

Algoritmos de Ruteo

31 Mayo 2005

Enrutamiento por Vector de Distancia

- Llamado tambien:
 - Bellman-Ford
 - Ford-Fulkerson
 - RIP

Usado Inicialmente en ARPANET

Componentes Principales

- Cada enrutador posee una tabla con
- una entrada por cada destino
conteniendo la distancia a el mismo
- cada enrutador comparte la tabla
con sus vecinos.

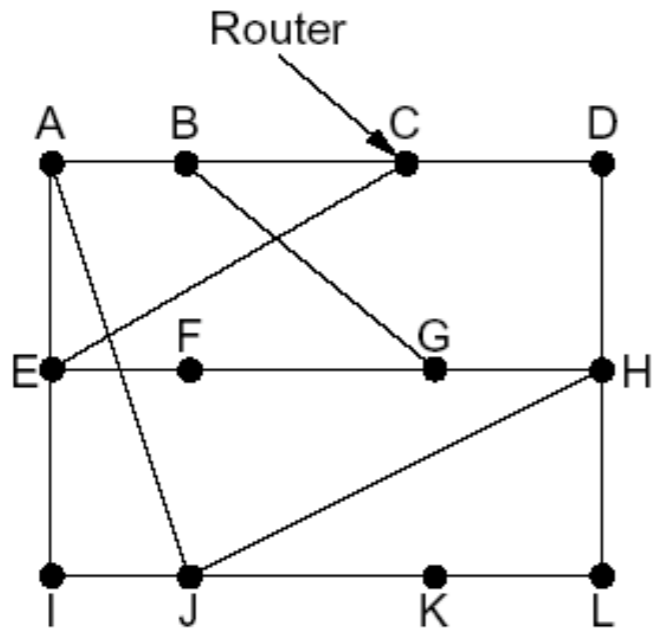
Metricas

- cantidad de saltos
- retardo
- numero total de paquetes encolados

Algoritmo

- Cada enrutador comparte su tabla con sus vecinos
- cada enrutador recibe de sus vecinos sus respectivas tablas
- cada enrutador calcula una nueva tabla utilizando las tablas de sus vecinos y la distancia los vecinos

Ejemplo



To	A	I	H	K	New estimated delay from J	
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K

JA delay is 8 JI delay is 10 JH delay is 12 JK delay is 6

Vectors received from J's four neighbors

New routing table for J

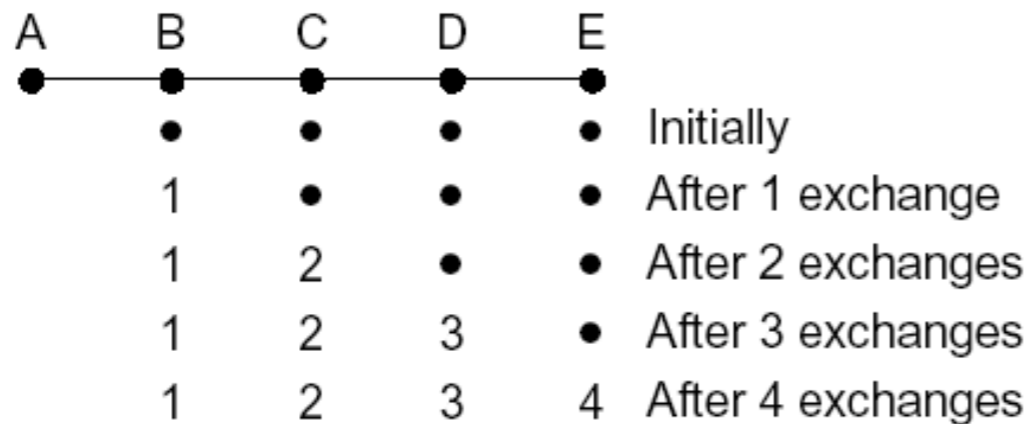
(a)

(b)

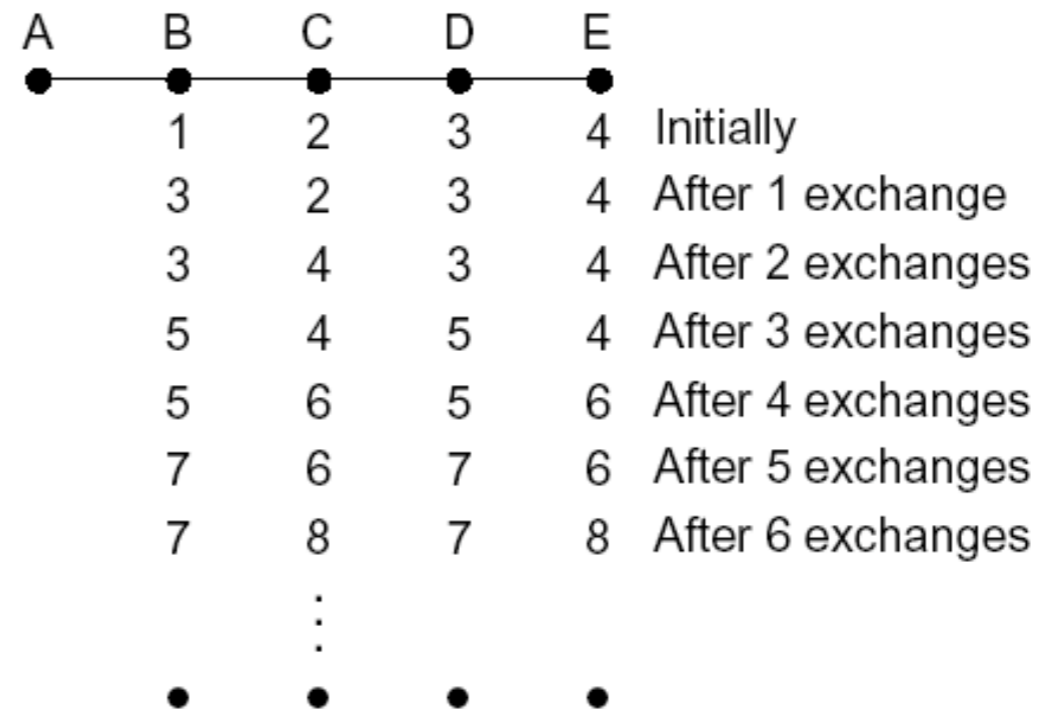
Calcular la Distancia de J a G

Defectos y Virtudes

- las buenas noticias viajan rapido
- las malas no tanto...



(a)



(b)

Enrutamiento por estado de enlace

- Descubrir a sus vecinos y conocer sus direcciones de red
- medir el retardo a sus vecinos
- contruir un paquete que indique esto
- enviar este paquete a todos los demas
- calcular la ruta mas corta a todos

conociendo vecinos

- Paquete HELLO por cada linea punto a punto
- respuesta con identificadores
- Identificadores de enrutadores únicos!

midiendo la linea

- enviar paquetes ECHO a los vecinos
- calcular el tiempo de ida y vuelta
- enviar varias veces para obtener valores mas útiles
- tener o no en cuenta la carga: pros y contras.

Construcción de paquetes

- una edad,
- una lista de vecinos
- identificador del emisor,
- y un numero de secuencia,
- una edad,
- una lista de vecinos

Cuando Construir?

- Faciles de construir, pero cuando?
 - Periodica: a intervalos regulares
 - cuando ocurra un evento (caida de un enlace, nuevo vecino, etc)

distribucion de paquetes

- Inundación: como contralarla?
 - Cada enrutador guarda (número de secuencia, origen)
 - Se descartan viejos y ya vistos
 - una edad se utiliza para evitar problemas con los numeros de secuencia

