

Algoritmo di Dijkstra's: dato un grafo, dati i nodi e rami trovare il cammino ottimo.

1) Initialization:

2)  $N' = \{a\}$

3) for all nodes  $x$

4) if  $x$  adjacent to  $a$

5) then  $D(x) = c(a, x)$  and  $p(x) = a$

6) else  $D(x) = \infty$

7) Loop

8) find  $y$  not in  $N'$  such that  $D(y)$  is a minimum

9) Add  $y$  to  $N'$

10) Update  $D(z)$  for all  $z$  adjacent to  $y$  and not in  $N'$ :

11)  $D(z) = \min \{D(z), D(y) + c(y, z)\}$

12) if  $D(y) + c(y, z) < D(z)$  then  $p(z) = y$

13) /\* new cost to  $z$  is either old cost to  $z$  or know

14) shortest path cost to  $y$  plus cost from  $y$  to  $z^*$ /

15) Until all nodes in  $N'$ .

Esempio:

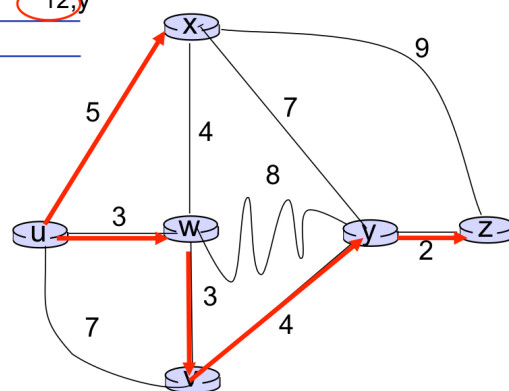
tea path: cammino  
da u scegliamo  
quella che costa meno

Step	$N'$	$D(v)$ $p(v)$	$D(w)$ $p(w)$	$D(x)$ $p(x)$	$D(y)$ $p(y)$	$D(z)$ $p(z)$
0	$u$	7, $u$	3, $u$	5, $u$	$\infty$	$\infty$
1	$uw$	6, $w$	5, $u$	11, $w$	$\infty$	$\infty$
2	$uwx$	6, $w$		11, $w$	14, $x$	
3	$uwxv$			10, $v$	14, $x$	
4	$uwxvy$				12, $y$	
5	$uwxvyz$					

no quando non e' direttamente connesso  
non conviene passare per w che ha peso 7

notes:

- construct shortest path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)



La difficoltà dell'algoritmo è che aumentando sempre di più il numero di nodi  $n$  diventa sempre più difficile utilizzarlo.

Inoltre i costi possono rappresentare la lunghezza dei rami o quantità dinamiche che variano nel tempo come l'utilizzazione della capacità o tempo di percorrenza, carico in

termini di traffico. Quando si ha a che fare con una grandezza tempo variante e un problema di ottimizzazione la usa come input per trovare l'instradamento ottimo, deve fare attenzione a non creare situazioni di instabilità.

Il link state si chiama globale perché è in grado di comunicare con tutti i nodi.

## Local distance-vectors

Si scambiano i messaggi solo tra vicini e poi il tutto procede attraverso il passaparola e si inviano i messaggi più volte secondo un processo iterativo.

Ogni nodo  $x$  si mantiene in memoria un vettore  $D_x$  che rappresenta le distanze dei nodi da  $x$  ai suoi vicini (i costi). Il vettore manterrà le distanze ottime da  $x$  a qualunque altro nodo.

Inoltre  $x$  mantiene un vettore  $D_v$  ossia il vettore delle distanze dei suoi vicini ossia il vettore dei costi per raggiungere ogni altro nodo.

## Bellman-Ford algorithm

Di volta in volta, ogni nodo manda il suo  $D_v$  ai vicini (può venire anche in modo asincrono).

Quando un nodo  $x$  riceve un nuovo  $D_v$  da un vicino, aggiorna il suo  $D_v$  usando l'equazione di B-F:

$$d_x(y) = \min_v \{c(x, v) + d_v(y)\}$$

Per ogni nodo nella rete.

Per poter fare questo calcolo il nodo  $x$  ha bisogno dei costi da  $x$  a ogni vicino, dei vettori distanze dei propri vicini (che gli li trasmettono loro).

