

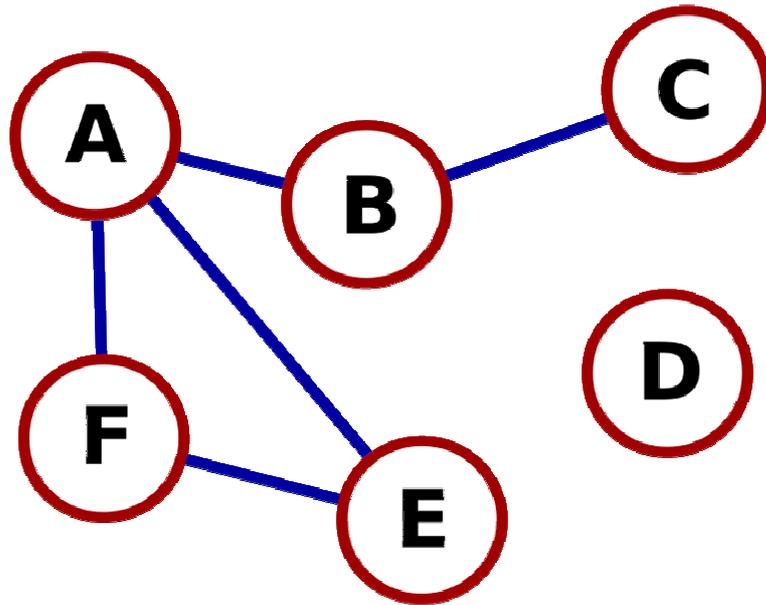
DAL LIVELLO DI LINEA AL LIVELLO DI RETE: IL PROBLEMA DELL'INSTRADAMENTO NEL LIVELLO DI RETE	1
Principi generali	1
Instradamento globale e instradamento centralizzato	3
Instradamento decentralizzato o distribuito	3
Instradamento statico o dinamico	3
Link state algorithm. L'algoritmo di Dijkstra	3
Algoritmo di instradamento distribuito: distance vector	16
L'instradamento isolato	19
Instradamento gerarchico	20
La congestione	21

Dal livello di linea al livello di rete: il problema dell'instradamento nel livello di rete

Principi generali

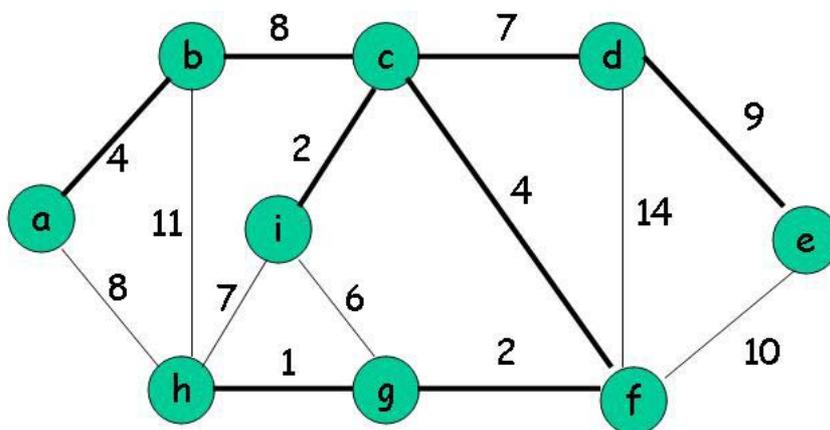
Il compito principale del livello di rete è la determinazione del miglior percorso (path o rotta (route) che i pacchetti di dati devono seguire per giungere a destinazione. La risoluzione di questa problematica è affidata a protocolli di instradamento o routine protocol. Il nucleo di un protocollo di instradamento è un algoritmo di instradamento cioè un metodo per la determinazione di un percorso dal costo minimo in termini ad esempio di ritardo di consegna, di distanza fisica delle tratte, di livelli di congestione dei tratti di rete coinvolti. La definizione di costo dipende naturalmente dalle politiche generali seguite dalla rete in questione.

Una rete di computer viene schematizzata mediante lo strumento matematico del grafo: un grafo è un oggetto costituito da nodi o vertici



I nodi rappresentano i router che costituiscono la intelaiatura portante della rete mentre gli archi di congiunzione fra i nodi rappresentano le varie connessioni esistenti fra i singoli router.

Se associamo dei valori numerici ad ogni arco si parla di grafo pesato.



In riferimento ad una rete di computer i pesi associati ai vari archi rappresentano la "distanza" o "costo" fra i due nodi collegati dall'arco.

Dal punto di vista del problema dell'instradamento allora si tratta di individuare nel grafo un percorso fra due nodi in modo tale che la somma dei pesi associati agli archi facenti parte del percorso sia minima.

Instradamento globale e instradamento centralizzato

Si parla di instradamento globale quando il calcolo del percorso migliore avviene essendo a conoscenza dello stato globale dell'intera rete cioè conoscendo tutti i collegamenti esistenti ed i loro costi. Si parla allora di global routine algorithm. Se l'algoritmo viene eseguito su di un solo nodo si parla di instradamento centralizzato. Si parla anche di algoritmi basati sullo stato dei link (link state algorithm).

Instradamento decentralizzato o distribuito

In questo caso il calcolo del percorso avviene in modo distribuito ed interattivo. Nessun router ha informazioni sullo stato globale della rete ma è a conoscenza inizialmente soltanto del costo dei link direttamente collegati ad esso e calcola gradualmente il percorso mediante uno scambio continuo di informazioni con i nodi vicini.

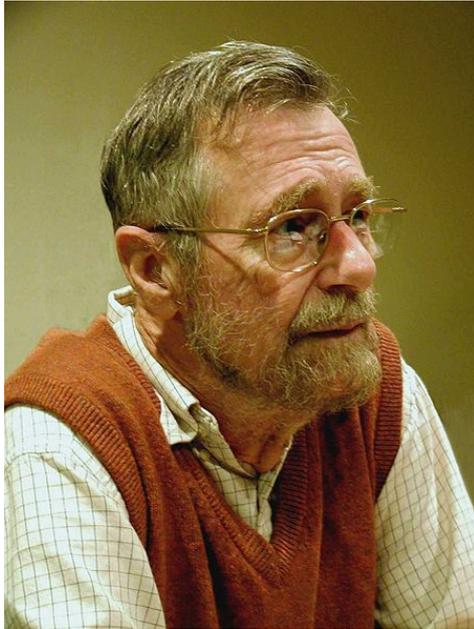
Instradamento statico o dinamico

Si parla di instradamento statico quando i percorsi ottimi vengono calcolati poco frequentemente presupponendo che le condizioni della rete cambino lentamente.

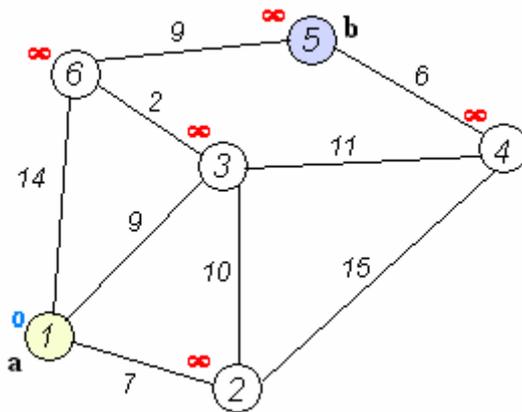
Gli algoritmi di instradamento dinamici vengono ricalcolati invece appena cambiano le condizioni della rete in termini di traffico o topologia. I percorsi vengono ricalcolati o con periodicità predefinita o in seguito allo stimolo provocato da una variazione dello stato della rete.

