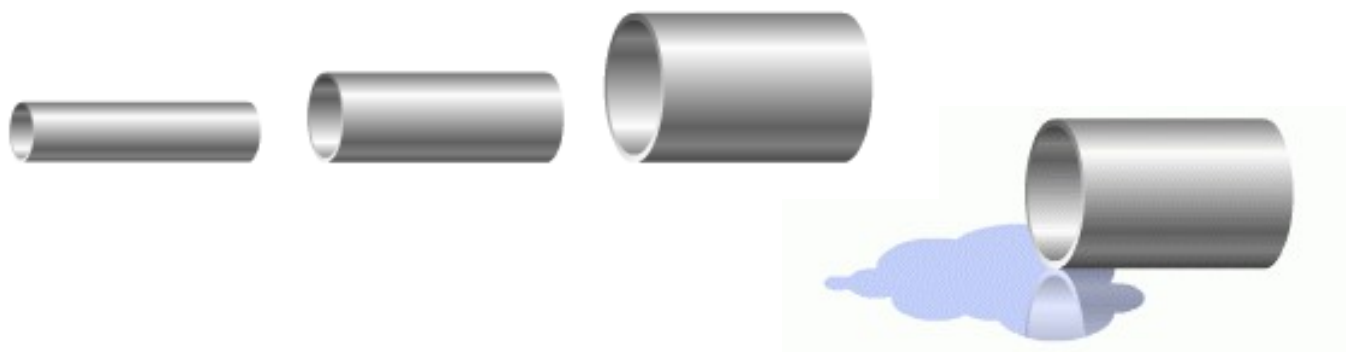


# Introducción a las Redes de Datos

## Prestaciones

### Ancho de Banda

Es comparable al ancho de una tubería  
¿Cuánto líquido puede pasar por la tubería?

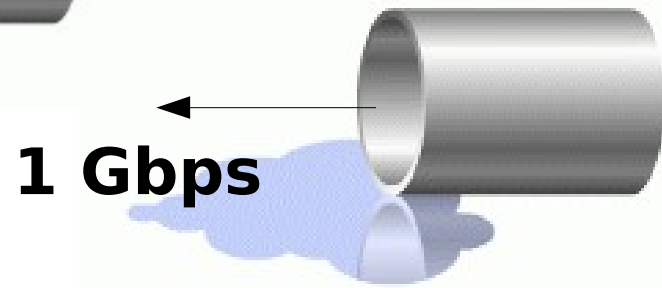
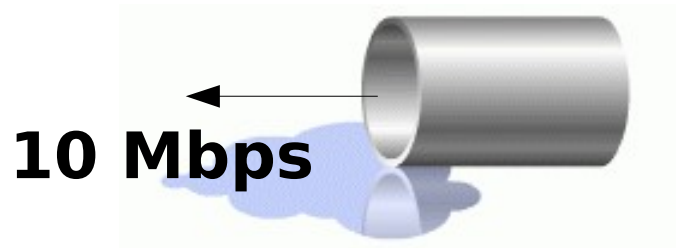
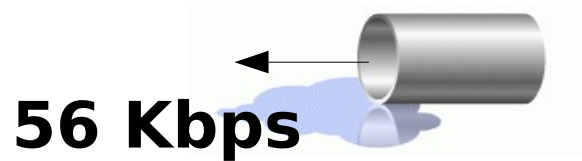


# Introducción a las Redes de Datos

## Prestaciones

### Ancho de Banda

Por ejemplo 10 Mbps significa que cada segundo se envía 10 millones de bits.

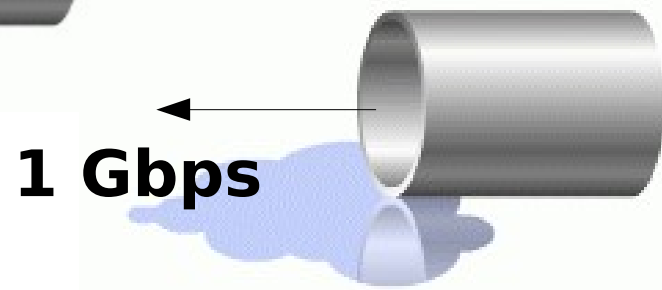
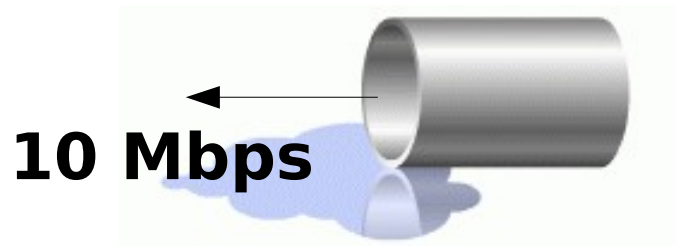
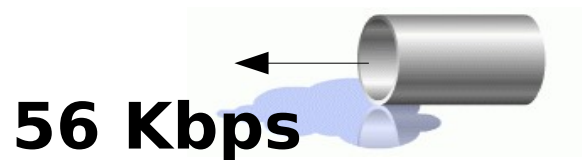


# Introducción a las Redes de Datos

## Prestaciones

### Ancho de Banda

A veces es útil pensar en el ancho de banda en términos de cuanto tiempo dura en transmitirse un bit de datos



# **Introducción a las Redes de Datos**

## **Prestaciones**

### **Ancho de Banda y Throughput**

Estos dos términos son utilizados de forma indistinta en ciertas ocasiones.

Podemos ver el primero como el valor nominal especificado de los equipos y medios de transmisión

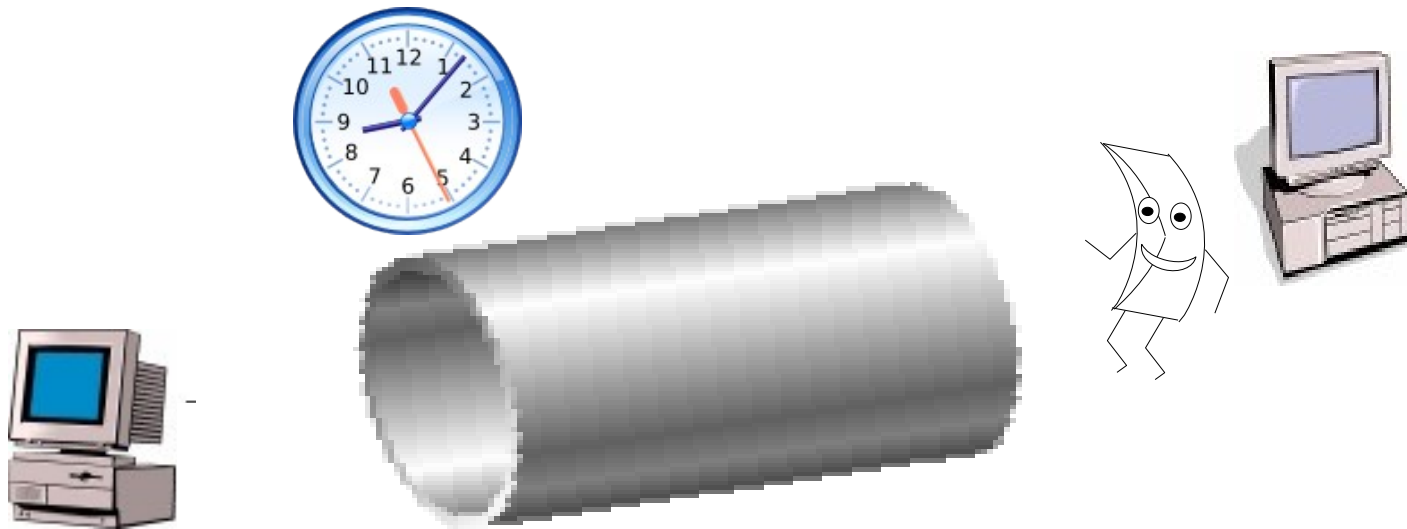
El segundo sería el valor real obtenido en la transmisión de datos por una aplicación determinada

# Introducción a las Redes de Datos

## Prestaciones

### Latencia

Es el tiempo que le toma a un sólo paquete en viajar desde el nodo fuente hasta el nodo destino

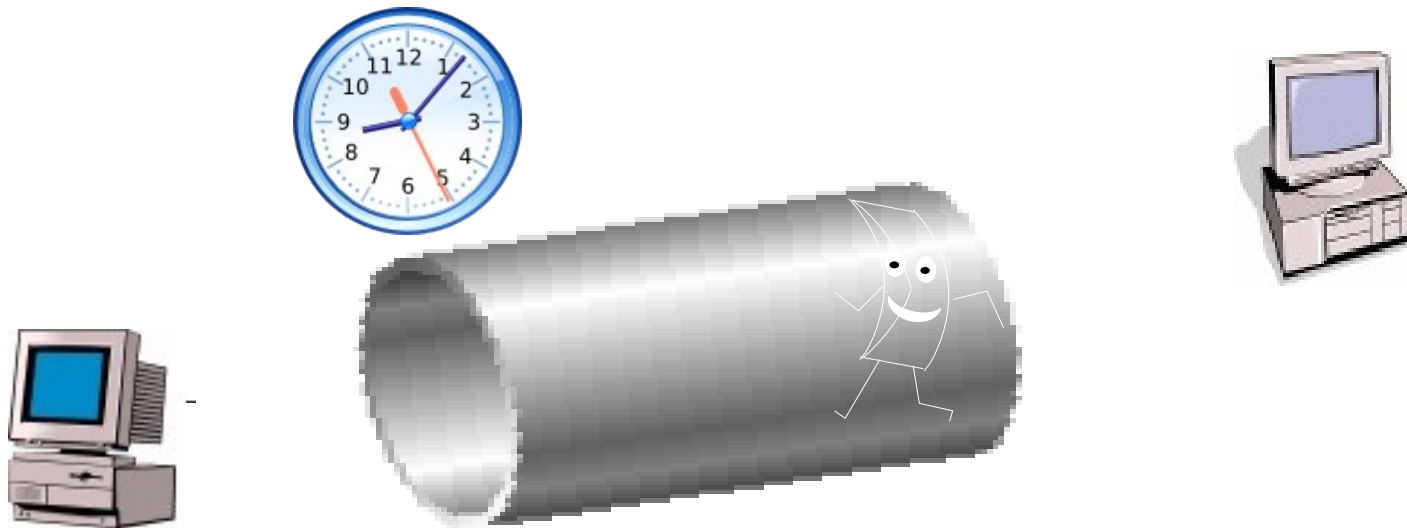


# Introducción a las Redes de Datos

## Prestaciones

### Latencia

Es el tiempo que le toma a un sólo paquete en viajar desde el nodo fuente hasta el nodo destino