## Comparison of LS and DV algorithms

### message complexity

• **LS:** with  $n$  nodes,  $E$  links,  $O(nE)$  msgs sent

• **DV:** exchange between neighbors only  
• convergence time varies

### speed of convergence

• **LS:**  $O(n^2)$  algorithm requires  $O(nE)$  msgs

• **DV:** convergence time varies  
• may be routing loops  
• count-to-infinity problem

**robustness:** what happens if router malfunctions?

**LS:**

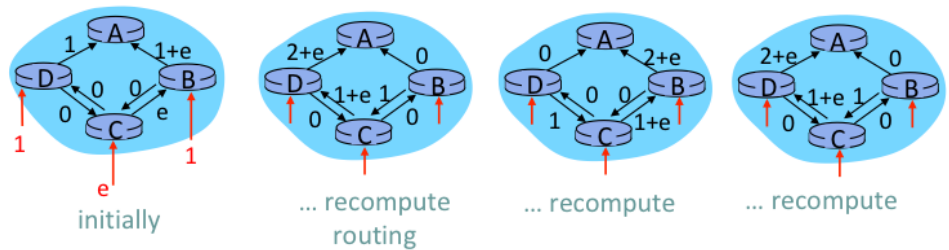
- node can advertise incorrect link cost
- each node computes only its own table

**DV:**

- DV node can advertise incorrect path cost
- each node's table used by others  
• error propagate thru network

L'approccio locale porta una minore complessità nel trasmettere i messaggi.

Nel DV c'è il problema di incompletezza delle informazioni. Per questo viene mandato sia il vettore distanze che il vettore next hop.



## Il routing di internet: OSPF

OSPF è il protocollo più usato attualmente per fare instradamento. Segue la filosofia globale ossia ogni nodo manda a tutti i nodi, ma internet è molto grande quindi si creerebbe problema di traffico. Quindi si introduce con OSPF il concetto di sistema autonomo. Si divide la rete in sottoinsiemi (non le sottoreti con ognuna uno specifico net id) in modo tale da poter applicare la filosofia globale. Un sistema autonomo è un insieme di sottoreti.

All'interno del sistema autonomo ci sono gli algoritmi intra-AS (ognuno ha un proprio algoritmo di instradamento), e l'algoritmo Inter-AS ossia per comunicare con altri sistemi autonomi.

aggregate routers into regions known as “autonomous systems” (AS) (a.k.a. “domains”)

### intra-AS routing

- routing among hosts, routers in same AS (“network”)
- all routers in AS must run *same* intra-domain protocol
- routers in *different* AS can run *different* intra-domain routing protocol
- gateway router: at “edge” of its own AS, has link(s) to router(s) in other AS'es

### inter-AS routing

- routing among AS'es
- gateways perform inter-domain routing (as well as intra-domain routing)

All'interno del sistema autonomo si hanno tante informazioni, dei sistemi autonomi al di fuori si sa poco quindi si snellisce lo scambio di informazione necessario riducendo la complessità e il pericolo di traffico.

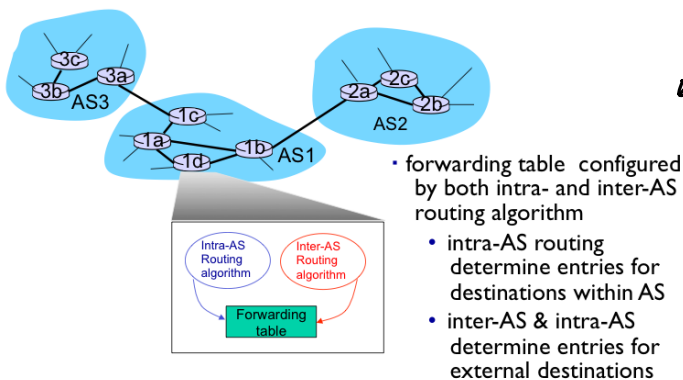
Ogni sistema autonomo trova il cammino ottimo minimo personale, quindi un subottimo.

Quindi il prezzo che si paga è che le tabelle di inoltra non sono più caratterizzati dai cammini ottimi globali (in quanto rinunciamo a scambiare le informazioni per trovarlo) ma da una collezione di cammini che sono a costo minimo di ogni sistema autonomo ma la somma di questi costi minimi non è il cammino ottimo globale.

