מבוא לרשתות תקשורת | 67594

הרצאות | פרופ' מיכאל שפירא

כתיבה | נמרוד רק

תשפ"ג סמסטר א'

תוכן העניינים

שבוע $\mathbb{I}$ | מבוא

הרצאה

נעסוק בעקרונות מרכזיים ברשתות תקשורת, ובראשם האינטרנט ואתגרים שיש להתמודד איתם בקשר אליו. האינטרט מאוד מורכב ואי אפשר להחליף אותו, אפשר לכל היותר לבצע בו שינויים שקשה מאוד לעשות והם איטיים.

אפליקציות באינטרנט

דוגמה הרשת משרתת מטרה כלשהי למשתמש כלשהו, לדוגמה סטרימינג של סרט. כיום סטרימינג קורה ע"י בקשה של הלקוח מהשרת של כמה שניות של סרט בכל פעם, באופן אינקרמנטלי. כך, אם הרשת לא יכולה להעביר (בין אם יש צוואר בקבוק אצל המשתמש או בכל מום אחר) מספיק סרט מספיק מהר, נקבל תקיעות ונצטרך להוריד רזולוציה כדי לקבל חוויה יותר טובה.

אם ידוע שיש בעיה ברשת, חלק מהשירותים שמספקים ווידאו בלייב שומרים באפר של כמה שניות מהשידור האמיתי כדי שאם יש תקיעה פתאומית באינטרנט החוויה תראה עדיין אחידה.

סטרימינג היא דוגמה לאפליקציה מעל הרשת - שימוש בפרוטוקולים שבקרוב נלמד. חווית הצפיה שתוארה למעלה נקראת Quality of Experience ואלג' הסטרימינג שהצגנו נקרא Dynamic Streaming over HTTP. העניין של QoE הוא בעייתי מאוד כי כולם מתחרים על אותו רוחב פס ושירותים רבים דורשים latency נמוך (לדוגמה שיחת זום, איכות יכולה להיות נמוכה אבל חייב להגיע מהר) לעומת throughput (כשקיבולת הרבה יותר חשובה מהזמן שלוקח שזה יגיע).

האינטרנט בעצמו הוא די פרוץ והרבה זמן מנסים לשפר את האבטחה של הרשת אבל זה מאוד קשה כי יש הרבה שחקנים והרשת מבוזרת ולהגיע להסכמה זה כמעט בלתי אפשרי.

בקרוב נדבר על מודל 7 השכבות של אבל לעתה נכווץ אותו ל-3 שכבות: החומרה (ראוטרים, מחשבים, סוויצ'ים וכו'), פרוטוקולים (הדרך שבה מדברות החומרות ביניהם, פרוטוקלים כמו TCP, BGP וכו') ואפליקציות (אתרים, אפליקציות וכו'). השכבה העליונה והתחתונה משתנות כל הזמן ומשתפרות בקצב תדיר, אבל השכבה האמצעית נשארת מאחורה כבר מאז שנות ה-90.

מהירות מעבר מידע

הרבה מהמידע באינטרנט עובר במהירות האור (לדוגמה בין יבשות) ולכן לכאורה אפשר לשלוח מידע מירושלים לניו יורק במהירות האור. הבעיה עם זה היא שראשית החומרה עצמה מאפשרת $\frac{2}{3}$ מהירות האור, רוחב הפס יכול להיות מוגבל, מספר תחנות הביניים בין היעדים יכול להיות משמעותי ויכול להיות שנלחם על מקום עם פקטות אחרות, לכן במקום ה-$30ms$ ההיפותטי של מהירות האור, זה לוקח כמעט פי 2.

בזמן 70 המילישניות האלה שנדמה שהן כלום זמן, מעבד ממוצע מספיק לעשות מיליוני סייקלים ולכן מבחינתו האינטרנט הוא סופר איטי. הזמן הזה הוא קריטי כשמדובר במניות, או דברים שדורשים תיאום בין אנשים שונים, וזו בעיה כי האינטרנט בפועל מיתרגם לתקשורת א-סינכרונית.

האינטרנט ועניינים נוספים

הרשת מחלוקת לשלושה משתתפים - קודקודים (ראוטרים, סוויצ'ים), קשרים (סיב אופטי וכו') ומשתמשי קצה (מחשב, טלפון אבל גם מקרר, טלוויזיה וכו').

כיום מספר המשתמשים ברשת גדל אקספוננציאלית, בגלל שלכל בן אדם יש הרבה יותר ממחשב אחד או טלפון אחד, אלא מקרר ומיקרו וטלוויזיה וכו'. יש הרבה סוגים שונים של משתמשי קצה, המון סוגים של קודקודים וקישורים, וגיוון גדול מאוד בדרישות אפליקציות (האם צריך דו-כיווניות, האם צריך מהירות, האם צריך אמינות וכו' וכו').

הרשת היא התשתית להעברת מידע בין משתמשים, ורק בה נעסוק. מעליה אפשר לבנות רשתות מבוזרות שונות שהן ברמת האפליקציה וזה א מה שיעניין אותנו.

רשת הטלפוניה

כדי להבין איך האינטרנט עובד, נבין קודם איך רשת הטלפוניה עבדה.

כשרוצים לדבר עם מישהו, קודם כל נרצה לדעת האם אפשר להגיע אליו באמצעות circuit, מסלול שהוא רק שלנו ושאי אפשר לעצור אותו באמצע. אחרי שיש קו, נעביר מידע על הקו עד שתסתיים השיחה. בסוף השיחה, נפנה את המשאבים.

המרכזן הוא זה שיוצר את הקו ומבצע switching - ניתוב של תקשורת שמגיעה מכניסה מסוימת ליציאה מסוימת. כיום טלפוניה מנותבת אוטומטית באמצעות אלג' מורכבים.

כיצד נוכל לנתב שתי כניסות לאותה היציאה?

- Time-Division - נחלק את הזמן לשברירים מחזוריים ונבצע round robbin על רוחב הפס.
- Frequency-Division - נחלק את הפס לתדרים שונים, כאשר כל תדר אחראי על משתמש אחד.

למה האינטרנט לא נראה כמו הטלפוניה

מה קורה אם אם משתמש לא משתמש בסלוט שלו (שותק בשיחה)? בטלפונים זה עבד בסדר, אבל באינטרנט זה לא פרקטי.

הרשת שתיארנו זה עתה היא מאוד יציבה, פשוטה, מהירה (העיכוב הוא רק המרחק הפיזי על הפס), צפויה, קל לגלות איפה יש בעיה.

אז למה לא להשתמש בה גם באינטרנט?

- אי-עמידות בפני כישלון: אם אין רוחב פס או אם יש בעיה באמצע, נוותר ולא נקבל שום תקשורת.
- בזבוז רוחב פס: טלפוניה לא שורדת תחת עומס כי רוחב הפס נגמר מתישהו ובאינטרנט זה לא מתקבל על הדעת. בטלפניה נשמור את רוחב הפס המקסימלי הנדרש לקוים הפתוחים עליה (P), לעומת האינטרנט שבו התקשורת המקסימלית אולי גבוהה אבל ממוצע הפקטות שמועבר (A) משמעותית יותר נמוך ולכן לא צריך לשמור את כל רוחב הפס המקסימלי כל הזמן.

  בטלפניה היחס $\frac{P}{A}$ הוא $1:3$. באינטרנט היחס הוא יותר מ-$1:100$.
- עיצוב מותאם אפליקציה: טלפוניה עוצבה לטלפונים, בניגוד לאינטרנט שנועד לאפליקציות מסוג אחר.
- זמן הכנה: בטלפוניה לוקח זמן לא זניח ליצור את הקו, בעוד באינטרנט צריך שזה יהיה (כמעט) מידי.

כדי להתגבר על כל הבעיות האלה, ב-64' הוצע השימוש בפקטות במקום קו שמור, ביזור הרשת והעברת הפקטות דרך הקודקודים ברשת ובהמשך גם ניתוחים סטיסטיים של מערכות כאלה.

רשתות ברודקאסט

ברשת ברודקאסט כל המידע מועברה לכל קודקוד וכולם חייבים להקשיב לפקטות. זה קורה ב-WiFi ולרוב בשאר LAN-ים (Local Area Networks), ומבחינה אינטואיטיבית גם בהרצאה.

יש כמה בעיות בשיטה זו, ביניהן:

- קשה להעביר את הפקטות מרחקים גאוגרפיים גדולים;
- קשה לתאם גישה של כמה משתמשים לרשת (בהרצאה רק אחד מדבר, אבל מה אם שני סטודנטים רוצים לדבר באותו הזמן?), זו בעיית ה-Multiple Access Problem.
- אין פרטיות בתקשורת (למרות שאפשר להצפין את המידע).