calculo de rutas

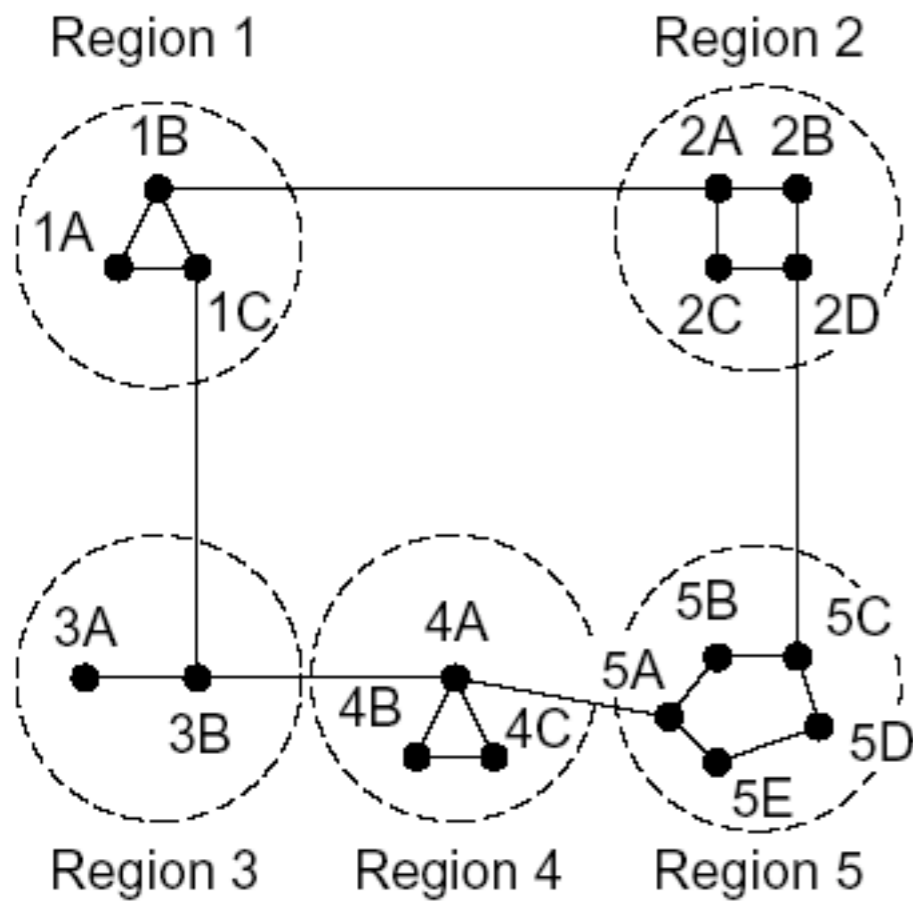
- se puede correr el algoritmo de Dijkstra localmente
- en grandes redes puede ser un problema: necesita demasiada memoria, el grafo puede ser erroneo.

Enrutamiento Jerarquico

Ninguno de los algoritmos es escalable:
todos requieren que cada enrutador sepa de todos los demas => demasiado exigente.

Solución: Buscar lineas suboptimas definiendo regiones y separar algoritmos intra-regiones e inter-regiones

Ejemplo



Full table for 1A

Dest.	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

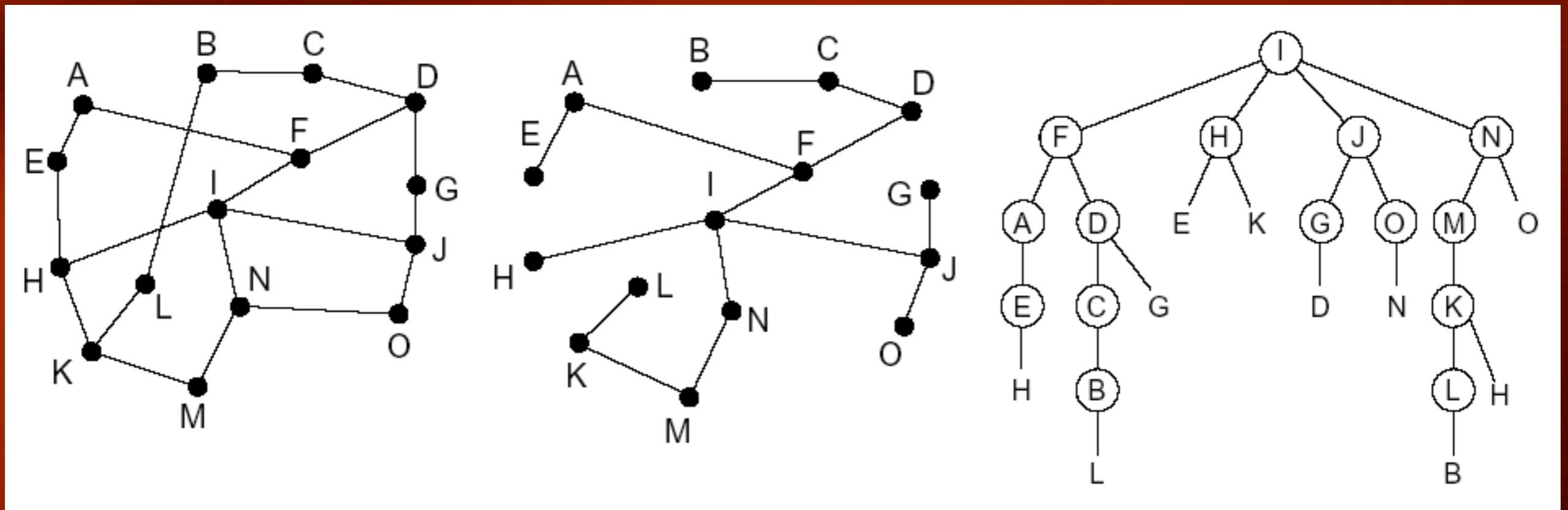
Enrutamiento Broadcast

Problema: Queremos enviar un mensaje a casi todas las máquinas

- 1) Enviar un mensaje a cada maquina individualmente
- 2) Usar inundación: Funciona si se puede controlarla
- 3) Usar mapas de bits: Un mapa va con cada paquete
- 4) Construir un arbol 'sink', al estilo de Dijkstra

construyendo el arbol

Como construir un arbol de 'sink' de **i**



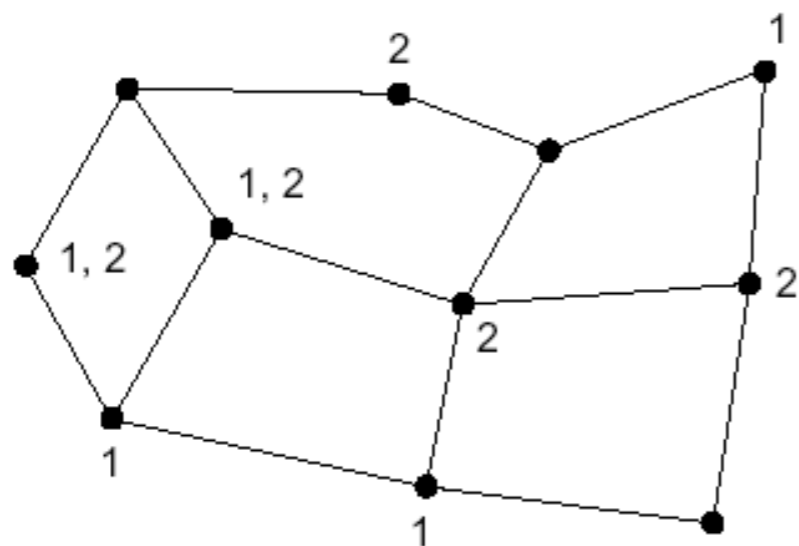
Reenviar un paquete si viene por la ruta preferida

Enrutamiento por multidifusión

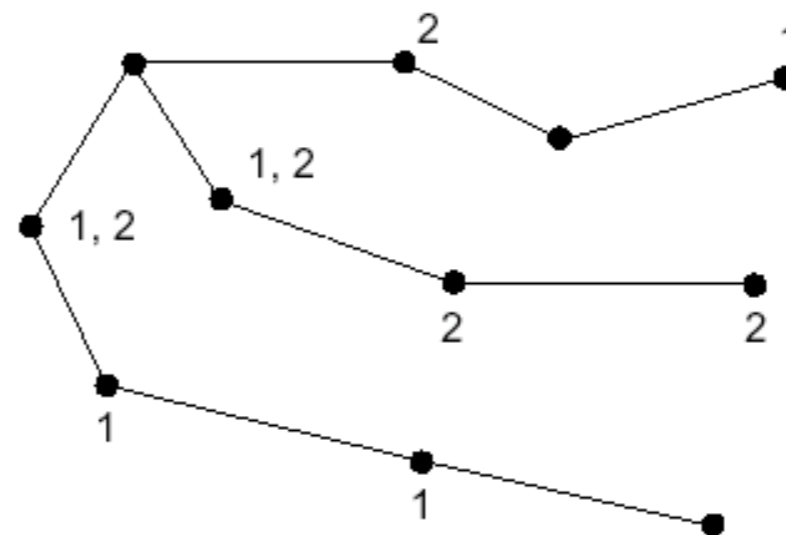
Problema: Queremos mandar un mensaje solo a un subconjunto de nodos. Necesitamos saber cuando entramos y salimos de un nodo en la subred

Solución: Construir un árbol de expansión (en cada enrutador) para toda la red. Usar el id-grupo para cortar caminos del árbol.