Link state algorithm. L'algoritmo di Dijkstra

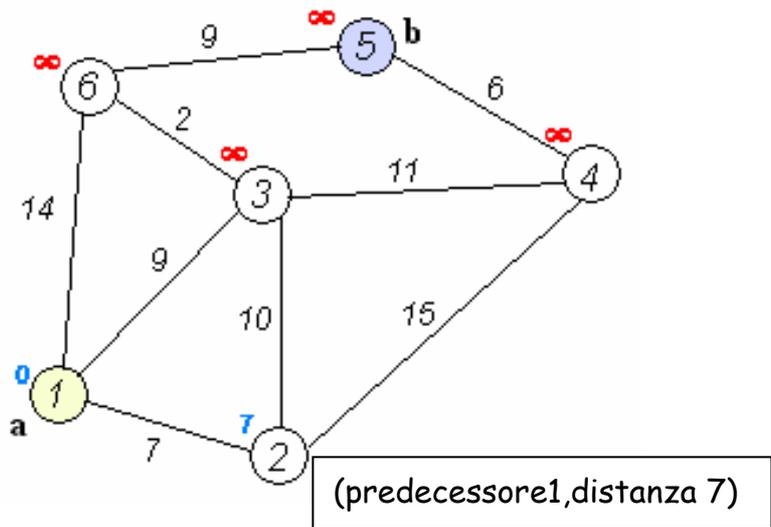
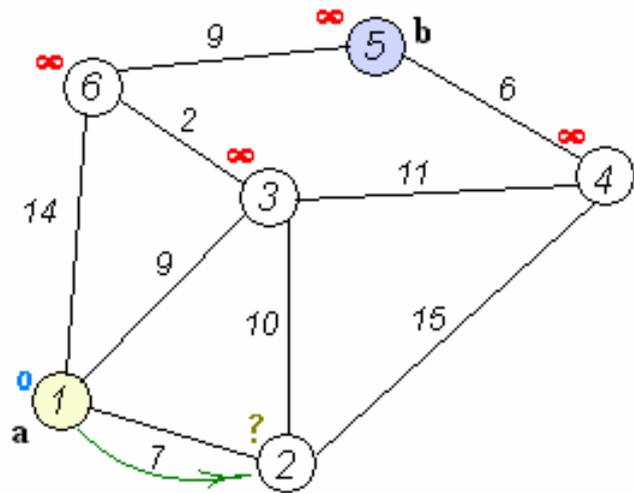
Presentiamo un algoritmo di instradamento globale dovuto ad Edsger Dijkstra.



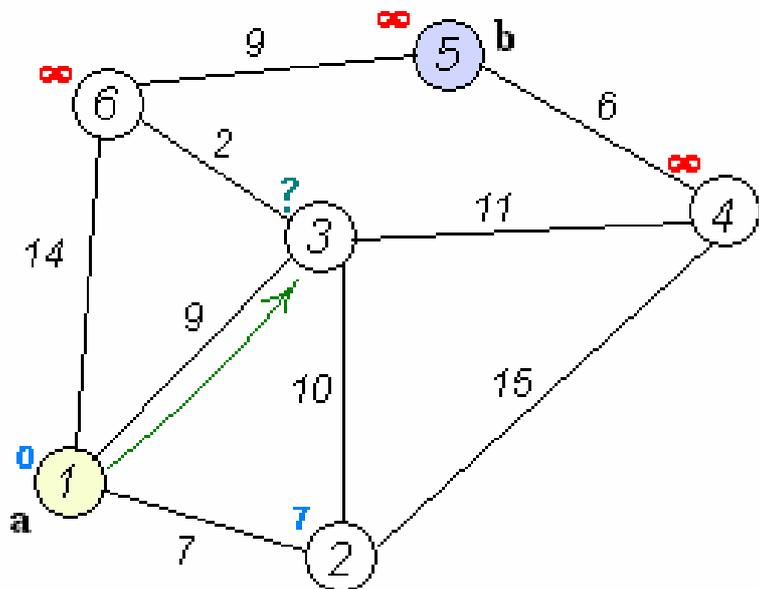
Consideriamo un grafo pesato rappresentante una rete di computer.



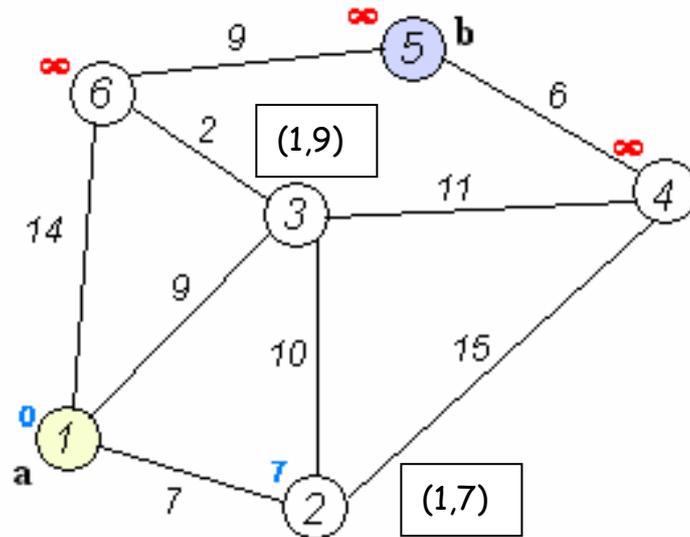
Supponiamo di voler calcolare il miglior percorso fra i nodi 1 e 5. Attribuiamo ad ogni nodo un valore costituito dalla distanza dal nodo di partenza. All'inizio assegniamo un valore indeterminato ad ogni nodo che segniamo per convenzione pari ad infinito. Analizziamo ogni nodo immediatamente raggiungibile dal nodo 1 e gli assegniamo come distanza il peso dell'arco che rappresenta il link fra i due router. Dal nodo 1 si può raggiungere il nodo 2 con distanza 7.



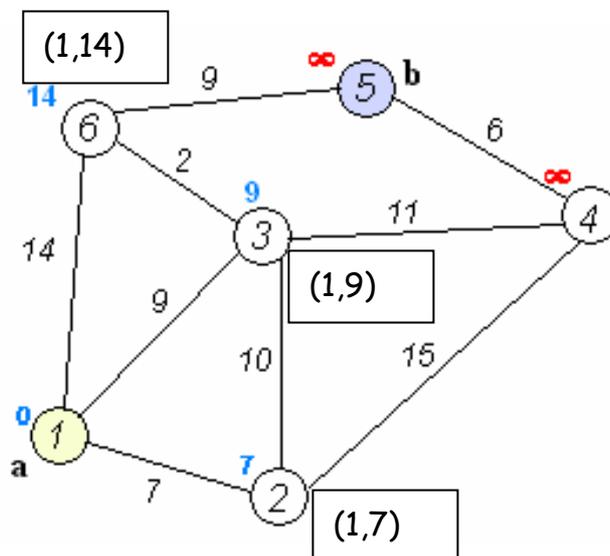
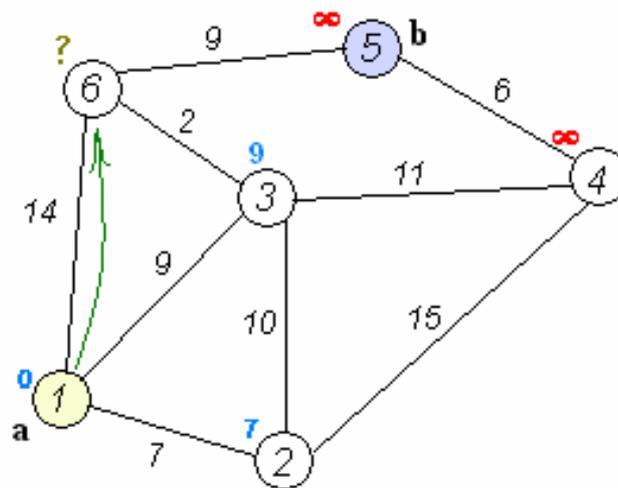
Dal nodo 1 si può raggiungere anche il nodo 3



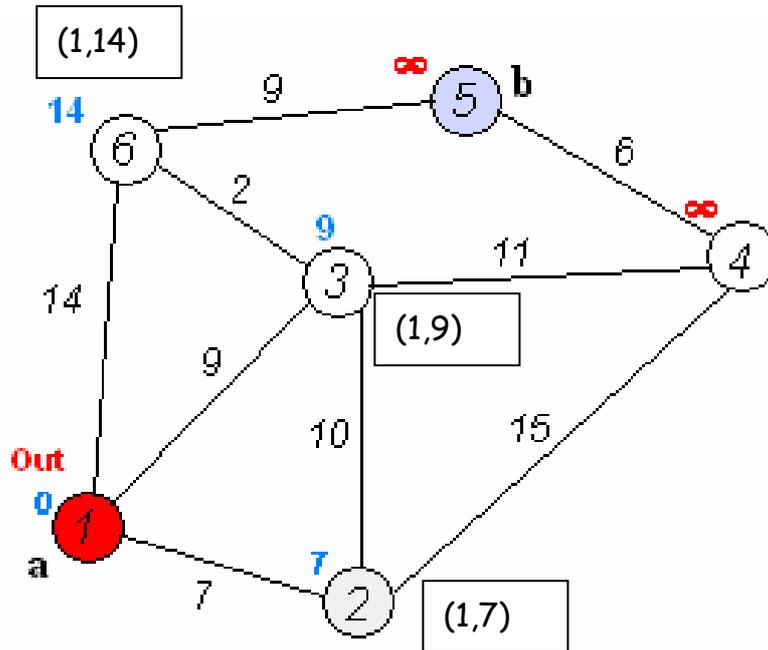
La distanza p è data dal peso dell'arco che rappresenta il link diretto cioè 9. Il predecessore del nodo 3 è dunque 1 con distanza 9