שבוע III | האינטרנט ועקרונות יסוד

הרצאה

רשתות switched

בניגוד לרשתות ברודקאסט, רשתות שהן switched מאפשרות לרבים לחלוק את אותו המשאב ולאנשים שונים לדבר עם אנשים אחרים באותו הזמן. מתחת לסיווג הזה יש עוד שני סוגי רשתות:

- circuit-switched - נקצה משאבים לקיבולת מקסימלית לאורך כל השיחה.
- packet-switched - נעביר פקטות בהתאם לכמה מידע נרצה להעביר ולא נקצה מראש את כל המשאבים שנדרשים.

פקטות

ברגע שלפקטה לא מוקצים משאבים, מסלול וכו', היא חייבת להכיל עליה את המידע שמספר לרשת מאיפה ולאן היא צריכה להגיע - כי אף אחד אחר לא יעשה את זה בשבילה.

הפקטה מכילה את גוף הפקטה (payload), שהוא המידע שאנחנו רוצים להעביר, ו-header שמכיל מידע שנועד רק לרשת (כמו פרטים על מעטפה).

כל פקטה מנתובת באופן עצמי ולכן היא יכולה לעבור מסלול שונה גם אם יצאה מיד אחרי פקטה אחרת. נצטרך לדאוג שהעובדה הזו לא תהפוך תקשורת לבלתי אפשרית ואיכותית.

כשפקטה יוצאת, היא עוברת דרך כמה תחנות ביניים וכל תחנת ביניים לכאורה מתעלמת מהגוף (אלא במקרים של אבטחה שלא נעסוק בהם), ומעבירה הלאה את הפקטה לתחנה הבאה. לקודקוד יש שיקול דעת לאן לנתב או לא לנתב פקטה (ואז היא תיפול). הרעיון הזה של שליחי ביניים נקרא store-and-forward - הקודקוד שומר את המידע וכשיכול (או לא), מעביר את המידע הלאה.

הערה בכל תחנת ביניים נוסף עוד דיליי של זמן עיבוד בקודקוד (וזה יכול להצטבר במרחקים ארוכים או קודקודים חלשים).

פרצים

מהתיאור הנ"ל, פקטות הן פתרון לא מוצלח ולא יציב בלי הבטחה של ביצועים כלשהם. אם כן, למה זה עדיין חשוב לנו?

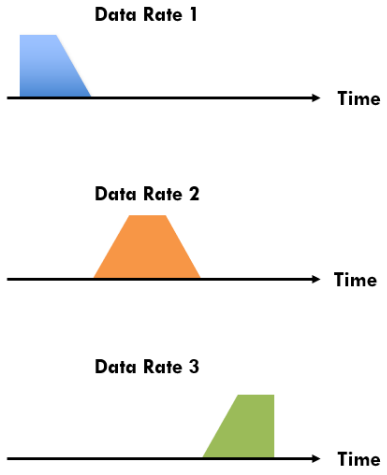
תקשורת באינטרנט הרבה פעמים היא בפרצים, כלומר דממה להרבה זמן ואז הרבה מידע יותר מרוחב הפס האפשרי, לדוגמה חיפוש בגוגל - נחפש, נסתכל על התוצאות, ואז שוב נחפש - וקבלת המידע מגוגל היא הפרץ. לעומת זאת, סטרימינג מגיע באופן יותר רציף מעל האינטרנט.

כיצד נטפל בפרצים? פתרון אחד הוא פשוט להפיל כל פקטה שאנחנו לא יכולים להעביר מיד. לחלופין, נוכל להוסיף באפרים ששומרים פקטות שמחכות לעבור. עם זאת, תמיד יהיו לנו יותר פקטות מאשר מקום בבאפר, ולכן תמיד נצטרך להפיל חלק מהפקטות.

דוגמה נניח שיש לנו שלושה פרצי מידע כבאיור (כחלק משיחה שלוקחת את כל ציר הזמן), כאשר במקסימום כל שיחה דורשת קצת יותר משליש מרוחב הפס.

אם היינו בטלפוניה, היינו מקצים לראשון ולשני מקום והשלישי היה נזרק בחוץ, והיינו מבזבזים בכל רגע נתון שני שליש מרוחב הפס.

עם זאת, היו מספקים משאבים לכולם כי הם לא רצו לדבר במקביל. ההנחה כאן היא שעומס התקשורת בא והולך אבל מתפלג באיזו שהיא צורה לא מתאומת שמאפשרת מעבר של רוב התקשורת בעומס סביר.



איור 1: דוגמה לתקשורת עם פרצים

Statistical Multiplexing

Statistical Multiplexing הוא הרעיון שלפיו נוכל לשרת משתמשים שונים על אותה התשתית גם אם רשמית אין לכולם מקום ביחד (לא כולם יכולים לחפש באותה השנייה בגוגל משהו, אבל בגלל שזה קורה בזמנים שונים במקומות שונים אנחנו כן מצליחים). גם ביטוח והלוואות עובדים ככה - הבנק לווה כסף בידיעה שלא כולם הולכים למשוך את הכסף שלהם ולכן מרשה לעצמו "להמציא" כסף, או בביטוח, שבו אם כל אחד היה בתאונת דרכים הביטוח היה בבעיה, אבל סטטיסטית זה לא סביר.

אפשר להסתכל על Stat Mux גם בתור חלוקה של הזמן לפריימים (שמחולקים אף הם לסלוטים). בכל פריים נייצר לכל היותר P פקטות ובממוצע נייצר A פקטות. אם P גדול מ-A משמעותית זה לא חכם להקצות משאבים לשיחה ספציפית כמו שהזכרנו כבר הרבה פעמים.

עם זאת, חוק המספרים הגדולים אומר לנו שסביר שערכים מהתפלגות יהיו קרובים לתוחלת, יותר מאשר לערך הכי גדול שאפשר - נקבל הרבה יותר התנהגות דומה ל-A פקטות לפריים מאשר P פקטות לפריים.

השוואה בין רשתות מנותבות Circuit ל-Packet

לרשת circuit-switched יש כמה יתרונות וחסרונות, ביניהם

יתרונות	חסרונות
רוחב פס מובטח	רוחב פס מבוזבז בפרצים
אבסטרקציה נוחה	חיבורים חסומים כשאין משאבים
מסלול העברת מידע פשוט	קודקודים חייבים להיות מודעים לשיחות שעוברות דרכם
תקורה נמוכה פר-פקטה	דיליי בהתחלת השיחה

החסרונות האלה הם קריטיים מדי לשירותים שלנו באינטרנט שבגללה אנחנו לא משתמשים בטלפוניה לאינטרנט.

קריטריון	packet	circuit
אמינות	אם יש כשל הרשת תחשב מסלול מחדש	אם הקו נופל לא נשחזר אותו
יעילות	ניצול של Stat. Mux לרוחב פס יעיל יותר	רוחב פס מוגבל ע"י הקוים שהוקצו להם משאבים
קלות מימוש	רשתות יכולות להתחבר בקלות כי הן לא מקצות אחת לשנייה משאבים	חיבור שתי מרכזיות דורש הקצאת משאבים מכל צד
טיפול בעומסים	צריך להתמודד עם עומסים (לזרוק פקטות, להוריד את הקצב,...)	לא רלוונטי

מודולריות

לחלק את הקוד לאבסטרקציות ומודולריות זה חשוב מכל מיני סיבות. בין היתר, אפשר לשנות את המימוש למימוש שונה/יותר יעיל בלי שאחרים ידעו. בנוסף, אפשר להוסיף עוד פונקציונאליות לקוד עם מודולים חדשים נפרדים, וזה בעיקר מאפשר להמשיך לשדרג ולהתקדם בחזיתות שונות באותו הזמן.

ברשתות מודולריות מקבלת נדבך נוסף - המימוש הוא מבוזר בין הרבה מכונות. לכן הושמו כמה עקרונות כדי לפתור בעיות כאלה.

- שכבות היררכיות - כל שכבה מדברת רק עם השכבה מעליה ומתחתיה ולא מודעת/מתעניינת באחרות.
- עקרון קצה לקצה - הרשת מאפשר חיבוריות וזהו, כל דבר אחר (כמו הצפנה) יעשה בידי המשתמשים עצמם. בנוסף, היא לא מבטיחה שפקטה תגיע, או שתגיע בסדר הנכון ולכן אפליקציות צריכות לדאוג גם לזה.

  העקרון הזה נועד כדי לאפשר אפליקציות שאנחנו עוד לא יודעים שיהיו להן דרישות אחרות לעשות מה שהן רוצות בצורה המיטבית ביותר.
- גורל משותף - בשיחה שמתקיימת, המקום היחיד שבו נשמר מידע על השיחה הוא בנקודות הקצה ולא בשום ראוטר או קודקוד באמצע.

הערה העקרון השני והשלישי מופרים ע"י ספקי אינטרנט באמצעות Firewalls וכל מיני מנגנונים אחרים כמו בדיקה של מעבר פקטות בלינקים חלשים מאוד (כמו סלולר). עם זאת, העקרונות האלה הם כן מנחים, רק לא ממומשים במציאות.