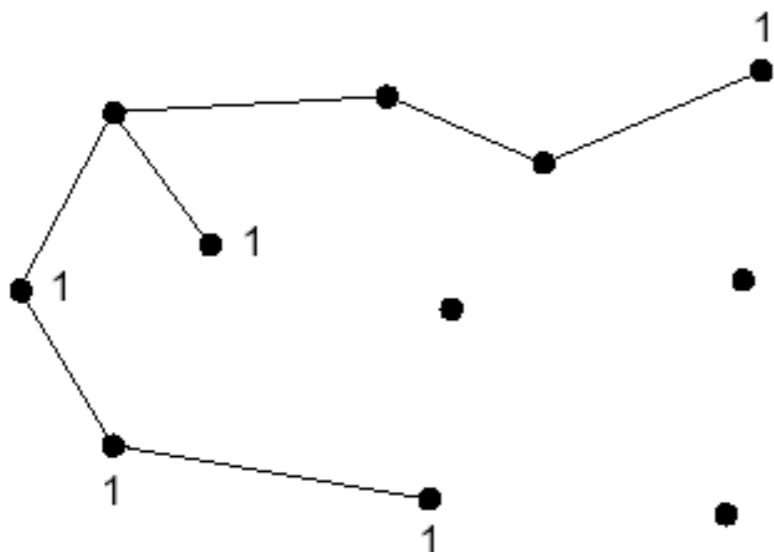
Ejemplo



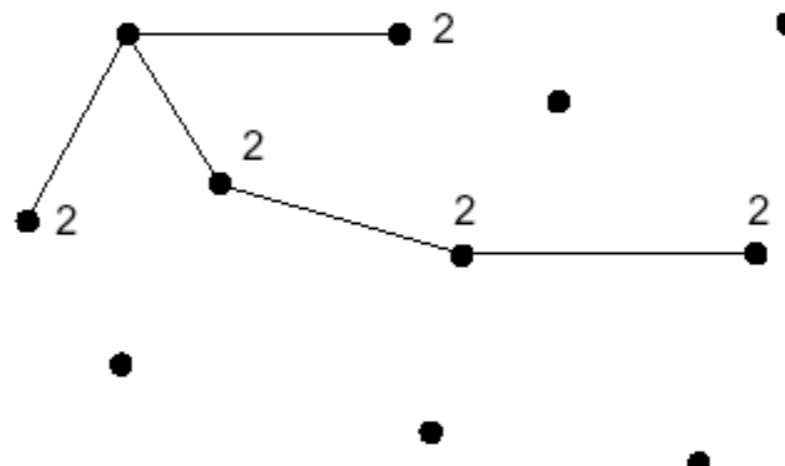
(a)



(b)



(c)



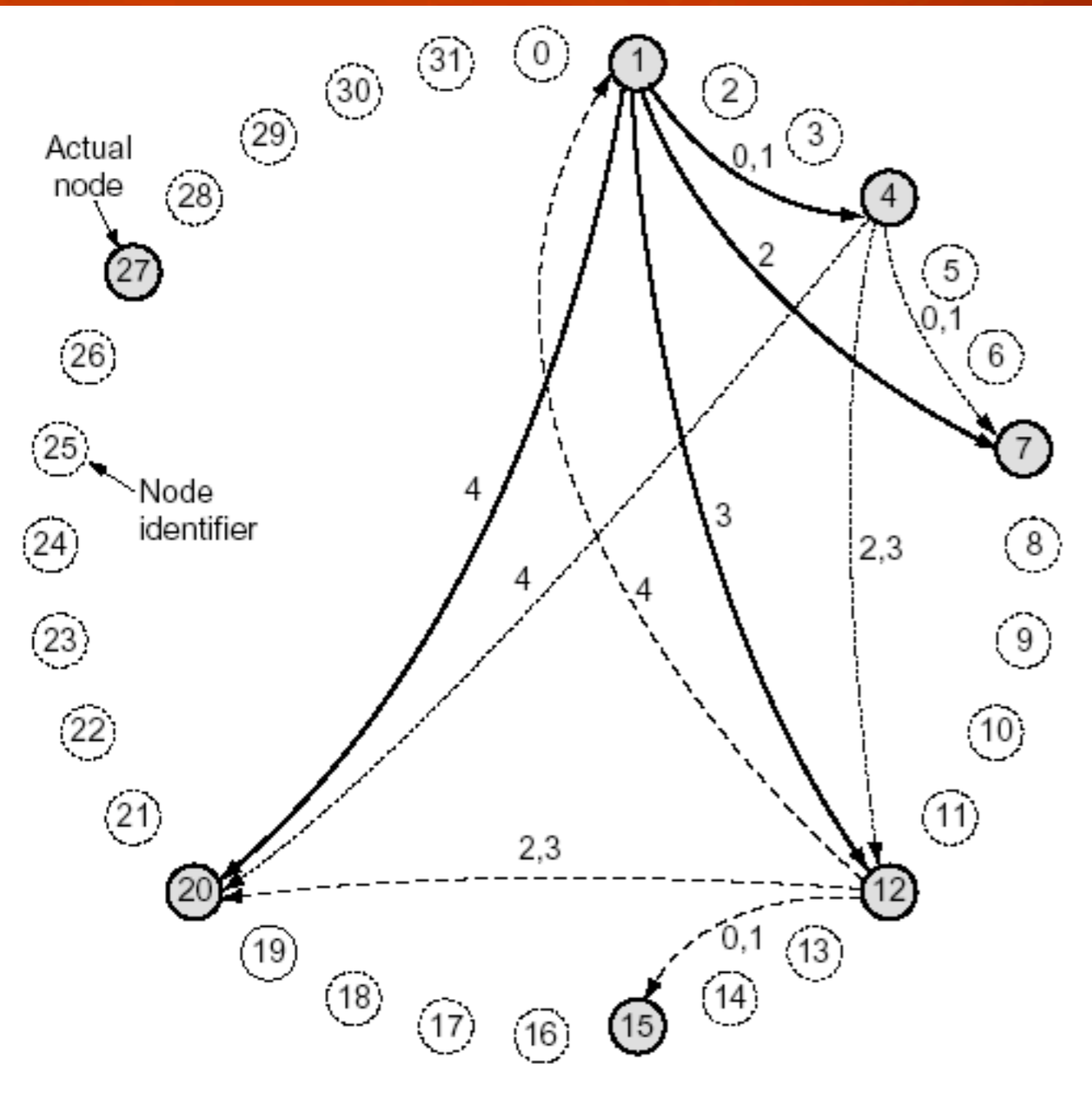
(d)

Redes peer-to-peer

Preliminares: Consiremos una colección grande de nodos, cada uno con una red IP. Cada nodo tiene un identificador de 160 bit. Logicamente todos los 2^{160} nodos se organizan en un anillo.

Importante: Cada maquina almacena archivos. Cada archivo tiene una llave de 160 bits. Se anume que el nodo con el ID mas bajo mayor que *llave* almacena el archivo (identificado con *succ(llave)*).

Busquedas en redes peer-to-peer



los nodos conocen
a su sucesor

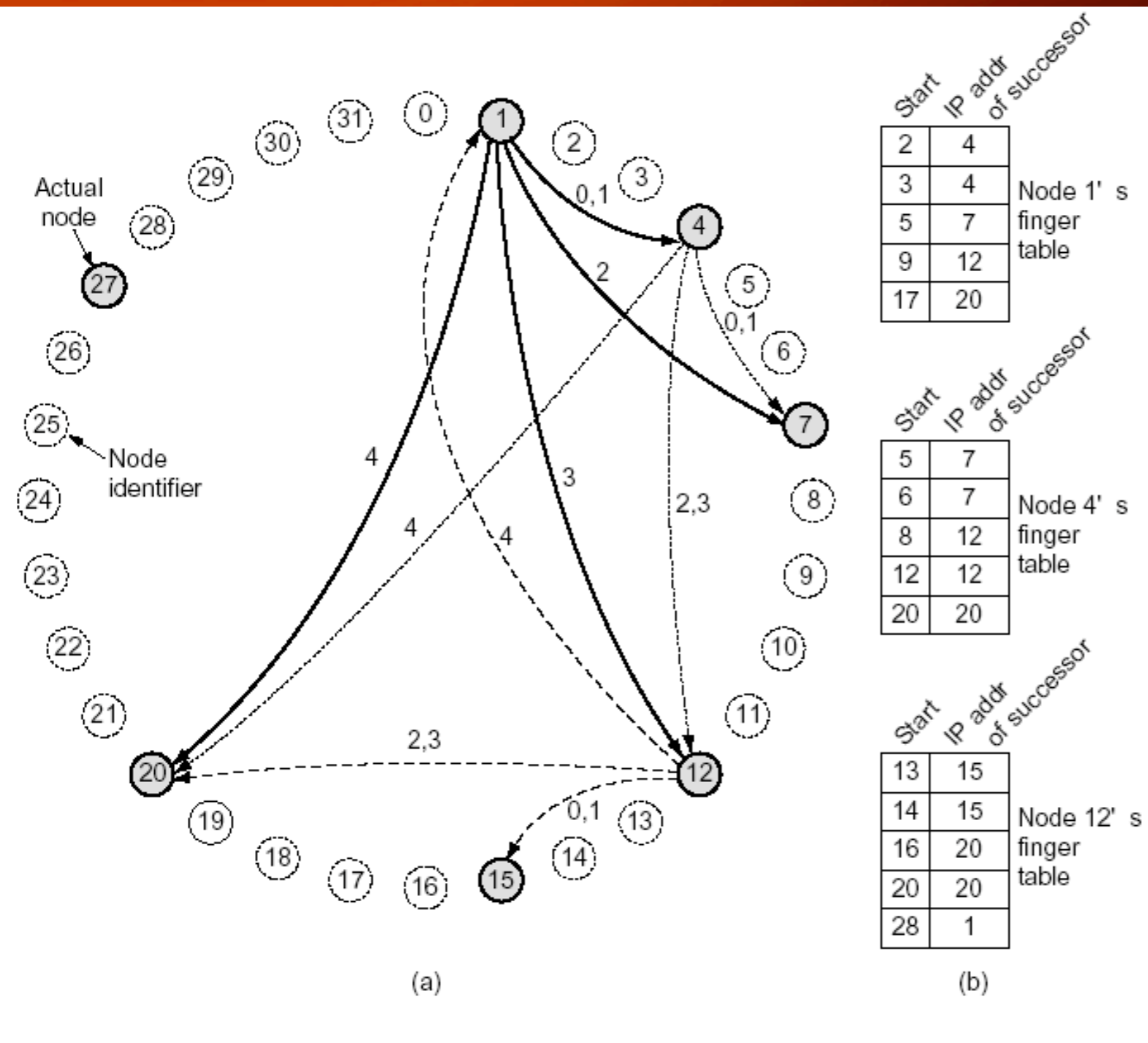
se envia IP del
solicitante y clave

Los paquetes de busqueda
se propagan en el anillo
hasta que se encuentra el
buscado

