**Il router:** Trova il percorso che permette ad un pacchetto di partire da una sorgente e arrivare a una destinazione, svolgendo la **forwarding.** Al router arriva un pacchetto in ingresso e deve decidere quale è l’interfaccia d’uscita per inoltrare il pacchetto verso destinazione, il router per **decidere l’interfaccia corretta** si rivolge alla **routing table.**

**Tabella di instradamento:** elenca le rotte di destinazione di una datarete e contiene:

* **L’indirizzo della rete di destinazione**
* **Next** **hop**
* **Interfaccia d‘uscita:** interfaccia del router a cui deve essere inoltrato il pacchetto per raggiungere il next hop.
* **Metric**: costo del percorso, coincide con il numero di hop che devono essere fatti per raggiungere la destinazione

**Instradamento statico**

Le **tabelle di instradamento** si **fanno manualmente** e quindi **non richiedono** ai router **molte prestazioni** **limitando** così **l’uso di banda**.

**Vantaggi:** Consigliabile per reti piccole e non richiede molte prestazioni

**Svantaggio**: ci vuole una persona che inserisca le rotte di instradamento e se cambiano spesso, il lavoro diventa costoso o impossibile.

**Instradamento dinamico**

Permette ai vari **router di scambiarsi informazioni** per determinare i possibili percorsi per raggiungere le destinazioni mediante i **protocolli di routing** che **usano** gli **algoritmi di routing** (Link State).

**Svantaggi**:

* **Maggiore** **prestazione** della **CPU**: il router oltre ad instradare i pacchetti, dovrà creare e aggiornare la sua tabella di instradamento da solo. Lo fa continuamente
* **Si porta via della banda**: i router devono comunicare tra di loro le informazioni sulle rotte

**Obiettivi di un protocollo routing**

* **Ottimalità:** deve essere fornito il migliore percorso
* **Imparzialità:** deve utilizzare tutte le linee disponibili per distribuire il traffico
* **Flessibilità:** deve garantire capacità di adattarsi ai cambiamenti della rete
* **Convergenza veloce:** tutte le tabelle di instradamento di tutti i router devono aggiornarsi quando avviene un cambiamento nella rete
* **Robustezza:** deve funzionare anche se ci fossero dei guasti
* **Semplicità**

**Algoritmo di routing Link State (Dijkstra):**

La **caratteristica principale** è che **ogni router conosce tutta la rete.**

Il router quando viene acceso non ha nessuna conoscenza della rete, ma solo a chi è connesso e il costo di quel link.

Ogni router manda fuori un **LSP** (**Link State Packet**), il quale permette lo scambio di informazioni relative alla rete, in modalità **flooding.**

