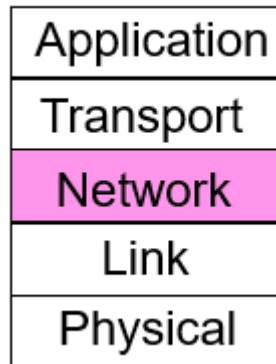


# Inundación

## Capa de Red



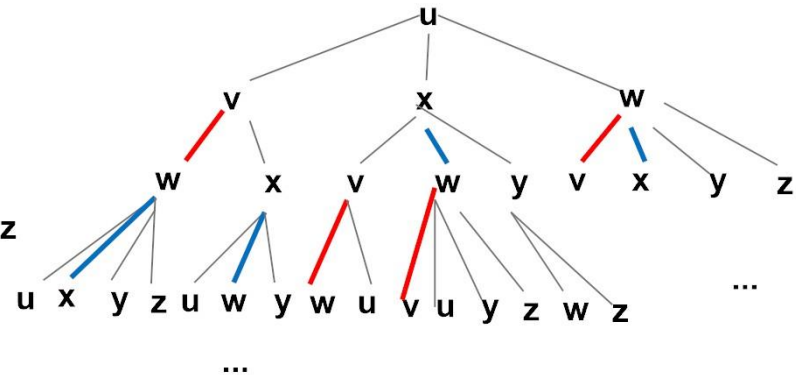
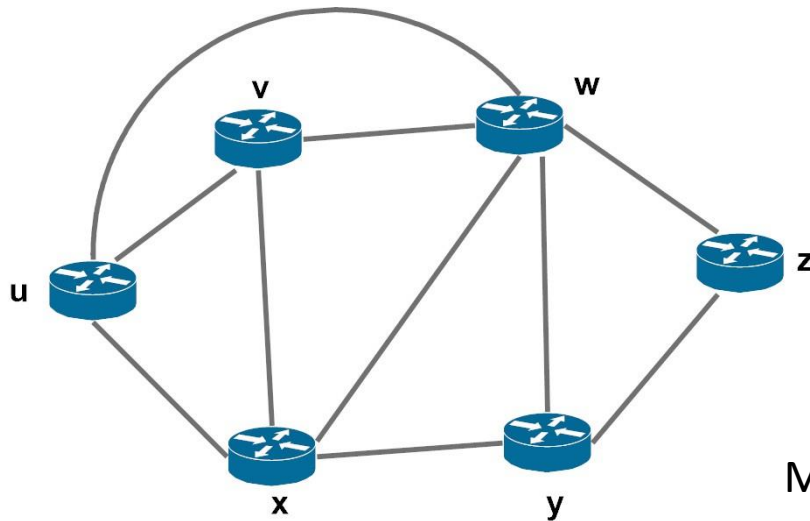
# Algoritmos de enrutamiento

- **Algoritmos de enrutamiento:** buscan determinar las rutas a seguir para enviar paquetes de un origen a un destino.
  - Ese conjunto de rutas se puede definir por medio de un conjunto de reglas a respetar.
  - Los algoritmos de enrutamiento se preocupan de actualizar las tablas de reenvío de mensajes en los enrutadores.

# Inundación

- **Idea de inundación:** para enviar un paquete de un origen  $u$  a un destino  $v$  los caminos usados son aquellos que respetan las siguientes reglas:
  - $u$  manda el mensaje por todas las líneas de salida.
  - Cada paquete que llega a un enrutador distinto de  $v$  se reenvía por cada una de las líneas excepto aquella por la que llegó.

# Inundación



Mandar paquete de *u* a *z*.

- **Problemas de la idea anterior:**

- La inundación genera grandes cantidades de *paquetes duplicados*;
  - Árbol de envío de paquetes. Cada arco representa un paquete que se envía.
    - P.ej. arcos rojos y azules en el árbol de arriba corresponden a paquetes duplicados.
  - Árbol de envío de paquetes es infinito con infinitos duplicados. O sea se generan infinitas rutas.
    - La causa es la presencia de ciclos en el grafo de la subred.
- a menos que se tomen algunas medidas para limitar el proceso.

# Inundación con registro de paquetes difundidos

- Hace falta limitar un poco el proceso de inundación dado en la idea anterior para resolver el problema.
- **Solución:** cada enrutador recuerda **los paquetes difundidos** previamente por él para decidir si acepta un paquete.

# Inundación con registro de paquetes difundidos

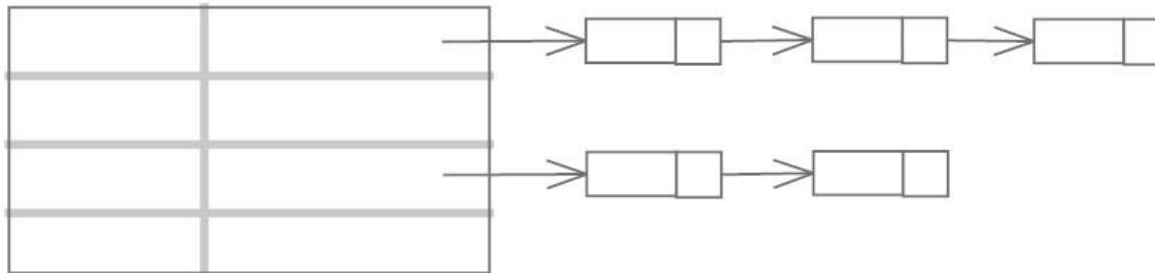
- **Refinamiento de la solución anterior:**

- ❑ El enrutador de origen pone un **número de secuencia** en cada paquete que recibe de sus hosts (así se distingue entre paquetes distintos del mismo enrutador de origen).
- ❑ Un enrutador recuerda para cada enrutador de origen los números de secuencia recibidos – i.e. pares <enrutador de origen, n° secuencia>
- ❑ Si llega un paquete a un enrutador con par <enrutador de origen, número de secuencia> recibido antes, no se lo reenvía.

# Inundación con registro de paquetes difundidos

- **Implementación:** Para cada enrutador usar **tabla de registro de paquetes difundidos**.

Nº enrutador    Lista de Nº de  
de origen    sequencia vistos



# Inundación con registro de paquetes difundidos

- ¿Cómo se puede evitar que las listas enlazadas *crezcan sin límites*?

- ☐ Agregar una columna **contador** que indica el mayor número de secuencia tal que:
- ☐ Llegaron paquetes con todos los números de secuencia anteriores desde ese enrutador de origen.





# Inundación

- **Inundación con contador de saltos:** integrar un **contador de saltos** en el encabezado de cada paquete, que disminuya con cada salto y el paquete se descarte cuando el contador llega a 0.
- **¿Cómo se determina el contador de saltos?**
  - ☐ Lo ideal es inicializar el contador de saltos a la **longitud de la ruta entre el origen y el destino**.
  - ☐ Si el emisor desconoce el tamaño de la ruta, puede inicializar el contador al peor caso, es decir, al **diámetro total de la subred**.

# Inundación

- **Inundación Selectiva:** una idea para la inundación bastante práctica, es la **inundación selectiva:**
  - ❑ Los enrutadores no envían cada paquete de entrada por todas las líneas, sino solo por aquellas que van aproximadamente en la dirección correcta.
  - ❑ **¿Qué tipo de información necesita almacenar un enrutador para poder aplicar inundación selectiva?**
    - Se necesita saber en qué dirección va cada línea.
    - Se necesita saber en qué dirección está el destino.