הדגמה ומוטיבציה למודל השכבות

נבנה את המשימות שלנו מלמטה למעלה.

- העברת אלקטורנים על כבל.
- העברת ביטים על כבל.
- העברת פקטות על כבל.
- העברת פקטות על רשת מקומית (LAN) עם כתובות מקומיות באמצעות ברודקאסט.
- העברת פקטות בין רשתות מקומיות שונות עם כותבות גלובליות.
- להבטיח שהפקטות מגיעות ליעד.
- לעשות משהו עם המידע הזה.

יש פה רבה משימות שונות ולשם כך אנחנו צריכים מודולים שונים - שכבות - שיהיו אחראים על כל משימה בנפרד.

דוגמה גם במציאות יש איזושהי שכבתיות. מנכ"ל א' שולח למנכ"ל ב' מכתב. המזכירה לקחה את המכתב, שמה אותו במעטפה, שלחה לחברת שליחויות.

זו שמה בעוד מעטפה למעבר פנימי בתוך תשתיות החברה, מעבירה למזכירה של המנכ"ל השני כשהיא הורידה את המעטפה הנוספת, המזכירה מוציאה מהמעטפה ונותנת את המכתב למנכ"ל.

סה"כ המנכ"ל שלח מידע והמנכ"ל האחר קיבל מידע. כל מה שנוסף הוא מטא-דאטא והמסלול שנוצר כאן הוא אנלוגי לחלוטין למה שקורה לפקטות באינטרנט.

מעבר לכך, כל שכבה תיקשרה באמצעות שפה ייחודית לה (מכתב, מעטפה ומעטפה פנימית) וזה אנלוגי לפרוטוקולים השונים של השכבות. אף שכבה לא צריכה להבין את השפה של אף שכבה אחרת.

שכבה	שפה	סוג מידע רלוונטי
מנכ"ל	מכתב	תוכן טקסטואלי
מזכירה	מעטפה	זהות הנמען
חברת שליחויות	מעטפה פנימית	מיקום הנמען

שבע המשימות שתיארנו למעלה מתאימות לחמש שכבות במודל השכבות באינטרנט.

- Pysical Layer - העברת ביטים (ואלקטרונים) בכבל.

- Link Layer - העברת פקטות ברשת מקומית (ובכבל).
- Network Layer - העברת פקטות בין רשתות מקומיות.
- Transport Layer - ווידוא הגעת הפקטות.
- Application Layer - לעשות משהו עם המידע.

תרגול

בתרגול חזרנו על הרבה מאוד הגדרות ומשפטים, אזכיר כאן את המרכזיים, להשלמות ראו קובץ התרגול

הערה בווי-פיי כולם שולחם הודעות לכולם וכדי למדל נכון את הרשת ולפתור את בעיית ההתנגשות, צריך הרבה הסת'.

הגדרה ההסת' של A בהינתן B היא $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$ עבור $P(B) > 0$.

דוגמה הטלנו קוביה וקיבלנו מספר זוגי. מה ההסת' לקבל 2? $P(\{2\} \mid \{2,4,6\}) = \frac{P(\{2\} \cap \{2,4,6\})}{P(\{2,4,6\})} = \frac{\frac{1}{6}}{\frac{1}{2}} = \frac{1}{3}$.

הגדרה A ו-B הם ב"ת אם $P(A \cap B) = P(A)\,P(B)$.

משפט (בייס) $P(A \mid B) = \frac{P(B|A)P(A)}{P(B)}$.

משפט (נוסחת ההסת' השלמה) עבור $\bigcup B_i = \Omega$, מתקיים $P(A) = \sum P(A \mid B_i)\,P(B_i)$.

דוגמה נתון סיב אופטי ששולחים עליו הודעות משני סוגים: הודעה טובה שהסיכוי שלא תיפול אחרי זמן T היא e^{-T} והודעה רעה עם e^{-1000T}. ההסת' ליצירת פקטה טובה היא p.

1. מה ההסת' שהודעה שנוצרה ב-$T = 0$ תשרוד עד $T = t$? מנוסחת ההסת' השלמה התשובה היא $pe^{-t} + (1-p)\,e^{-1000t}$.
2. בהינתן שהודעה שרדה t שניות, מה ההסת' שהיא פקטה טובה? מבייס,

$$P(\text{טובה} \mid t \text{ שרדה}) = \frac{P(\text{טובה})\,P(t \text{ שרדה} \mid \text{טובה})}{P(t \text{ שרדה})} = \frac{p \cdot e^{-t}}{pe^{-t} + (1-p)\,e^{-1000t}}$$

הערה ישנם כל מיני מ"מ בדידים ידועים, להלן סיכום שלהם בטבלה

	Bernoulli	Binomial	Geometric	Poisson
Intuition	We make an experiment, and we could either fail or succeed	We make n independent Bernoulli experiments. We mark by X the sum of successes	We do Bernoulli experiments until we succeed. We denote by X the number of tries	Counting the number of events that occurred during some period of time
Probability mass function	$P(X=1)=p$ $P(X=0)=1-p$	$P(X=k)$ $=\binom{n}{k}p^k(1-p)^{n-k}$	$P(X=k)$ $=(1-p)^{k-1}p$	$P(X=k)$ $=\frac{\lambda^k}{k!}e^{-\lambda}$
Notation	$X\sim Ber(p)$	$X\sim Bin(n,p)$	$X\sim Geo(p)$	$X\sim Pois(\lambda)$
Exp.	$E(X)=p$	$E(X)=np$	$E(X)=1/p$	$E(X)=\lambda$
Var.	$Var(X)=p(1-p)$	$Var(X)=np(1-p)$	$Var(X)=(1-p)/p^2$	$Var(X)=\lambda$

איור 2: טבלת להשוואת מ"מ בדידים

דוגמה נניח שאנחנו שולחים הודעה מ-S ל-D דרך $n-1$ תחנות ביניים, כאשר ההסת' שפקטה תיפול היא p.

1. מה ההסת' שהפקטה תגיע ל-D? $(1-p)^n$ ההסת' שכל הקפיצות כן יצליחו.
2. בהינתן שאנחנו שולחים הודעה שוב ושוב עד שהיא תגיע, מה תוחלת מספר ההודעות שנשלח?

נגדיר X מספר ההודעות שנשלח עד שההודע התגיע. מתקיים $X\sim \text{Geo}\left((1-p)^n\right)$ שתוחלתו $\frac{1}{(1-p)^n}$.

3. עתה כל תחנת ביניים (לא כולל S) שולחת את הפקטה שוב ושוב עד שהפקטה מגיעה, מה ההסת' שנצליח?

$1-p$ כי זה תלוי רק בהאם S הצליח להעביר לתחנה הראשונה.

הגדרה נאמר כי $X\sim\exp(\lambda)$ אם $f_X(x)=\begin{cases}\lambda e^{-\lambda x} & x>0\\ 0 & \text{אחרת}\end{cases}$.

דוגמה במטח מטאורים, מגיעים מטאורים בקצב של $\lambda=50\frac{\text{מטאורים}}{\text{שעה}}$. נוכל לאמר שהמטח הוא תהליך פואסוני בקצב λ. בהתאם, מספר המטאורים ב-t שעות הוא מ"מ פואסוני עם פרמטר λt, כלומר $M\sim\text{Pois}(50t)$ (ל-$50t$ אין יחידות).

מה ההסת' שנצלם שני מטאורים במצלמה שנחשפה ל-15 שניות? נבחר $t=\frac{15}{60\cdot 60}$ ונחשב $P(M=2)\approx 0.018$ או בסימון חלופי $P_{k=2}\left(t=\frac{15}{60\cdot 60}\right)$.

מה הסהת' שנצלם יותר ממטאור אחד? נסמן $T=\frac{15}{3600}$, לחשב $P(M>1)$ ישירות דורש חישוב טור שזה לא נוח, לשם כך נחשב

$$\begin{aligned}P(M>1)&=1-P(M\le 1)\\&=1-P_{k=0}(t=T)-P_{k=1}(t=T)\\&\approx 0.019\end{aligned}$$

ונשים לב כי כל האפשרויות מעבר ל-$M=3$ תורמות לנו מעט מאוד להסת' (ככל שמרתחקים מהתוחלת דועכת ההסת' להיות שם).

הערה נסמן $P_{k=i}(t=T)=P(M=i)$ כאשר $M\sim\text{Pois}(\lambda T)$, ההסת' שהגיעו i פקטות בתהליך פואסוני עם פרמטר λ אחרי זמן T.

הגדרה תהליך סופר (counting process) מוגדר ע"י $\{N_t : t \geq 0\}$ כאשר $N_0 = 0$ ו-$N_t \in \mathbb{N}_0$, $\forall t \geq 0$ סופר את מספר האירועים שקרו עד זמן t (לכן זו פ' מונוטונית).

