STP · יתירות · RSTP · bridge ID · BPDU והגנות

שבוע 18 · 21/01–17/01 · **פלנר עדי**

שני כבלים בין שני מתגים. נכון — כי אם אחד נקרע, השני מחזיק. ואסון — כי מסגרת `broadcast` תסתובב ביניהם לנצח (אין `TTL` בשכבה 2). `STP` פותר את הסתירה: חוסם פורט אחד, ופותח אותו כשהאחר נופל. שלוש שאלות בחינה בפרק הזה.

1. הבעיה — לולאה בשכבה 2

```
PC-A ── SW1 ═══ SW2 ── PC-B          two links, no STP:
            ═══
1. PC-A sends a broadcast (ARP)
2. SW1 floods it out BOTH links to SW2
3. SW2 floods each copy out the OTHER link back to SW1
4. SW1 floods again… forever. no TTL at layer 2.
-> broadcast storm: 100% CPU, links saturated
-> MAC table instability: PC-A's MAC flips between two ports every microsecond
```

שלוש תוצאות: סופת שידור, טבלת `MAC` לא יציבה, עותקים כפולים. ובכל זאת — יתירות היא דרישה. לכן `STP`.

2. הפתרון — עץ בארבע בחירות

```
BID (bridge ID) = priority (default 32768) + MAC address     -> LOWEST wins

1. ROOT BRIDGE      the switch with the lowest BID. all priorities equal? lowest MAC.  (exam 2023)
2. ROOT PORT        on every OTHER switch: the one port with the cheapest path to root
                    cost: 10 Gbps=2 · 1 Gbps=4 · 100 Mbps=19 · 10 Mbps=100
3. DESIGNATED PORT  on every segment: the one port closest to root (root's ports are all DP)
4. BLOCKED          everything else. receives BPDUs, sends nothing

BPDU: the hello (every 2 s) that carries the root's BID and the path cost
```

`root` → `root port` → `designated` → `blocked`. תמיד בסדר הזה. ל-`root` עצמו אין `.root port`

3. דוגמה — שלושה מתגים

```
SW1  32768 / 0001.0001.0001      SW2  32768 / 0002.0002.0002      SW3  32768 / 0003.0003.0003
       |                                 |                                 |
      SW1 ─────(1G)───── SW2
        \                 /
        (1G)            (1G)
           \            /
                SW3

root      = SW1 (lowest MAC, all priorities equal)
SW2 RP    = port to SW1 (cost 4)          SW3 RP = port to SW1 (cost 4)
SW2–SW3 segment: DP = SW2's port (SW2 has lower BID) · SW3's port = BLOCKED
SW1's ports: all DP
```

הפורט החסום: בצד של המתג עם ה-`BID` הגבוה יותר על המקטע שאינו לכיוון ה-`root`.

4. מצבי הפורט וזמנים — ומה `RSTP` שינה

802.1D (`STP`)	זמן	מה קורה	802.1W (`RSTP`)
`blocking`	20 שנ׳ (`max age`)	מקבל `BPDU`, לא שולח, לא לומד	`discarding`
`listening`	15 שנ׳	שולח/מקבל `BPDU`, לא לומד `MAC`	`discarding`
`learning`	15 שנ׳	לומד `MAC`, לא מעביר	`learning`
`forwarding`	—	מעביר נתונים (שאלת בחינה 2026)	`forwarding`
`disabled`	—	כבוי	`discarding`

30–50 שניות עד `forwarding` ב-`STP` — ולכן מחשב 'מחכה ל-`DHCP`'. `RSTP` (802.1W): התכנסות ב-1–2 שניות, `alternate port` מוכן מראש. **`PVST/Rapid PVST`** (`Cisco`): עץ נפרד לכל `VLAN` — `VLAN 10` חוסם קישור א׳, `VLAN 20` חוסם ב׳ — איזון.

> **טיפ** · `Cisco` ברירת מחדל: `PVST`+. `spanning-tree mode rapid-pvst` — תמיד.

