

Communication Networks (0368-3030) / Spring 2011

The Blavatnik School of Computer Science,
Tel-Aviv University

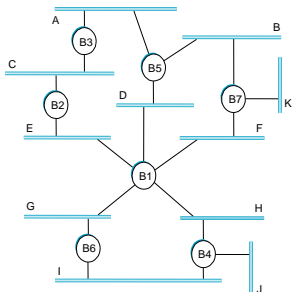
Allon Wagner

More Link Layer Switching & STP

Exercise (Peterson & Davie 5th ed.)

What happens in the spanning tree algorithm if bridge B1 does not participate and:

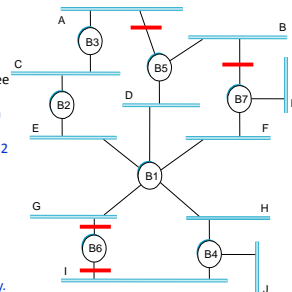
- a) Simply forwards all spanning tree algorithm messages?



Exercise (Peterson & Davie 5th ed.)

What happens in the spanning tree algorithm if bridge B1 does not participate and:

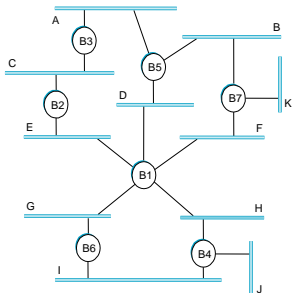
- a) Simply forwards all spanning tree algorithm messages?
- D, E, F, G would be considered a single LAN segment.
 - The resulting tree would have B2 as its root.
 - Blocked ports:
 - B5 to A (B5 has a direct link to B2)
 - B7 to B
 - B6 on either side
 - If B1 also forwards all data frames, network works correctly.



Exercise (Peterson & Davie 5th ed.)

What happens in the spanning tree algorithm if bridge B1 does not participate and:

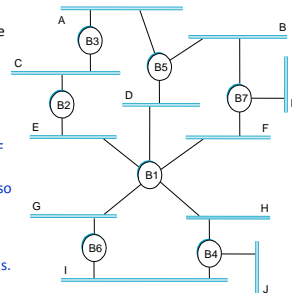
- b) Drops all spanning tree messages?



Exercise (Peterson & Davie 5th ed.)

What happens in the spanning tree algorithm if bridge B1 does not participate and:

- b) Drops all spanning tree messages?
- Then the LAN is actually disconnected into two parts: A-F and G-H.
 - Neither part has a redundancy, so STP blocks no ports.
 - If B1 still forwards non-STP messages, then we still have loops! The network malfunctions.



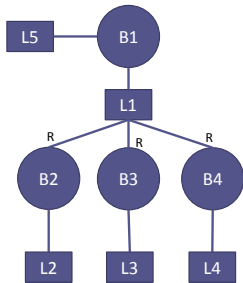
שאלה ממבחן – מועד א' 2010/11

- השאלה הבאה מתייחסת ל- STP ברשת שכבה 2, כלומר אוסף של LANs המקושרים ע"י גשרים (Bridges) למערכת קשירה (Connected).
 - הסבירו את מטרתו של STP, וציינו בעיית תקשורת חשובה שהוא בא למנוע
 - STP נועד לפתור את הבעיות שעלולות להווצר כאשר אלגוריתם ה- self-learning של switches מופעל ברשת שיש בה מעגלים.
 - מצד שני, אנחנו רוצים לאפשר מעגלים פיזיים (יתירות ברשת). הפתרון הוא להריץ אלגוריתם שינתק לוגית קישורים מסויימים באופן שישאיר את הרשת קשירה וחסרת מעגלים (כלומר, עץ).
 - האלגוריתם מבוזר לגמרי, ופועל כ- plug & play בהתאם לעקרונות של switches שכבר ראינו.

שאלה ממבחן – מועד א' 2010/11

- אם על LAN מסוים ברשת יושבים K פורטים של גשרים (Bridges), כמה מהם יכולים להיות Designated Ports ברגע מסוים, לאחר שכל הגשרים הישימים קיבלו את הודעות התצורה מעמיתיהם שעל LAN זה? האם כולם יודעים מי הוא?
 - לתשובות החיוביות נמקו מדוע
 - לתשובות השליליות תמ דוגמה
- כל הגשרים המחוברים ל- LAN מקבלים את אותן הודעות וביניהן יש יחס סדר מלא.
- כל גשר שומר את ההודעה הכי טובה שקיבל על הפורט, ואת ההודעה שהוא עצמו שלח על הפורט. אם הוא זה ששלח את ההודעה הכי טובה – הוא מחליט שהפורט הוא designated.
- לפיכך, לאחר שכל הגשרים החליפו את ההודעות שלהם ייתכן בדיוק designated port אחד, וכולם יודעים מיהו.

שאלה ממבחן – מועד א' 2010/11



- עבור ה- LAN שנידון בחלק (ב), כמה מהפורטים הנדונים יכולים להיות Root Ports?
 - אם התשובה היא: "תמיד 1", נמקו מדוע
 - אם התשובה היא: "יתכן שיותר מאחד", תמ דוגמה
- ייתכן שיותר מאחד.
- למשל, ניקח רשת שהיא כבר עץ. כל הפורטים שמחברים את הרמה השנייה לכיוון השרשר הם root ports.