**Esempio:**

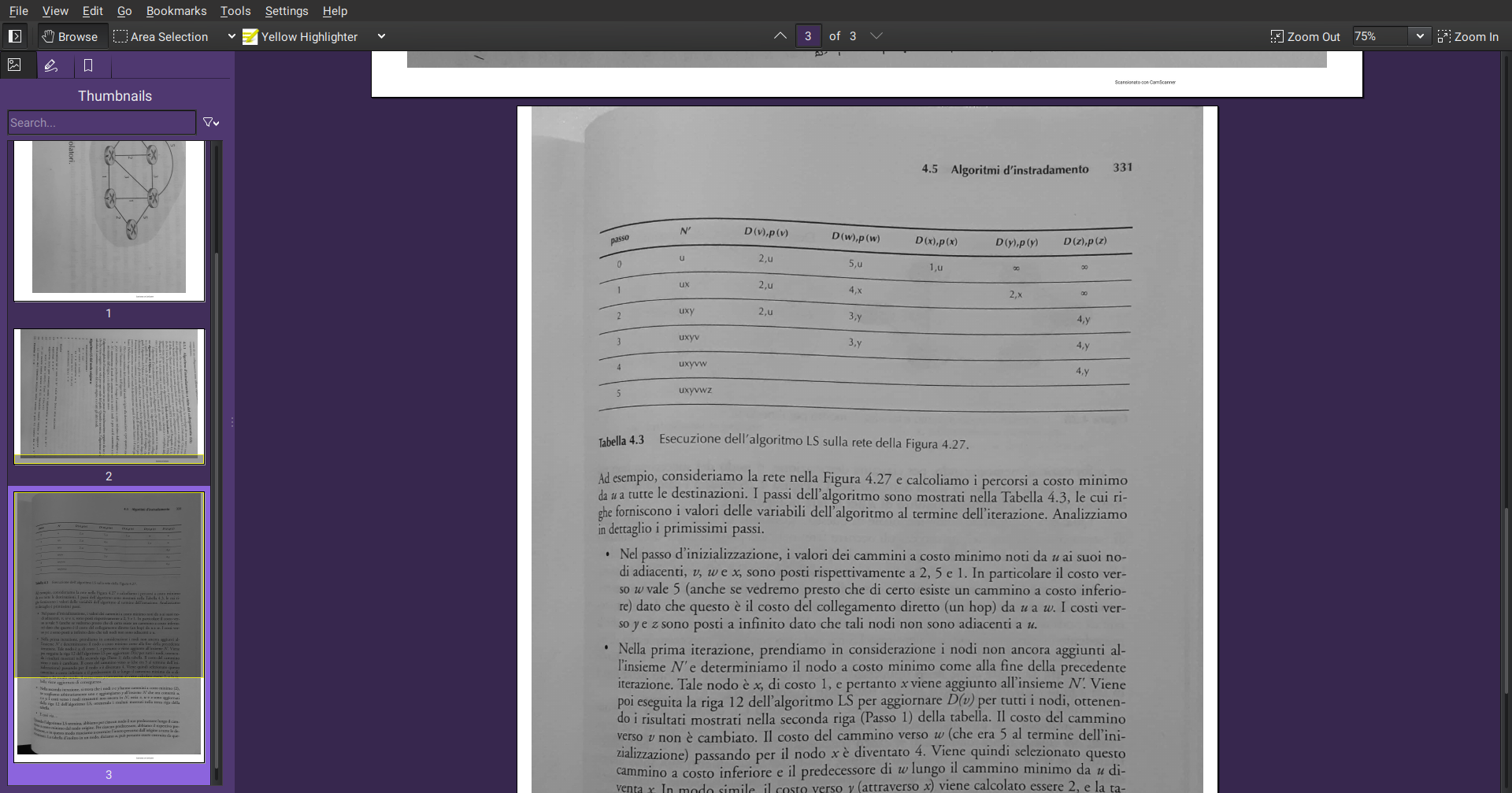
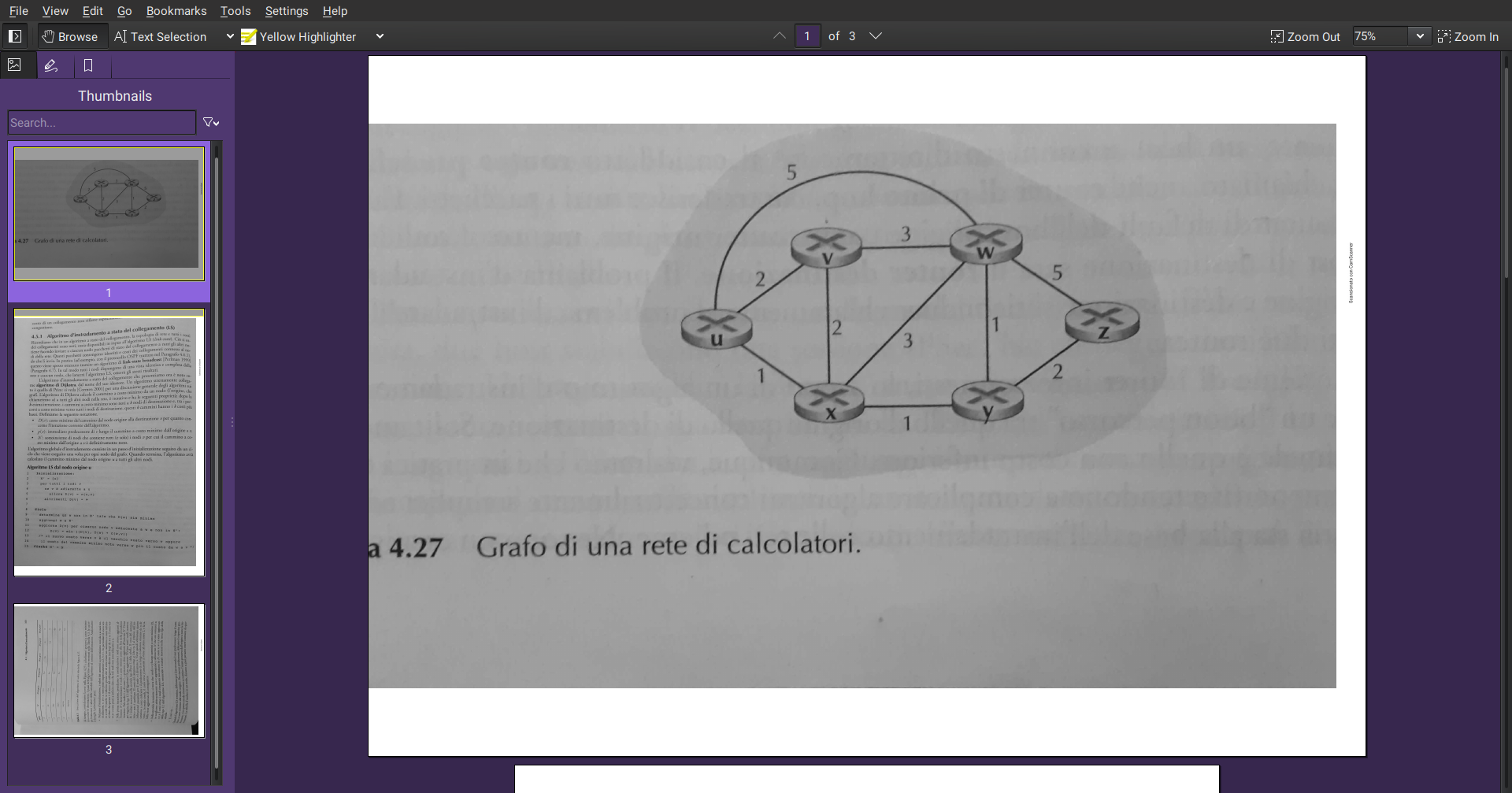
Router A riceve dal Router B le informazioni che gli dice cosa vede lui e se lui vede il Router C, il Router A che prima non lo vedeva adesso lo sa e che è raggiungibile tramite il Router B. E queste informazioni il router A li trasmette ai suoi vicini.