Optimización

- Tablas Finger:
 - m entradas, indexadas de 0 a $m-1$
 - dos campos por entrada: inicio y $\text{succ}(\text{inicio})$
 - $\text{inicio} = k + 2^i \pmod{2^m}$

Ejemplo



Como buscar?

Si clave *llave* esta entre el nodo k y $\text{succ}(k)$ la busqueda se manda a $\text{succ}(k)$

Sino, se busca en la tabla de finger el nodo mas cercano inferior a *clave*

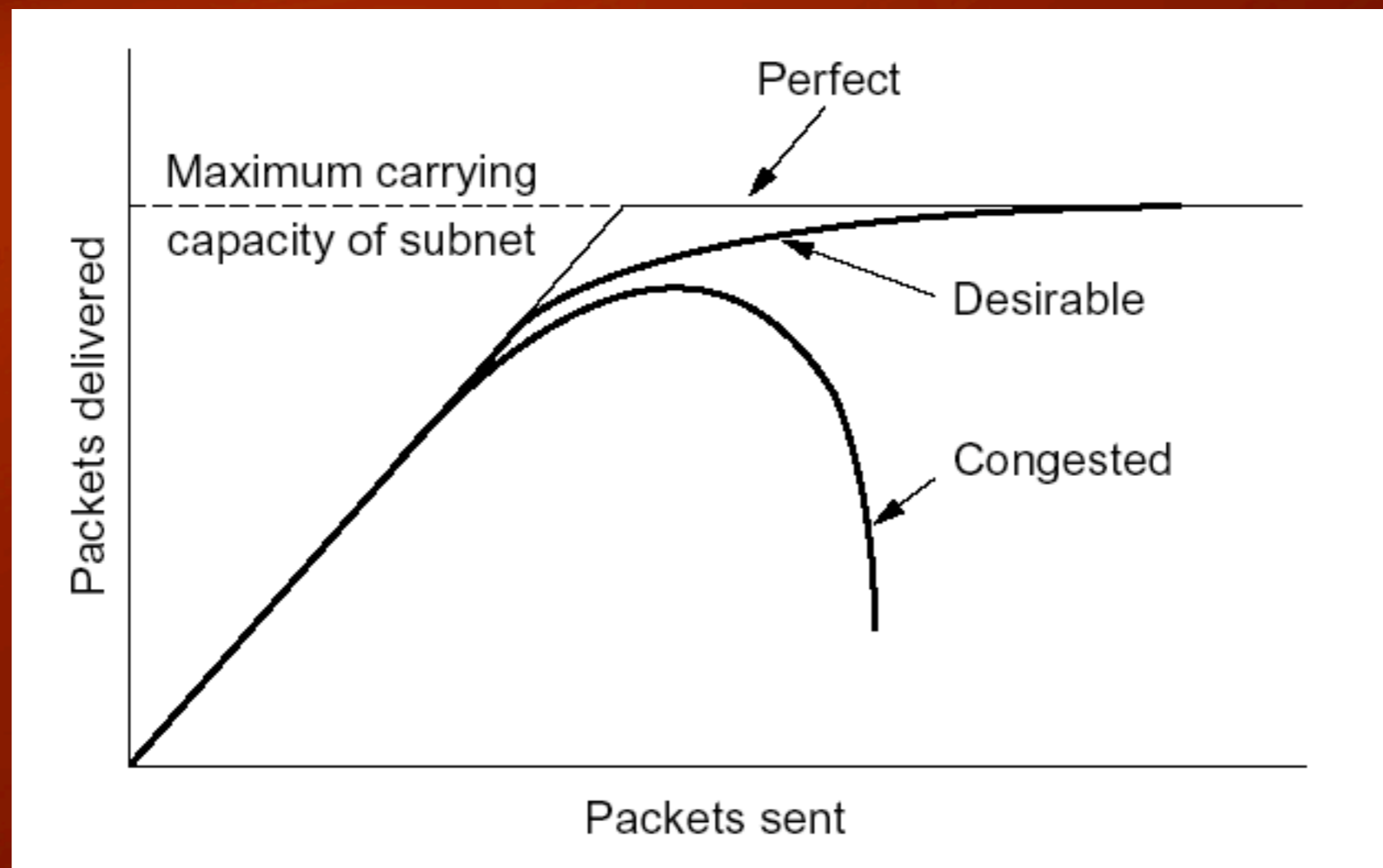
Ejemplos: Buscar 3, 14, 16 desde el nodo 1

Insertando Nodos

- nodo r desea unirse
- contacta algún nodo y pide $\text{succ}(r)$
- r solicita a $\text{succ}(r)$ su predecesor
- r informa a estos dos que quiere entrar y así entra
- el nuevo predecesor de r entrega las claves que correspondan

Congestión

Cuando hay demasiados paquetes hay una degradación del desempeño



Razones para la congestión

- procesadores lentos,
- poco ancho de banda,
- desequilibrio en el sistema,
- llegan paquetes por varias líneas que deben salir por la misma
 - paquetes se encolan
 - si no hay suficiente memoria se descartan
 - la memoria infinita no resuelve el problema!!

Control de flujo vs. de la congestión

- Flujo: Se refiere a líneas punto a punto entre un emisor y un receptor. **Perspectiva:** las máquinas
- Congestión: A toda la red, de manera que la red transmita a su máxima capacidad. **Perspectiva:** la red

Tipos Soluciones

- **Ciclo Abierto:** Buen diseño, una vez implementado no se toca. Ejemplos: Cuando crear nuevos ciclos, que paquetes descartar, etc.
- **Ciclo Cerrado:** Retroalimentación: Monitorear, tomar decisiones y modificar el comportamiento.

CG en circuitos virtuales

- Control de Admisión: Una vez detectada la congestión no admitir nuevos circuitos.
- Control de Ruteo: Aceptar nuevos circuitos pero rutearlos cuidadosamente.
- Control de Calidad: negociar la calidad de la conexión antes, de manera de anticipar las necesidades

CG en datagramas

- Bit de advertencia:
 - Un ruteador marca al paquete con un bit de advertencia.
 - Este bit se copia la paquete de ACK que retorna al emisor
 - El emisor regula su caudal de datos

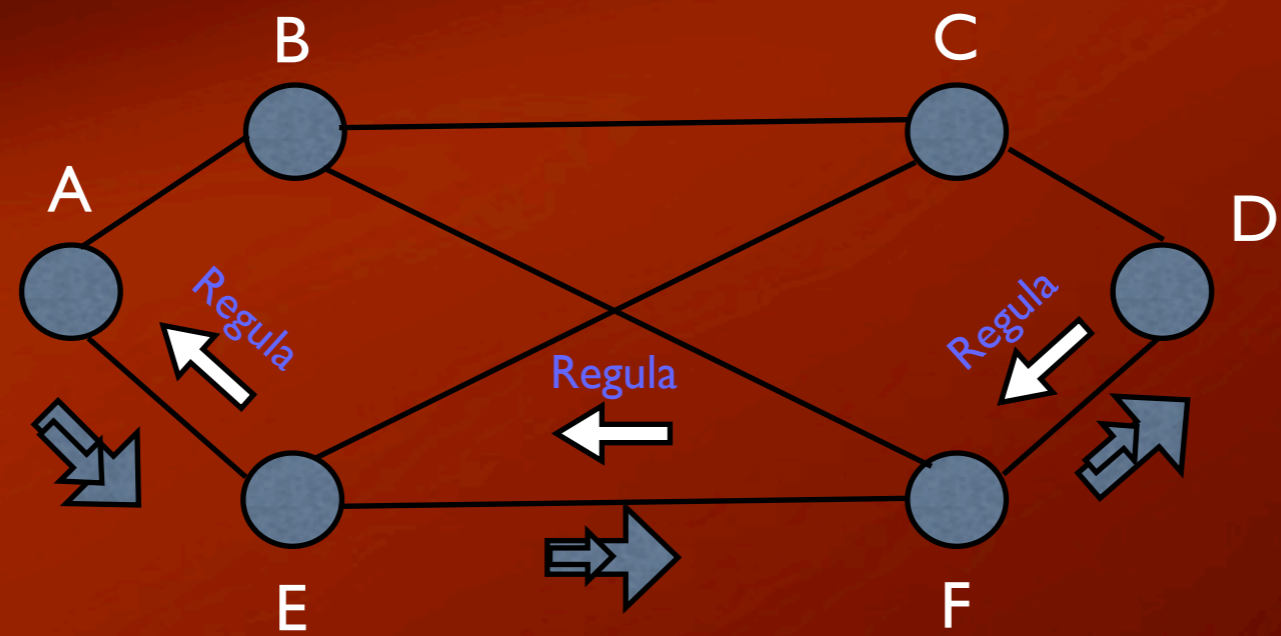
CG Datagramas

- El anterior es un método indirecto: Esto es, no genera paquetes ad hoc.
- Métodos directos con paquetes reguladores.