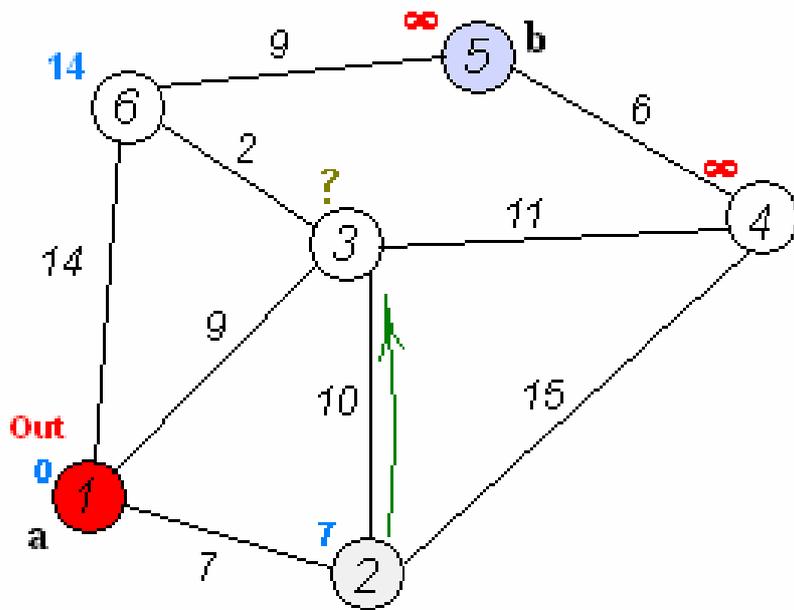
Dal nodo 1 posso raggiungere direttamente anche il nodo 6 con distanza 14



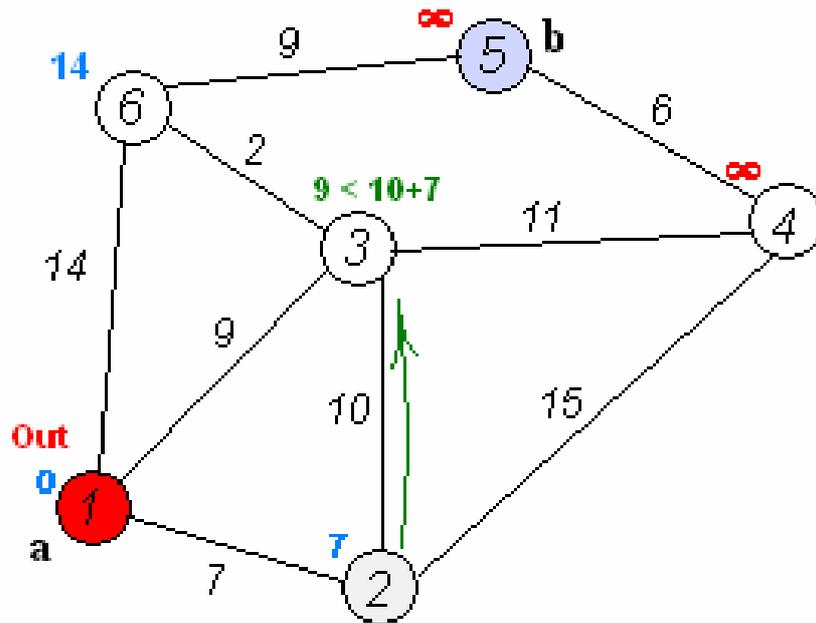
Avendo testato tutti i link diretti utilizzabili dal nodo 1 lo si segna in qualche modo per ricordare che è stato testato. Il primo passo dell'algoritmo è terminato



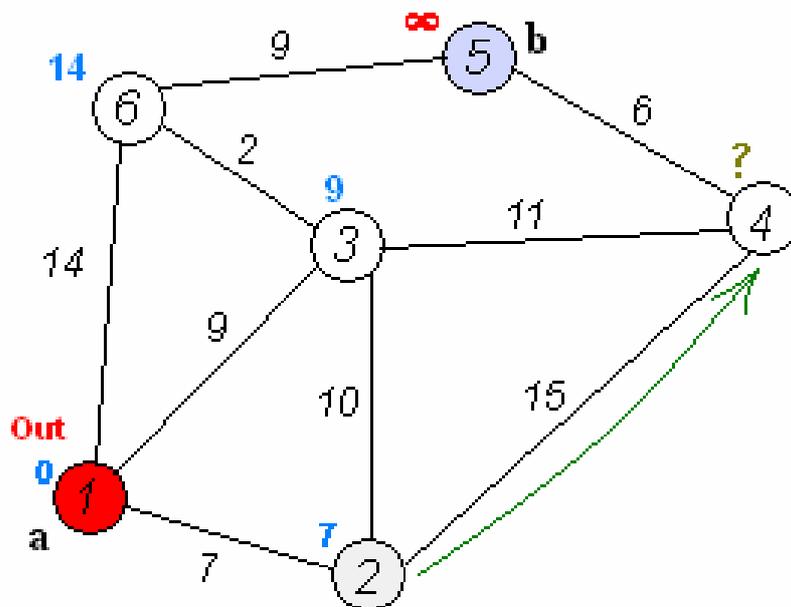
Per ognuno dei nodi direttamente raggiungibili dal nodo 1 abbiamo calcolato le distanze minime. Analizzando le distanze appena calcolate notiamo che il nodo più promettente tra quelli cui è stata attribuita una distanza perché attualmente a distanza minima è il nodo 2. Dal nodo 2 possiamo raggiungere direttamente il nodo 3



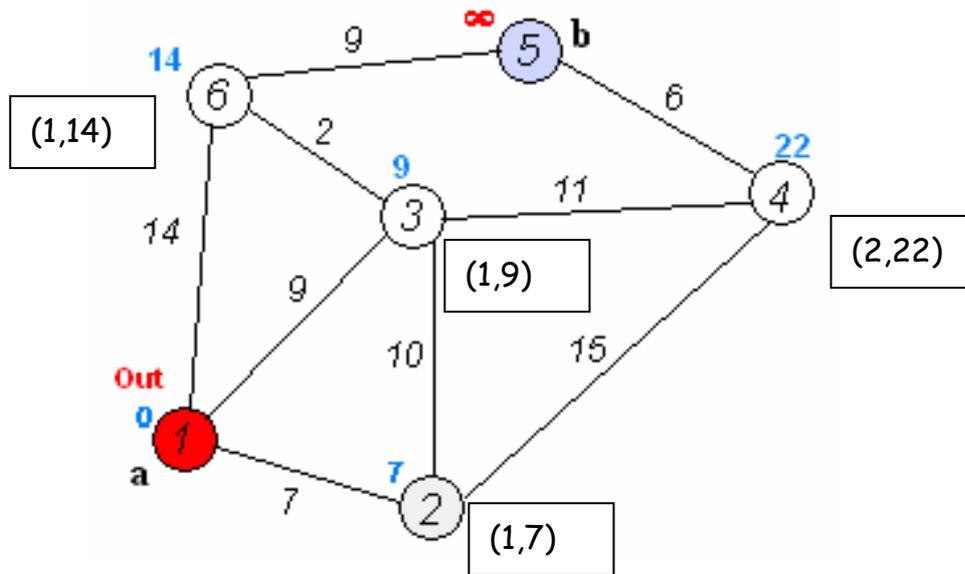
Il percorso dal nodo 1 al nodo 3 attraverso il nodo 2 avrebbe un costo pari alla somma della distanza del link fra 1 e 2 e del link fra 2 e 3.