Gli LSP, nella fase iniziale, in teoria dovrebbero viaggiare solo a fronte di cambiamenti, poi viaggiano a intervalli regolari.

**Caratteristiche** del **Link State**:

* Conosce tutta la rete
* Ha una convergenza rapida
* Difficilmente genera loop ed è in grado anche di identificarli e interromperli
* Tutti i nodi hanno basi di dati identiche
* È facilmente scalabile (all’aumentare del numero di router)

**Algoritmo di Dijkstra**



**Algoritmo di routing Distance Vector (Bellman-Ford)**

All’accensione i router memorizzano le interfacce e le distanze con i loro vicini e tra di loro si scambiano le loro tabelle di instradamento aggiornandosi.

**Vantaggi:**

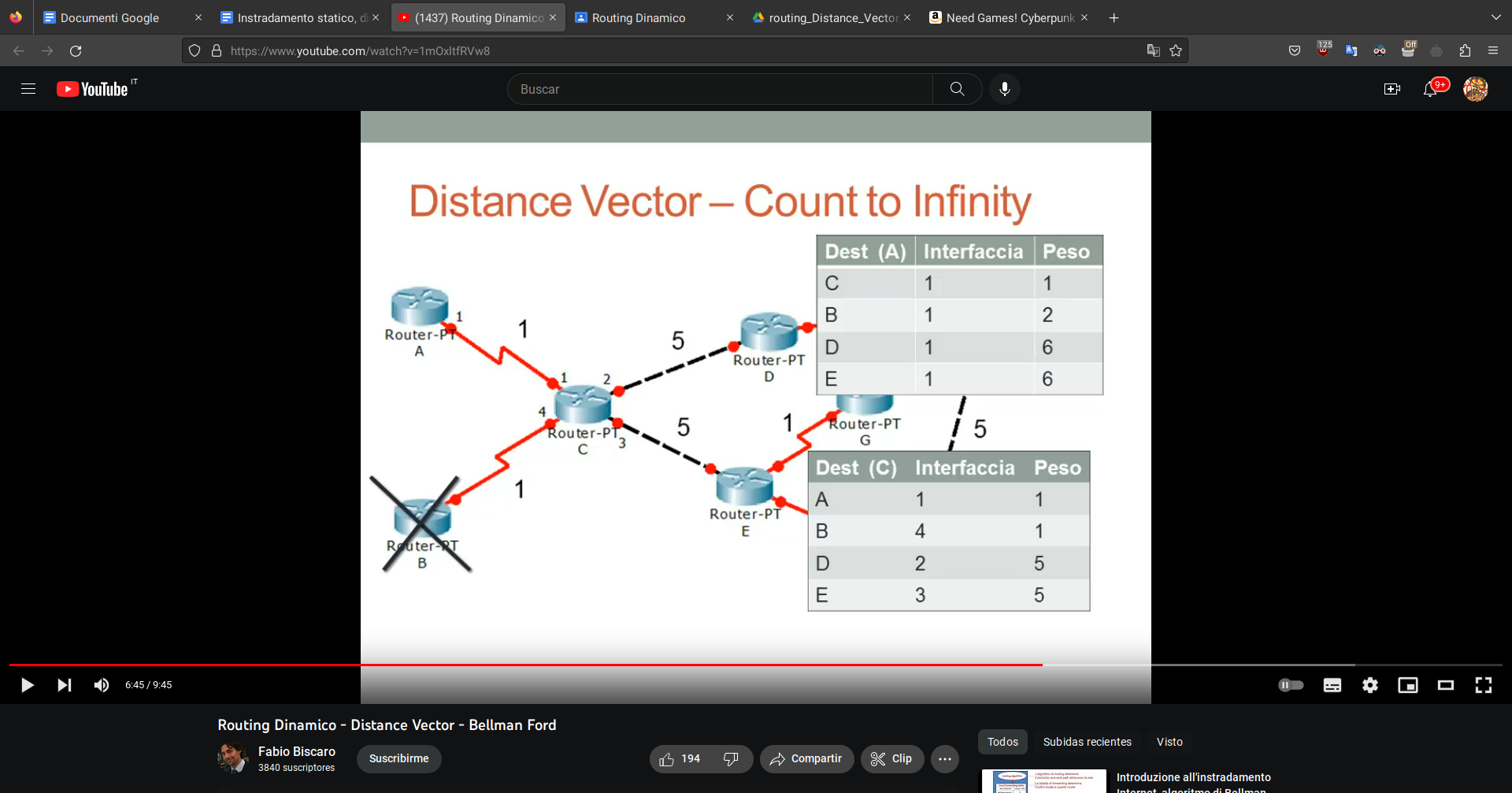
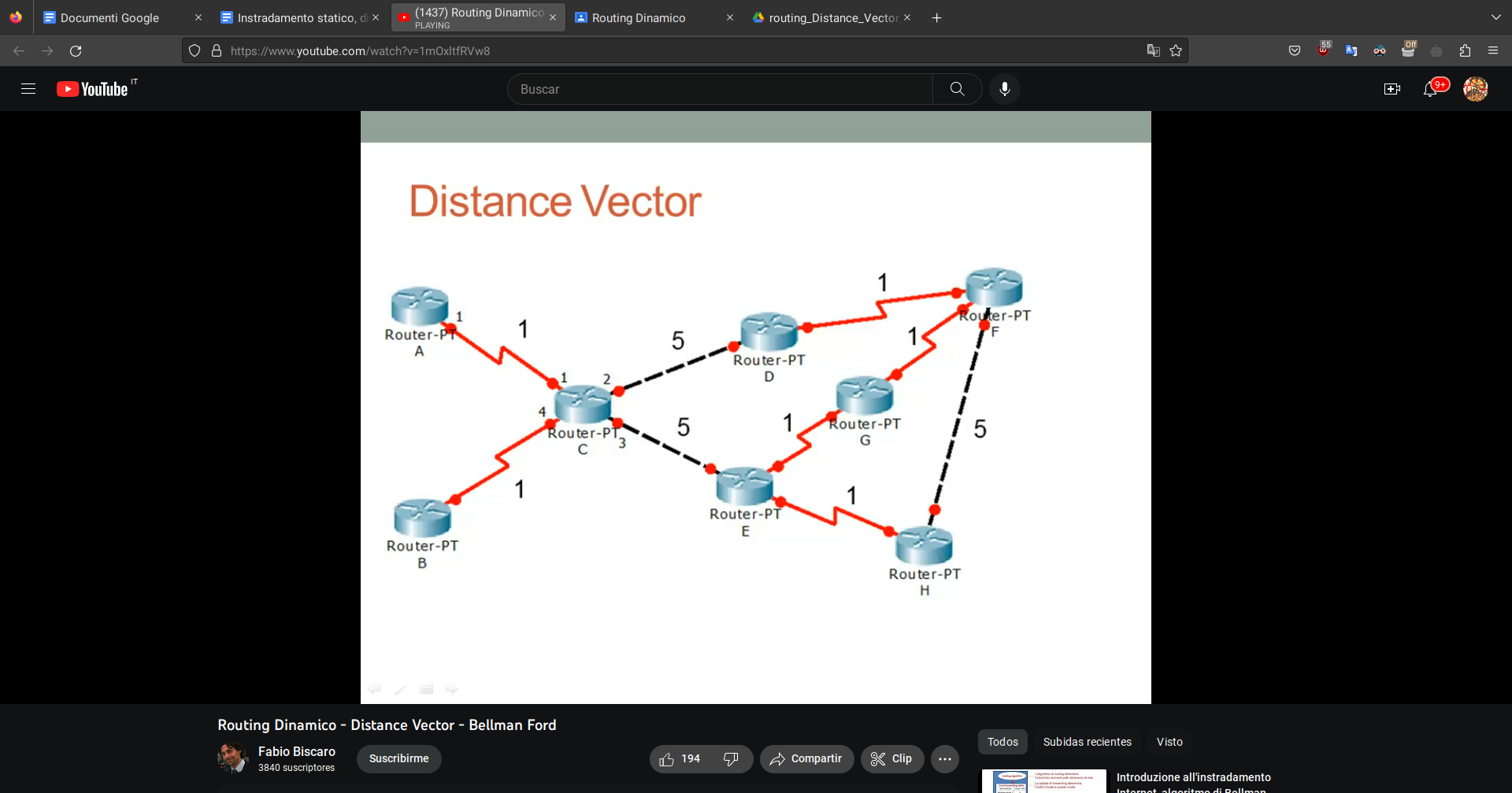
* È semplice
* Si possono trasmettere le modifiche nella rete