5. פקודות — `root` ו-הגנות

```
SW1(config)# spanning-tree vlan 1 root primary        <- priority 24576 (or 4096 below current root)
SW2(config)# spanning-tree vlan 1 root secondary      <- priority 28672
SW1(config)# spanning-tree vlan 1 priority 4096       <- manual (multiples of 4096)
SW1(config)# spanning-tree mode rapid-pvst

! access ports (PCs) — skip listening/learning, and shut if a switch shows up
SW1(config)# interface range fa0/1-24
SW1(config-if-range)# spanning-tree portfast              exam 2026
SW1(config-if-range)# spanning-tree bpduguard enable
! or globally: spanning-tree portfast default · spanning-tree portfast bpduguard default

! port toward a foreign switch — it may never become root port
SW1(config-if)# spanning-tree guard root

SW1# show spanning-tree
VLAN0001
  Root ID    Priority 24577   Address 0001.0001.0001   This bridge is the root
  Bridge ID  Priority 24577 (priority 24576 sys-id-ext 1)
Interface   Role Sts Cost  Prio.Nbr Type
Gi0/1       Desg FWD 4     128.25   P2p
Gi0/2       Desg FWD 4     128.26   P2p
```

`root primary` לפני שמתג ישן עם `MAC` נמוכה תופס. `portfast` + `bpduguard` על כל פורט מחשב.
`root guard` לכיוון מתגים שלא שלכם.

⚠ **שימו לב ·** `portfast` על `trunk` = לולאה זמנית. רק פורטי גישה.

דוגמה פתורה

השאלה

ארבעה מתגים בריבוע, כולם `priority 32768. MAC: SW1` ...0A, `SW2` ...03, `SW3` ...0F, `SW4` ...07. כל הקישורים 1G. מי `root` , איזה פורט חסום, ומה הפקודה שתהפוך את `SW1` (המתג החדש והחזק) ל- `root` ?

הפתרון צעד אחר צעד

1. **`root`** : כל ה- `priority` זהים → `MAC` נמוכה → `SW2` (...03).
2. **`root ports`** : `SW1` → הפורט ל- `SW2` (`cost 4`). `SW3` → הפורט ל- `SW2` (4). `SW4` : שני מסלולים ב- `cost 8` (דרך `SW1` או דרך `SW3`) — תיקו → `sender BID` נמוך: `SW1` (...0F... < ...0A)? לא — `SW3` ,0F... `SW1` → 0A... `SW1` נמוך → `RP` של `SW4` = לכיוון `SW1` .
3. **`designated`** : כל הפורטים של `SW2` · `SW1` – `SW4` : `DP` ב- `SW1` (קרוב ל- `root`) · `SW3` – `SW4` : `DP` ב- `SW3` (`cost 4` מול 8 של `SW4`).
4. **`blocked`** : הפורט של `SW4` לכיוון `SW3` .
5. **הפקודה**: `SW1` (`config`)# `spanning-tree vlan 1 root primary` → `priority 24576` > 32768 → `SW1 root.` העץ מתחשב מחדש.
6. **אימות**: `show spanning-tree` ב- `SW1` → ' `This bridge is the root` '.

תשובה · `root` = `MAC` נמוכה כשה- `priority` שווה — בדיוק המתג שלא רצינו. `root primary` מתקן. (בחינה 2023 ז2 + ו3)

תרגילים — פתרו לבד לפני שאתם בודקים בכיתה

1. כל ה- `priority 32768.` מי `root` ?
2. הפקודה שהופכת מתג ל- `root` ?
3. `STP` מונע לולאות איך?
4. `portfast` — למה ולאיזה פורט?
5. איזה מצב מעביר נתונים?
6. חמשת מצבי 802.1D בסדר?
7. `RSTP` — מה משתנה?
8. `bpduguard` — מה עושה?

9. `root guard` — מה עושה?

10. `cost` של 1 `Gbps` ? 100 `Mbps` ?