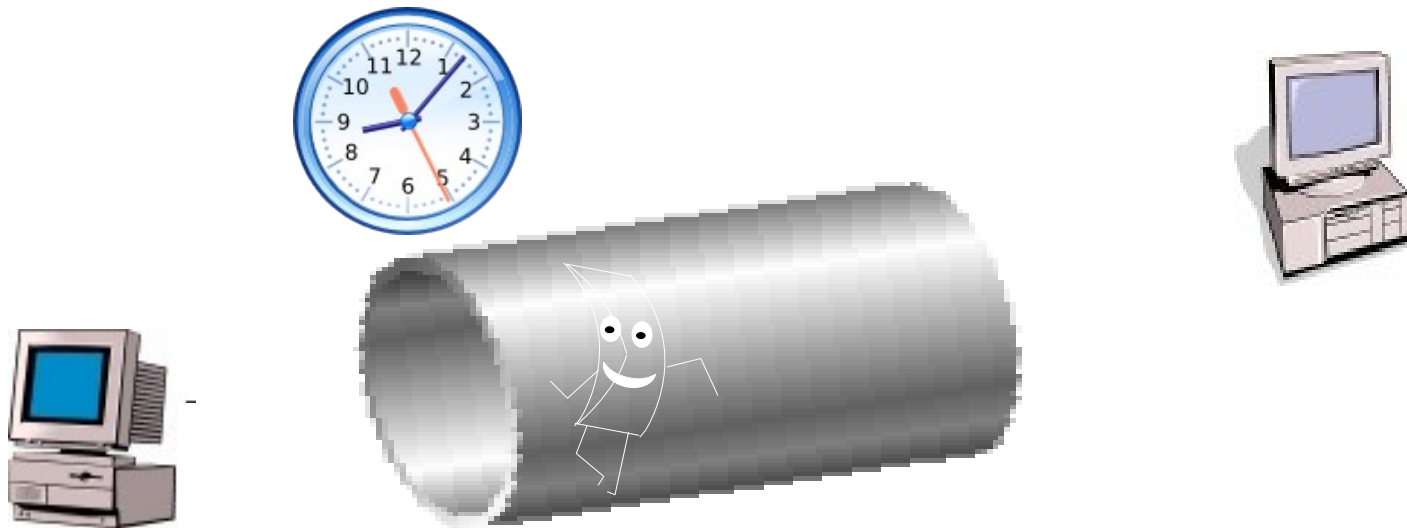


# Introducción a las Redes de Datos

## Prestaciones

### Latencia

Es el tiempo que le toma a un sólo paquete en viajar desde el nodo fuente hasta el nodo destino

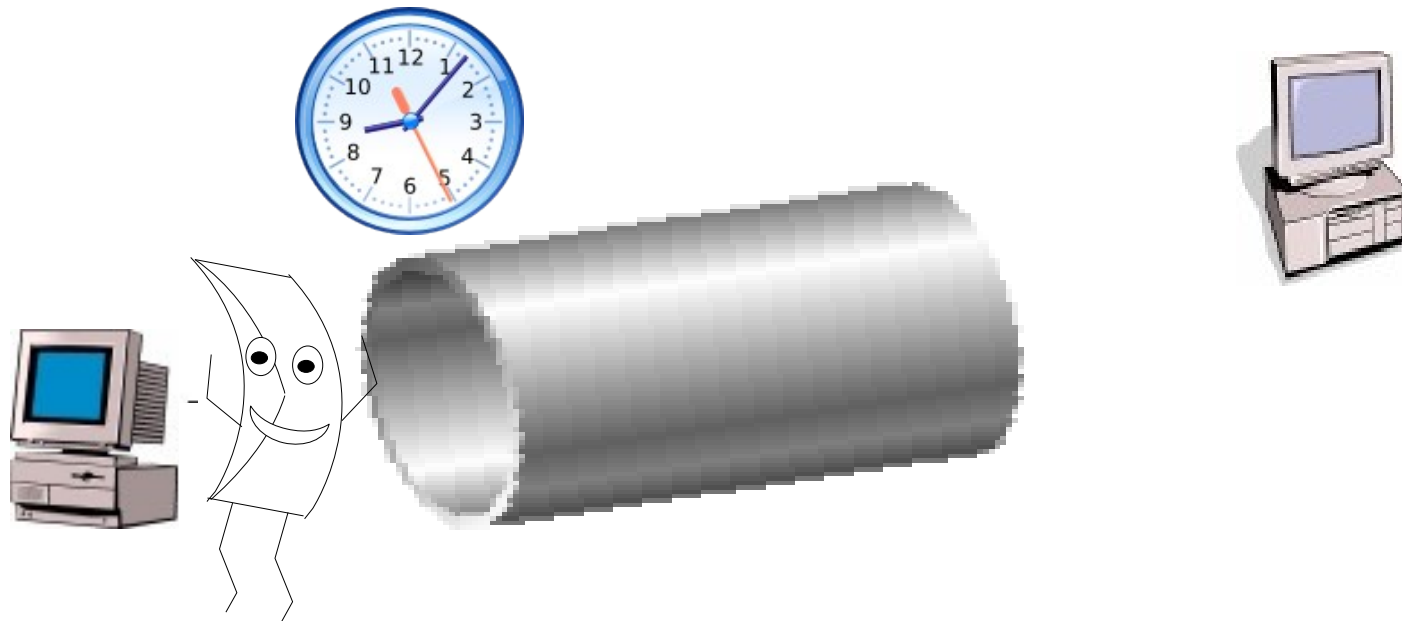


# Introducción a las Redes de Datos

## Prestaciones

### Latencia

Es el tiempo que le toma a un sólo paquete en viajar desde el nodo fuente hasta el nodo destino



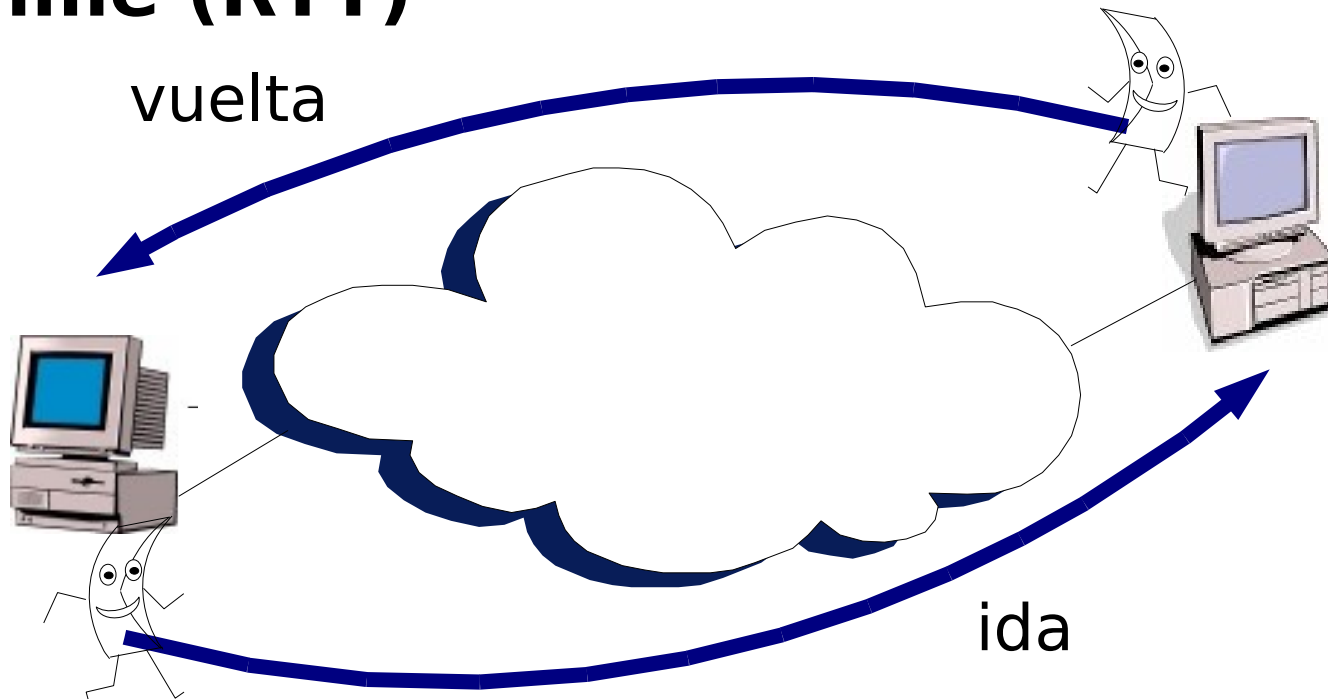


# Introducción a las Redes de Datos

## Prestaciones

### Latencia

Hay situaciones en las que es más interesante conocer el tiempo en que tarda un paquete en ir y regresar. A ese tiempo se le conoce como **Round Trip Time (RTT)**



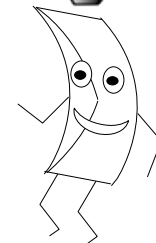
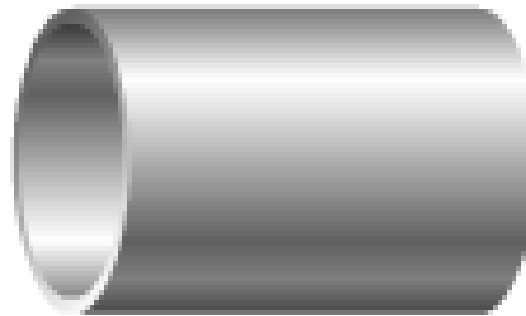
# Introducción a las Redes de Datos

## Prestaciones

**Latencia** Podemos ver la latencia como la suma de tres elementos

1) Propagación: es la velocidad de propagación de la luz. Nada es más rápido que la luz.