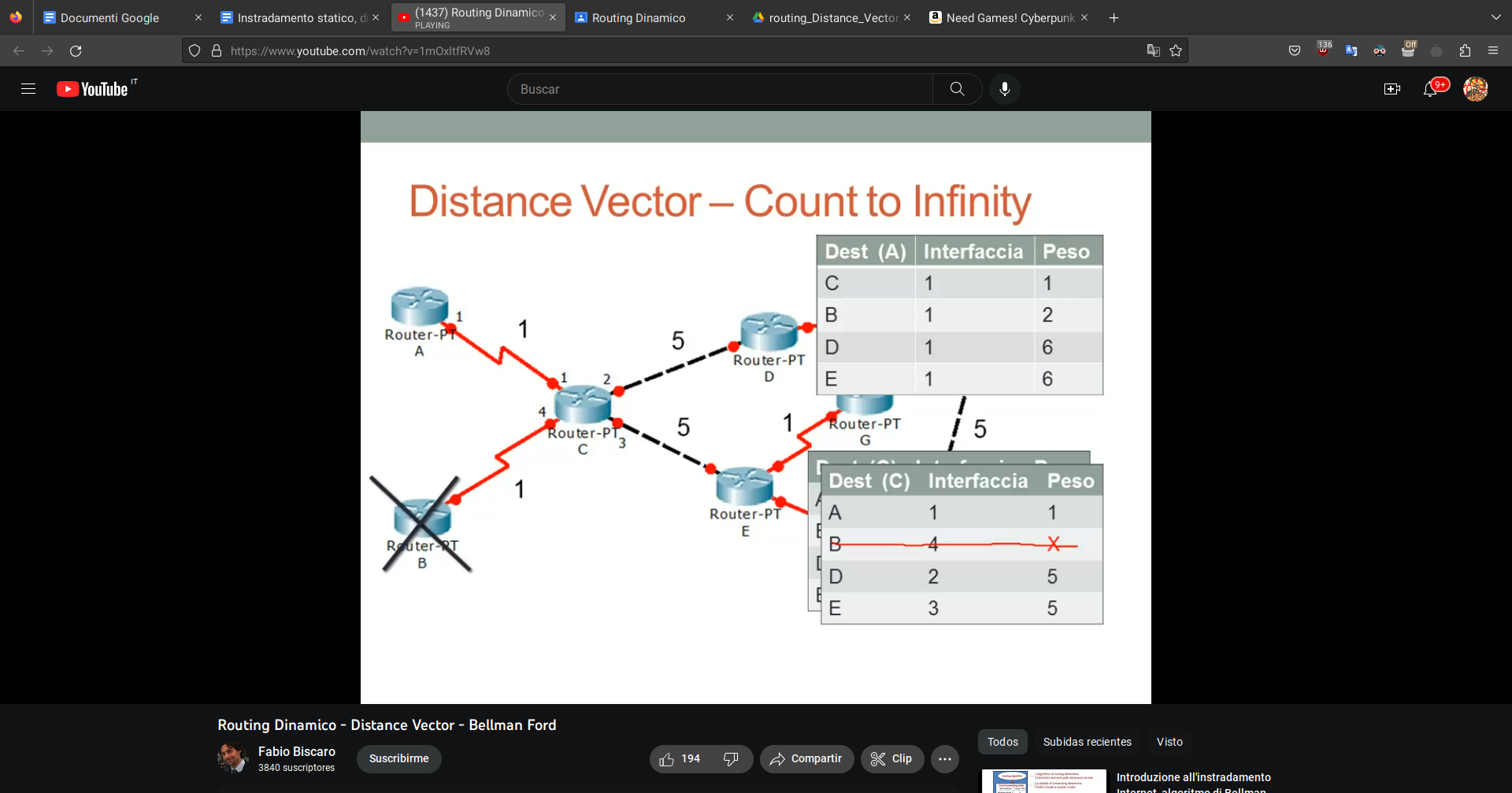
**Svantaggi:**

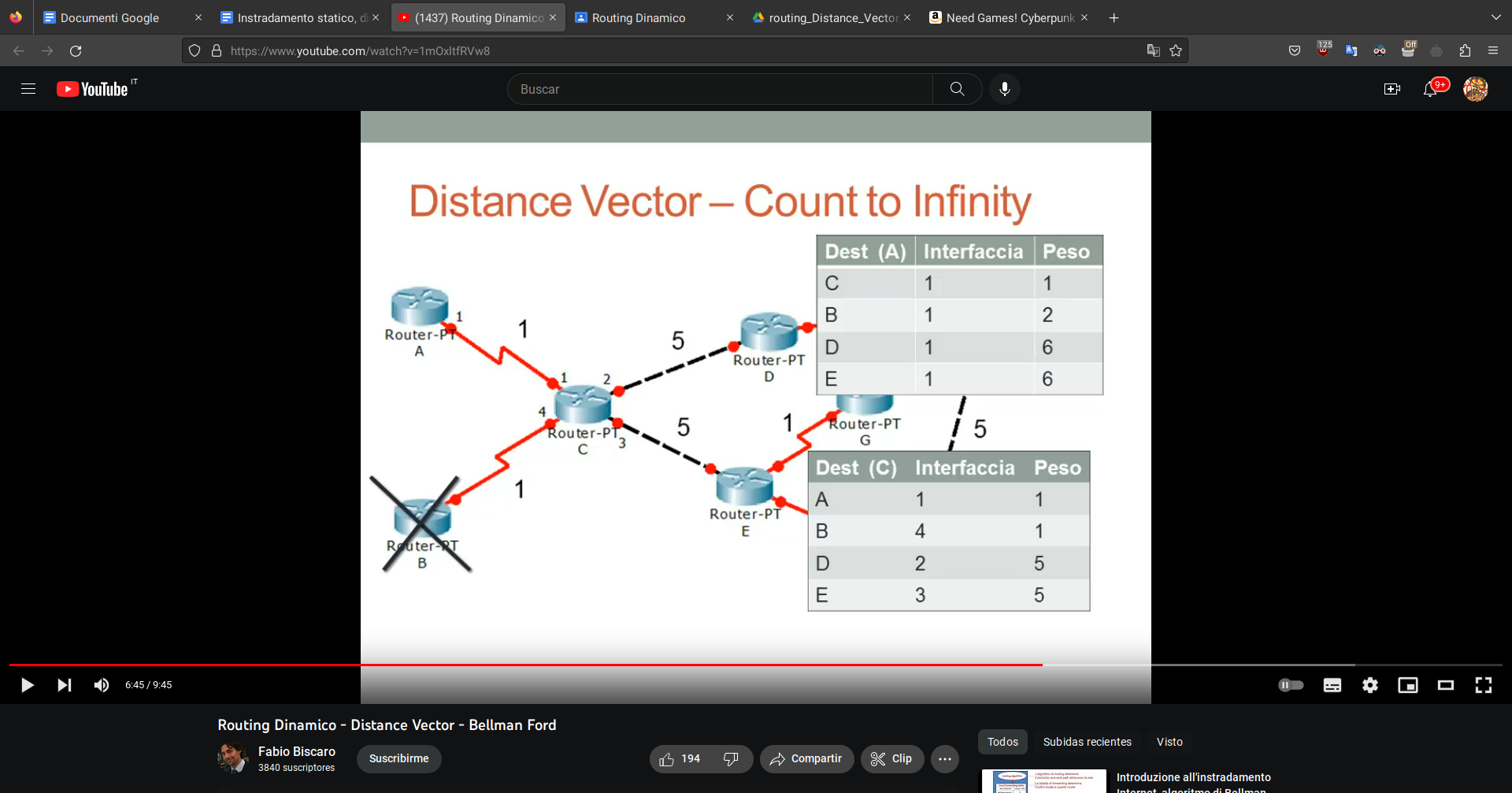
* Si scambiano molti dati ed è più lento
* I router fanno fatica a riconoscere un loop
* La convergenza richiede molti messaggi, dopo NxN messaggi scambiati
* Count to infinity