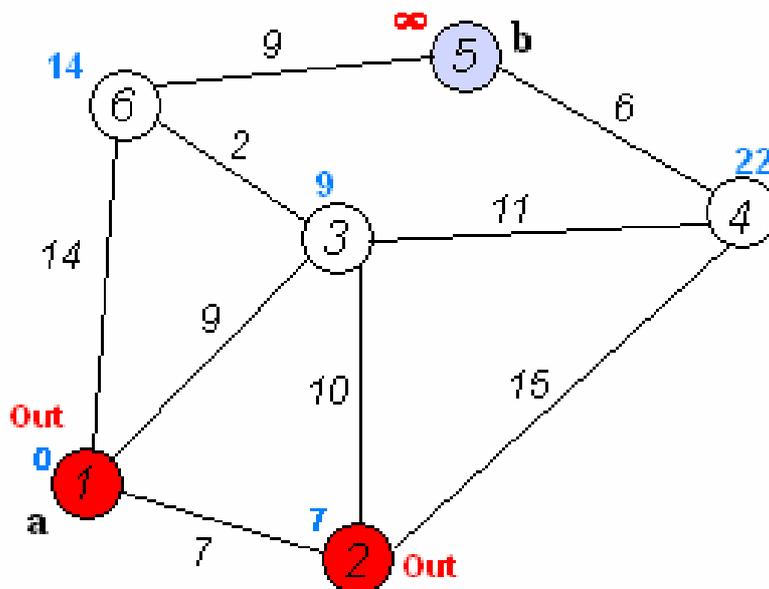
Otteniamo un peggioramento rispetto al percorso diretto da 1 per cui si lasciano inalterati i dati assegnati al nodo 2 nell'iterazione precedente. Si noti che non è detto che debba essere così: se ad esempio il costo del link diretto fra 1 e 3 fosse stato pari a 20, sarebbe diventato più conveniente raggiungere 3 attraverso il nodo 2. Dal nodo 2 possiamo raggiungere direttamente il nodo 4 con distanza 11.



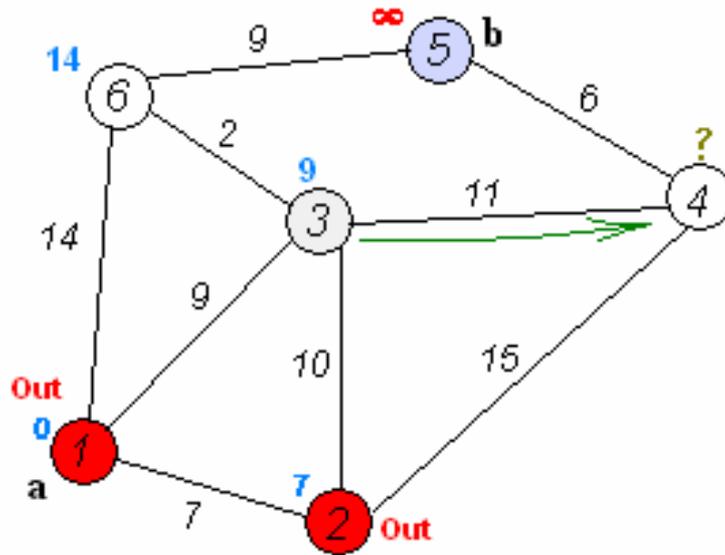
Al nodo 4 attribuiamo predecessore 2 e distanza pari al costo del link 1-2 e del link 2-4, cioè $7+15 = 22$.



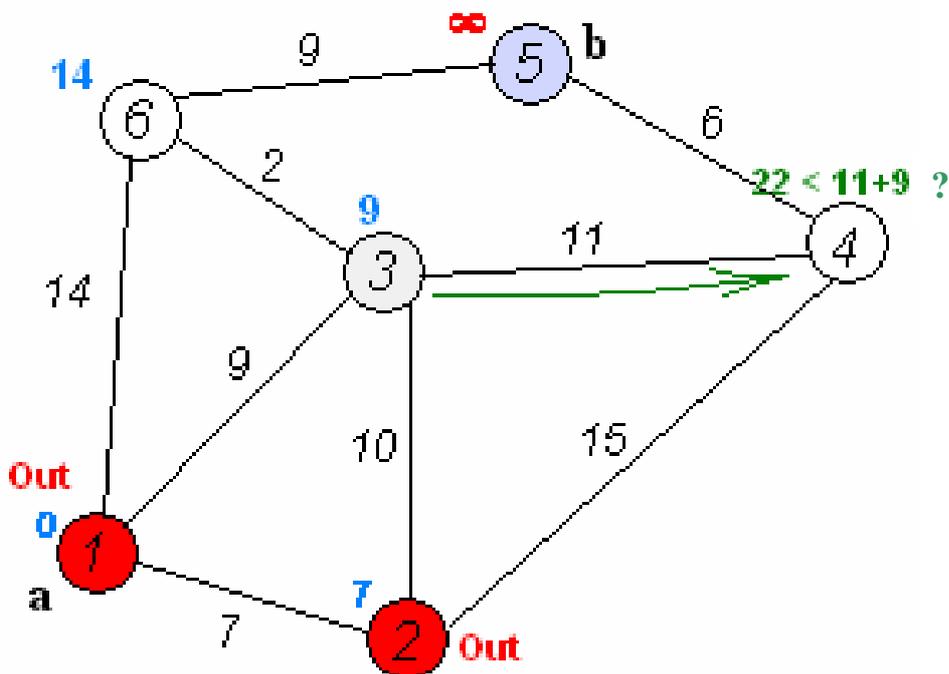
Segniamo che il nodo 2 è stato testato: il secondo passo dell'iterazione è completato.



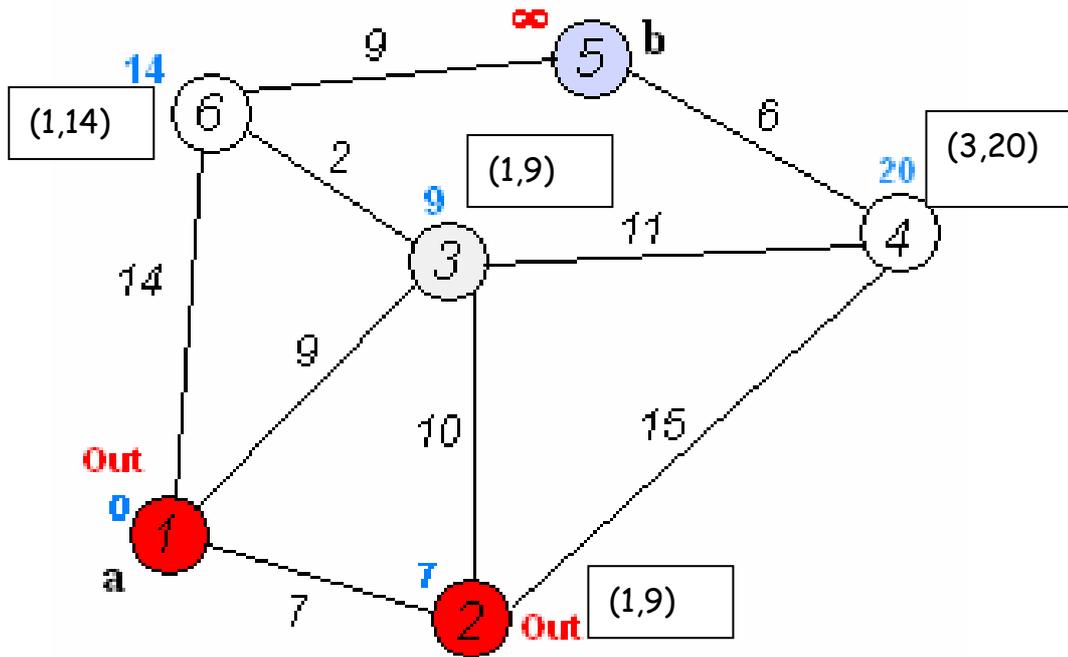
Fra i nodi non testati a cui è stata attribuita una distanza risulta attualmente più promettente il nodo 3 dal quale possiamo raggiungere direttamente i nodi 6 e 4.