$3.0 \times 10^8$  m/s en el vacío



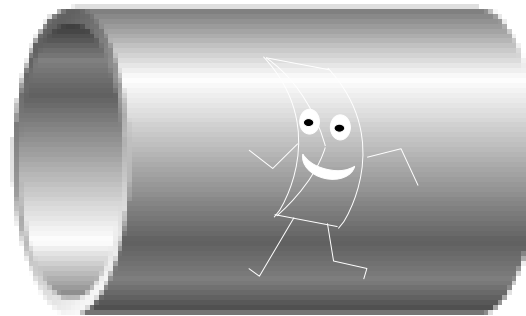
# Introducción a las Redes de Datos

## Prestaciones

**Latencia** Podemos ver la latencia como la suma de tres elementos

1) Propagación: es la velocidad de propagación de la luz. Nada es más rápido que la luz.

$2.3 \times 10^8$  m/s en un cable



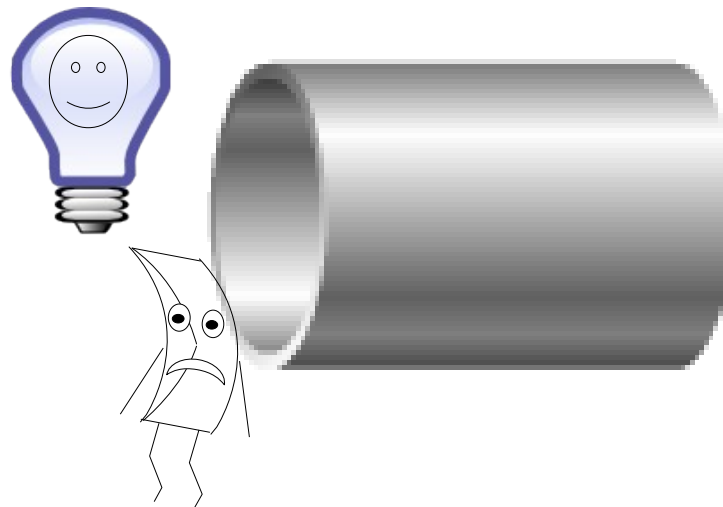
# Introducción a las Redes de Datos

## Prestaciones

**Latencia** Podemos ver la latencia como la suma de tres elementos

1) Propagación: es la velocidad de propagación de la luz. Nada es más rápido que la luz.

$2.0 \times 10^8$  m/s en  
fibra óptica

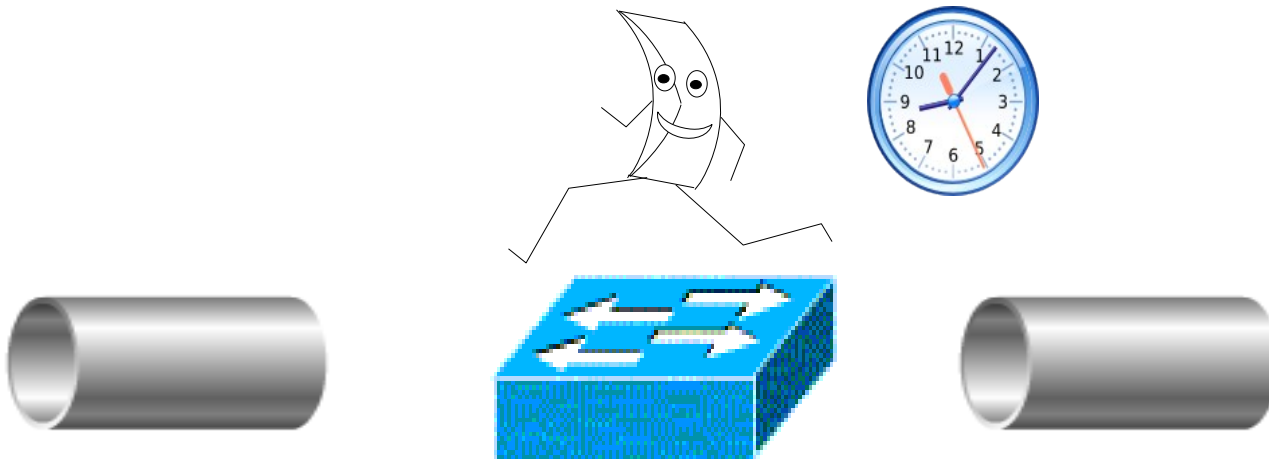


# Introducción a las Redes de Datos

## Prestaciones

**Latencia** Podemos ver la latencia como la suma de tres elementos

2) Tiempo en Cola: A los dispositivos de comunicación les toma cierto tiempo en retransmitir un paquete .

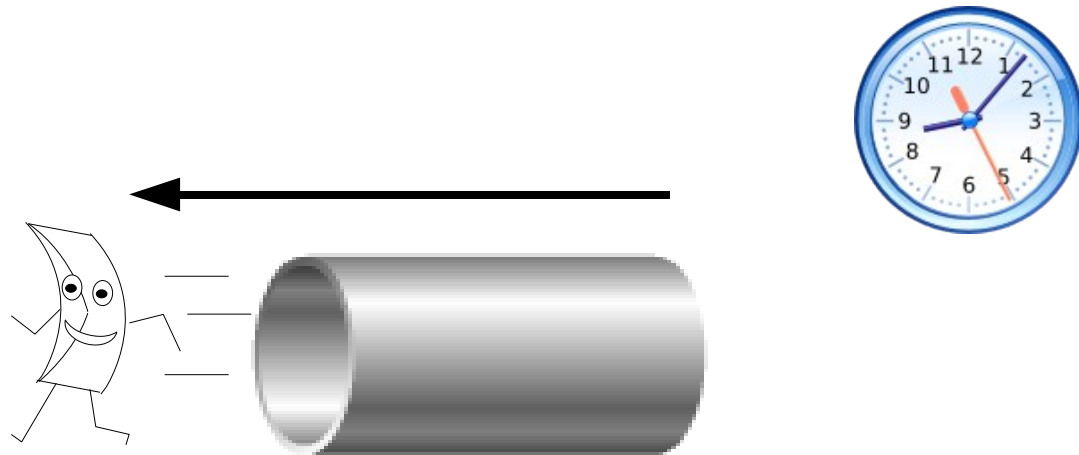


# Introducción a las Redes de Datos

## Prestaciones

**Latencia** Podemos ver la latencia como la suma de tres elementos

3) Transmisión: la cantidad de tiempo que toma transmitir una unidad de datos a través de un enlace.



# Introducción a las Redes de Datos

## Prestaciones

**Latencia =**

**Propagación + T. en cola + Transmisión**



+



+

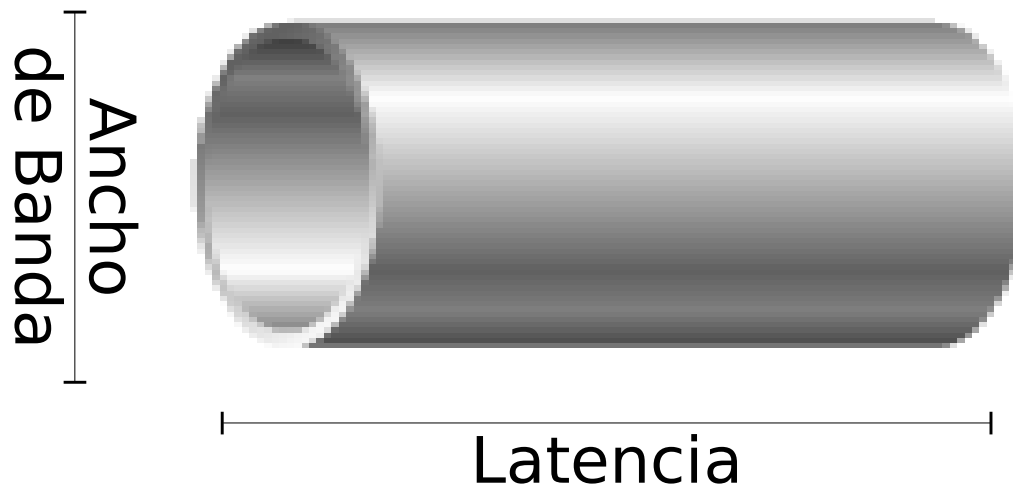


# Introducción a las Redes de Datos

## Prestaciones

### Producto Ancho de Banda por Latencia (PABL)

Es útil considerar esta métrica en ciertas ocasiones la cual puede ser considerada como el volumen del canal



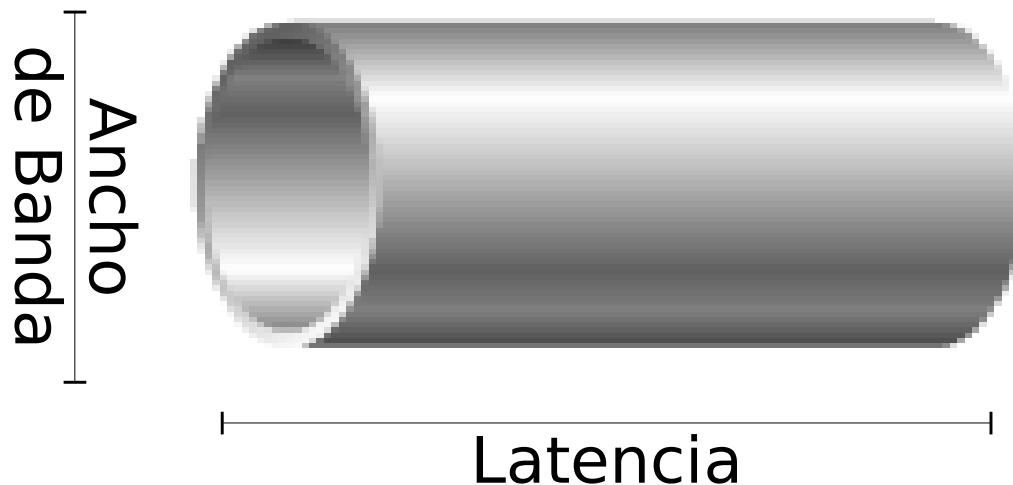


# Introducción a las Redes de Datos

## Prestaciones

### Producto Ancho de Banda por Latencia (PABL)

Si medimos los bits en unidades de tiempo podemos determinar cuantos de ellos caben en el canal