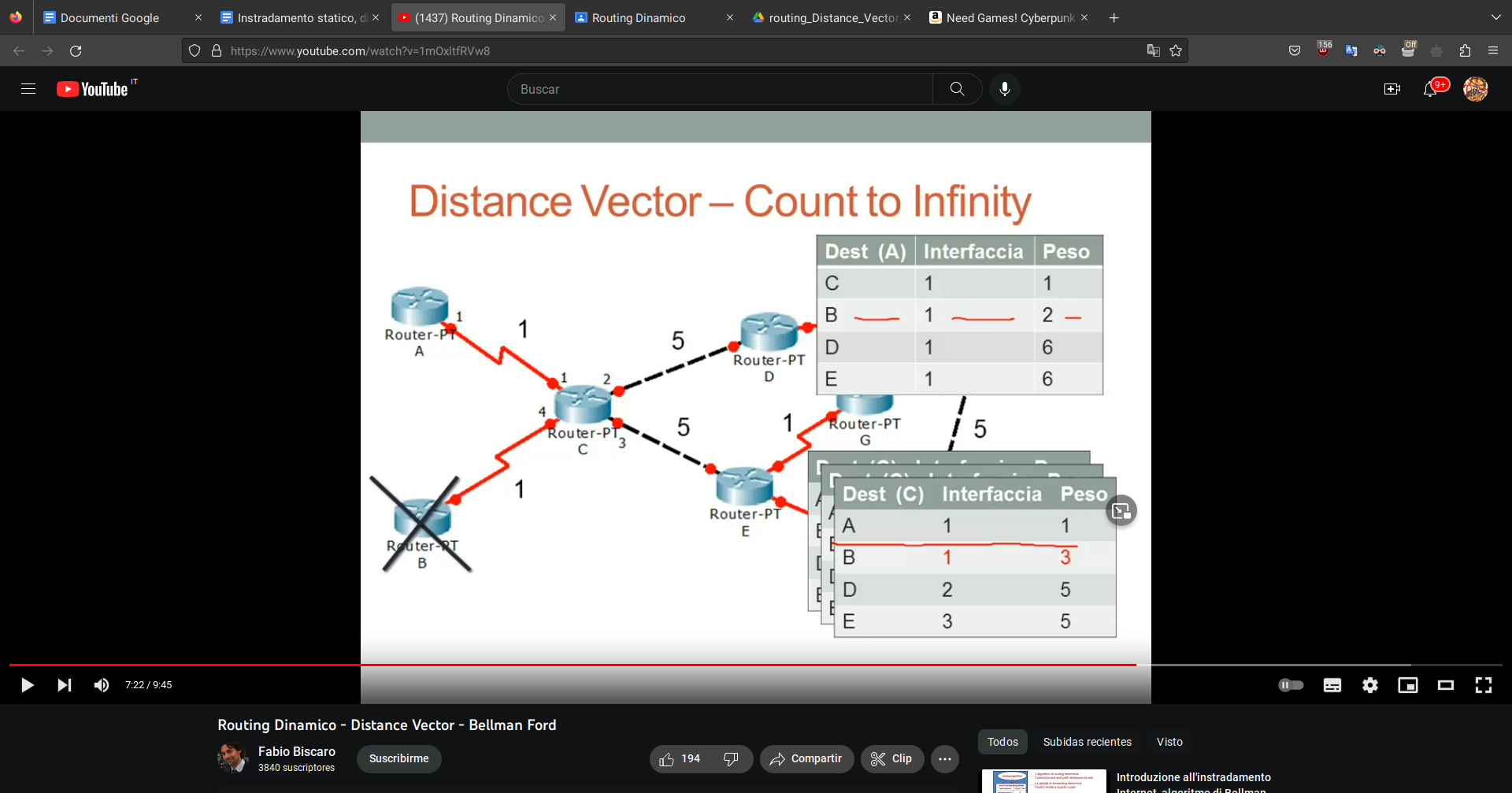
**Problemi:**

**Count to Infinity:**



Supponiamo che router B non c’è più, la tabella del router C si aggiorna:



Ma visto che nella tabella del router A esiste ancora B, questa manderà la sua tabella al router C e passerà a questo:

Diventa 3 perché si sommano i pesi da router A per arrivare al router B (che non esiste più) e il peso tra router A e C. Questa situazione continuerà all’infinito.

**Routing loop:** il pacchetto rimane in un ciclo senza mai giungere a destinazione

**Soluzioni:**

* **TTL**
* Un **timer** che fa scadere le tabelle routing
* **Split** **horizon**: non viene rimandato un pacchetto a chi ce l’ha inviato
* **Poison reverse**: se una rete diventa irraggiungibile, il suo **peso** viene **impostato all’infinito**
* **Route poisoning:** blocca i percorsi che aumentano il costo supponendo che si tratti di un loop. Lo svantaggio è che potrebbe una congestione temporanea
* **Triggered updates:** invia aggiornamenti appena si verifica un cambiamento nella rete
* **Hold down:** per un periodo di tempo, un router non accetta aggiornamenti relativi ad un link che è stato rimosso dal vettore

La **differenza** tra i due algoritmi è che il **LS utilizza l'intera topologia della rete** per prendere decisioni di instradamento, mentre **DV si basa sulle informazioni scambiate tra router vicini** per prendere decisioni di instradamento.