Distinguiamo protocolli di instradamento:

- intra instradamento AS: ossia il cammino ottimo preciso per il sistema autonomo
- Inter instradamento AS: il cammino ottimo tra sistemi autonomi.

Border gateway protocol è come è conosciuto intra-AS routing.



in ogni sistema autonomo si distinguono i router di confine.

I router di confine si fanno carico di annunciare all'esterno l'insieme di sottoreti che appartengono al sistema autonomo di cui fa parte.

Intra-AS Routing è conosciuto anche come interior gateway protocols (IGP), i più comuni intra-AS routing protocols sono:

- RIP routing Information Protocol
- OSPF
- IGRP

## OSPF

È "aperto"

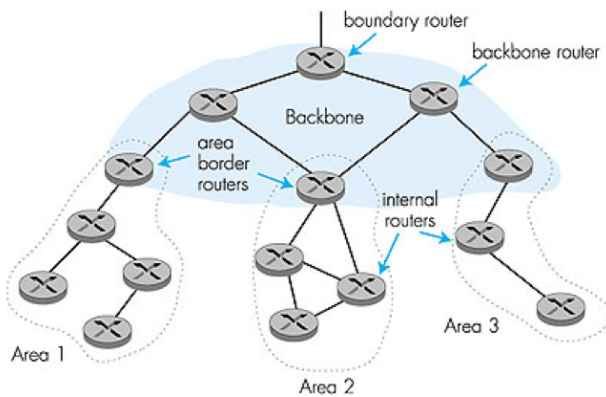
Consta di due fasi:

- flooding (allegamento): quella in cui ogni router scrive ad un altro router. (È un algoritmo tale che ogni nodo rilancia i suoi messaggi a ogni nodo, ma deve fare in modo che non inoltri troppe volte gli stessi messaggi).
- Usa link-state algorithm

## OSPF caratteristiche avanzate

- Sicurezza: tutti i messaggi di OSPF sono autenticati, si instaura una fiducia tra i router (per evitare intrusioni "malefiche", ossia chi si spaccia per router o per forzare a fare un inoltra errato).
- Sono permessi cammini multipli con lo stesso costo (mentre in RIP solo uno)

## La gerarchia dell'OSPF



OSPF stesso all'interno fornisce metodi per suddividere gli stessi sistemi autonomi in diversi gruppi, i router di confine comunicano tra gruppi.

Due livelli di gerarchia: local area, backbone.

- **two-level hierarchy:** local area, backbone.
  - Link-state advertisements only in area
  - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.
- **area border routers:** "summarize" distances to nets in own area, advertise to other Area Border routers.
- **backbone routers:** run OSPF routing limited to backbone.
- **boundary routers:** connect to other AS' s.

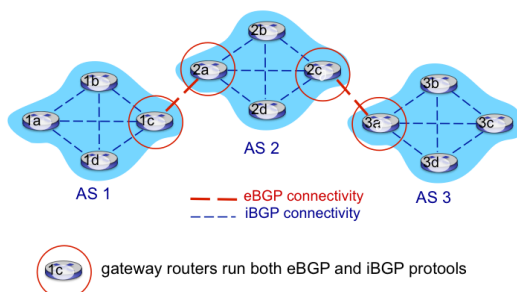
## Internet Inter-AS routing: BGP

Tra sistemi autonomi il principio che si vuole utilizzare è diffondere le informazioni di connettività.

BGP si trova al livello applicativo, quindi al di sotto c'è il livello di trasporto. I messaggi tra BGP a ogni AS sono:

- eBGP: messaggi scambiati tra router di AS diversi
- iBGP: messaggi scambiati tra router di stesso AS

I router di confine ricevono i messaggi e poi lo comunicano ai router della sottorete.



Abbiamo 3 sistemi autonomi (1,2,3).

Supponiamo che il sistema autonomo 3 aggiunga una sottorete (un router) e deve annunciarlo a tutti.