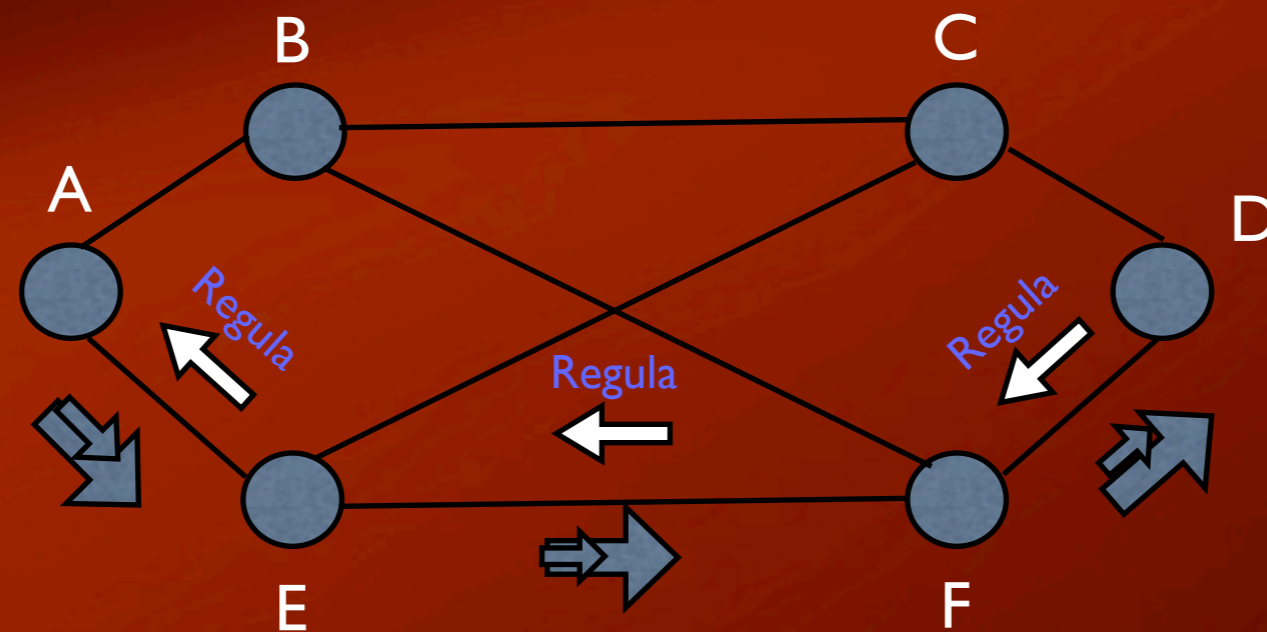
Paquete Regulador

- El enrutador regresa un paquete regulador al destino.
- Los paquetes de tráfico son marcados indicando que ya se genero un regulador
- el origen disminuye su trafico al recibir un regulador
- el origen espera cierto tiempo para volver a transmitir

Ejemplo



Salto a Salto



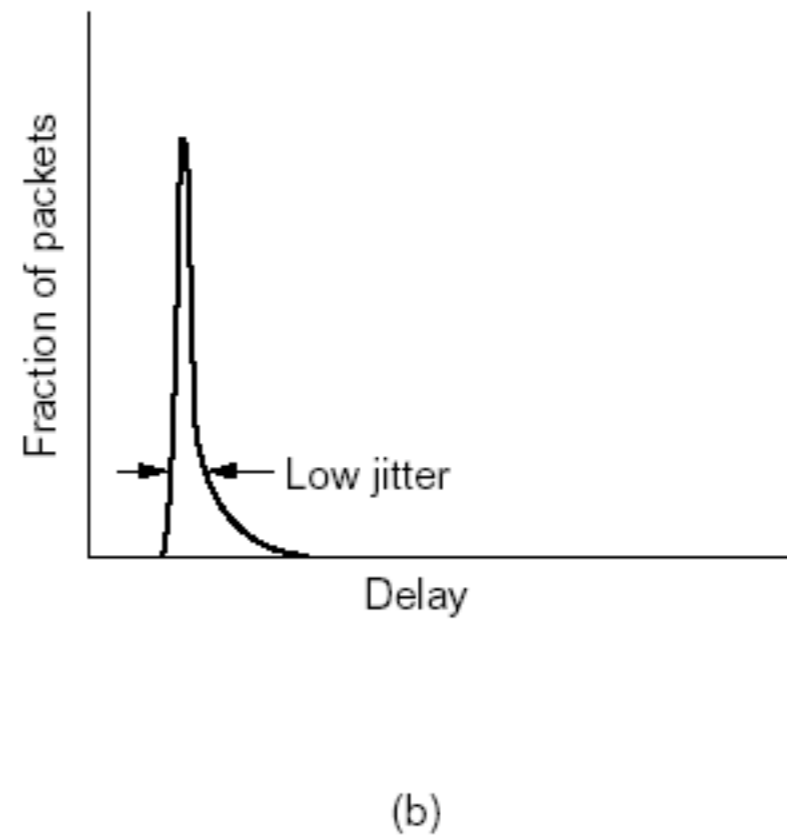
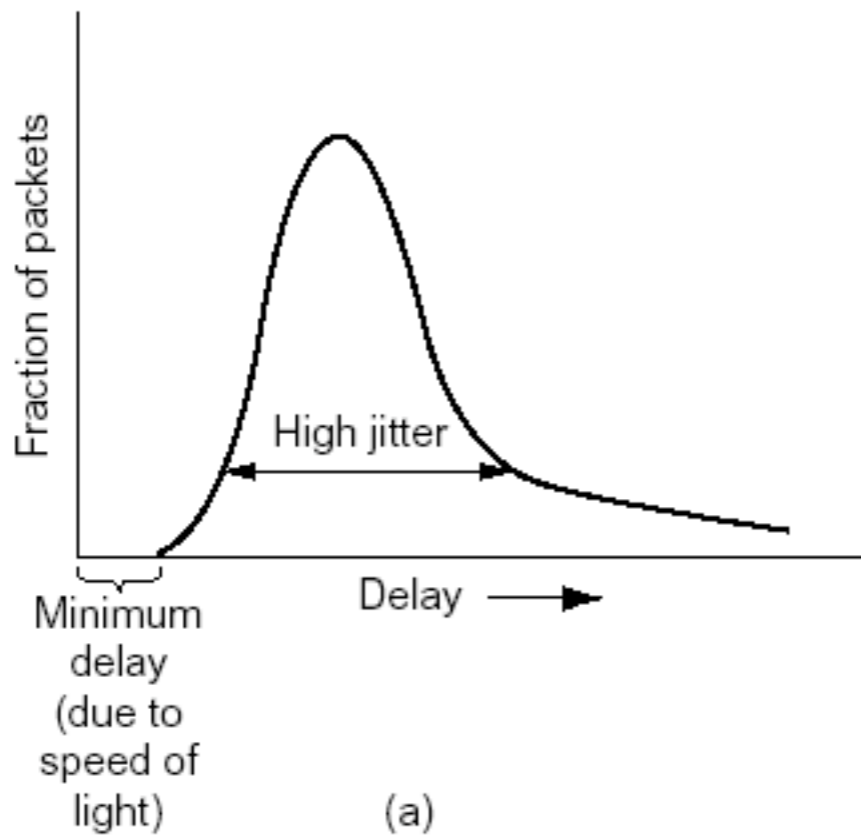
Desprendimiento de Carga

- Cuando el enrutador no puede con todos los paquetes los tira
- como seleccionar los paquetes a tirar?
 - transmision de archivos: los mas nuevos.
 - transmision de video: lo mas viejos

Fluctuación (Jitter)

- Fluctuación es la desviación en el retardo de los paquetes
- Retardo: Es el tiempo de transito de los paquetes.
- Fluctuacion es un concepto global: cuanto nos separamos de la media.