# Introducción a las Redes de Datos

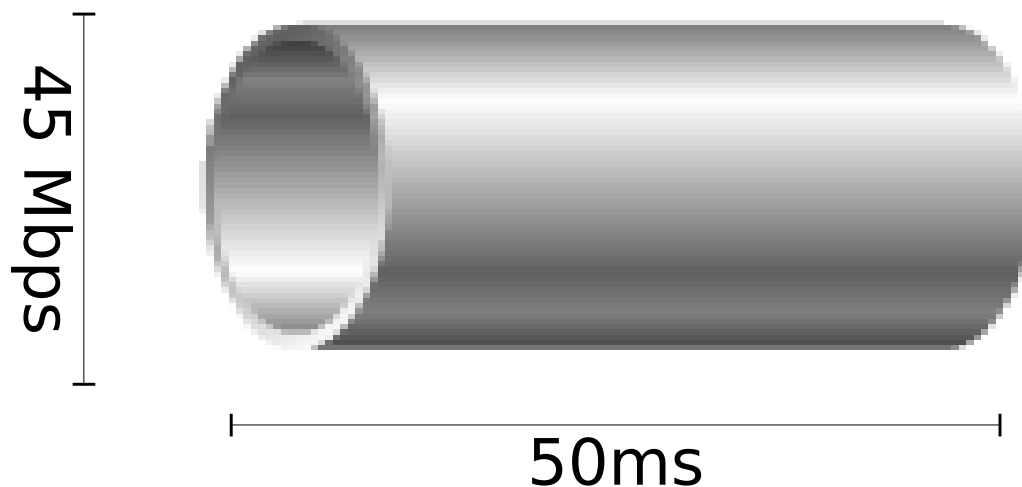
## Prestaciones

### Producto Ancho de Banda por Latencia (PABL)

Por ejemplo:

$$\text{No bits} = (50 \times 10^{-3} \text{ seg}) \times (45 \times 10^6 \text{ bits/seg})$$

$$\text{No bits} = 2.25 \times 10^6 \approx 280 \text{ KB}$$



# Introducción a las Redes de Datos

## Prestaciones

### **Producto Ancho de Banda por Latencia (PABL)**

Este producto es muy importante en el diseño de las redes de alto rendimiento pues corresponde a **cuantos bits puede enviar la fuente antes de que el primer bit alcance el destino.**

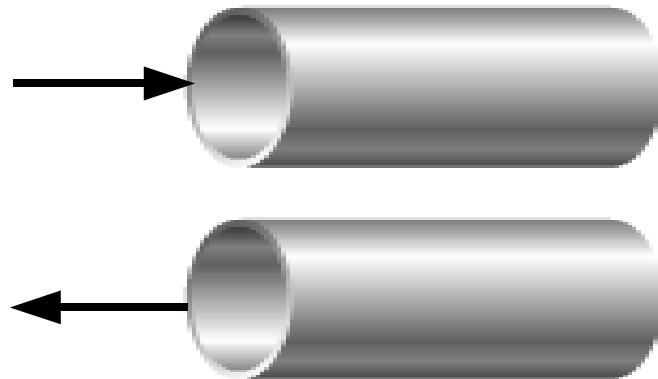
# Introducción a las Redes de Datos

## Prestaciones

### Producto Ancho de Banda por Latencia (PABL)

Supongamos que la fuente desea confirmación de la recepción de cada bit.

Para ello contamos con otro canal con iguales características. (Nótese que estamos interesados en el RTT)



# Introducción a las Redes de Datos

## Prestaciones

### Producto Ancho de Banda por Latencia (PABL)

Entonces la fuente puede enviar  $2 \times \text{PABL}$  antes de que comience a recibir confirmación del primer bit enviado

En nuestro ejemplo anterior esto corresponde a  $5.5 \times 10^6$  bits, aproximadamente 560 KB.

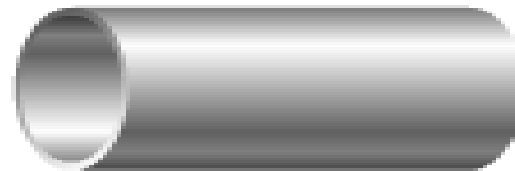
# Introducción a las Redes de Datos

## Prestaciones

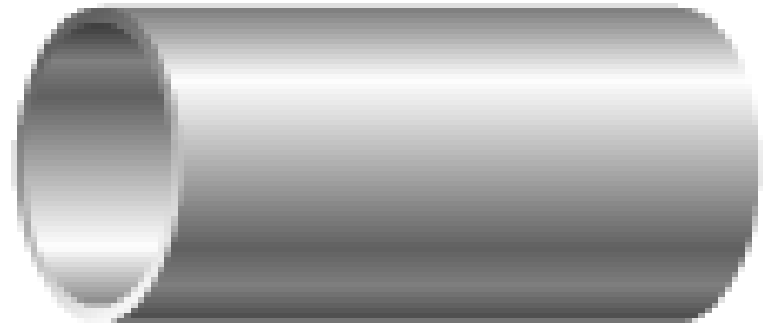
Ejemplo 1: consideremos dos enlaces, ambos con un RTT de 100ms



1-Mbps



1-Gbps



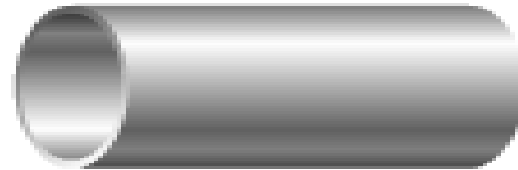
# Introducción a las Redes de Datos

## Prestaciones

Ejemplo 1: consideremos dos enlaces, ambos con un RTT de 100ms



1-Mbps



Al archivo le toma 100 RTTs para ser transmitido

# Introducción a las Redes de Datos

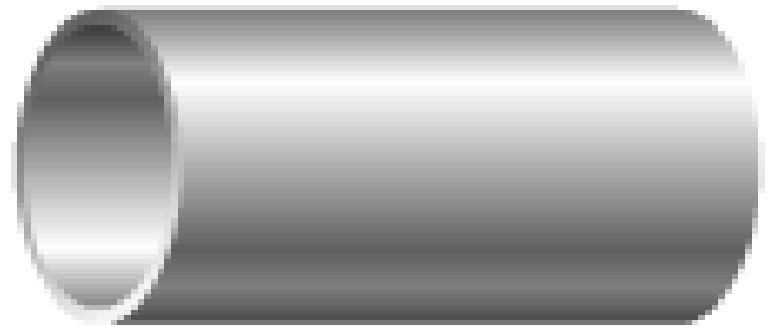
## Prestaciones

Ejemplo 1: consideremos dos enlaces, ambos con un RTT de 100ms



Al archivo le toma 1 RTT para ser transmitido

1-Gbps





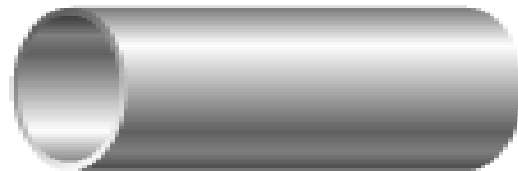
# Introducción a las Redes de Datos

## Prestaciones

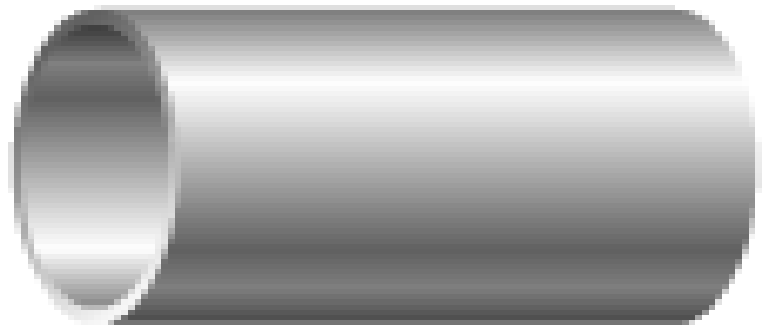
Ejemplo 2: consideremos los mismos enlaces, ambos con un RTT de 100ms. Al archivo le toma el mismo tiempo en los 2 enlaces (1 RTT)



1-Mbps



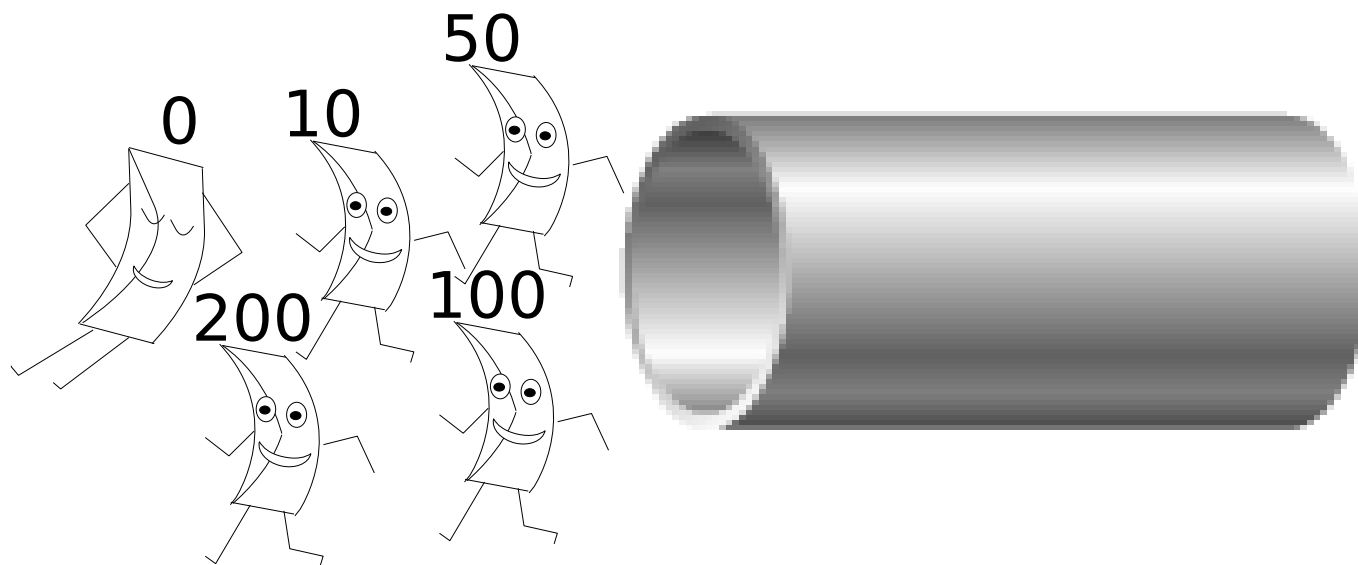
1-Gbps



# Introducción a las Redes de Datos

## Prestaciones

**Jitter:** se refiere a la variación (dispersión estadística o varianza) de las demoras de los paquetes. Las causas pueden ser diferentes comportamientos en las colas de los dispositivos de comunicación, problemas en los enlaces, etc.



