

Ethernet: stp e bpdu

- da questo pdf non ho capito lo stp.

Certo, analizziamo insieme lo Spanning Tree Protocol (STP). È un concetto cruciale che risolve un problema molto specifico che può accadere nelle reti Ethernet.

Prima di vedere *come* funziona, cerchiamo di capire *perché* è necessario.

Le reti sono spesso costruite con collegamenti ridondanti per garantire che, se un cavo si rompe, ce ne sia un altro di riserva. Immagina tre switch (A, B e C) collegati tra loro a triangolo.

Ora, ripensando a come uno switch impara gli indirizzi e a cosa fa quando non conosce una destinazione (il "flooding" o broadcast), cosa pensi che succeda se un computer collegato allo switch A invia un messaggio broadcast in questa rete a triangolo?

- Looppa

Esatto, la ridondanza crea un loop (un ciclo infinito).

Un messaggio di broadcast inviato da A verrebbe inoltrato da B a C e da C a B. Entrambi lo rimanderebbero ad A, che lo inoltrerebbe di nuovo, e così via, all'infinito. Questo fenomeno è chiamato "broadcast storm" e può bloccare completamente una rete in pochi secondi.

È qui che entra in gioco lo Spanning Tree Protocol (STP). Il suo unico scopo è prevenire questi loop. Lo fa in modo molto intelligente: esamina la rete e disattiva temporaneamente i collegamenti ridondanti, trasformando una topologia con cicli in una topologia ad albero (uno "spanning tree"), dove esiste un solo percorso attivo tra due punti qualsiasi.

Per fare questo, il protocollo segue tre passaggi fondamentali, come descritto nel PDF. Il primo passo è l'elezione di un "capo". Sai come si chiama?

- root. cerca di andare avanti un po' veloce per favore, non mi servono proprio tuttissimi i passaggi, solo i principali

Ottima domanda. La fase di convergenza è il risultato finale, ma il processo di inizializzazione è dove avviene tutta la "magia". Quando accendi una rete di switch, all'inizio sono tutti uguali e non sanno nulla della topologia. Ognuno di loro pensa di essere il più importante. Per iniziare, ogni switch comincia a inviare dei messaggi speciali chiamati **BPDU (Bridge Protocol Data Units)** da tutte le sue porte. È come se ognuno gridasse: "Sono io il capo! Il mio ID è X!".

A questo punto, cosa succede quando uno switch riceve un BPDU da un vicino che ha un ID "migliore" (più basso) del suo?

1. Continua a gridare di essere lui il capo.
2. Smette di gridare di essere il capo e inizia a inoltrare i messaggi del vicino che ha l'ID migliore.

➤ 2

Esatto, la numero 2 è la mossa giusta. Questo è il cuore del processo di inizializzazione. Appena uno switch sente un BPDU da un vicino "più forte" (con un ID più basso), capisce di aver perso la gara. Smette di proporsi come capo e diventa un semplice messaggero per quello che, al momento, sembra essere il Root Bridge. Questo processo si ripete a cascata in tutta la rete. Ogni switch ascolta i suoi vicini, confronta gli ID e inoltra solo il BPDU del "capo" migliore che ha sentito finora. Alla fine di questa "elezione a catena", quale messaggio pensi che tutti gli switch della rete si staranno scambiando?

➤ Il root

Alla fine di questo processo, tutti gli switch della rete si sono messi d'accordo su chi sia il Root Bridge. A quel punto, tutti inoltreranno solo i suoi BPDU. Questo stabilisce un punto di riferimento comune per tutti. Il passo successivo è usare questo riferimento per costruire la "strada" più efficiente e senza loop. Come fa uno switch a decidere quale delle sue porte offre il percorso migliore per raggiungere il Root? Pensa a OSPF: quale metrica usava per decidere la strada migliore?

➤ meno hop, costo minore

Proprio come OSPF, anche STP assegna un "costo" a ogni collegamento, solitamente basato sulla sua velocità (un link più veloce ha un costo più basso). Uno switch guarda tutte le sue porte e calcola il costo totale per raggiungere il Root Bridge attraverso ciascuna di esse. Il costo totale è la somma del suo costo interno e del costo che gli viene comunicato dal vicino su quella porta. La porta con il costo totale più basso diventa la sua Root Port, ovvero la porta "preferita" per comunicare con il capo. Una volta che ogni switch ha trovato la sua Root Port, manca solo un ultimo passo per evitare i loop. Cosa bisogna decidere su ogni singolo cavo che collega due switch?

➤ l'instradamento?