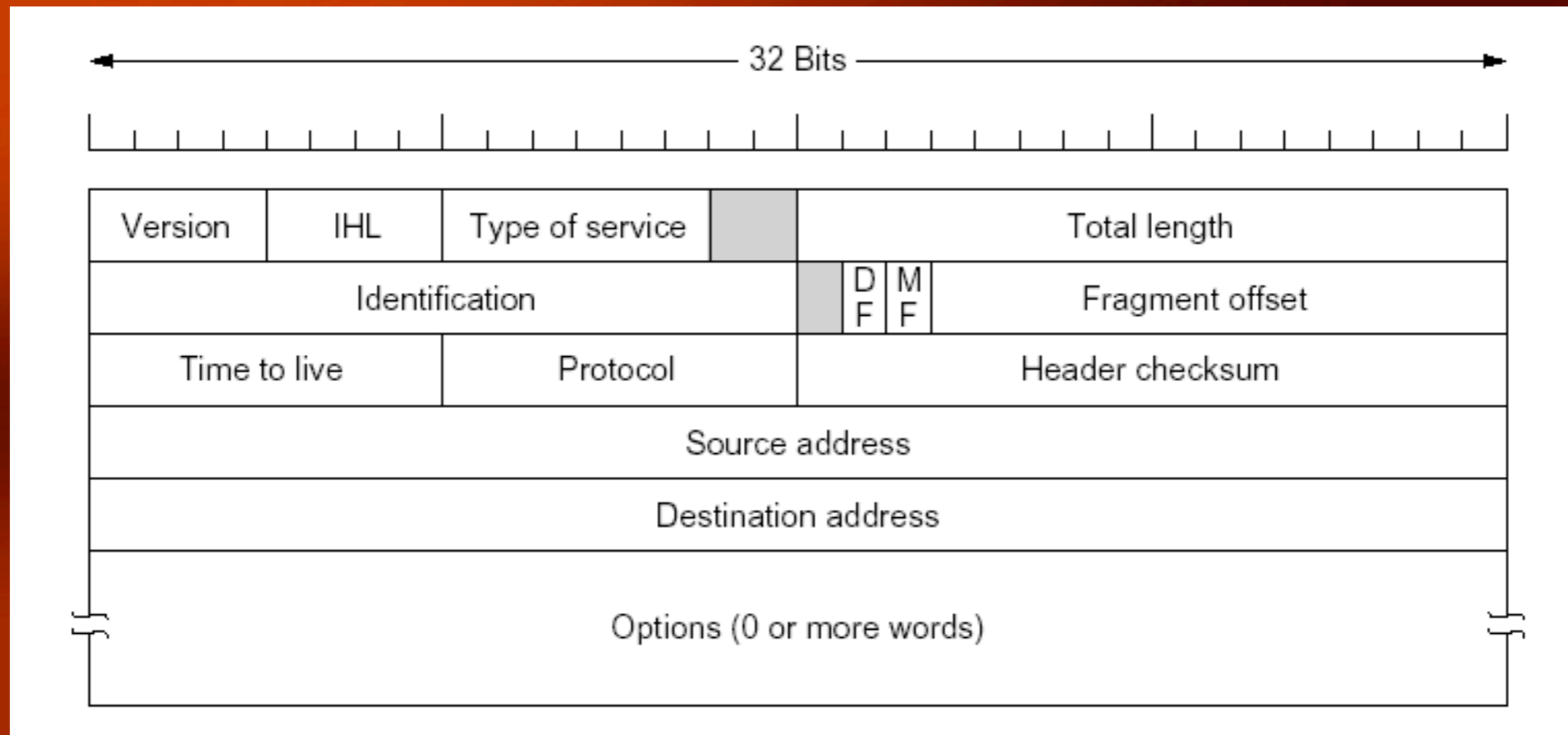
Fluctuación



Calidad de Servicio

Entra y no lo doy, hay preguntas para guiarlos.

Direcciones IP



IHL: Longitud del encabezado

TOS: 3 bit priority mas 3 bit flag

TLen: Longitud del encabezado mas datos

ID: id datagrama

DF: No fragmentar

MF: Hay mas fragmentos

FOff: Offset en datagrama original

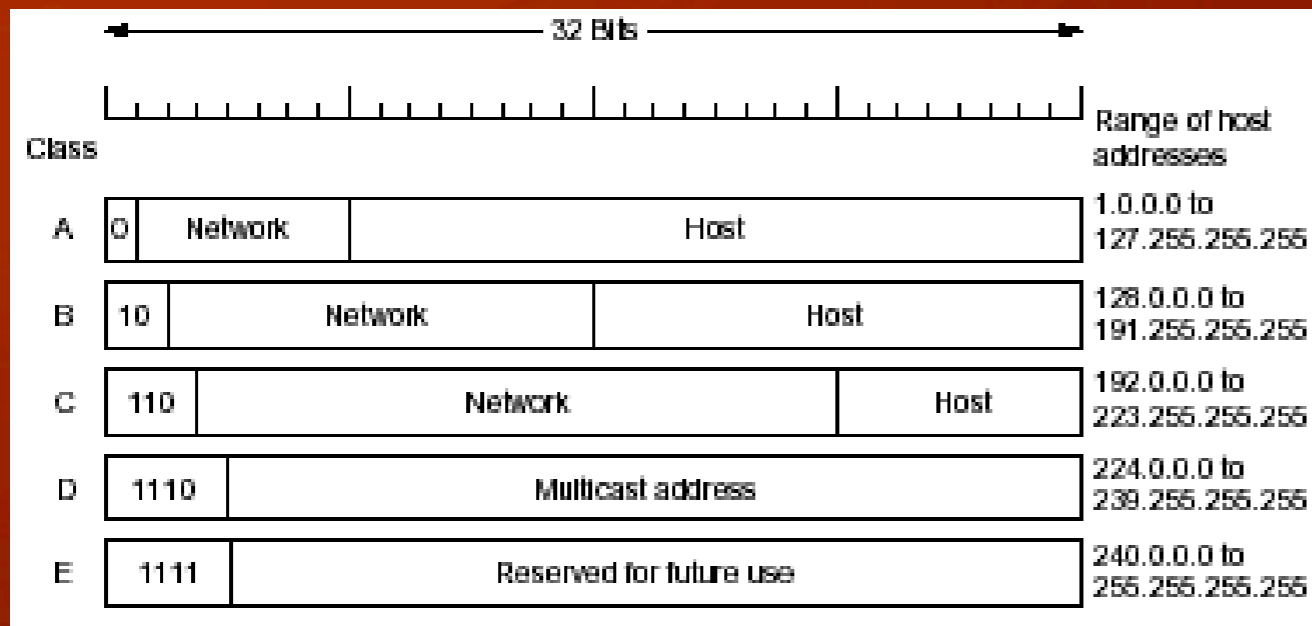
TTL: Maximo numero de hops

Prot: Identificador de protocolo

HCS: Chequea el encabezado

Direcciones IP

- Cada interface tiene una dirección IP única.
- Por varias décadas existieron 5 categorías:



Clase	Cant. Redes	Cant. Hosts
A	126	16,777,214
B	16.382	65,536
C	2.097.150	254

Direcciones Especiales

- 0.0.0.0: usada por hosts al ser arrancados
- 0 para numero de red: refiere a la red actual
- 127.0.0.0: loopbacks, pruebas
- **net.unos**: difusión en la red **net**
- todos unos: difusión local.

Enrutamiento Interdominios sin Clases

Por que hace falta?

- Internet sin números IP :(
- Direcciones desperdiciadas por la estructura jerarquica de redes.
- Los enrutadores saben la dirección de todas las redes: Aumentar la cantidad de redes no es viable. Por que?
- Un nuevo espacio de direcciones?
E.g., pais, provincia, etc?

CIDR: Conceptos Básicos

- Asignar Direcciones IP en bloques de tamaño variable independientemente de las clases.
- Antes: Cada entrada tenía como llegar a cada red. El número de redes era fija.
- Ahora: Cada entrada tiene un IP base más una máscara de red.

Funcionamiento

Supongamos llegue un paquete.

- Obtener el número IP del destino.
- Para cada entrada en la tabla
 - despejar número de red del destino usando la máscara en la entrada (operación AND)
 - si el resultado concuerda con el número base de la entrada, tenemos una ruta!
 - si hay más de una concordancia elegir la que tiene la máscara más grande. Por qué?