# Introducción a las Redes de Datos

## Prestaciones

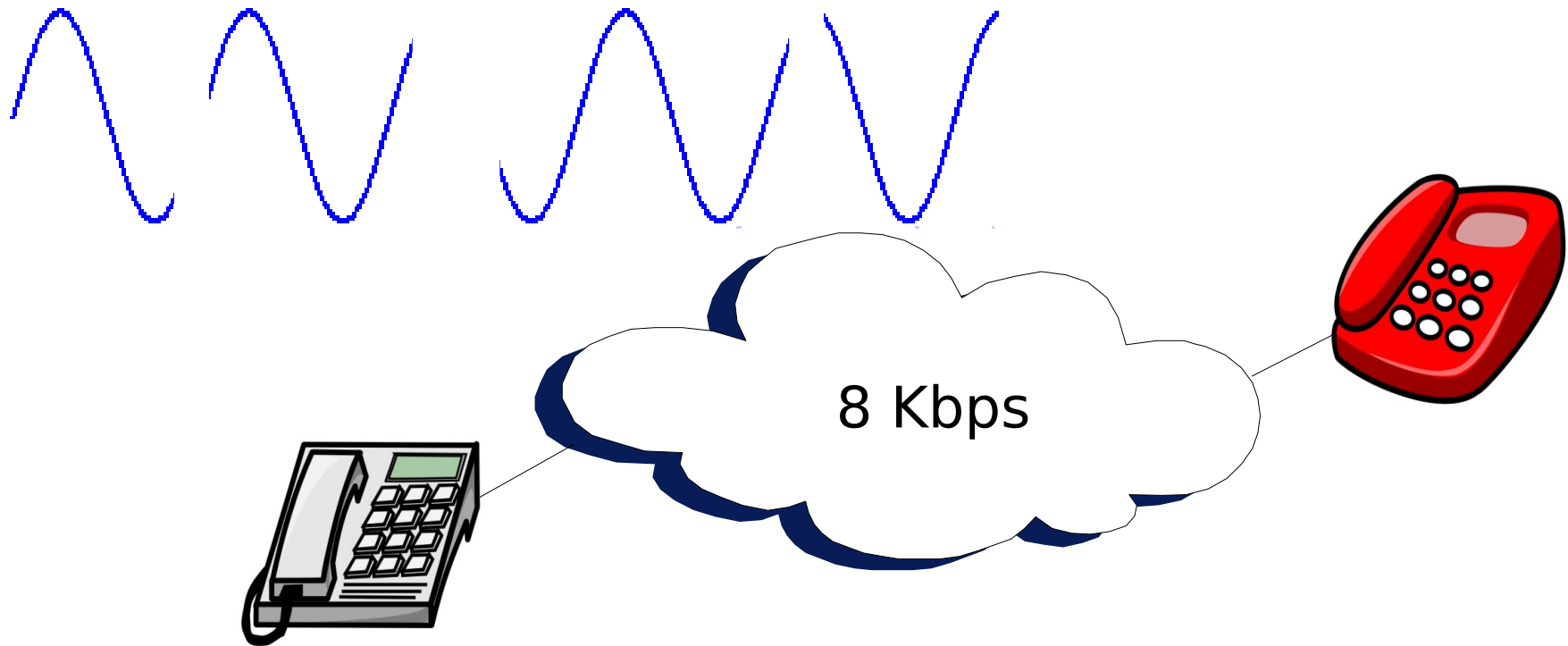
Los términos **Ancho de Banda** y **Latencia** son muy importantes a la hora de medir el rendimiento de una red

Sin embargo, la **aplicación** es quien marca la pauta en la decisión de cual de ellos es más importante.

# Introducción a las Redes de Datos

## Prestaciones

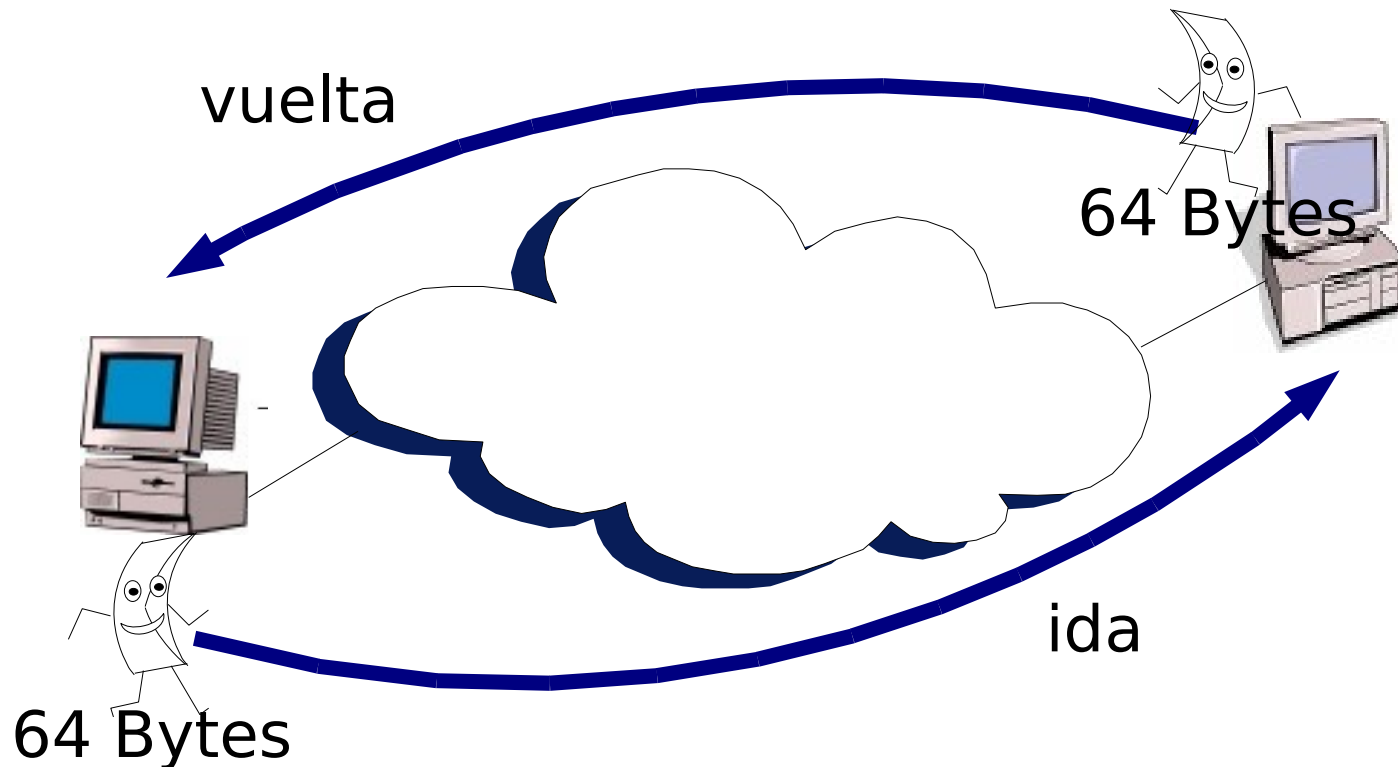
Por ejemplo, para transmitir voz por la red es mucho más importante el retardo de los paquetes que la cantidad de ellos que se pueda transmitir.



# Introducción a las Redes de Datos

## Práctica

Comando ping: este nos da una idea del estado del enlace entre la máquina local y la máquina remota



# Introducción a las Redes de Datos

## Práctica

Comando ping: Envía una secuencia de paquetes hasta la máquina destino y los recibe de vuelta

```
gilberto en odie>ping atlas
```

```
PING atlas.cecalc.ula.ve (150.185.138.1) 56(84) bytes of data.
```

```
64 bytes from atlas.cecalc.ula.ve (150.185.138.1): icmp_seq=1 ttl=61 time=0.936 ms
```

```
64 bytes from atlas.cecalc.ula.ve (150.185.138.1): icmp_seq=2 ttl=61 time=0.673 ms
```

```
64 bytes from atlas.cecalc.ula.ve (150.185.138.1): icmp_seq=3 ttl=61 time=0.783 ms
```

```
64 bytes from atlas.cecalc.ula.ve (150.185.138.1): icmp_seq=4 ttl=61 time=0.685 ms
```

```
64 bytes from atlas.cecalc.ula.ve (150.185.138.1): icmp_seq=5 ttl=61 time=0.679 ms
```

```
--- atlas.cecalc.ula.ve ping statistics ---
```