הגדרה תהליך פואסוני עם פרמטר λ הוא תהליך סופר שבו מספר האירועים באינטרוולים זרים ב"ת מספר, ושמספר האירועים בכל אינטרוול באורך t מתפלג לפי מ"מ פואסון עם פרמטר λt.

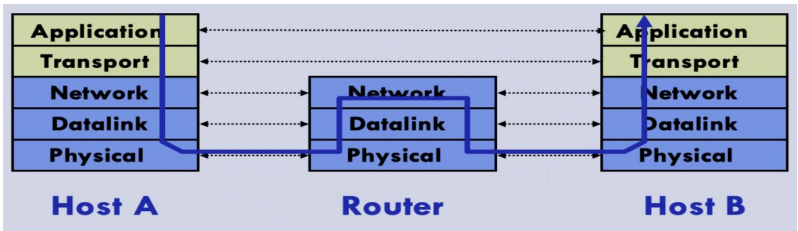
דוגמה מספר האנשים שעולים על אוטובוס הוא לא תהליך פואסוני, כי מספר האנשים תלוי בזמן (ולכן המ"מ שמייצג אינטרוול לא יכול להיות תלוי רק באורך הקטע אלא גם במיקום שלו על ציר הזמן).

שבוע $\mathbb{III}$ | מודל השכבות ותקשורת ב-LAN

הרצאה

הערה מתוך חמש השכבות, השתיים העליונות (Transport, Application) ממומשות רק בקצוות ולא בשום מקום אחר.

מסלול טיפוסי בין שני משתמשים נראה כך, כאשר העלייה והירידה באמצע מתארים את העובדה שכל שני קודקודים ברשת (שאינם משתמשי קצה), פותחים וסוגרים מעטפות גם הם כדי להעביר הלאה את ההודעה. כאן המעטפות הן header-ים שנוספים לתוכן הפקטה.



איור 3 : גרף להדגמת מעבר פקטה בין משתמשי קצה

פרוטוקולים

הגדרה פרוטוקול הוא הסכם איך לתקשר כדי להעביר מידע, או לתאם שימוש במשאב כלשהו. פרוטוקולים מגדירים סינטקס (מה הפורמט שבו מעבירים מידע) וסמנטיקות (איך מגיבים לאירועים והודעות מסוימות).

הערה פרוטוקול, בשונה מאלג', לא מתאר איך בדיוק נעשה משהו, אלא מה אנחנו צריכים שיעשה בכל מימוש של הפרוטוקול.

דוגמה הפרוטוקול הכי פופולרי בווב הוא HTTP, שיש לו סינטקס שמכיל הדרים, גוף הודעה, endpoint בכתובת כלשהי וכו'.

דוגמה פקטת IP מכילה הרבה מאוד שדות ולכל אחד מהם גודל ומיקום מגודר. אם נרצה להוסיף עוד שדה, להרחיב שדה וכו' וכו', נגיע לבעיה של interoperability - יש הרבה מאוד שחקנים באינטרנט וקשה להגיע להסכמה כדי שכל השחקנים בדרך ישתפו פעולה עם השינוי.

הערה ה-IETF הוא ארגון שכל מטרתו לאגד וליצור סטנדרטים לפרוטוקולים.

לכל שכבה יש כמה פרוטוקולים שונים (באפליקציה יש הרבה מאוד, בתעבורה יש הרבה וביניהם TCP, UDP וכו'), למעט בשכבה 3, שבה יש רק את IP.

חייבת להיות שכבה אחת שבה יש הסכמה בטוח על השפה שמדברים כי רק באמצעותה אפשר לבסס הסכמה על הפרוטוקלים האחרים שבהם יש שונות.

כל שכבה מקיימת את עקרון האנקספולציה - כל שכבה מכילה בתוך ההדרים שהיא מוסיפה את המידע שהיא צריכה ומשתמשת בשכבות מתחתיה כקופסה שחורה.

אפיון חמשת השכבות

נאפיין כל שכבה במודל חמשת השכבות באמצעות כמה קריטריונים.

- שירות - מה השכבה עושה.
- ממשק השירות - איך ניגשים לשירות (לצורך השכבה שמעל).
- פרוטוקול - איך peers מתקשרים (ממשק הפרוטוקול, איך ה-peer-ים משיגים את השירות, איך השכבה ממומשת בין קודקודים אבל לא בתוכם).

לכל שכבה יכולים להיות כמה מימושים שונים (כאן נכנסים חידושים והתאמות לצרכים שונים).

השכבה הפיזית

- שירות - מעבירה ביטים בין שני קודקודים שמחוברים בקשר פיזי.
- ממשק - מגדיר איך להעביר ולקבל ביטים (יש לספק את גודל ההודעה וכו').
- פרוטוקולים - קידודים שמאפשרים ייצוג ביטים (עוצמת זרם חשמל וכו').
- דוגמאות - סיב אופטי, גלי רדיו וכו'.

שכבת ה-Link

- שירות - לאפשר מעבר הודעות בין משתמשים באמצעות כתובות אבסטרקטיות מקומיות ולא דרך חיבור פיזי בלבד.
- ממשק - שליחת הודעות (מסגרות של ביטים) בין משתמשי קצה, קבלת ההודעות המיועדות למשתמש הקצה הנכון.
- פרוטוקולים - routing, כתובות MAC.
- דוגמאות - אתרנט, ווי-פי.

שכבת הרשת

- שירות - העברת פקטות לכתובת ספציפית עם כתובות אבסטרקטיות אבל גלובליות בין רשתות (העברת פקטה מרשת אחת בשכבה 2 לרשת אחר בשכבה 2).
- ממשק - איך לשלוח הודעות לרשתות אחרות ואיך לקבל הודעות שמיועדות לי.
- פרוטוקלים - יצירת רשתות ניתוב שמכילות מידע על לאן להעביר הלאה הודעה שמיועדת לכל כתובת גלובלית.
- דוגמאות - IP בלבד.

שכבת התעבורה

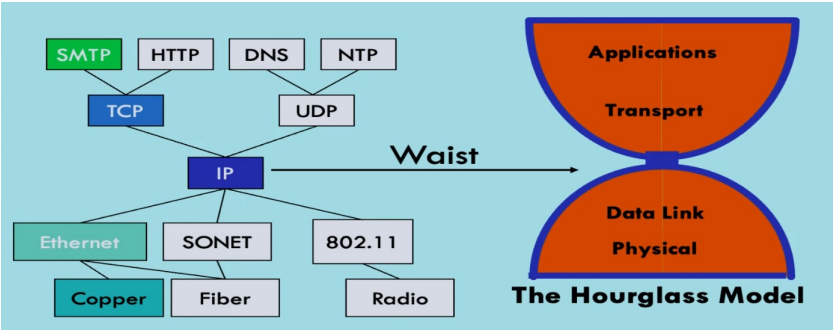
- שירות - תקשורת בין תהליכים, שמבחינה בין שיחות שונות שמתקיימות במקביל בין משתמשים, מספקת אולי אמינות בשליחת הודעות, וויסות קצב הודעות.
- ממשק - שליחת הודעות לתהליך ספציפי ביעד מסוים, העברת ההודעה המתקבלת לתהליך בתוך המכונה.
- פרוטוקול - הגדרת אמינות, פאקטיזציה של הודעות ארוכות, בקרת רצפים (שיגיעו לפי הסדר).
- דוגמאות - TCP, UDP אבל גם SCTP, T/TCP ועוד.

שכבת האפליקציה

שכבה זו היא השכבה השביעית, כאשר דילגנו על שכבות 5 ו-6 מסיבות אנכרוניסטיות.

- שירות - כל שירות שמוצע למשתמש הקצה.
- ממשק - תלוי באפליקציה.
- פרוטוקל - תלוי באפליקציה.
- דוגמאות - HTTP, SMTP, BitTorrent, Skype ועוד.

באיור הדגמה של "מודל שעון החול" - שכבה אחת חייבת להיות קבועה כדי שהשאר יוכלו להיות interpolable.



איור 4: גרף להדגמת מעבר פקטה בין משתמשי קצה

תקשורת ב-LAN

נתון לינק יחיד (בין אם כבל פיזי או תווך לוגי כמו ווי-פיי), כיצד נאפשר גישה מתואמת של כמה אנשים שונים?

הערה כדי לפתור את הבעיה של תקשורת ב-LAN (שכבת ה-datalink) נניח לעתה שיש לנו שכבה פיזית (נניח שקיימת), שכבת ה-datalink, ומעליה רק את האפליקציה.

הערה נקרא למשתמשי קצה קודקודים, לתווך הלוגי לינק ולפקטה בשכבה 2 פריים.

הפתרון שלנו צריך לאפשר כמה שירותים כאמור, ביניהם גישה ללינק עם כתובות בהדר הפריים והעברת מידע אמינה בין קודקודים (error detection וכו').