Vicinissimo! Non si tratta di decidere l'instradamento (quello è il lavoro del router), ma qualcosa di più semplice. Su ogni cavo ci sono due switch. Per evitare un loop, solo uno dei due può essere "responsabile" di quel cavo. Quindi, l'ultimo passo è eleggere su ogni

segmento il **Designated Bridge**, ovvero lo switch che si assume la responsabilità di inoltrare il traffico su quel cavo. L'altro switch, su quel segmento, metterà la sua porta in stato di **blocco**. Così, alla fine del processo:

1. Abbiamo un **Root Bridge** (il capo).
2. Ogni altro switch ha una **Root Port** (la sua via migliore verso il capo).
3. Su ogni cavo, c'è un **Designated Bridge** (il responsabile del cavo), e l'altro switch ha la porta **bloccata**.

Il risultato è una rete pulita, senza loop, ma pronta a riattivare i collegamenti di scorta se qualcosa va storto. Ti torna questo schema finale?

➤ com'è fatto il bpd, segui il pdf

Certo, guardiamo come è fatto un messaggio BPD secondo il documento. È più semplice di quanto sembri, perché contiene esattamente le quattro informazioni chiave che servono agli switch per prendere le loro decisioni. Un BPD è come un biglietto da visita che ogni switch invia ai suoi vicini. Contiene queste quattro informazioni fondamentali:

1. **Root ID:** L'ID dello switch che, secondo il mittente, è il "capo" della rete (il Root Bridge).
2. **Cost to Root:** Il costo totale del percorso dallo switch che sta inviando il BPD fino al Root Bridge.
3. **Sender Bridge ID:** L'ID dello switch che sta inviando fisicamente questo messaggio.
4. **Port ID:** L'ID della porta da cui viene inviato il messaggio.

Ora, immagina di essere uno switch. Ricevi un BPD da un vicino. Quale di questi quattro campi guardi per primo per decidere se il tuo vicino ha una visione "migliore" della rete rispetto alla tua?

➤ Root

Esattamente. Il Root ID è la prima cosa che si guarda, ed è la più importante. La regola è semplice: il Root ID più basso vince. Se ricevi un BPD da un vicino che ti annuncia un Root ID migliore (più basso) di quello che conosci tu, ignora tutte le altre informazioni. Sai già che quel vicino ha una strada migliore verso il "vero" capo della rete. Solo se il Root ID che ricevi è identico al tuo, allora passi al secondo criterio per spareggiare. Quale pensi che sia, guardando gli altri tre campi?

➤ Costo

Certo, costo! È proprio come in OSPF. ****Il costo totale più basso vince.**** Se ricevi un BPDU da un vicino che ti annuncia lo stesso Root ID che conosci tu, ma con un costo inferiore per raggiungerlo, capisci che quel vicino ha una strada più corta. Quindi adotti il suo percorso. E se anche il costo è identico? C'è un ultimo spareggio. Secondo te, cosa si usa per decidere chi dei due vicini (che ti offrono lo stesso capo allo stesso costo) è il "migliore"?

➤ id più basso, anche se non so perché

Hai assolutamente ragione a chiederti "perché". Il ****Sender Bridge ID**** (l'ID di chi ti sta parlando) è lo spareggio finale. Il motivo è puramente pratico: la rete ha bisogno di un modo deterministico e senza ambiguità per prendere una decisione quando tutte le altre condizioni sono pari. Usare l'ID più basso come regola finale garantisce che ci sarà ****sempre e solo un vincitore****. Non ci possono essere pareggi, il che evita situazioni di stallo in cui due switch continuano a discutere all'infinito su chi sia il migliore. È una regola semplice per rompere l'uguaglianza e garantire che il sistema arrivi sempre a una conclusione stabile. Ricapitolando la gerarchia delle decisioni:

1. ****Vince il Root ID più basso.****
2. ***A parità di Root ID*, **vince il costo minore.****
3. ***A parità di costo*, **vince il Sender Bridge ID più basso.****
4. ***Come ultima risorsa (raro)*, si usa il **Port ID più basso**.**

Questo processo garantisce che, non importa quanto complessa sia la rete, STP riuscirà sempre a eleggere un capo e a costruire un percorso senza loop. Ti è chiaro ora come queste quattro informazioni nel BPDU permettono agli switch di mettersi d'accordo?

➤ sì. spiegami meglio la storia dell'age

Certo, analizziamo il concetto di "age". È un'idea molto importante per garantire che la rete non si basi su informazioni vecchie e potenzialmente sbagliate. Pensa al BPDU inviato dal Root Bridge come a un "decreto ufficiale" con una data di emissione. Il campo Message Age nel BPDU è un contatore, misurato in secondi, che tiene traccia di quanto tempo è passato da quando quel decreto è stato emesso. Il Root Bridge invia i suoi BPDU con un Message Age pari a 0. Ogni volta che uno switch riceve questo BPDU e lo inoltra ai suoi vicini, incrementa leggermente questo contatore. Questo ci porta a un altro timer fondamentale: il Max Age. Di solito è impostato a 20 secondi. Se uno switch smette di ricevere BPDU "freschi" sulla sua porta principale (la Root Port) per un tempo superiore al Max Age, cosa pensi che concluda riguardo al suo collegamento con il "capo"?