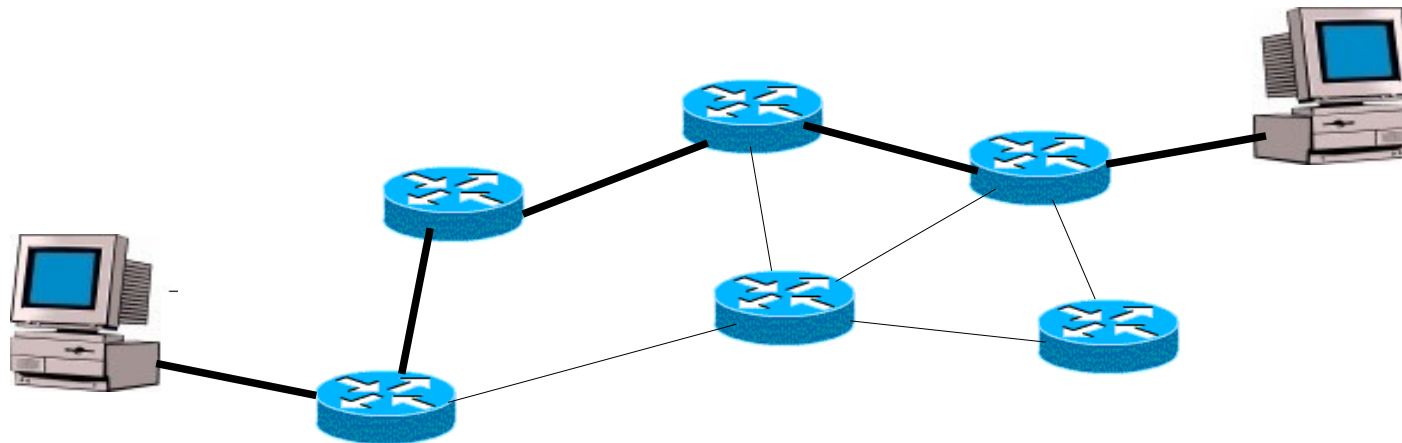
```
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
```

```
rtt min/avg/max/mdev = 0.673/0.751/0.936/0.102 ms
```

# Introducción a las Redes de Datos

## Práctica

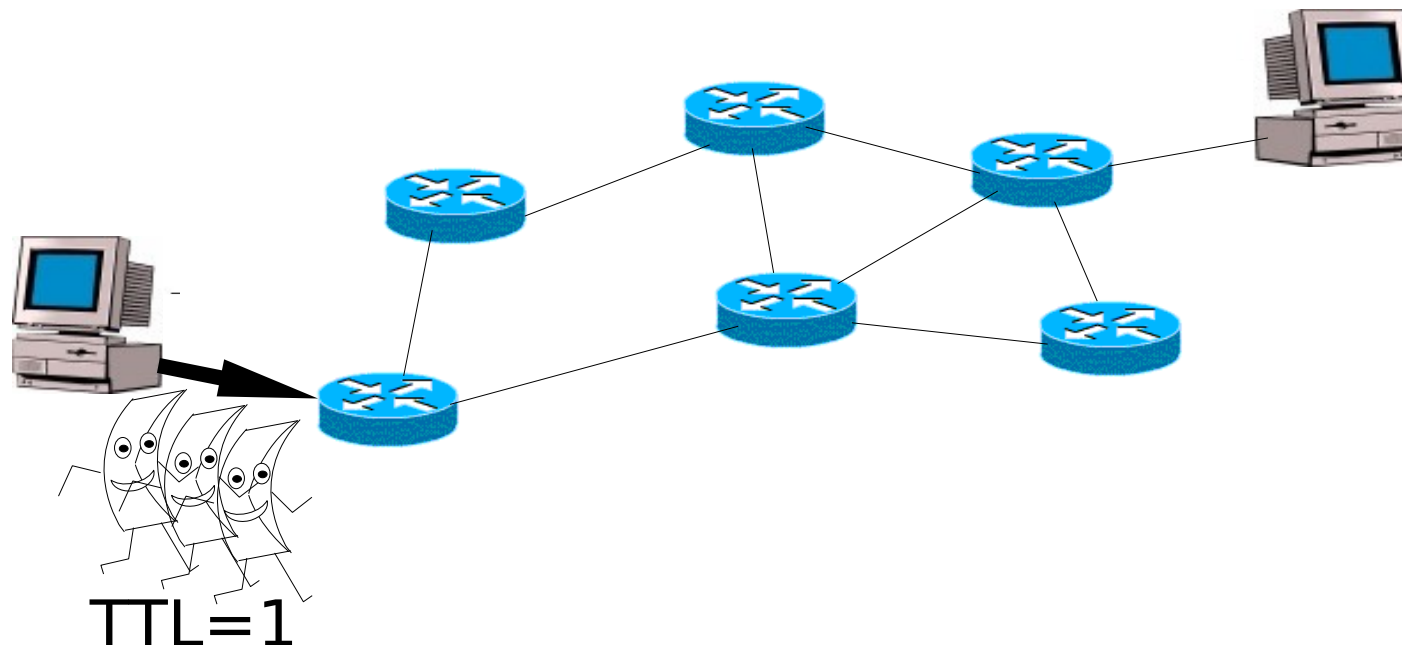
Comando traceroute: muestra la ruta que siguen los paquetes para alcanzar la máquina destino.



# Introducción a las Redes de Datos

## Práctica

Comando traceroute: Utiliza paquetes UDP con un TTL (Time To Live) muy pequeño

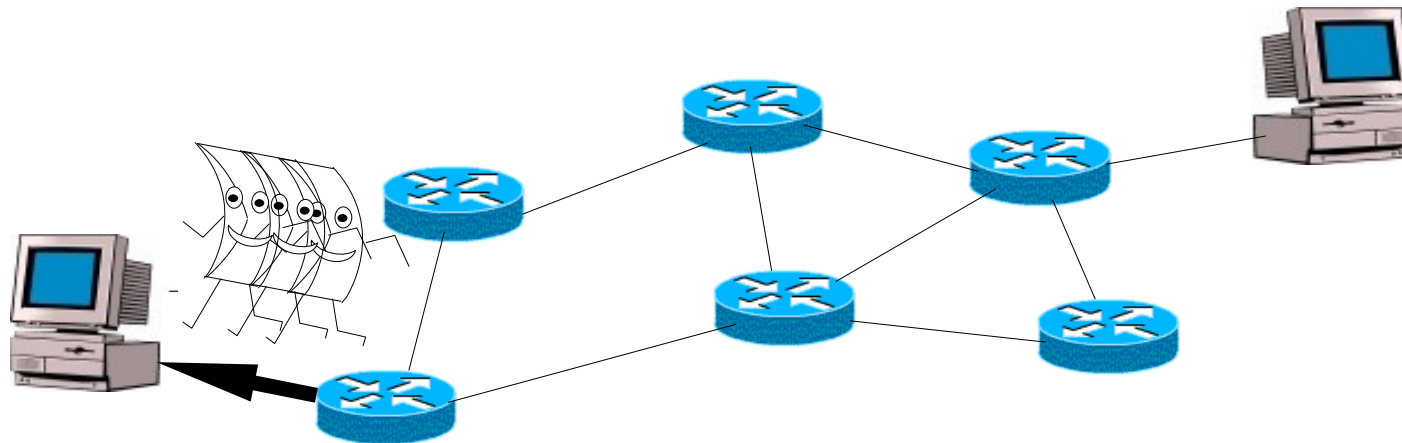




# Introducción a las Redes de Datos

## Práctica

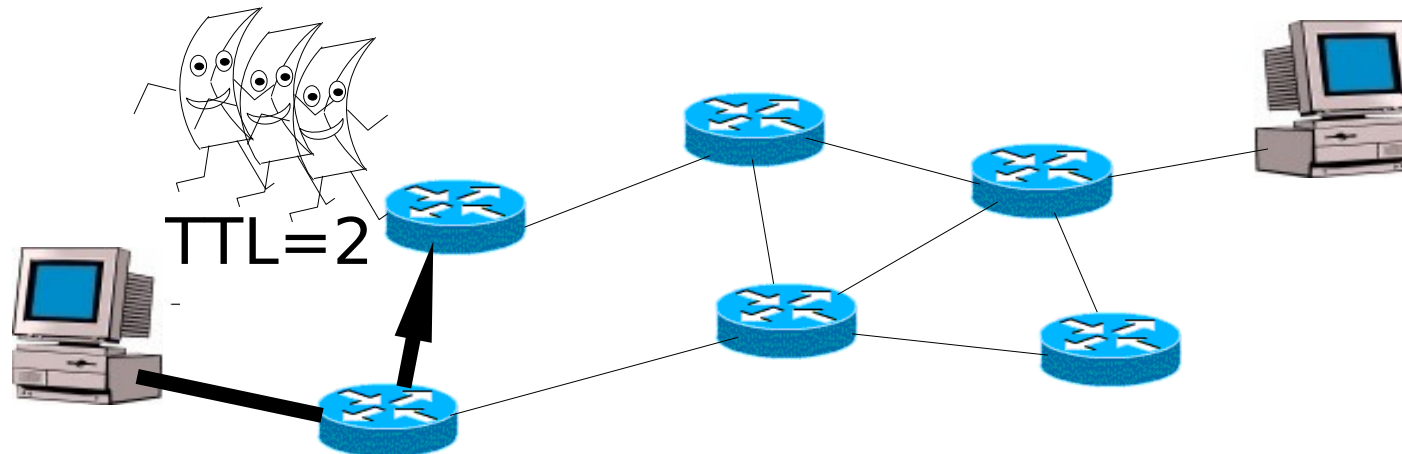
Comando traceroute: Luego espera la respuesta del router (ICMP TIME\_EXCEEDED)



# Introducción a las Redes de Datos

## Práctica

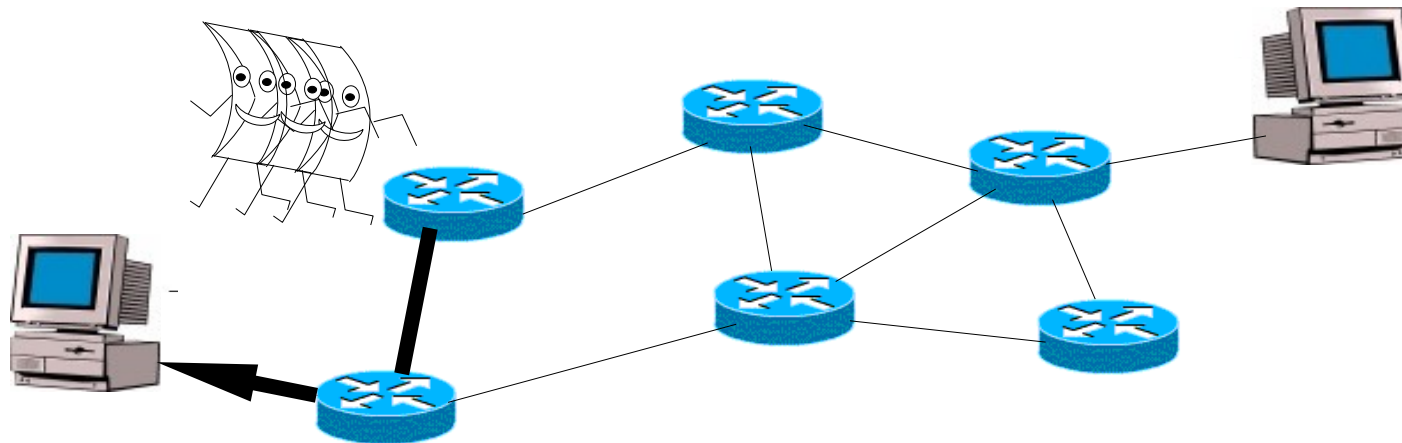
Comando traceroute: Aumenta el TTL de los tres paquetes