שכבת הלינק ממומשת בכרטיס הרשת שיש לנו במחשב/מכשיר, כך שרק כרטיסי רשת מדברים ביניהם בשכבה הזו ואין לאף גורם אחר חלק כאן.

הערה שכבת הלינק פותרת את הבעיה שפתר ה-switch רק בלי ה-switch. כשיש switch, אין בעיה של שיתוף המשאב של הלינק כי כולם מחולקים כבר לשיחות. המקום היחיד שבו יש עדיין דמוי broadcast domain זה בחלק של הכבל מכל קודקוד לסוויץ' (שם המחשב לכאורה מקבל ושולח מידע לכולם, עד שהסוויץ' ממיין הכל).

אנחנו מחפשים Multiple Access Protocol - אלג' מבוזר שקובע איך קודקודים מתקשרים ב-broadcast domain.

ה-MAP האידאלי

- כשקודקוד רוצה לדבר, הוא יכול לדבר בקצב R (רוחב הפס המלא).
- כש-M קודקודים רוצים לדבר, כל אחד יכול לדבר בקצב $\frac{R}{M}$.
- אין גורם ריכוזי שמתאם את השיחות.
- אין תיאום של שעונים, יישור של ביטים, והאלג' פשוט.

פרוטוקולים לגישה אקראית

כשלקודקוד יש פקטה לשלוח הוא שולח אותה בקצב המקסימלי האפשרי. אם שני קודקודים משדרים יש התנגשות ואז פרוטוקולים מסוג זה צריכים להגדיר איך לזהות התנגזויות ואיך לשחזר מהם מידע.

דוגמה אחת לפרוטוקול כנ"ל הוא Slotted ALOHA. נניח לשם פשטות (1) שכל הפריימים באותו הגודל; (2) שהזמן מחולק לסלוטים בגודל שווה (הזמן לשדר פריים יחיד); (3) קודקודים מתחילים לשדר רק בתחילת סלוט; (4) קודקודים הם מסונכרנים; (5) ואם שני קודקודים מתנגשים, כולם מזהים התנגשות.

תחת כל ההנחות הללו, כשנרצה לשלוח פקטה, נשלח אותה בסלוט הקרוב ואם הצלחנו בלי התנגשות יופי, ואם יש התנגשות נשדר את הפריים מחדש בהסת' p עד שנצליח לשדר ללא התנגשות.

יתרונות
אם רק קודקוד אחד מדבר הוא משדר בקצב המקסימלי
מאוד מבוזר, כל מה שצריך זה שהקודקודים יהיו מסונכרנים
פשוט

חסרונות
כשיש התנגשויות מתחילים לבזבז הרבה סלוטים
יש סלוטים ריקים
אפשר לזהות התנגשות לפני שנגמר הסלוט אבל לא נעשה עם זה כלום
אין לנו באמת יכולת לסנכרן

הגדרה goodput הוא היחס של סלוטים מוצלחים (ללא התנגשות ועם פריים) מסך הסלוטים. throughput הוא יחס הסלוטים עם פריים מסך הסלוטים.

ננתח את האלג' כאשר עתה תמיד נשלח פריים בהסת' p, גם בפעם הראשונה. נניח שלכל קודקוד יש אינסוף פריימים לשלוח, יש לנו N קודקודים והסת' שליחה p.

ההסת' שקודקוד יצליח לשלוח הודעה ללא התנגשות היא $p\left(1-p\right)^{N-1}$ (הוא שולח וכל השאר לא). ההסת' שקודקוד כלשהו יצליח הוא $Np\left(1-p\right)^{N-1}$ כי המאורעות שאנחנו מאחדים זרים. בשביל יעילות מקסימלית נמצא p^* שממקסם את הנ"ל, וכש-N שואף לאינסוף הערך הזה נותן יעילות של $\frac{1}{e} \approx 0.37$ שזה די גרוע.

כל הניתוח הזה לא פרקטי כי אין לנו סלוטים במציאות כי אין סינכרון כאמור, לשם כך יש את Pure ALOHA - כל קודקוד מתנהג כאילו אנחנו כן ב-slotted ALOHA כאשר ייתכן שהסלוטים של הקודקודים השונים לא מסוכנרים. במקרה כזה, ההסת' להתנגשות עולה כי לא נקבל רק התנגשויות בתוך פריימים, אלא גם בין פריימים.

תרגול

דוגמה בדומה לאנלוגיה למנכ"ל והמזכירה, אפשר להסתכל על מודל השכבות בדומה לפעילות של דואר ישראל.

נרצה לשלוח הודעה מירושלים, ישראל לטוקיו, יפן. קודם כל נצעק בשכונה האם מישהו מכיר את הנמען בטוקיו. אם לא, נשלח למרכז איסוף עירוני ושם יבדקו האם הנמען נמצא בעיר אחרת בארץ. אם לא, המכתב יועבר לנתב"ג ומשם יעבור את המסלול ההפוך ביפן.

כעת נעסוק בשכבה 2 ובפרט בפרוטוקולי גישה אקראית.

הערה נניח כמה הנחות משפטות: (1) כשקודקוד משדר הוא משדר ברוחב פס מלא; (2) אם יש התנגשות כל המידע אבד; (3) אפשר להתעלם מרעש בערוץ (אם יש כזה).

הגדרה רוחב הפס של ערוץ תקשורת כלשהו הוא כמות המידע שאנחנו יכולים לשלוח ליחידה זמן, נסמנו B ויחידותיו $\frac{bit}{sec}$.

הערה נסמן goodput ב-η.

דוגמה כמה זמן יקח לשדר פקטה כשפקטה היא בגודל 30 ביטים ורוחב הפס הוא 5 ביט/שנייה. כמה זמן יקח לשדר את ההודעה? $\frac{|packet|}{B} = \frac{30}{5} = 6$.

מה רוחב הפס בערוץ שמצליח לשדר 3 פקטות באורך 30 ביטים בתוך שתי שניות? $\frac{3\cdot 30}{2} = 45$.

לצורך ניתוח ALOHA, נסמן T הזמן לשליחת פקטה אחת, $X_p \sim Z$ מספר הפקטות שנשלחות באינטרוול בגודל T.

ב-ALOHA כפי שלמדנו בהרצאה, ב-ALOHA שולחים פריימים בסלוטים, כאשר ב-slotted מניחים שלכל הקודקודים יש שעון מסונכרן ולכן כל הסלוטים של כולם מתואמים, ואילו ב-pure יתכן שלכל אחד שעון אחר.

ננתח את האלג' במקרה הבינומי - $X_p \sim \text{Bin}(n,p)$ (כל קודקוד משדר בהסת' p) ובמקרה הפואסוני - $X_p \sim \text{Pois}(gT)$ (יש אינסוף קודקודים, ובאופן טיפוסי נצפה ל-g הודעות בשנייה, T גודל האינטרוול בשניות).

גישה בינומית (slotted)

דוגמה אם יש שלושה קודקודים, ההסת' ששניים יתנגשו היא $p^2(1-p)$, ההסת' לסלוט ריק הוא $(1-p)^3$ וכו'.

ראשית נחשב את ההסת' לפקטה מוצלחת (p_{suc}) ללא התנגשויות. לאחר מכן נסתכל על P_{suc} כמ"מ אינדיקטור על האם סלוט היה ללא התנגשויות ועם הודעה ואז נחשב את תוחלתו, וזה יהיה ה-goodput. נחזור על זה לארבעת המקרים

כמו שראינו כבר כמה פעמים, ההסת' p_{suc} היא $np(1-p)^{n-1}$ (בינומי קלאסי).

$T_{suc} = TP_{suc}$ הוא המ"מ המייצג את זמן השידור המוצלח בסלוט מסוים, כאשר $P_{suc} \sim \text{Ber}\left(\frac{np(1-p)^{n-1}}{p_{suc}}\right)$. מתקיים $E[T_{suc}] = Tp_{suc}$ ולכן ה-goodput הוא $\frac{T \cdot p_{suc}}{T} = p_{suc}$.

גישה בינומית (pure)

נסתכל על סלוט כלשהו, ונשים לב שעתה יכול להיות שמישהו נכנס לסלוט שלנו באמצע (או שאנחנו נכנסו לו). נצטרך שגם ביחידת הזמן הזו (סלוט מהזווית שלנו) וגם בזו הקודמת אף אחד לא שלח שום דבר, כלומר $p_{suc} = np(1-p)^{2(n-1)}$. ומשם שאר הניתוח זהה. במקרה כזה p האופטימלי הוא $\frac{1}{2n}$ שנותן $\frac{1}{2e}$ שזה חצי מה-goodput של ה-slotted.

גישה פואסונית (slotted)

דוגמה מה ההסת' שעל פני שני סלוטים לא נשלחה אף הודעה? $P_{k=0}(t=2T) = \frac{g^0 e^{-2gT}}{0!} = e^{-2gT}$, לחלופין נוכל לחשב את זה כמכפלת מ"מ ב"ת על שני הסלוטים. נשים לב שבתהליך הפואסוני שלנו אכן אינטרוולים זרים הם בעלי התפלגות ב"ת על מספר ההודעות (כיאה לתהליך פואסוני).