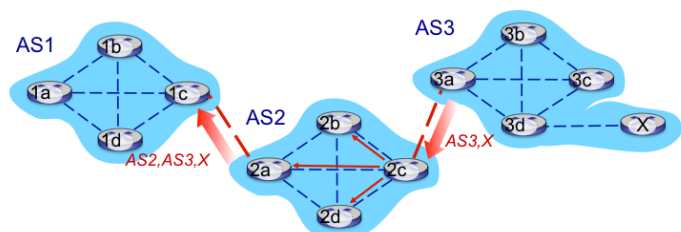
Chiamando x il nuovo router, manda un annuncio attraverso il router di confine 1A. Questa informazione viene diffusa dal router 2C a tutti i router del sistema autonomo 2A, quest'ultimo la diffonde verso il sistema autonomo 1.

Tre degli elementi fondamentali di BGP sono:

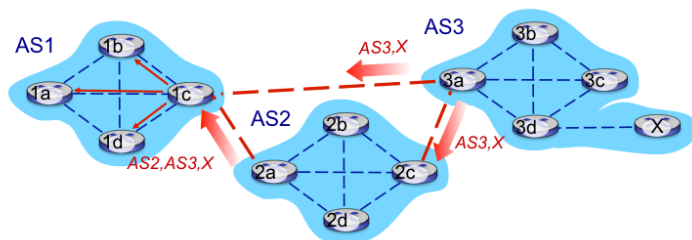
- prefisso
- AS-PATH
- NEXT-HOP

Ossia in ogni annuncio si dice: questa sottorete (e si dà il prefisso) la si può raggiungere attraverso una sequenza ordinata di sistemi autonomi (AS-PATH) e il NEXT hop per raggiungere la prossima sottorete è x.

Ad esempio:



- AS2 router 2c receives path advertisement **AS3,X** (via eBGP) from AS3 router 3a
- Based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers
- Based on AS2 policy, AS2 router 2a advertises (via eBGP) path **AS2,AS3,X** to AS1 router 1c



gateway router may learn about **multiple** paths to destination:

- AS1 gateway router 1c learns path **AS2,AS3,X** from 2a
- AS1 gateway router 1c learns path **AS3,X** from 3a
- Based on policy, AS1 gateway router 1c chooses path **AS3,X**, and **adverts path within AS1 via iBGP**

la scelta dei percorsi alternativi dipende dalla politica

BGP non cerca il cammino ottimo ma si basa solo su politiche: ha come obiettivo garantire la connettività globale.

## BGP messaggi

- I messaggi BGP scambiati tra pari avviene attraverso la connessione TCP

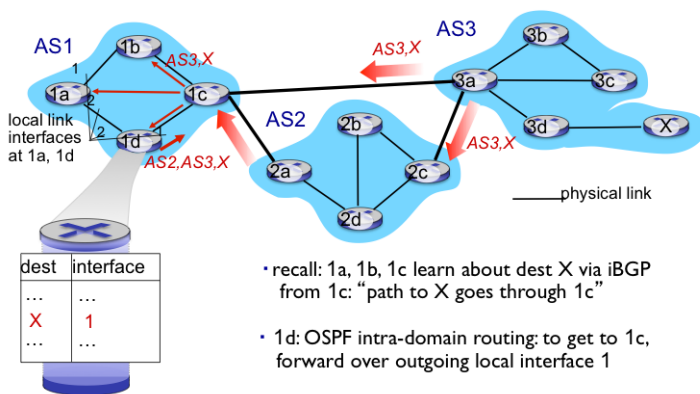
I messaggi BGP:

- OPEN: apre la connessione TCP con BGP
- UPDATE: advertises new path, annunciano nuovi prefissi o cambiamenti su prefissi esistenti.
- KEEPALIVE: mantiene la connessione attiva anche quando non viene utilizzata.
- NOTIFICATION : annuncia errori

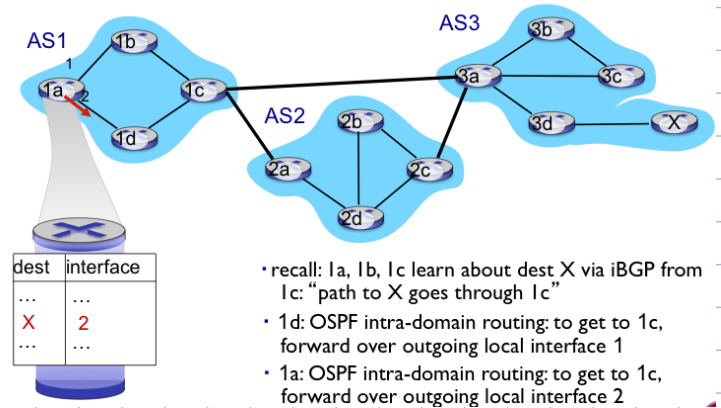


## Esempio:

Q: how does router set forwarding table entry to distant prefix?