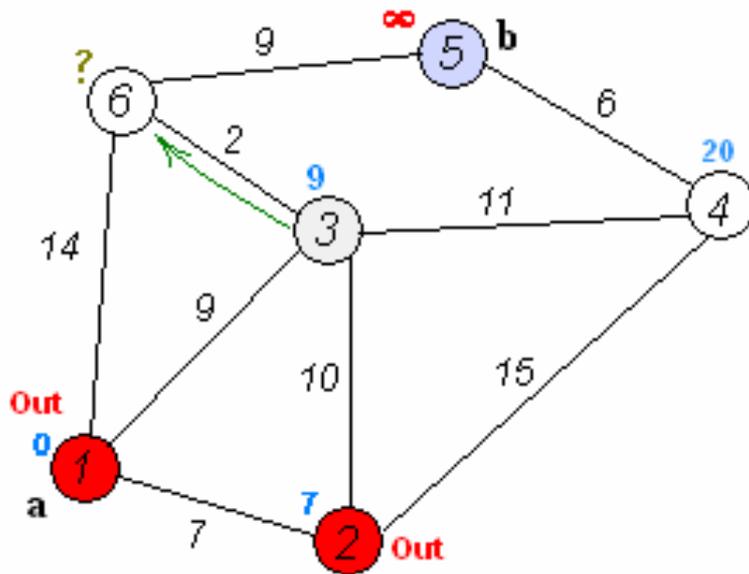
La distanza del nodo 4 dal nodo di partenza attraverso il nodo 3 è pari al costo del link 1-3 sommato al costo del link 3-4: $9+11 = 20$. Si prova se la distanza precedentemente attribuita al nodo 4 sia inferiore alla nuova distanza calcolata.



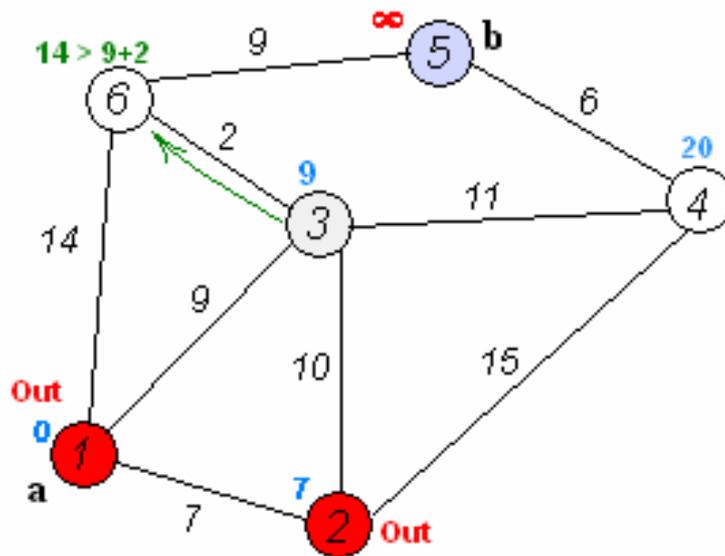
Poiché il risultato del test è falso, aggiorniamo la distanza fra 1 e 4 con il nuovo dato calcolato e imponiamo che il predecessore di 4 sia il nodo 3 in luogo del nodo 2.



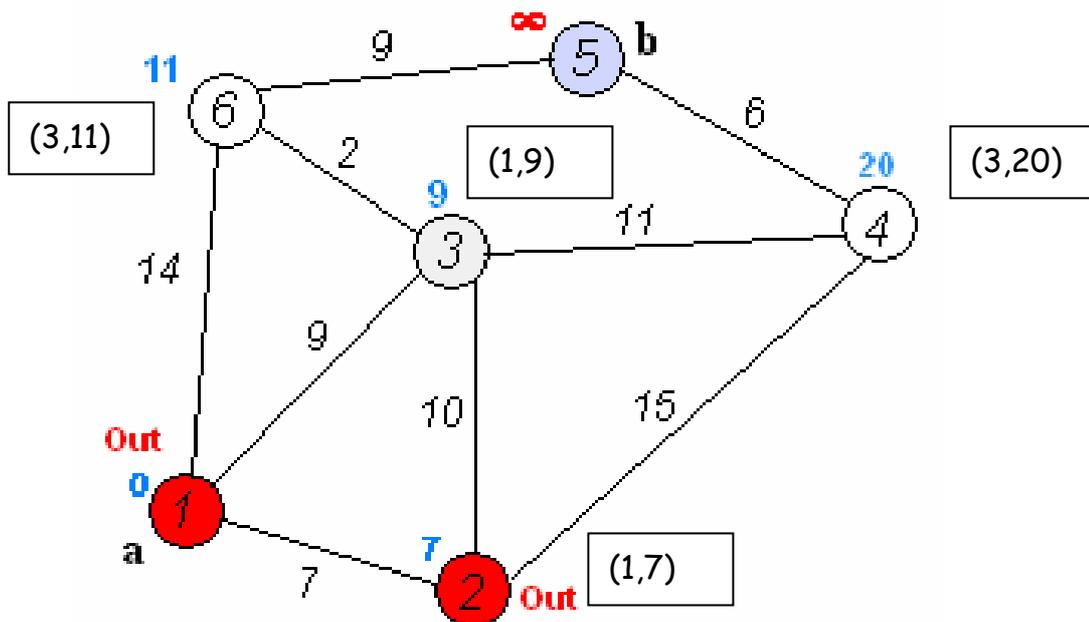
Dal nodo 3 possiamo raggiungere anche il nodo 6.



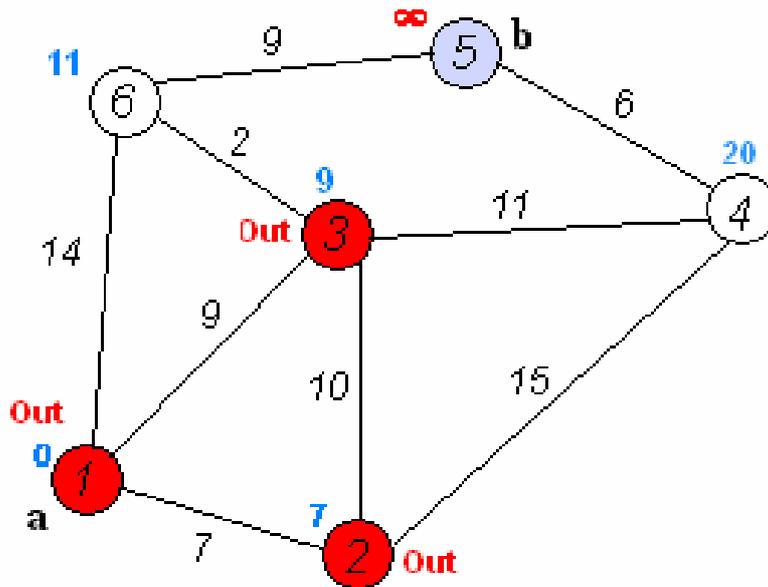
La distanza fra nodo 1 e nodo 6 attraverso il nodo 3 è pari a $9+2$ e risulta inferiore alla distanza precedentemente assegnata: risulta più conveniente raggiungere il nodo 6 attraverso il nodo 3 piuttosto che direttamente.



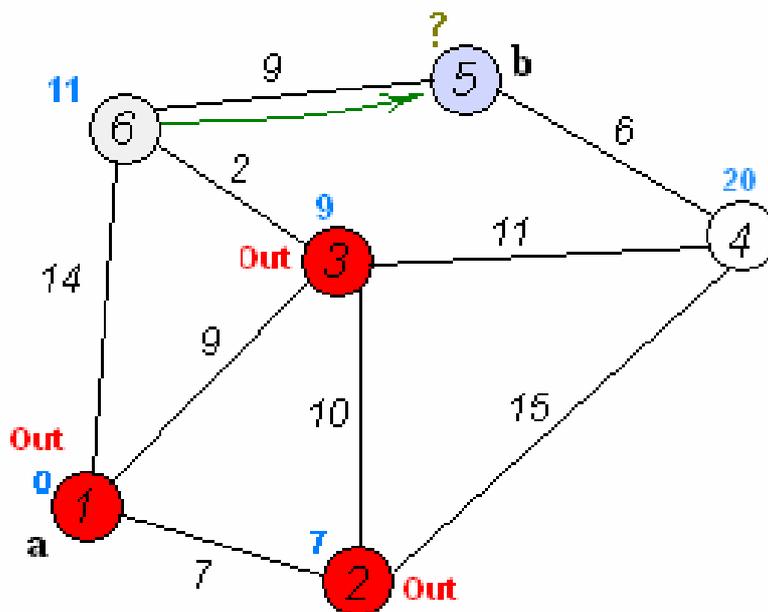
Vengono aggiornate le informazioni sul nodo 6: il nuovo predecessore è 3 e la distanza è ora 11.