מה ההסת' p_{suc} הפעם? על פני סלוט אחד, אנחנו מחפשים את ההסת' שנשלחה הודעה אחת רק, כלומר $p_{suc} = P_{k=1}(t=T) = gTe^{-gT}$ ומאותה סיבה כמו לפני, ה-goodput הוא $\frac{E[T_{suc}]}{T} = \frac{TE[P_{suc}]}{T} = \frac{Tp_{suc}}{T} = p_{suc}$.

גישה פואסונית (pure)

מה היא p_{suc} הפעם? נסתכל שוב על יחידה אחד בגודל סלוט אחד, ונרצה שתשלח הודעה אחת בסלוט הזה ו-0 באחד שלפני, ובגלל שזה תהליך פואסוני זה ב"ת ונוכל להכפיל את ההסת', כלומר

$$p_{suc} = P_{k=0}(t=T)\,P_{k=1}(t=T) = e^{-gT}gTe^{-gT} = gTe^{-2gT}$$

וה-goodput בהתאם כמו לפני, הפעם $\eta_{max} = \frac{1}{2e}$ בדומה ל-pure בבינומי.

שאלה הודעות מגיעות כתהליך פואסוני עם פרמטר g. חלק מההודעות באורך S וחלק באורך $L > S$ כאשר ההסת' ל-S היא p (ו-$1-p$ ל-L).

- נניח שאנחנו משתמשים ב-slotted ALOHA עם סלוט בגודל L. מה ה-goodput?

ההסת' לפקטה מוצלחת היא $p_{suc} = P_{k=1}(t=L)$, ואז

$$T_{suc} = \begin{cases} L & p_{suc,L} \\ S & p_{suc,S} \\ 0 & \text{אחרת} \end{cases}$$

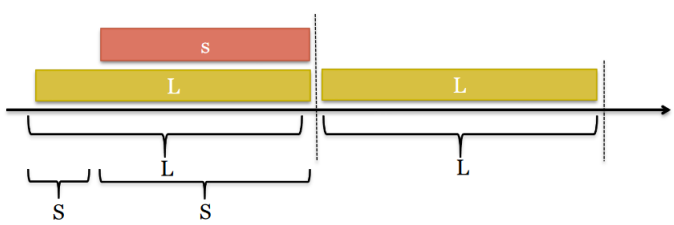
כאשר $p_{suc,L}$ זו ההסת' ששלחנו הודעה ארוכה והצלחנו ו-$p_{suc,S}$ בהתאמה. לכן $E[T_{suc}] = Lp_L + Sp_S$ ואז ה-goodput הוא

$$\eta = \frac{E[T_{suc}]}{L} = \frac{1}{L}\left(Lp_{suc,L} + Sp_{suc,S}\right) = gLe^{-gL}\left((1-p) + \frac{S}{L}p\right)$$

- נניח שאנחנו משתמשים ב-pure ALOHA. מה ה-goodput?

עתה כדי שנשלח הודעה מוצלחת בסלוט בגודל L, נחלק למקרה הקצר והארוך.

לשליחת הודעה ארוכה מוצלחת, נרצה (1) שתשלח הודעה אחת ארוכה בסלוט בגודל L שלנו; (2) 0 הודעות בסלוט בגודל S לפני; (3) ו-0 הודעות בסלוט בגודל $L-S$ לפניכן שהן ארוכות כי רק הן מתנגשות עם הסלוט הנוכחי, הקצרות לא (כמו באיור).



איור 5: הדגמה של הסלוטים המועדים להתנגשות, ה-S הימני הוא טעות ואמור להיות $L-S$

לכן סה"כ נקבל

$$\begin{aligned} p_{suc,L} &= \underbrace{P_{k=1}(t=L)(1-p)}_{(1)} \cdot \underbrace{P_{k=0}(t=S)}_{(2)} \cdot \underbrace{P_{k=0}(t=(L-S)(1-p))}_{(3)} \\ &= (1-p)\,gLe^{-gL} \cdot e^{-gS} \cdot e^{-g(1-p)(L-S)} \\ &= (1-p)\,gLe^{-g(2L(S-L)p)} \end{aligned}$$

כאשר הכפלנו ב-$(1-p)$ בחוץ ב-(1) כי אנחנו שולחים הודעה ורק בהסת' $1-p$ היא באורך L ובפנים ב-(3) כי אנחנו מעוניינים רק בהודעות באורך L שיכולות להשלח ולכן קצב ההודעות הרלוונטי משתנה מ-g ל-$g\,(1-p)$.

עתה ההסת' שנשלח הודעה מוצלחת ביחידת זמן בגודל S מחושבת לפי ההסת' (1) שתשלח הודעה אחת אינטרוול בגודל S; (2) שיהיו 0 הודעות בסלוט בגודל S לפני; (3) שיהיו 0 הודעות ארוכה בסלוט $L-S$ לפני הסלוט הקודם.

כלומר סה"כ אותו חישוב כנ"ל רק עם $P_{k=1}\,(t=S)$ במקום $P_{k=1}\,(t=L)$.

לסיכום,

$$\eta = \frac{L \cdot p_{suc,L}}{L} + \frac{S p_{suc,S}}{S} = p_{suc,L} + p_{suc,S} = \dots$$

כאשר הפעם חילקנו כל אחד באורך ההודעה כי הודעה באורך S תורמת לנו S לרוחב הפס המנוצל בהצלחה והודעות ב-L תורמות L.

שבוע $\mathbb{IV}$ | שכבת הלינק לעומק

תרגול

דוגמה נניח שאנחנו מתקשרים עם סיב אופטי שלו 3 כבלים. מחשב אחד לא יודע לתקשר בשלושת הסיבים בו זמנית אלא רק באחד, ובו הוא משתמש בהצלחה $\frac{1}{2}$ מהזמן, אז ה-goodput יהיה $\frac{1}{6}$ (למעשה הוא חסום מלמעלה ע"י $\frac{1}{3}$).

הערה מעתה, η (ה-goodput) יהיה יחס הזמן המנוצל בהצלחה מסך הזמן, כפול השיקולים הפיזיקלים שנתונים לנו מבחוץ.

ראינו בתרגול הקודם את ה-goodput של גרסאות שונות של ALOHA, להלן טבלה שמסכמת אותם

	הסת' תחת תהליך פואסוני	הסת' כתלות ב-g, T
Slotted	$P_{k=1}\,(t=T)$	gTe^{-gT}
Pure	$P_{k=1}\,(t=T)\,P_{k=0}\,(t=T)$	gTe^{-2gT}

שאלה נתונים תשעה כפרים על מאדים בסידור הבא

1	2	3
4	5	6
7	8	9

כל כפר יכול לשמוע את ארבעת הכפרים הסמוכים לו (למעלה, למטה, ימינה ושמאלה), אפע"פ שהודעות מכוונות אך ורק לנמענים בתוך אותו הכפר.

הכפרים האי-זוגיים מתקשרים (בתוך הכפר) עם slotted ALOHA עם פרמטר g_o והזוגיים ב-pure ALOHA עם פרמטר g_e.

ניתן להניח כי בכל כפר יש אינסוף קודקודים (ולכן אפשר להזניח כל מיני תופעות כמו העובדה שלא משנה איזו יחידת זמן בגודל T נבחר, עדיין קצב ההודעות יהיה gT).

- מה ה-goodput של כל המושבה אם $g_o = 0$?

ה-goodput של כל מושבה זוגית הוא p_{suc} כפי שראינו בעבר, כאשר p_{suc} הוא (מהטבלה הנ"ל) $g_o T e^{-2g_o T}$. סה"כ ה-goodput הוא $4 \cdot \left(\frac{1}{4} g_o T e^{-2g_o T}\right) = g_o T e^{-2g_o T}$ כאשר אנחנו מכפילים ב-4 כי הניצול המוצלח שנוסף לנו הוא ב"ת לכן אפשר לסכום את כל הכפר והרבע שם כי כל כפר היא רבע מכלל המושבה.

- מה ה-goodput של כל מושבה אם $g_e = 0$?

בדומה לנ"ל, $g_e T e^{-g_e T}$ כי אנחנו ב-slotted.

- מה ה-goodput במקרה הכללי?

נחלק את הכפרים לשלוש קטגוריות: פינה, צלע ומרכז.

  - פינתי: ההסת' ש-1 (בה"כ) ישדר בהצלחה היא

$$p_{suc,corn} = \underbrace{P_{k=1}^{g_o}(t=T)}_{\text{הפינה משדרת}} \cdot \underbrace{\left(P_{k=0}^{g_e}(t=T)\right)^2}_{\text{השכנים לא משדרים כרגע}} \cdot \underbrace{\left(P_{k=0}^{g_e}(t=T)\right)^2}_{\text{השכנים לא משדרים קודם}} = g_o T e^{-g_o T} e^{-4g_e T}$$
$$= g_0 e^{-T(g_0 + 4g_e)}$$

כאשר נשים לב להבחנה בין זוגיים לאי-זוגיים מבחינת ה-pure/slotted לצורכי חישוב ההסת'.