שאלה ממבחן – מועד א' 2010/11

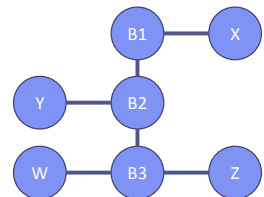
- הוכיחו שאחרי התנסות STP, Designated Port של LAN אינו יכול להיות Root Port של הגשר שלו.
- בסעיף א' הסברנו שמבין כל הפורטים המחוברים ל- LAN, בדיוק אחד הוא – designated port: נתבונן בגשר B ובפורט P שלו אשר הוא ה- designated port של ה- LAN.
- נשים לב ש- B שולח את אותה הודעה על כל הפורטים שלו. נסמנה M.
- B שלח את ההודעה הכי טובה על ה- LAN הזה (אחרת, הוא לא היה מחליט שהוא ה- designated).
- נתבונן כעת בהודעה M' הכי טובה ש- B קיבל על כל הפורטים שלו:
 - M טובה יותר מ- M' אזי B מחליט שהוא עצמו ה- root ולכן אין לו בכלל root port.
 - סיימנו.
- אחרת, M' טובה יותר מ- M והפורט עליו התקבלה הוא ה- root port. פורט זה אינו יכול להיות (כי עליו B הוא זה ששלח את ההודעה הכי טובה). סיימנו.

שאלה ממבחן – מועד א' 2010/11

- השתמשו ב- (ד) להראות כי מסגרת הנשלחת מאחד ה- LANs במערכת עם כתובת יעד לא ידועה לגשרים במערכת תגיע בין השאר אל ה- Root.
- אבחנה: כאשר האלגוריתם מתייבב, כל הגשרים מכירים את אותו השרשר. ה- root port של כל גשר B מחובר אותו ל- LAN אליו מחובר גם גשר B', כך שהמרחק של B' מהשרשר קטן ממש מהמרחק של B אל השרשר.
- כשה- host שולח מסגרת עם יעד לא ידוע, הגשר שהוא designated על ה- LAN (נסמנו B) אוסף אותו ועושה לו flooding. קרי, B שולח אותו על כל הפורטים שאינם זה שקיבל את המסגרת. מכאן, ש- B שולח אותו גם דרך ה- root port שלו (שאינו יכול להיות זה שדרונו התקבלה המסגרת לפי הסעיף הקודם). נניח שכך הוא מגיע ל- LAN L.
- יהא B' הגשר ששלח את ההודעה הטובה ביותר על L (במילים אחרות, הוא זה שגרם ל- B להחליט שה- root port שלו הוא זה שיוצא ל- L). הפורט של B' ל- L הוא בהכרח designated (מדוע?), ולכן אוסף את המסגרת. חזרו למצב ההתחלה.
- נמשיך ככה באינדוקציה. בכל פעם המסגרת מגיעה לגשר בעל מרחק קטן ממש מהשרשר מאשר הגשר הקודם, ולכן בסוף נגיע למרחק 0, כלומר לשרשר.

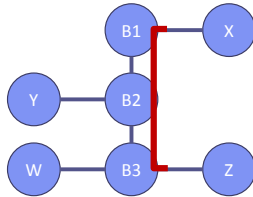
Exercise (Peterson & Davie 5th ed.)

- W, X, Y, Z – hosts. B1, B2, B3 – bridges
- All switch tables are initially empty.
- X sends a message to W. Which bridges learn where X is? Does Y's network interface see this packet?
- The frame is flooded. All the bridges see it, and so does Y's interface.



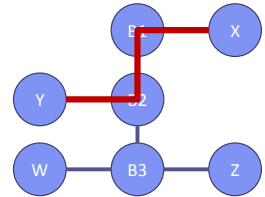
Exercise (Peterson & Davie 5th ed.)

- W, X, Y, Z – hosts. B1, B2, B3 - bridges
- All switch tables are initially empty.
- Now, Z sends a message to X. Which bridges learn where Z is? Does Y's network interface see this packet?
- All the bridges know where X is, so the frame is simply forwarded from each bridge on its link to X.
- All bridges learn where Z is.
- Y doesn't see the frame.



Exercise (Peterson & Davie 5th ed.)

- W, X, Y, Z – hosts. B1, B2, B3 - bridges
- All switch tables are initially empty.
- Y sends a message to X. Which bridges learn where Y is? Does Z's network interface see this packet?
- B1, B2 already know where X is so they are forwarding the frame properly. They learn where Y is.
- Z doesn't see the frame.



Exercise (Peterson & Davie 5th ed.)

- W, X, Y, Z – hosts. B1, B2, B3 - bridges
- All switch tables are initially empty.
- W sends a message to Y. Which bridges learn where W is? Does Z's network interface see this packet?
- B3 doesn't know where Y is, so the frame is flooded by B3. Thus, Z sees the frame.
- When it arrives to B2, who knows where Y is, it is simply forwarded on the link to Y.
- B3 and B2 have learned where W is.