# Introducción a las Redes de Datos

## Práctica

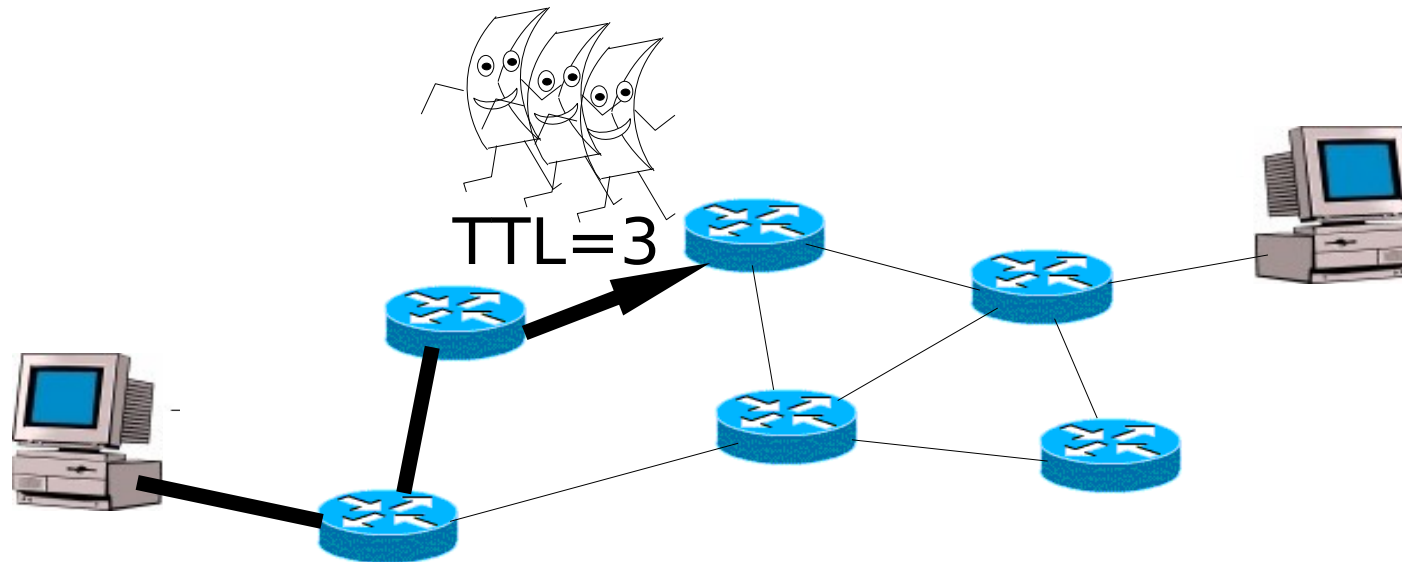
Comando traceroute: sucesivamente va aumentando el TTL



# Introducción a las Redes de Datos

## Práctica

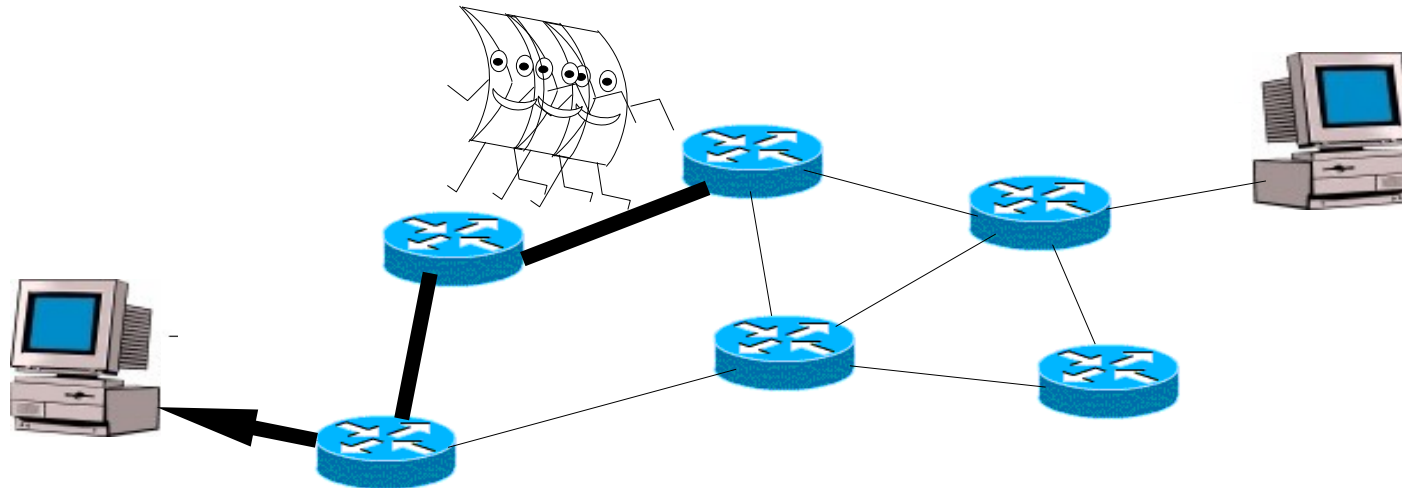
Comando traceroute: sucesivamente va aumentando el TTL



# Introducción a las Redes de Datos

## Práctica

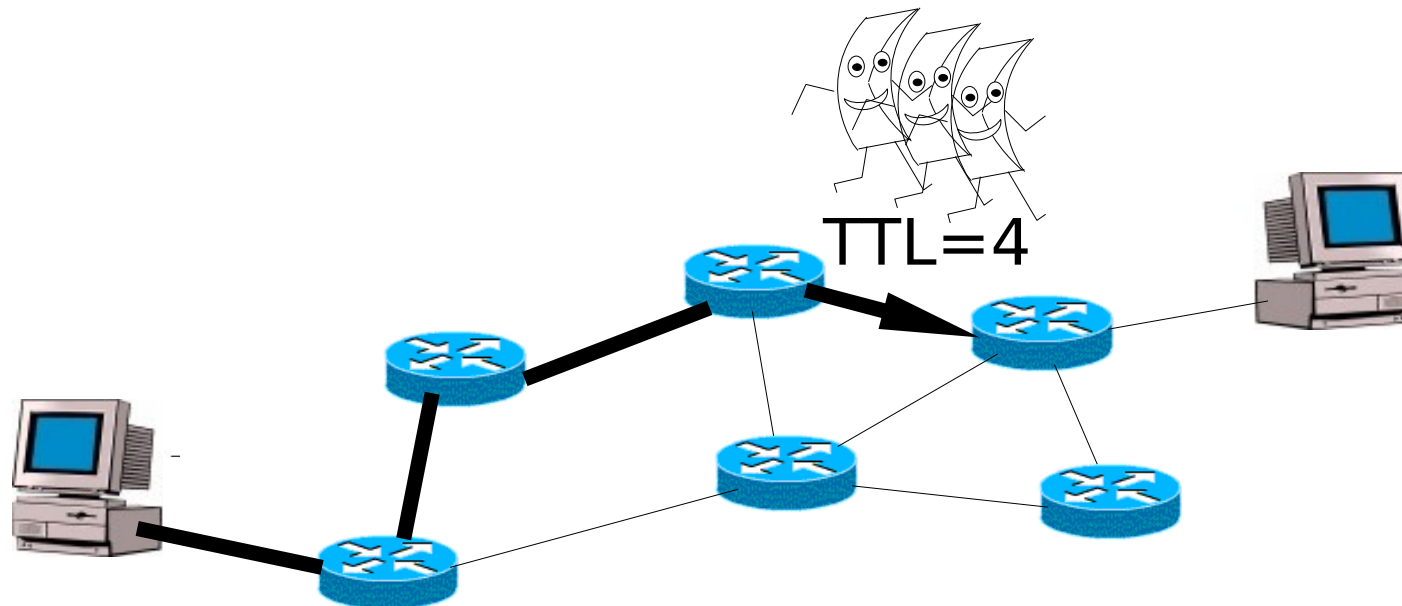
Comando traceroute: sucesivamente va aumentando el TTL



# Introducción a las Redes de Datos

## Práctica

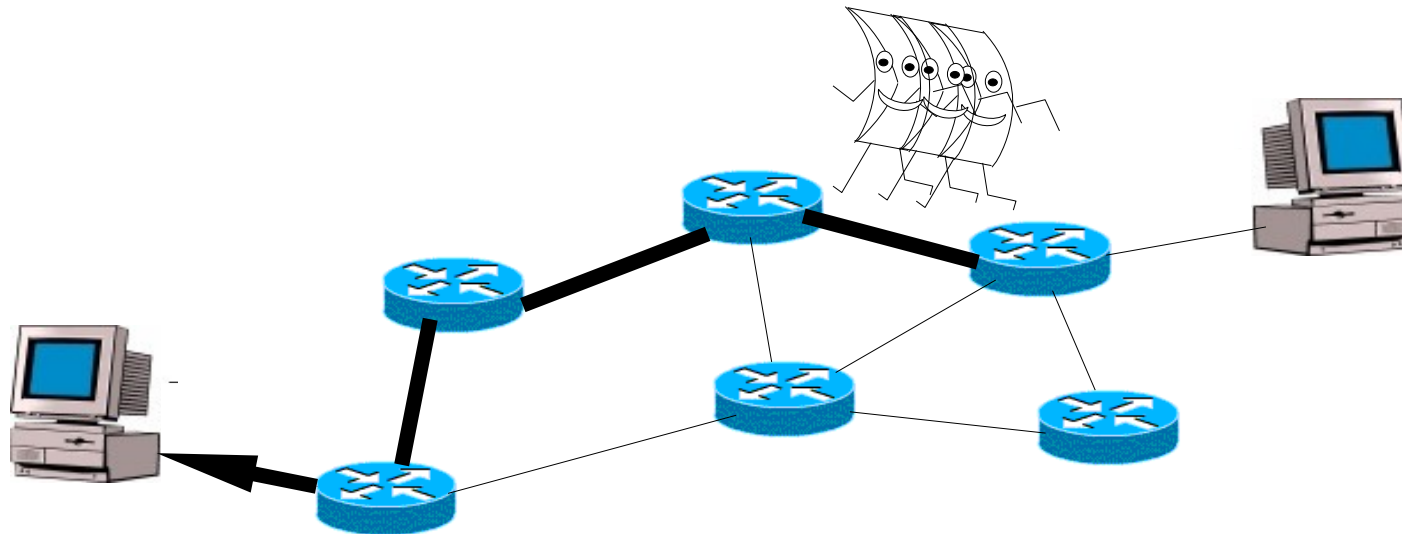
Comando traceroute: sucesivamente va aumentando el TTL