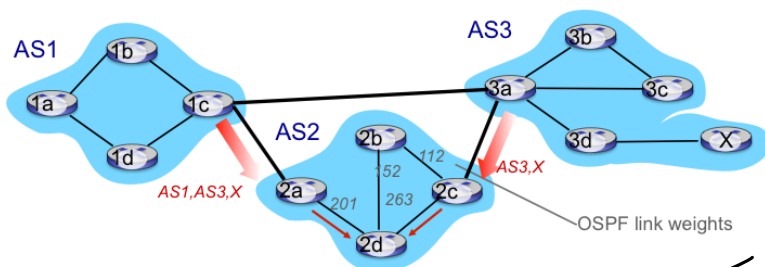
Q: how does router set forwarding table entry to distant prefix?



La selezione di router è basata su:

- valori di preferenze locali data dalla politica di decisione
- AS-PATH più corto
- NEXTHOP router più vicino : hot potato routing
- Criteri aggiuntivi

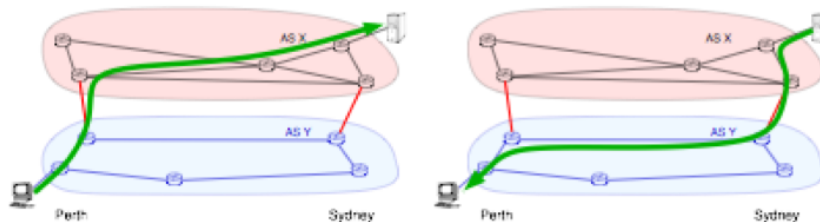
## Hot potato routing



- 2d learns (via iBGP) it can route to X via 2a or 2c
- **hot potato routing**: choose local gateway that has least intra-domain cost (e.g., 2d chooses 2a, even though more AS hops to X): don't worry about inter-domain cost!

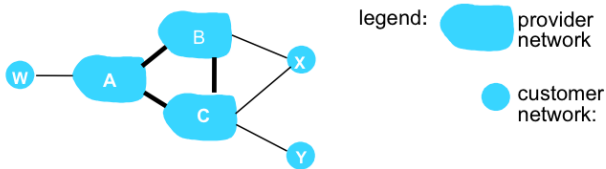
2d egoisticamente sceglie il cammino con costo più basso (2a)

si sceglie quella che carica il meno possibile l'operatore celeste (orina che si passa per meno tempo possibile)



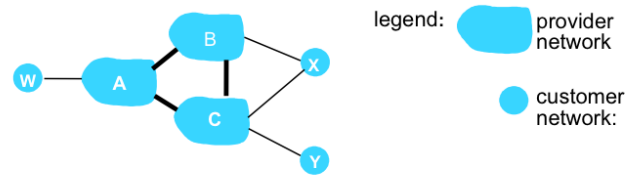
## Hot potato routing:

- each ISP aims at minimizing the path of the outbound traffic within his own AS

BGP: achieving policy via advertisements

Suppose an ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs)

- A advertises path Aw to B and to C
- B *chooses not to advertise* BAw to C:
  - B gets no "revenue" for routing CBAw, since none of C, A, w are B's customers
  - C does not learn about CBAw path
- C will route CAw (not using B) to get to w

BGP: achieving policy via advertisements

Suppose an ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs)

- A, B, C are *provider networks*
- X, W, Y are customer (of provider networks)
- X is *dual-homed*: attached to two networks
- *policy to enforce*: X does not want to route from B to C via X
  - ... so X will not advertise to B a route to C

Why different Intra-, Inter-AS routing ?*policy:*

- inter-AS: admin wants control over how its traffic routed, who routes through its net.
- intra-AS: single admin, so no policy decisions needed

*scale:*

- hierarchical routing saves table size, reduced update traffic

*performance:*

- intra-AS: can focus on performance
- inter-AS: policy may dominate over performance