Traducción de Dirección de Red

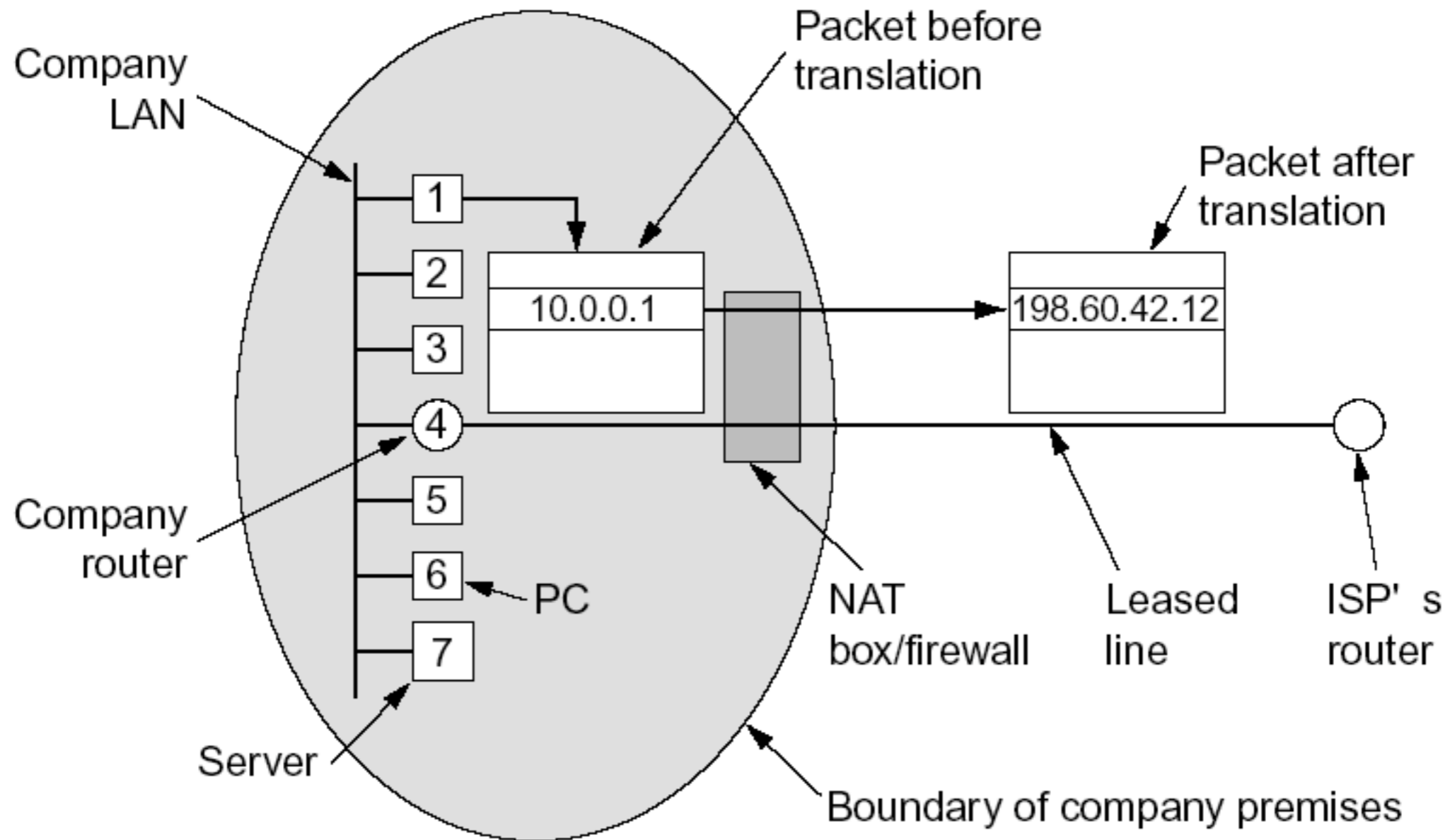
Que pasa cuando un ISP tiene mas clientes que direcciones IP asignadas?

- Direcciones Dinámicas?
- Solución: Traducción de Dirección de red.
- Creación de IP no-reales, o de uso interno:

De	A	Numero Hosts
10.0.0.0	10.255.255.255/8	16.777.216
172.16.0.0	172.31.255.255/12	1.048.576
192.168.0.0	192.168.255.255/16	65.536

Cada puerta de enlace sabe traducir los numeros ip

Ejemplo



Protocolos de Control (ICMP)

Idea Básica: Queremos informar hosts y enrutadores cuando algo no falló, o pedir información del estado.

1. Distinguir tipos de mensajes
2. Cada tipo de mensaje tiene un subtipo, e.g., un destino puede ser inalcanzable (codigo 3) por que el protocolo no esta soportado (código 3)
3. Un paquete ICMP contiene el paquete IP que produce al paquete.

Ejemplos

- Destination Unreachable: Cuando la subred o el enrutador no encuentra el destino, o cuando un paquete DF encuentra una “red de paquete pequeño”
- Time Exceeded: Cuando un paquete se cae porque el contador llega a cero.
- Parameter Problem: Se ha descubierto un valor ilegal en un campo de encabezado.
- Redirect: Cuando el enrutador se da cuenta que el paquete parece estar mal enrutado.
- Echo y echo reply: para testear si nun destino es inalcanzable. Se espera que una máquina envíe un echo reply cuando recibe un reply.

ARP

Como se transforma una dirección IP en una ethernet?

- 1.- **Enrutador:** Pregunta a cada host en la red quien tiene el IP en cuestión. Esto se hace encapsulando la pregunta como un paquete ARP en frame de la capa de datos y “broadcasting”
- 2.- **Host:** Detecta que el paquete es un paquete ARP. Si la dirección buscada es la suya, responde devolviendo su dirección ethernet.
- 3.- **Enrutador:** El enrutador reconoce la respuesta y cachea la respuesta.

RARP, BOOTP, DHCP

Esto no lo doy pero entra, hay preguntas para guiarlos.

Enrutamiento

Idea Basica: Hace la distinción entre enrutamiento **entre** sistemas autonomos y **en** sistemas autonomos

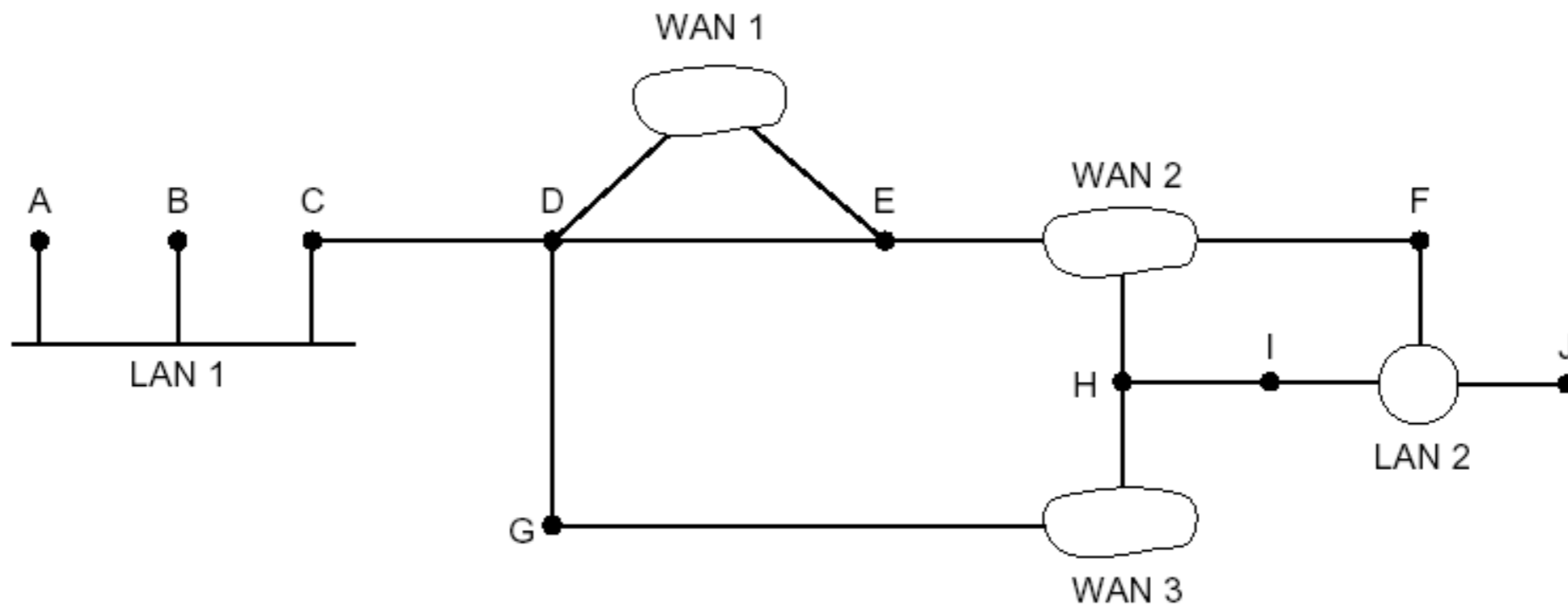
Intra SA se ocupa de hacer llegar los paquetes al destino lo mas rápido posible (OSPF)

Inter SA se ocupa de asuntos politicos. Por ejemplo: Algunos SA no deberian ser atravezados: los del eje del mal (BGP)