דף עבודה — `STP` — הבעיה, העץ, המצבים, הפקודות

שם: _____ תאריך: _____ חוליה: _____

המטרה: לבחור `root` ופורטים על טופולוגיה נתונה, ולכתוב 5 פקודות `STP` עם משפט לכל אחת.

מהלך — סמנו כל שלב שהושלם

- ☐ תרגיל `docx` : הטופולוגיה הנתונה — `root` ? `root port` בכל מתג? `designated` ? `blocked` ? נמקו כל בחירה.
- ☐ שנו 'המנהל רוצה ש- `SW-CORE` יהיה `root` ' — הפקודה. חשבו מחדש: מה משתנה?
- ☐ טבלת המצבים מהזיכרון: 5 מצבים, זמנים, מה קורה בכל אחד. ליד כל אחד — המקבילה ב- `RSTP` .
- ☐ 3 תרחישים → ההגנה: (א) מחשב מחכה דקה ל- `DHCP` (ב) מישהו חיבר מתג ביתי לשקע (ג) מתג של קבלן מחובר לרשת שלנו.
- ☐ כתבו 5 פקודות: `root primary` · `mode rapid-pvst` · `portfast` · `bpduguard` · `show spanning-tree` — משפט לכל אחת.
- ☐ 4 שאלות בחינה: 2023 ו2 · ז2 · ו3 · 16 2026 — ענו ונמקו.

רישום התוצאות

#	מה למלא	חישוב · הערות · ראיות (פלט פקודה)
1	`root` _____ · `RP` : _____ · `DP` : _____ · `blocked` : _____	_____
2	פקודה: _____ · מה משתנה: _____	_____
3	טבלה: _____	_____
4	(א) _____ (ב) _____ (ג) _____	_____
5	5 פקודות: _____	_____
6	בחינה: _____	_____

רפלקציה — שתי שורות

מה היה הכי קשה, ומה פתר את זה?

מה הייתי עושה אחרת בפעם הבאה?

אישור המורה: ☐ שאלון היחידה ב-Moodle (100) ☐ שיעורי הבית (30) · **XP** _____ · תג: ☐ Root ☐ Bridge · חתימה _____

כרטיס יציאה — 3 שאלות, 5 דקות, נמסר ביציאה

1. `BID` = ?

---

2. סדר הבחירות?

---

3. מצב שמעביר?

עוד לבדיקה עצמית — ענו בעל פה, בלי להסתכל אחורה

1. `portfast` על `trunk`?

מילון מונחים

עברית	אנגלית	הסבר קצר
פרוטוקול עץ פורש	`STP · 802.1D`	מונע לולאות בשכבה 2 על ידי חסימת פורטים
מזהה גשר	`Bridge ID (BID)`	`priority` + `MAC` — הנמוך הופך `root`
גשר שורש	`Root Bridge`	המתג שממנו נמדדים כל המסלולים
פורט שורש	`Root Port`	הפורט של המתג לכיוון ה-`root` — אחד למתג
פורט מיועד	`Designated Port`	הפורט של המקטע לכיוון ה-`root` — אחד למקטע
BPDU	`Bridge Protocol Data Unit`	הודעת `STP` — כל 2 שניות
מהיר STP	`RSTP · 802.1W`	התכנסות בשניות, 3 מצבים
מהיר לפורט	`PortFast`	דילוג על `listening/learning` בפורט גישה
שומר BPDU	`BPDU Guard`	כיבוי פורט גישה שקיבל `BPDU`
שומר שורש	`Root Guard`	פורט שלא יהפוך `root port`

להעמקה

בספר הקורס

- **SRWE 5.3.4** — PortFast and BPDU Guard · עמוד 165
- **SRWE 5.2.2** — 1. Elect the Root Bridge · עמוד 150
- **SRWE 5.1.5** — Layer 2 Loops · עמוד 142

קבצים שנעבוד איתם

- SRWE_Module_5.pptx
- Spanning Tree Protocol.docx
- תרגיל.docx
- STP.pkt
- stp.pkt
- 5.1.9 Packet Tracer - Investigate STP Loop Prevention.pka