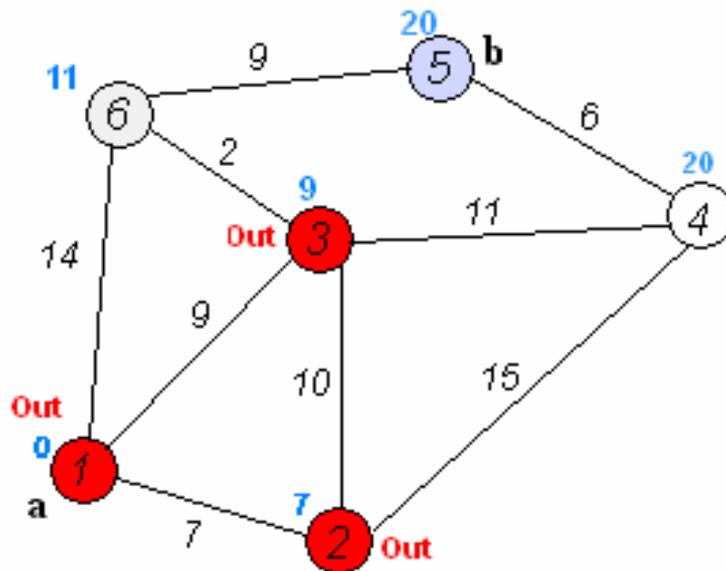
Segniamo il nodo 3 come testato: la terza iterazione è terminata.



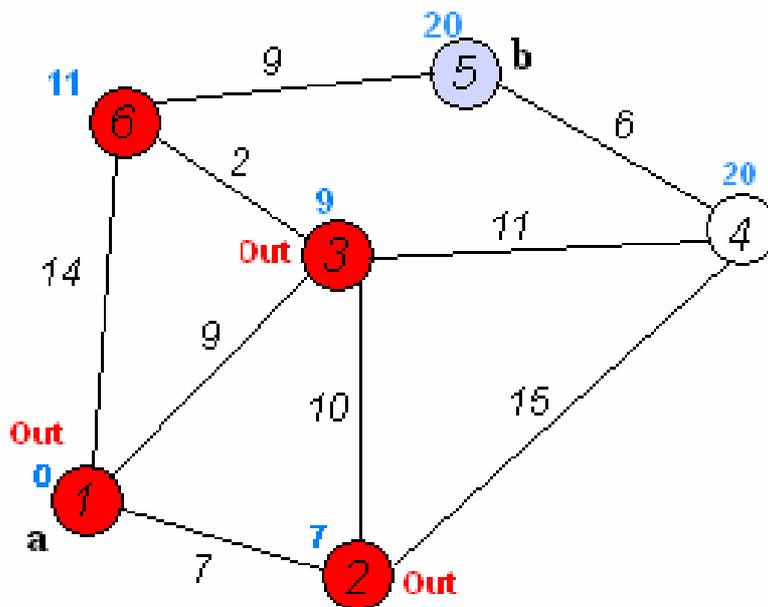
Fra i nodi non testati cui è stata attribuita una distanza risulta ora di distanza inferiore il nodo 6. Da esso si può raggiungere direttamente il nodo destinazione 5.



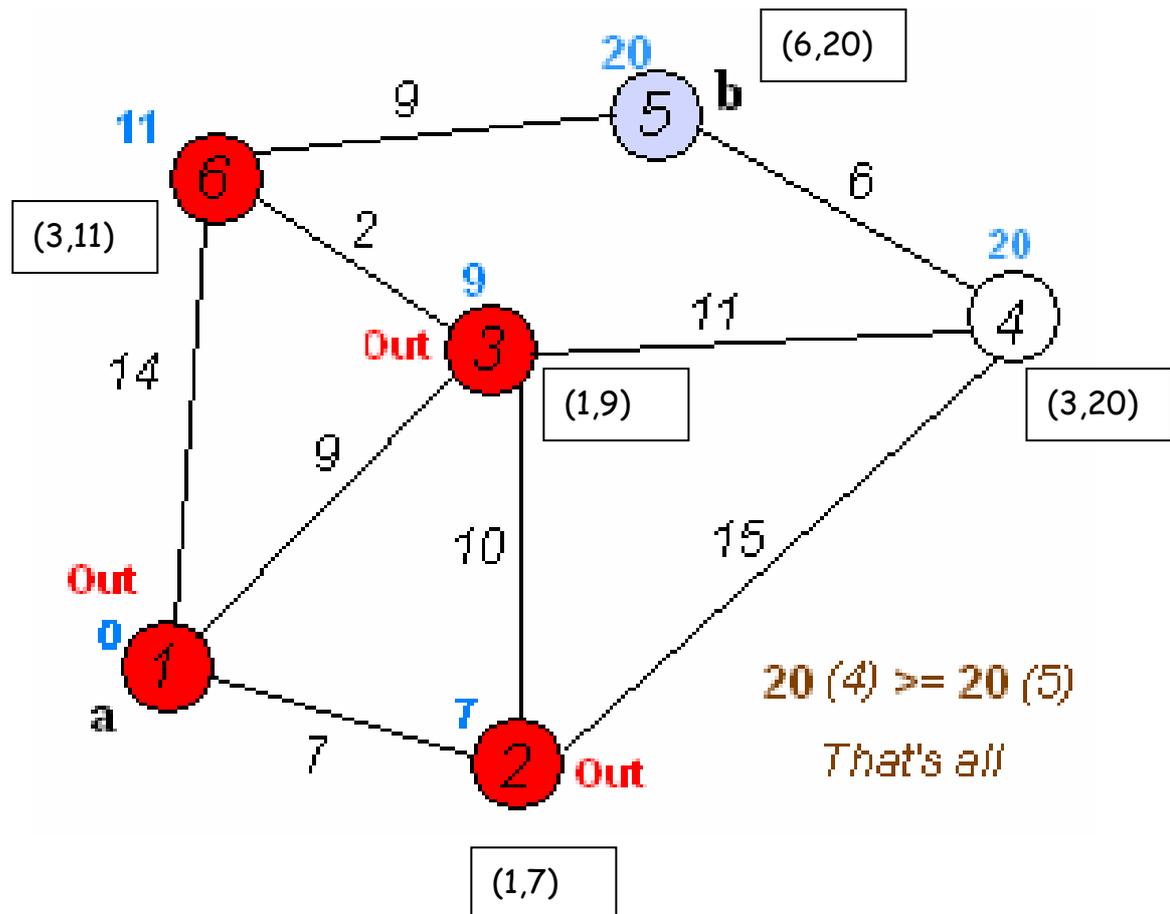
La distanza di 5 da 1 risulta pari a $11+9=20$.



L'iterazione è già terminata poiché dal nodo 6 si può raggiungere soltanto il nodo 5.