  - צלע: ההסת' ש-2 (בה"כ) ישדר בהצלחה היא

$$p_{suc,side} \underbrace{P_{k=1}^{g_e}(t=T)}_{\text{הצלע משדרת}} \cdot \underbrace{P_{k=0}^{g_e}(t=T)}_{\text{הכפר לא משדר לפני}} \cdot \underbrace{\left(P_{k=0}^{g_o}(t=T)\right)^3}_{\text{השכנים לא משדרים כרגע}} \cdot \underbrace{\left(P_{k=0}^{g_o}(t=T)\right)^3}_{\text{השכנים לא משדרים לפני}} = g_e T e^{-2T(g_e + 3g_0)}$$

כאשר כאן אנחנו מסתכלים על המקרה הכללי שבו חלון הזמן שבחרנו לא aligned בדיוק על הסלוט של ה-slotted (במקרה כזה לא היינו צריכים את הביטוי האחרון).

נוכל להתעלם מהמקרה הספציפי הזה כי יש אינסוף נקודות קצה ובגלל שהכל רציף לא מסתכלים על חלון נקודתי.

  - מרכזי: ההסת' ש-5 ישדר בהצלחה היא

$$p_{suc,mid} = \underbrace{P_{k=1}^{g_o}(t=T)}_{\text{המרכז משדר}} \cdot \underbrace{\left(P_{k=0}^{g_e}(t=T)\right)^4}_{\text{השכנים לא משדרים כרגע}} \cdot \underbrace{\left(P_{k=0}^{g_e}(t=T)\right)^4}_{\text{השכנים לא משדרים קודם}} = g_o T e^{-T(g_e + 8g_o)}$$

לכן סה"כ ה-goodput הכולל הוא $\frac{1}{9}\left(p_{suc,mid} + 4 \cdot p_{suc,corn} + 4 \cdot p_{suc,side}\right) = \ldots$.

הערה לכאורה אם נשלחה הודעה בידי אחד הקודקודים ביחידת הזמן הקודמת ב-pure ALOHA, ההסת' שתשלח הודעה ביחידת הזמן הנוכחית ירדה ולכן ההסת' תלויות וכל החישובים שלנו לא נכונים. נראה הוכחה למה זה לא נכון. נסמן $X_{suc,i}$ מספר ההודעות ששלח הקודקוד ה-i בהצלחה ביחידת זמן T (0 או 1). בממוצע, על יחידת זמן בגודל T, בממוצע ישלחו

$$\sum E\left[X_{suc,i}\right] = E\left[\sum X_{suc,i}\right] = E\left[X_{suc}\right] = p_{suc}$$

כאשר השתמשנו בלינאריות התוחלת שמתקיימת גם כשהמ"מ לא ב"ת ולכן סה"כ באמת קיבלנו שלא משנה איזה חלון נבחר, תמיד נקבל בממוצע p_{suc} הודעות בשנייה.

שאלה נתונה רשת עם שני סוגים של פקטות כאשר קצב השליחה מתנהג כתהליך פואסוני $X_p \sim \text{Pois}(gT)$. פקטות מסוג A הן בגודל S ופקטות מסוג B הן בגודל αS כאשר $\alpha > 1$. כולם משתמשים בפרוטוקול slotted ALOHA עם סלוט $T = S$. בהסת' p נשלחת פקטה מסוג A ובהסת' $1-p$ מסוג B. יש אינסוף קודקודים.

- מה הוא p_{suc}, ההסת' לפקטה מוצלחת?

נוכל להסתכל על התהליך כשני תהליכים נפרדים, אחד $S_p^\alpha \sim \text{Pois}((1-\alpha)g)$ ואחד $S_p \sim \text{Pois}(pg)$ בהתאם לסוגים.

ההסת' לפקטה מוצלחת באורך αS היא

$$\begin{aligned} p_{suc}^\alpha &= \underbrace{P_{k=0}^\alpha(t=(\lceil\alpha\rceil-1)S)}_{\text{אף ארוך לא יכנס לפני}} \cdot \underbrace{P_{k=1}^\alpha(t=S)}_{\text{אנחנו היחידים כרגע}} \cdot \underbrace{P_{k=0}^\alpha(t=(\lceil\alpha\rceil-1)S)}_{\text{אף ארוך לא מתחיל עד שנסיים}} \cdot \underbrace{P_{k=0}(t=\lceil\alpha\rceil S)}_{\text{אף קצר לא מתחיל עד שנסיים}} \\ &= e^{-g(1-p)(\lceil\alpha\rceil-1)S} \cdot (1-p)\,gSe^{-\alpha g(1-p)S} \cdot e^{-\alpha g(1-p)(\lceil\alpha\rceil-1)S} \cdot e^{-\alpha gp\lceil\alpha\rceil S} \end{aligned}$$

ההסת' לפקטה מוצלחת באורך S היא (בדומה לחישוב הנ"ל)

$$p_{suc} = P_{k=1}(t=S)\,P_{k=0}^\alpha(t=S)\,P_{k=0}^\alpha(t=(\lceil\alpha\rceil-1)S)$$

וההסת' לפקטה מוצלחת בכללי היא $p_{suc,tot} = p \cdot p_{suc} + (1-p)\,p_{suc}^\alpha$.

- מה ה-goodput?

ה-goodput הוא

$$\eta = \frac{E[T_{suc}]}{\text{סך הזמן}} = \frac{S\alpha p_{suc}^\alpha}{S\lceil\alpha\rceil} + \frac{Sp_{suc}}{S} = \frac{\alpha}{\lceil\alpha\rceil}p_{suc}^\alpha + p_{suc}$$

תיקון שגיאות

לעתים בערוץ יש רעש ואז צריך לתקן ולזהות את השגיאות. ישנן כמה שיטות לזיהוי ותיקון שגיאות.

- repetition code - נחזור על כל ביט שלוש פעמים ונתייחס לביט לפי החלטת הרוב, כך נזהה כל אי אחידות ונתקן נכון רק אם יש רוב לא הרוס.

נוכל להסתכל על שלושת הביטים כקוורדינטות בעולם תלת ממדי, ואז כל האפשרויות ל-3 ביטים יוצרים קוביה תלת-ממדית. יש רק שתי קוורדינטות נכונות ולמעשה כדי להכריע, עושים הטלה ולוקחים את התוצאה (דמיינו גאומטרית). כאן התקורה היא 200% כי יש לנו n ביטים ואנחנו שולחים $3n$ ביטים.

- parity code - נוסיף ביט אחד שהופך את ההודעה לזוגית (0 אם היא כבר זוגית ו-1 אם היא אי-זוגית). התקורה כאן היא $1 + \frac{1}{n}$.

  שיטה זו נותנת לנו זיהוי של עד שגיאה אחת אבל אין לנו יכולת לשחזר את המקור.

  עכשיו נחלק את ההודעה למטריצה ונוסיף לכל שורה ועמודה ביט זוגיות (2-d parity). כך נוכל לזהות שלוש שגיאות ולתקן אחת.

שבוע $\mathbb{V}$ | CSMA ואלג' MAP

הרצאה

אנחנו עדיין מנסים לתקשר בתווך משותף (רשתות ברודקאסט) שאין בו סוויץ' באמצע. התוצאה של Slotted ALOHA שמניח הרבה דברים הייתה goodput של $\frac{1}{e} = 37\%$, וב-Pure ALOHA קיבלנו goodput של $\frac{1}{2e} = 18\%$.

הערה אם אנחנו יודעים לזהות התנגשות, אנחנו יודעים גם שכשנתחיל לדבר באמצע סלוט של מישהו אחר אנחנו נתנגש איתו, לכן אין סיבה להתחיל לדבר.

Carrier Sense Multiple Access

ב-CSMA, נאזין לפני שנשדר, ואם נזהה שהערוץ שקט, נשדר פריים שלם, ואחרת נחכה עד שיתפנה.

אם יש איזושהי הסטה מינימלית בין אנשים תמיד, אז לכאורה לא נוכל אף פעם לקבל התנגשות (גם אם התחלנו לדבר באותו הזמן אחרי שהיה שקט אחד יזהה את האחר). עם זאת, ההאזנה ותחילת השידור לא קורים באותו הזמן, ואז יכול להיות שקודקוד אחד התחיל לשדר אחרי ששמע שקט, והאחר גם לא שמע שום דבר ϵ זמן אחריו כי השידור לא הגיע אליו עדיין ועכשיו שניהם מדברים.

מעבר לזה, אם מצב כנ"ל קורה, לא רציונאלי להמשיך את השידור כי הקודקוד השני יודע שהוא גורם להתנגשות.