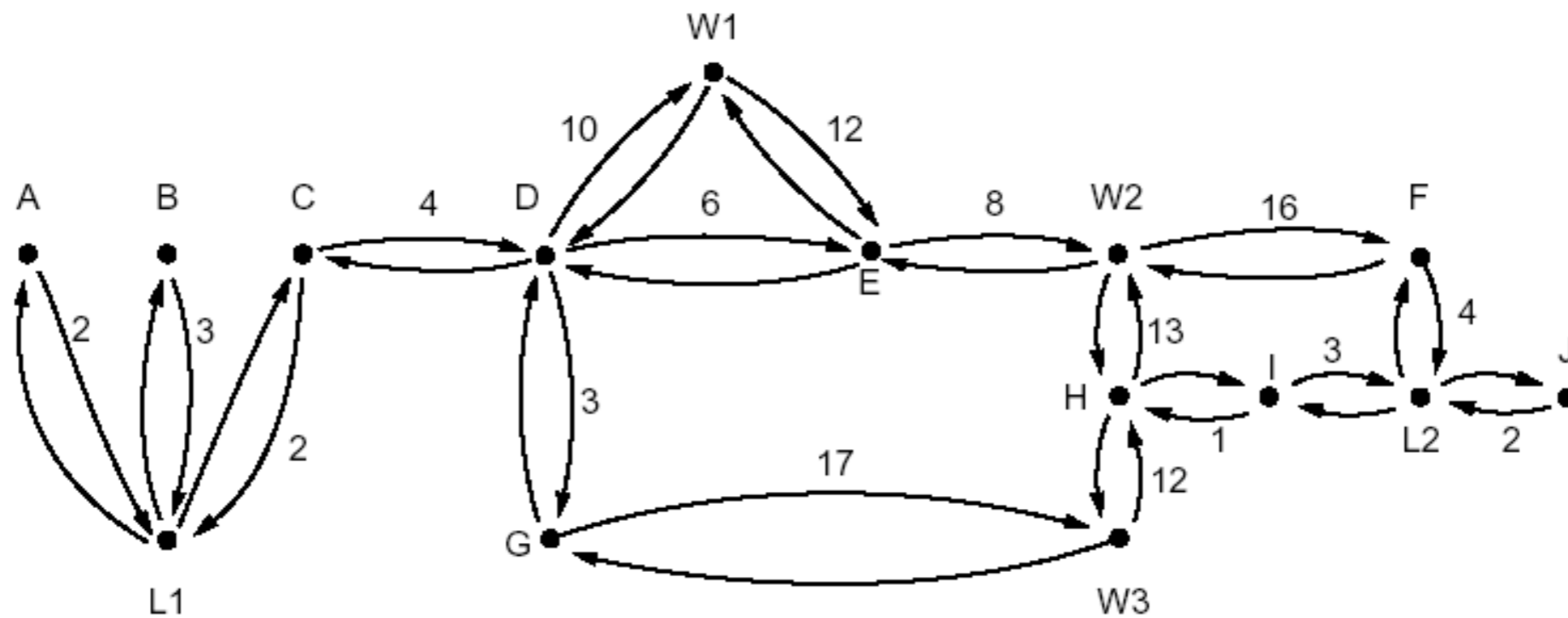
Enrutamiento de puerta de enlace interior (OSPF)

- En funcionamiento desde 1990 aprox. se siguieron las siguientes maximas para diseñarlo.
 - especificación abierto,
 - soporte para varias métricas,
 - algoritmo dinámico,
 - soporte para el tipo de servicio,
 - balancear la carga
 - no conocer toda la topologia,
 - seguridad, evitar recibir información maliciosa,
 - soporte para “tunneling”

Idea Principal: Construir una representación de la red



(a)

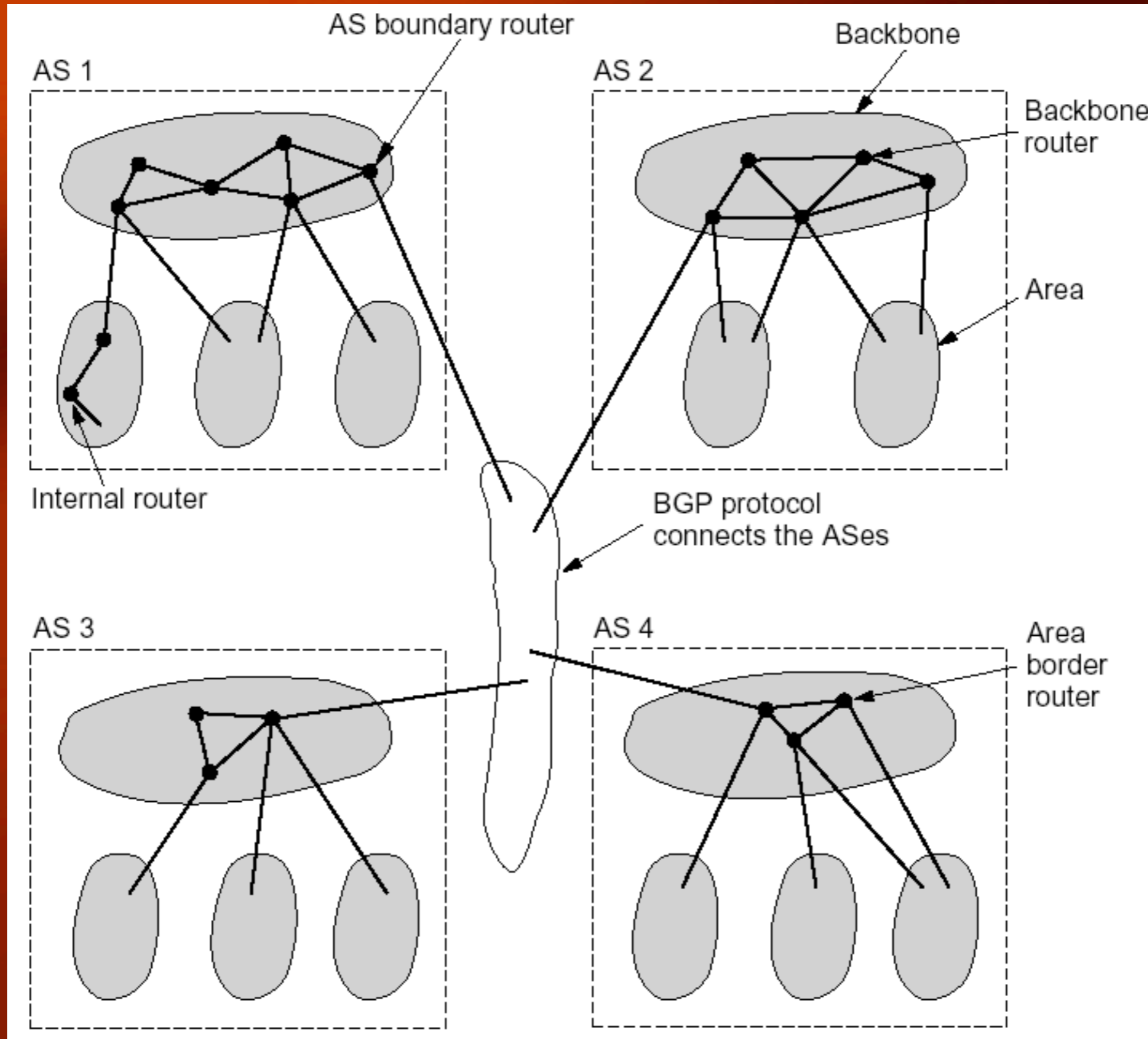


(b)

Construyendo el grafo.

- Definir areas de enrutadores, un enrutador puede no pertenecer a ningun area y ningun enrutador pertenece a dos areas.
- Fuera de un area sus detalles topologicos son desconocidos.
- Cada Sistema Autonomo tiene una conexion al “backbone”.
- Todas las areas se conectan al “backbone”
- Cada enrutador que se conecta a dos o mas areas es parte del “backbone”,
- El “backbone” es un area en si misma (llamada Area 0),
- Como el “backbone” es un area, su topologia es desconocida.

Ejemplo



Tipos de Conexión

OSPF maneja los siguientes tipos de conexión

- punto a punto (una conexión serial)
- multiaccess (Ethernet)
- non-broadcast multiaccess (Frame Relay)

En redes multiaccess se designan routers

El tráfico ocurre entre routers adyacentes solamente

Paquetes de OSPF

Tipo de mensaje	Descripción
Hello	Descubre quienes son los vecinos
Link State Update	Proporciona los costos del emisor a sus vecinos
Link State ACK	Confirma la recepción de la actualización del enlace
Database Description	Anuncia que actualizaciones tiene el emisor
Link State request	Solicita información del socio

Protocolo de Puerta de Enlace de Frontera (BGP)

Enrutamiento entre sistemas autonomos. Razones politicas y NO tecnicas:

- Ningún transito a través de ciertos autonomos.
- Nunca ponga Irak en una ruta que inicie el Pentagono
- No pasar por EEUU para llegar de la Columbia Britanica a Ontario
- El tráfico que pasa por IBM no debe pasar por MS

Las rutas se especifican a MANO

IPv6

Principales Características

- Direcciones mas largas: es virtualmente imposible quedarse sin números IP
- Encabezado simple: permite el manejo eficiente de paquetes
- Apoyo a opciones: Ahora algunos campos son opcionales
- Seguridad: mejor manejo de autenticidad.

IPv6 es incompatible con IP