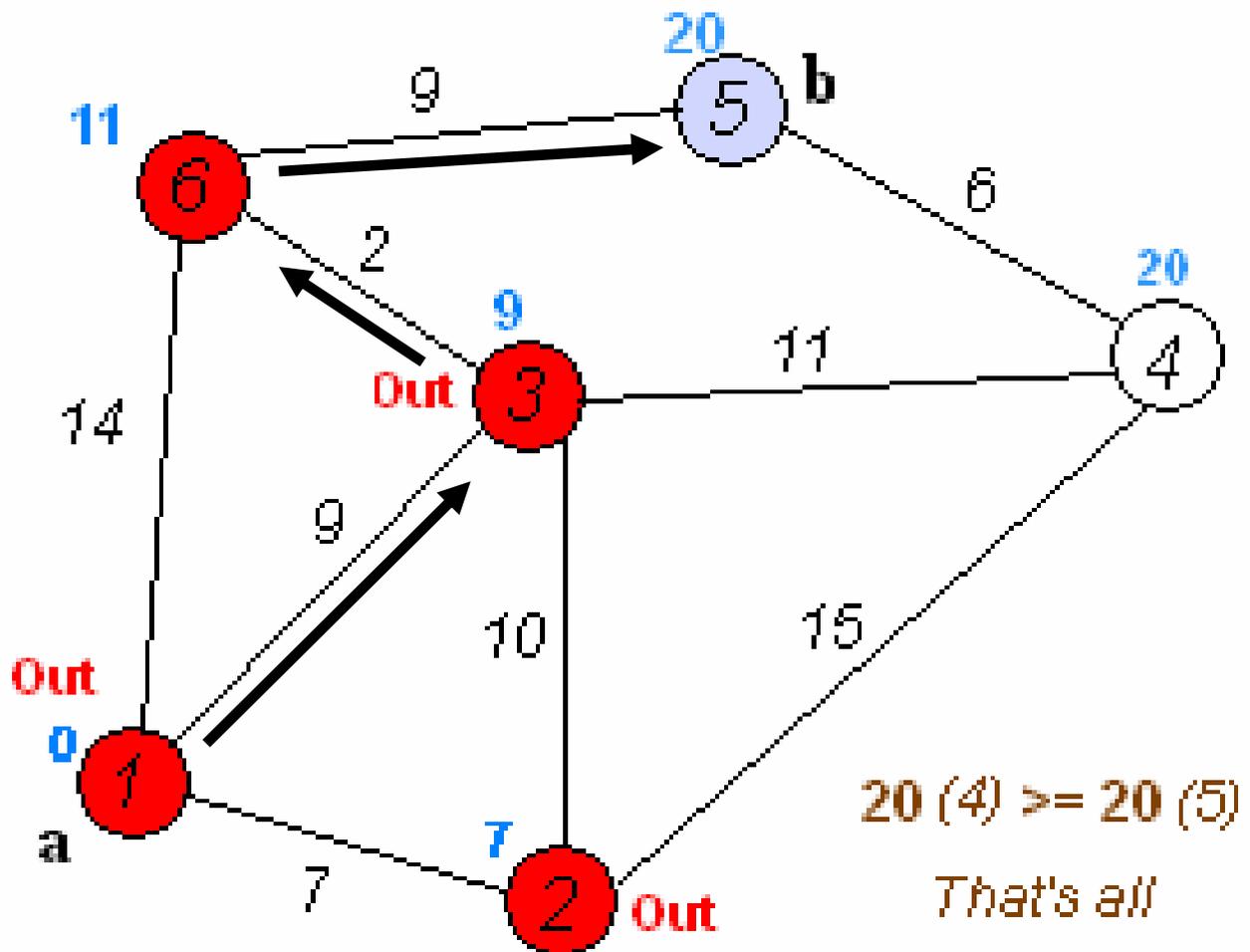
Resta da testare il nodo 4 dal quale si può raggiungere il nodo 5 con distanza $20+6=26$: non c'è miglioramento, l'iterazione finisce perché dal nodo 4 non si può raggiungere nessun altro nodo. Non ci sono altri nodi da testare: l'algoritmo si arresta



Abbiamo determinato il percorso minimo: la distanza la possiamo leggere dal secondo valore attribuito al nodo di destinazione 5. il percorso lo possiamo ricostruire leggendo a ritroso i vari predecessori:

- predecessore di 5:=nodo 6
- predecessore di 6:=nodo 3
- predecessore di 3:=nodo 1

Percorso : nodo 1 - nodo 3 - nodo 6 - nodo 5



Algoritmo di instradamento distribuito: distance vector

Vediamo ora un algoritmo di instradamento distribuito in cui i router non debbono avere conoscenza dello stato globale della rete ma costruiscono il percorso migliore scambiandosi informazioni con gli altri router presenti in rete. Esempio classico è l'algoritmo distanze vector (vettore delle distanze) detto anche algoritmo di Bellman-Ford.

Gli algoritmi di tipo Link State prevedono che ogni router sia informato dei cambiamenti nella topologia della rete. I protocolli basati su Distance Vector - come RIP ed IGRP - sono più leggeri. Ogni router misura la distanza (secondo una metrica che può includere vari fattori) che lo separa dai nodi adiacenti e riceve i dati dai router vicini. Utilizzando l'algoritmo di Bellman-Ford, il router costruisce una tabella che associa ad ogni destinazione conosciuta:

la stima della distanza che lo separa dalla destinazione

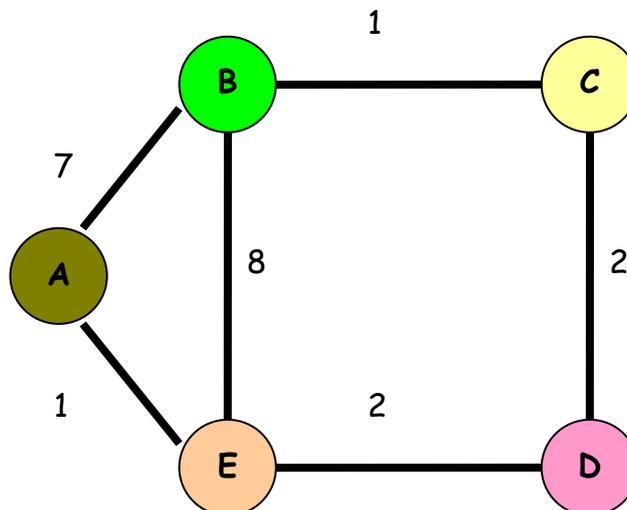
il primo passo del percorso calcolato

Periodicamente il router aggiorna le misure di distanza dai router adiacenti e comunica la propria tabella ai vicini. Dopo sufficienti scambi di informazioni, ciascun router potrà avere una riga per ogni altro nodo nella rete.

Ogni nodo tiene una tabella detta tabella delle distanze. Tale tabella contiene tante righe quanti sono gli altri nodi raggiungibili della rete e tante colonne quanti sono i nodi direttamente raggiungibili.

Data la tabella relativa al nodo X, nell'intersezione fra la colonna relativa al nodo Z e la riga corrispondente al nodo Y si inserisce un valore pari alla distanza fra il nodo X ed il nodo Z più la distanza minima del percorso migliore fra il nodo Z ed il nodo Y. Il primo dato è già noto al nodo X mentre il secondo dato esso lo ottiene come informazione dagli altri nodi.

Consideriamo il seguente esempio di rete



Supponiamo di voler costruire la tabella del nodo E. Essa dovrà essere del seguente tipo

D ^E	A	B	D
A			
B			
C			
D			

La prima cella deve contenere

Il costo o distanza per raggiungere A (riga di A) partendo da E e passando per A (colonna di A)

Chiaramente esso coincide con il costo del link diretto fra A ed E.

D ^E	A	B	D
A	1		
B			
C			
D			

La seconda cella deve contenere

Il costo o distanza per raggiungere A (riga di A) partendo da E e passando per B (colonna di B)

Esso sarà pari al costo del link diretto fra E e B pari a 8 più il costo del percorso a istanza minima fra B ed A. Tale informazione sarà fornita al router E da B ed è pari come si può vedere dalla figura a 6 (percorso B-C-D-E-A).

D ^E	A	B	D
A	1	14	
B			
C			
D			

La terza cella deve contenere

Il costo o distanza per raggiungere A (riga di A) partendo da E e passando per D (colonna di D)

Esso sarà pari al costo del link diretto fra E e D pari a 2 più il costo del percorso a istanza minima fra D ed A. Tale informazione sarà fornita al router E da D ed è pari come si può vedere dalla figura a 3 (percorso D-E-A).

D ^E	A	B	D
A	1	14	5
B			
C			
D			

La tabella definitiva è la seguente

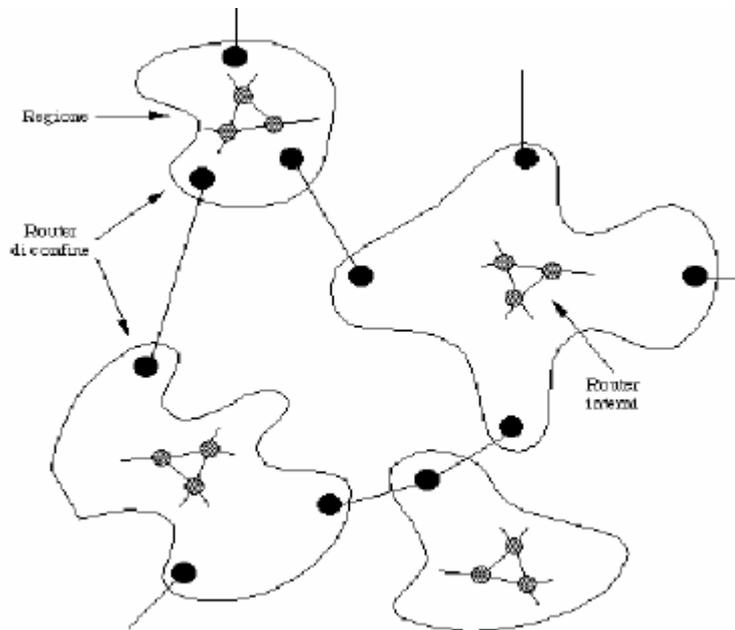
D ^E	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

Per ogni riga il valore minimo indica attraverso quale dei nodi direttamente collegati E deve instradare il messaggio per giungere al nodo di destinazione. Ad esempio sulla terza riga il valore inferiore è in corrispondenza della terza colonna, il che vuol dire che se E vuole mandare messaggi a C li deve inviare al nodo D.