➤ assume che non sia più disponibile e ricomincia la ricerca della root

Hai assolutamente ragione, è la conclusione logica. Se le informazioni dal "capo" smettono di arrivare, lo switch capisce che il percorso non è più valido. A quel punto, dichiara che le sue vecchie informazioni sono scadute e ricomincia da capo il processo di elezione, cercando un nuovo Root Bridge o un percorso alternativo. Questo meccanismo di "scadenza" (Max Age) è fondamentale perché garantisce che la rete si adatti ai cambiamenti. Se un cavo viene staccato o uno switch si spegne, la rete non rimane bloccata con una mappa vecchia, ma si riconfigura dinamicamente per trovare la nuova topologia migliore. Riesci a vedere come il "Message Age" e il "Max Age" lavorano insieme per mantenere la rete aggiornata e stabile?

- parlarmi della parte successiva. prima fammi un riassunto totale e poi un piccolo quiz (con domande difficili che entrino nei dettagli) per capire se ho capito

Perfetto, facciamo un riepilogo completo e poi ti metto alla prova con qualche domanda per vedere se è tutto chiaro. ### Riassunto Totale: Spanning Tree Protocol (STP)

Lo **Spanning Tree Protocol (STP)** è un protocollo fondamentale che agisce come un "vigile urbano" per le reti Ethernet, con un unico, importantissimo obiettivo: **prevenire i loop di rete**. **Il Problema** 🚫: Nelle reti costruite con collegamenti ridondanti per affidabilità, un messaggio broadcast (come una richiesta ARP) può rimanere intrappolato in un ciclo infinito, generando un "broadcast storm" che consuma tutte le risorse e blocca la rete. **La Soluzione** 🌳: STP analizza la topologia fisica della rete e la trasforma in una topologia logica a forma di "albero" (senza cicli). Lo fa **bloccando temporaneamente** le porte che creano ridondanza. Se il percorso principale si rompe, STP riattiva una delle porte bloccate per garantire la connettività. **Il Processo in 3 Passi**:

- Elezione del Capo (Root Bridge):** Tutti gli switch eleggono un unico "punto di riferimento" chiamato Root Bridge. Vince lo switch con l'ID più basso.
 - Trovare la Strada Migliore (Root Port):** Ogni altro switch calcola il percorso a costo minore per raggiungere il Root Bridge. La porta che si trova su questo percorso viene chiamata Root Port.
 - Gestire i Segmenti (Designated/Blocked Port):** Su ogni singolo cavo che collega due switch, viene eletto un "responsabile" (Designated Bridge). La sua porta rimane attiva (Designated Port), mentre la porta dello switch "perdente" su quel cavo viene bloccata.
- Il Meccanismo (BPDU):** Tutta la comunicazione avviene tramite messaggi speciali chiamati **BPDU**. Ogni BPDU contiene le 4 informazioni chiave per prendere decisioni: **Root ID, Costo per il Root, ID del Mittente e ID della Porta**. La decisione su quale

percorso sia migliore segue una gerarchia precisa: prima si confronta il Root ID, poi il costo, e infine l'ID del mittente come spareggio. * **Mantenere la Rete Aggiornata (Timers):** STP usa dei timer come **Max Age** per assicurarsi che le informazioni non siano obsolete. Se uno switch non riceve BPDU "freschi" entro il tempo limite (di solito 20 secondi), capisce che il percorso non è più valido e avvia un ricalcolo della topologia.

Quiz di Verifica

Ora vediamo se hai afferrato i dettagli. Queste domande sono pensate per essere un po' insidiose.

Domanda 1:

Due switch, **Switch A** (ID: 32768.**AA**) e **Switch B** (ID: 32768.**BB**), sono collegati alla tua porta. Entrambi ti inviano un BPDU annunciando lo **stesso Root Bridge** (ID: 32768.11) con lo **stesso identico costo** per raggiungerlo. Quale dei due percorsi sceglierai come tuo percorso preferito verso il Root e perché?

Domanda 2:

Il tuo switch ha una Root Port attiva e sta ricevendo regolarmente BPDU con un **Message Age** di 5 secondi. Improvvisamente, il cavo collegato a quella porta viene danneggiato e smetti di ricevere BPDU. Il timer **Max Age** è impostato a 20 secondi. Cosa farà esattamente il tuo switch e dopo quanto tempo?

Domanda 3:

Stai analizzando un BPDU ricevuto da un vicino. La tua attuale visione della rete è che il Root Bridge ha ID `32768.11` e il costo per raggiungerlo è `10`. Il BPDU che ricevi dal vicino annuncia un Root Bridge con ID `32768.22` ma con un costo incredibilmente basso di `2`. Prenderai in considerazione il costo più basso del vicino o c'è qualcos'altro che guardi prima e che ti fa prendere una decisione diversa?

- 1. switch A per ordine lessicografico
- 2. dopo 20 secondi smette di fare forwarding e ricomincia la costruzione dell'st
- 3. no, lascio la root con l'id più basso