



Algoritmi di routing

LIVELLO NETWORK



Connection Less - Connection Oriented

Connection Less: algoritmo applicato ex novo ad ogni pacchetto

Connection Oriented: algoritmo applicato soltanto nella fase di setup (**session routing**)

Come deve essere un algoritmo di routing?

1. Corretto
2. Semplice
3. Robusto
4. Stabile
5. Equo
6. Ottimale

Classi

Due classi di algoritmi di routing:

1. **NON adattivi** (*statici*): decisioni di routing prese in anticipo, all'avvio della rete, e comunicate ai router che si attengono ad esse.
2. **Adattivi** (*dinamici*): decisioni di routing riformulate molto spesso.

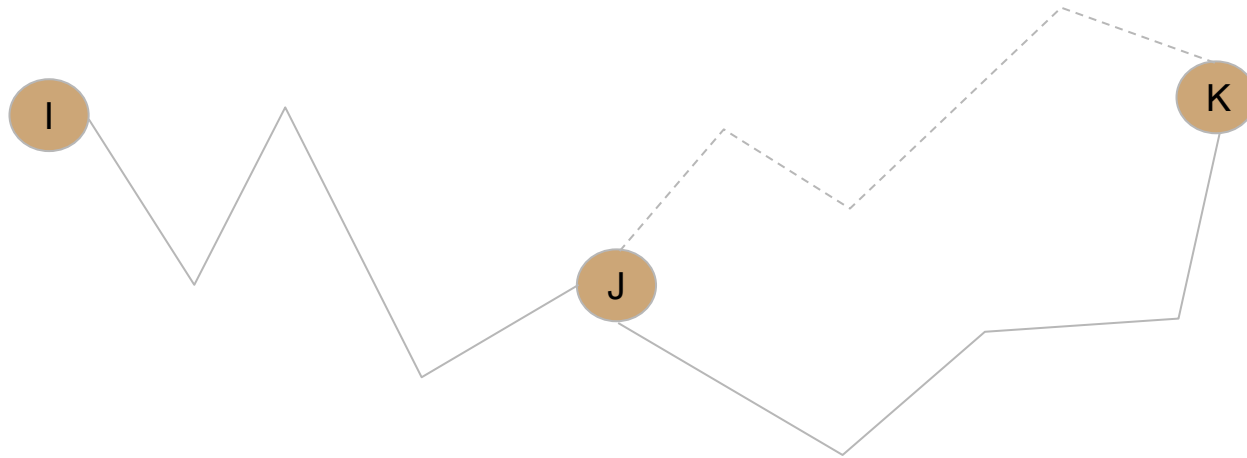
Algoritmi adattivi - scelte

Differiscono tra loro per i seguenti motivi:

1. *Come ricevono le informazioni:*
 - a. localmente
 - b. dai router adiacenti
 - c. da tutta la rete
2. *Quanto spesso rivedono le decisioni:*
 - a. a intervalli prefissati
 - b. quando il carico cambia
 - c. quando la topologia cambia
3. *Quale metrica di valutazione adottano:*
 - a. Distanza
 - b. Numero di hop
 - c. Tempo di transito stimato

Principio di ottimalità

< Se il router J è nel cammino ottimo fra I e K, allora anche il cammino ottimo fra J e K è sulla stessa strada. >



Sink Tree

Il **SINK TREE** per un router è l'**albero** costituito dall'insieme dei *cammini ottimi* da tutti i router a quel router di destinazione.

Gli algoritmi di routing cercano i *sink tree* relativi a tutti i possibili router di destinazione e instradano i pacchetti lungo tali *sink tree*.

Algoritmi statici

Eseguiti soltanto all'avvio della rete, le decisioni di routing relative sono applicate senza essere più modificate.

Di seguito vedremo:

1. Shortest path routing
2. Flooding
3. Flow-based routing

Shortest path routing

Un host di gestione della rete mantiene un grafo che rappresenta la subnet:

- I nodi rappresentano i router
- Gli archi rappresentano le linee punto a punto

All'avvio della rete (o a variazioni permanenti della topologia):

1. Applica al grafo un algoritmo per il calcolo del cammino minimo fra ogni coppia di nodi (per es. l'algoritmo di Dijkstra)
2. Invia tali informazioni a tutti i router

Flooding

Il flooding consiste nell'inviare ogni pacchetto su tutte le linee eccetto quella da cui è arrivato.

- Inconveniente: genera un numero enorme di pacchetti.
- Soluzioni:
 - **Contatore** inserito in ogni pacchetto
 - Coppia (**source router ID, sequence number**) in ogni pacchetto
 - Selective flooding: duplicare i pacchetti solo sulle linee che vanno all'incirca nella giusta direzione.

Flooding

Generalmente non è utilizzato come algoritmo di routing.

Tuttavia è utile:

- In campo militare
- Per l'aggiornamento contemporaneo di informazioni distribuite
- Come strumento di paragone per altri algoritmi

Flow-based routing

Basato sull'idea di traffico.

- Calcola in anticipo il traffico atteso su ogni linea
- Deriva una stima del ritardo medio atteso per ciascuna linea
- Elabora le decisioni di routing sui dati di cui sopra

Algoritmi dinamici

Non eseguiti solo all'avvio della rete: rimangono in esecuzione sui router.

Vediamo:

- Distance Vector routing
- Link State routing

Distance Vector routing

Ogni router ha una tabella contenente un elemento per ogni altro router.

Ogni elemento della tabella contiene:

- La distanza dal router in oggetto
- La linea in uscita da usare per arrivarci

Per i vicini immediati il router stima direttamente la distanza inviando speciali pacchetti *echo*.

Ogni router manda la sua tabella a tutti i vicini e riceve quella dei vicini.

Distance Vector routing

Problema: lento a reagire alle cattive notizie: **count to infinity**

Utilizzato in Internet col nome di RIP (routing internet protocol)

Link State Routing

- Ogni router tiene sott'occhio lo stato dei collegamenti tra sé e i suoi vicini immediati e distribuisce tali informazioni a tutti gli altri.
- Sulla base di tali informazioni ogni router costruisce localmente la topologia completa della rete e calcola il cammino minimo fra sé e tutti gli altri.

Link State Routing

Passi da seguire:

1. Scoprire chi sono i vicini
2. Misurare il costo delle relative linee
3. Costruire un pacchetto con tali informazioni
4. Mandare a tutti il pacchetto
5. Costruire la topologia dopo aver ricevuto i pacchetti dagli altri router
6. Calcolare il cammino minimo fra sé e gli altri router