L'instradamento isolato

IN questo tipo di instradamento si rinuncia a prendere una decisione in base ad informazioni sullo stato della rete nel decidere verso quale router instradare il pacchetto. Un esempio di metodo è il hot potato routing o instradamento a patata bollente nel quale un router che riceve un pacchetto tenta di liberarsene al più presto inviandolo ad esempio sul link più corto o meno congestionato.

Un'alternativa potrebbe essere l'instradamento a pioggia in cui il router invia una copia del pacchetto ricevuto a tutti i router direttamente collegati ad esso. Sono evidenti i difetti in termini di aumento di traffico.

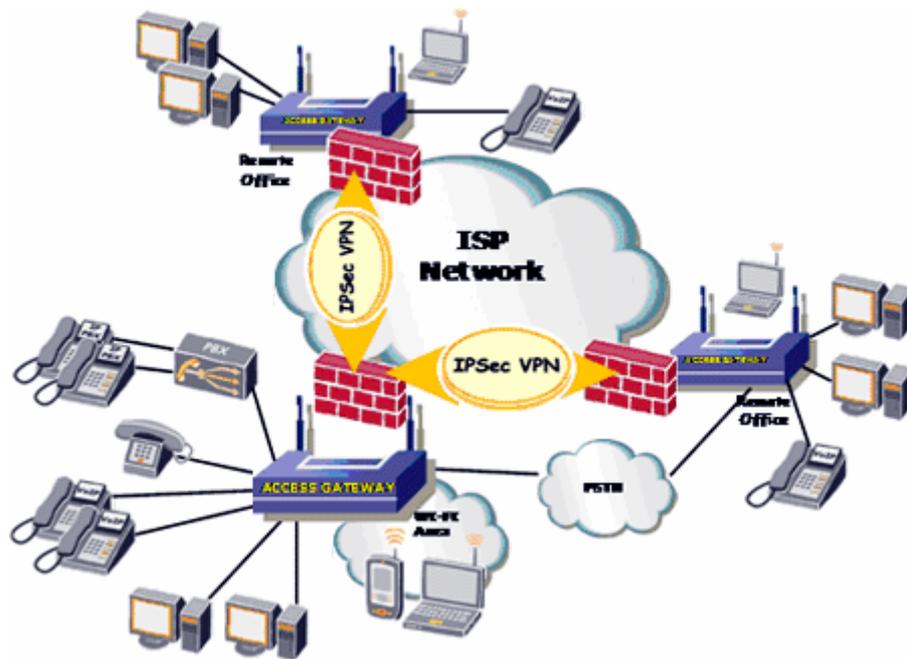


Instradamento gerarchico

Con il proliferare di router e host interconnessi in Internet il carico aggiuntivo o overhead per ogni router per eseguire calcoli, trasmettere informazioni agli altri router, immagazzinare dati sul traffico tende a diventare proibitivo. Si pone poi il problema che le società che gestiscono porzioni della rete sono interessate a mantenere una propria autonomia amministrativa rispetto alle altre realtà in rete.

Nell'instradamento gerarchico abbiamo aggregazioni di router dette regioni o sistemi autonomi (AS Autonomous system). I router presenti in un AS gestiscono il problema dell'instradamento soltanto per i percorsi presenti in quel AS. Il protocollo utilizzato all'interno del AS prenderà il nome di intra-autonomous system routing protocol.

Dei router speciali avranno il compito di instradare i pacchetti fra AS diversi. Essi prendono il nome di gateway. Essi utilizzeranno protocolli di instradamento fra AS diversi detti inter-autonomous system routing protocols.



La congestione

Un elevato numero di pacchetti in circolazione lungo la rete può congestionarla. Un nodo possiede delle aree di memoria o buffer di ingresso in cui immagazzina i pacchetti che gli giungono e dei buffer di uscita in cui immagazzina i pacchetti che stanno per essere spediti. Se i pacchetti giungono al nodo più velocemente di quanto questo riesca ad elaborarne, i suoi buffer di ingresso si esauriscono ed esso non può più accettarne altri che vanno così persi. La stessa cosa può succedere con i buffer di uscita.

La tecnica di preallocazione dei buffer.

Quando si chiede ad un nodo di diventare un nodo intermedio di un circuito di comunicazione virtuale esso assegna a questo circuito un buffer di ingresso ed un buffer di uscita. Una volta che il nodo esaurisce i buffer disponibili, se gli viene inviata una nuova richiesta di essere inserito in un circuito virtuale esso risponde con un messaggio di indisponibilità. Questa tecnica impedisce la congestione del nodo.

La tecnica a scarto di pacchetto

Con questa tecnica un nodo, se si trova in caso di necessità per evitare un sovraccarico di una linea scarta dei pacchetti in ingresso. Questa tecnica ha il seguente inconveniente. Se il nodo vuole inviare un pacchetto verso un altro nodo, esso

ha un buffer di uscita occupato da tale pacchetto. Il secondo nodo ha ricevuto tale pacchetto e manda al primo nodo un pacchetto di riscontro dell'avvenuta ricezione. Il primo nodo potrebbe scartare proprio questo pacchetto di riscontro e in tal modo non si renderebbe conto che la trasmissione è avvenuta in maniera corretta e così non libera il buffer che resta inutilmente occupato.

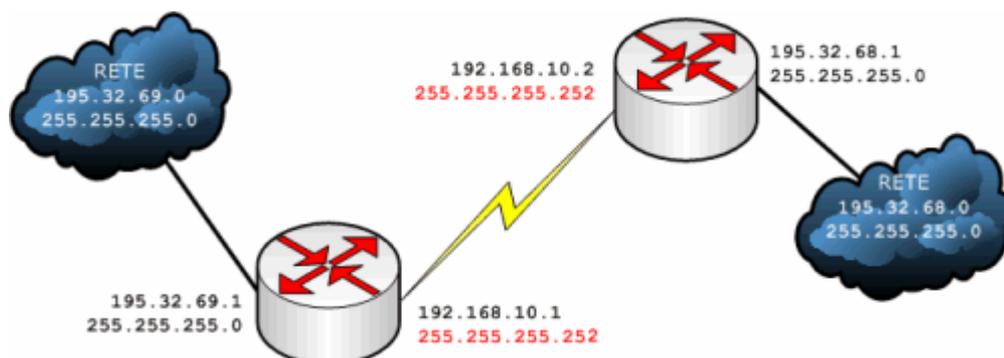
Occorre allora fare in modo che ogni nodo effettui la preallocazione di un certo numero di buffer per consentire almeno la ricezione dei pacchetti di riscontro. Va rilevato, inoltre, che, quando un pacchetto viene scartato, la stazione trasmittente deve ritrasmettere tale pacchetto. Vi sono allora più comunicazioni lungo la rete che non terminano e quindi si sovrappongono determinando un'eccessiva occupazione delle linee di trasmissione. Si cerca allora di migliorare la tecnica dello scarto dei pacchetti facendo in modo che vengano scartati più facilmente i pacchetti di trasmissioni appena incominciate. In tal modo si può portare a termine le trasmissioni che sono già iniziate da tempo e che quindi hanno maggior probabilità di concludersi in breve tempo liberando banda sulle linee.

La limitazione del numero di pacchetti circolanti

Questa tecnica tenta di evitare la congestione limitando il numero di pacchetti che circolano sulla rete. Sulla rete circolano dei particolari pacchetti detti permessi di trasmissione o token. Un nodo prima di poter trasmettere un pacchetto, deve catturare un permesso sulla rete. Quando il nodo ha terminato di trasmettere il pacchetto, rinvia sulla rete il token.

Tecnica di controllo dell'utilizzo delle linee di uscita

Supponiamo di avere tre nodi collegati in un circuito



Ogni nodo sorveglia il traffico sulle sue linee di uscita. Quando l'utilizzazione di una sua linea di uscita supera un valore prefissato e rischia di congestionarsi, esso la marca. Nel nostro caso supponiamo che ciò avvenga proprio per la linea che congiunga nodo B e nodo C. Se ora A invia un pacchetto a B affinché venga trasferito a C, B invia un pacchetto di allarme ad A. A viene costretto a ridurre il proprio traffico verso C.