# Introducción a las Redes de Datos

## Práctica

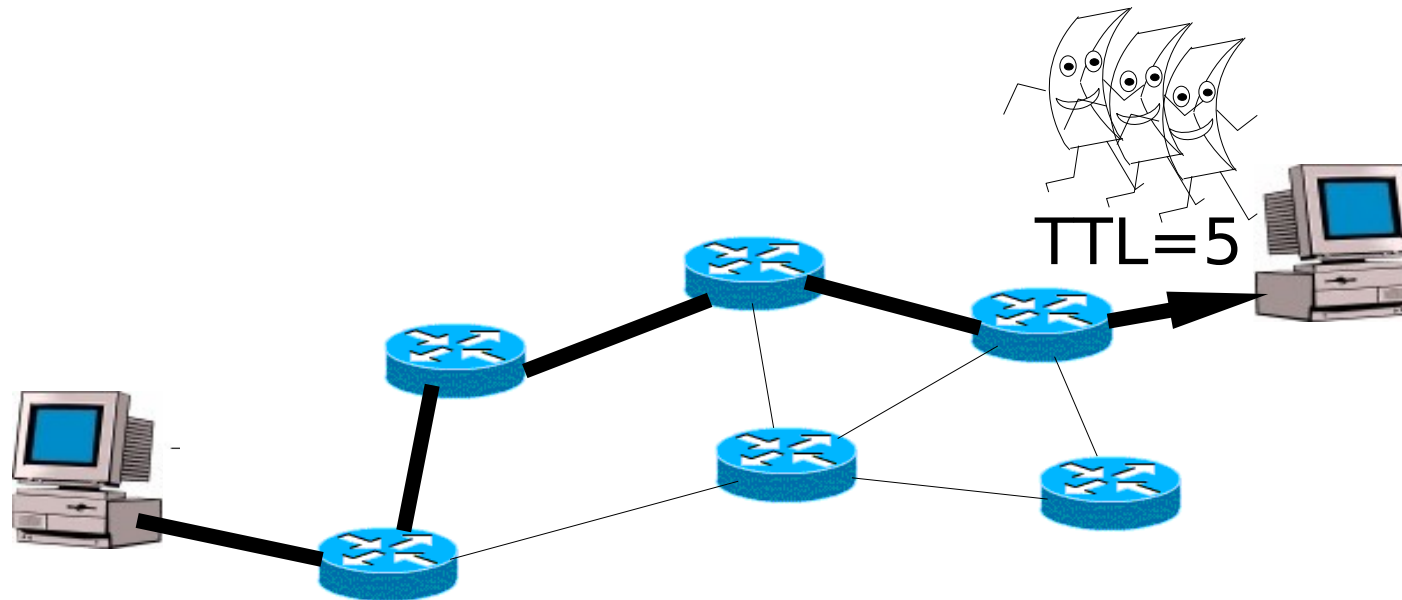
Comando traceroute: sucesivamente va aumentando el TTL



# Introducción a las Redes de Datos

## Práctica

Comando traceroute: sucesivamente va aumentando el TTL

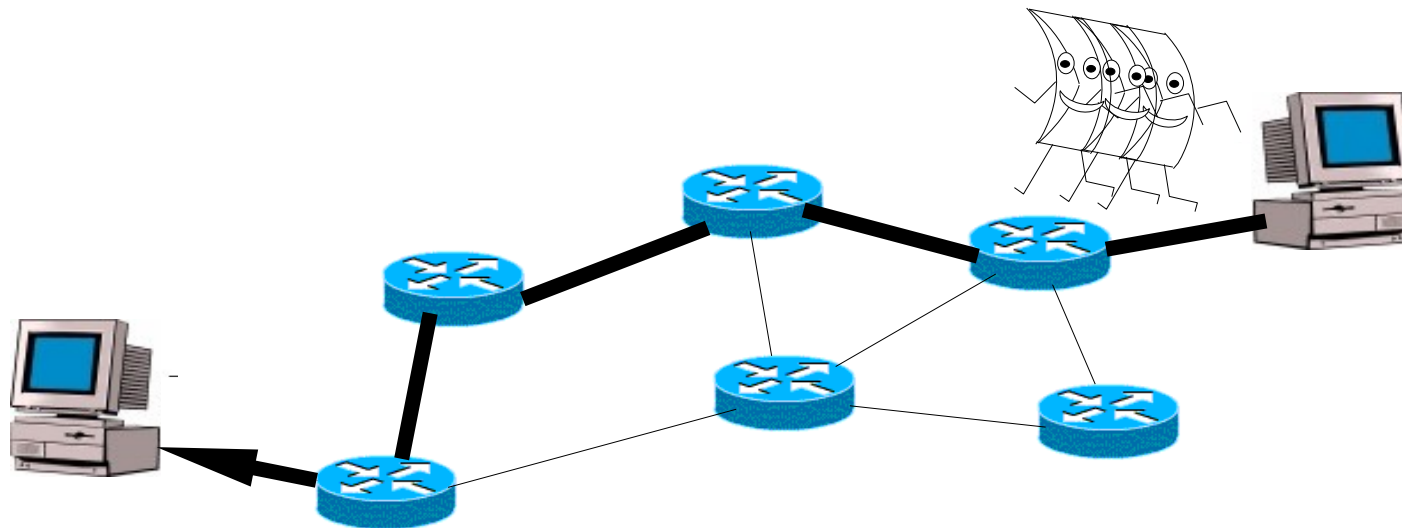




# Introducción a las Redes de Datos

## Práctica

Comando traceroute: hasta obtener un paquete **ICMP port unreachable** que significa que la máquina destino fue alcanzada



# Introducción a las Redes de Datos

## Práctica

Comando traceroute: muestra la ruta que siguen los paquetes para alcanzar la máquina destino. En cada salto se puede visualizar el tiempo que toma en llegar a ese tramo.

```
gilberto en odie>traceroute atlas
```

```
traceroute to atlas.cecalc.ula.ve (150.185.138.1), 30 hops max, 40 byte packets
```

```
1 150.185.181.254 (150.185.181.254) 1.187 ms 0.451 ms 0.443 ms
```

```
2 150.185.163.250 (150.185.163.250) 0.474 ms 0.395 ms 0.381 ms
```

```
3 150.185.141.1 (150.185.141.1) 0.599 ms 0.488 ms 0.724 ms
```

```
4 * * *
```

```
5 * * *
```

# Introducción a las Redes de Datos

## Práctica

Comando traceroute: Algunas veces existen firewalls que impiden este tipo de tráfico.

```
gilberto en odie>traceroute -I atlas
```

```
traceroute to atlas.cecalc.ula.ve (150.185.138.1), 30 hops max, 40 byte packets
```

```
1 150.185.181.254 (150.185.181.254) 0.520 ms 0.440 ms 0.433 ms
2 150.185.163.250 (150.185.163.250) 0.409 ms 0.377 ms 0.382 ms
3 150.185.141.1 (150.185.141.1) 3.834 ms 0.465 ms 0.444 ms
4 atlas.cecalc.ula.ve (150.185.138.1) 0.620 ms 0.492 ms 0.488 ms
```

# Introducción a las Redes de Datos

## Práctica

Comando `tcptraceroute`: Algunas veces existen firewalls que impiden este tipo de tráfico.

```
odie ~ # tcptraceroute atlas 80
```

```
Selected device eth0, address 150.185.181.44, port 60818 for outgoing packets
```

```
Tracing the path to atlas (150.185.138.1) on TCP port 80 (http), 30 hops max
```

```
1 150.185.181.254 0.541 ms 0.440 ms 0.433 ms
```

```
2 150.185.163.250 0.421 ms 0.384 ms 0.371 ms
```

```
3 150.185.141.1 0.500 ms 0.452 ms 0.443 ms
```

```
4 atlas.cecalc.ula.ve (150.185.138.1) [open] 0.606 ms 0.500 ms 0.497 ms
```

# Introducción a las Redes de Datos

## Práctica

Comando iperf: Sirve para medir el throughput entre una máquina fuente y una máquina destino. Se levanta un servidor en un extremo

```
odie ~ # iperf -s
```

```
-----  
Server listening on TCP port 5001  
TCP window size: 60.0 KByte (default)  
-----
```

```
[ 4] local <IP Addr node2> port 5001 connected with <IP Addr node1> port  
2357
```

```
[ ID] Interval      Transfer    Bandwidth
```

```
[ 4] 0.0-10.1 sec  6.5 MBytes 5.2 Mbits/sec
```

# Introducción a las Redes de Datos

## Práctica

Comando iperf: y en el otro extremo se ejecuta el comando como un cliente

```
node1> iperf -c node2
```

```
-----  
Client connecting to node1, TCP port 5001  
TCP window size: 59.9 KByte (default)  
-----
```

```
[ 3] local <IP Addr node1> port 2357 connected with <IP Addr node2> port  
5001
```

```
[ ID] Interval      Transfer    Bandwidth
```

```
[ 3] 0.0-10.0 sec   6.5 MBytes 5.2 Mbits/sec
```