CSMA/CD וזיהוי התנגשויות רחוקות

CSMA כנ"ל, כאשר נניח שאנחנו מזהים התנגשויות בזמן קצר, ומבטלים שידורים אם מזהים התנגשות כדי למנוע בזבוז של הערוץ. לאחר ביטול השידור נגריל זמן המתנה ונחזור לשלב ההמתנה לשקט.

הערה מודל כזה אפשרי ב-LAN קווי אבל באלחוטי אין לנו דרך לדעת האם יש התנגשות בהכרח כי הסיגנל שלנו משמעותית יותר חזק מזה שמגיע אלינו.

במקרה שבו אנחנו מאוד רחוקים מקודקוד אחר שמתנגש איתנו, נוכל לשמוע על ההתנגשות הרבה אחרי שסיימנו לשדר את הפריים וחשבנו שהוא היה מוצלח.

במקרה הכי גרוע תחילת השידור של הקודקוד הרחוק היא ב-$t = PROP - \epsilon$ כאשר $PROP$ זמן פעפוע ביט מאיתנו לקודקוד הרחוק ביותר ברשת. במקרה כזה הרחוק יזהה התנגשות ב-$t = PROP$, ונגלה אנחנו שהייתה התנגשות רק ב-$t = 2PROP - \epsilon$.

אם כן, לאחר $2PROP$ אם לא התוודענו על התנגשות ניתן להניח ששידרנו בהצלחה. מכאן שכדי להשתמש ב-CSMA/CD צריך אורך מינימלי של פריים כדי שיתקיים $B \cdot L > 2 \cdot PROP$ כלומר שהזמן שלוקח לשדר פקטה (אורך כפול רוחב פס) יהיה יותר מ-$2 \cdot PROP$, ואז נזהה את ההתנגשות לפני תום שידור הפריים ונדע אם צריך לשדר שוב או לא ולא נאלץ לשמור פריימים לשידורים עתידיים וכו'.

חישוב ה-goodput של CSMA/CD

נחשב את ה-goodput במקרה של CSMA/CD. נניח שהזמן מחולק לסלוטים ושסלוט הוא פעמיים זמן הפעפוע המקסימלי ברשת (רק בשביל החישוב התאורטי, לא קשור לפרוטוקול).

בכל רגע נתון, נשלח פריים בהסת' p (גם בפעם הראשונה וגם בשידורים מחדש). ההסת' שרק אחד שידר בסלוט היא $\binom{N}{1} p (1-p)^{N-1} = \alpha(p)$ וזה מקסימלי עבור $p = \frac{1}{N}$, שאז נקבל $\alpha_{max} = \frac{1}{e} \approx 0.4$. נסמן A מספר הסלוטים המבוזבזים בתוחלת לפני ששידרנו בהצלחה פריים, אז מתקיים $A = \alpha \cdot 0 + (1-\alpha)(1+A)$ וכאשר $\alpha = \alpha_{max}$ נקבל $A = 1.5$ (מדובר בתוחלת של מ"מ גאומטרי). לכן ה-goodput שלנו הוא

$$
\begin{aligned}
\eta_{CSMA/CD} &= \frac{\text{זמן השידור של סלוט אחד}}{\text{כמות הסלוטים שנדרשו עד לשידור מוצלח כולל}} \\
&= \frac{TRANSP}{TRANSP + E\left[\text{מספר הסלוטים המבוזבזים לשידור מוצלח של פריים}\right]} \\
&= \frac{TRANSP}{TRANSP + A\left(2 \cdot PROP\right)} \\
&= \frac{TRANSP}{TRANP + 3 \cdot PROP} \\
a = \frac{PROP}{TRANSP} \quad &= \frac{1}{1+3a}
\end{aligned}
$$

כאשר $TRANSP$ הוא זמן השידור של הודעה אחת (גודל הסלוט). במציאות מתקבל $\eta_{CSMA/CD} \approx \frac{1}{1+5a}$. כדי למקסם את ה-goodput נרצה למזער את a, כלומר להקטין את היחס. אם כן, ברשתות LAN קטנות, $PROP$ הוא קטן ואז אפשר להרשות לעצמנו $TRANSP$ לא עצום (לא יקח שעה לשדר הודעה) שייתן goodput טוב.

הערה אם כל האינטרנט היה ברדוקאסט, היינו צריכים $TRANSP > 2 \cdot PROP$ כלומר הרבה זמן לשידור הודעה, וגם $a \gg 1$ והיה לנו goodput נורא.

כתובות MAC

לכל כרטיס רשת צרובה כתובת MAC באורך 48 ביטים. כתובות כאלה נקראות כתובות פיזיות כי הן לא מספרות לנו שום דבר על המיקום שלנו בעולם, אלא רק על מי ייצר את הכרטיס שלנו וכיוצ"ב. נשתמש ב-MAC-ים ברשתות LAN כדי לקבוע את המען של הפריים שלנו. יש גוף שמרכז את חלוקת כתובות ה-MAC לחברות שמייצרות NIC-ים.

הערה כיום כבר אין יותר מדי רשתות ברודקאסט טהורות, כשלרוב יש סוויצ'ים ששוברים את התקשורת החופשית באמצע. הפרוטוקול המרכזי שמתמשים בו בשכבת הלינק כיום הוא Ethernet.

Ethernet

האלג' של את'רנט עובד בדומה ל-CSMA/CD, עם שינויים קלים.

1. כרטיס הרשת מקבל פקטה משכבת הרשת ויוצר פריים מתאים לה.
2. אם כרטיס הרשת מזהה שהערוץ פנוי, הוא משדר, אחרת יחכה שיהיה פנוי.
3. אם הכרטיס שידר פריים שלם בלי לזהות התנגשות, סיימנו.
4. אם זוהתה התנגשות, נפסיק לשדר ונשלח jam signal שמוודא שאם שידרנו קצת מדי זמן האחרים יבינו שזה לא רעש אלא שידור שהפסיק באמצע.
5. אחרי ביטול השידור בהתנגשות, נחכה זמן אקראי $k \in \{0, \ldots, 2^{m-1}\}$ לפני חזרה לשלב 2, כאשר m מספר הפעמים שהתנגשנו עבור הפריים הנוכחי.

תרגול

הבעיה ב-ALOHA כאמור היא שאנחנו לא מתנהלים עם התנגשויות כמו שצריך. לצורך כך הומצא CSMA, שמטפל בהתנגשויות באמצעות הפסקת שידור לאחר זיהוי התנגשות ובדיקה האם הערוץ פנוי לפני שמתחילים לשדר. נשווה בין כמה תתי-פרוטוקולים כאלה

פרוטוקול	הערוץ פנוי?	הערוץ תפוס?	השלכות
1-persistent	שלח הודעה	המתן עד שפנוי ושלח מיד	יש הסת' גבוהה להתנגשויות בערוצים עמוסים
non-persistent	שלח הודעה	חכה פרק זמן ונסה שוב	נקבל ניצולת נמוכה בעומס נמוך
p-persistent	שלח בהסת' p	המתן עד שפנוי ושלח בהסת' p	פשרה בין השניים הנ"ל

Exponential Backoff

כמה זמן ראוי לחכות ב-non-persistent? הדרך הכי פופלרית לחישוב זמן ההמתנה הוא Exponential Backoff.

יהי פרמטר c קבוע מראש (לרוב 2). אחרי הניסיון ה-k לשידור פריים, נחכה jT יחידות זמן כאשר j מוגרל אחיד על $[0, c^k - 1]$. בכל פעם שנצליח לשדר, נאפס את k שלנו.

ברגע שזיהינו התנגשות עם אחרים, נשדר jam signal כדי שאחרים ידעו שהתנגשנו ושאין מה להתחייס לפריים הנוכחי.

חישוב ה-goodput של CSMA/CD (יותר לעומק)

נבצע את אותו הניתוח שעשינו בהרצאה, רק קצת יותר מפורט. נניח שאין לנו exponential backoff, שיש לנו N תחנות, זמן השידור המקסימלי ברשת הוא T_{prop} וזמן השידור של פריים הוא T_{trans}.

נניח שהפרוטוקול משדר בהסת' p, שהזמן (קונספטואלית) מחולק לסלוטים בגודל $2T_{prop}$ יחידות זמן, ושלקודקודים תמיד יש מה לשדר.

הפרוטוקול שלנו ישדר אם פנוי, אחרת יחכה לסלוט הבא, וכשיזהה התנגשות ישלח jam signal, יפסיק את השידור וינסה בסלוט הבא בהסת' p.

לרשת יש שלושה מצבים אפשריים.

- מתיחות - שני קודקודים משדרים ויגלו את ההתנגשות רק לאחר זמן לכל היותר $2T_{prop}$.
- שקט - אף אחד לא משדר
- משדר - קודקוד אחד משדר בהצלחה פריים.

ההסת' לשידור מוצלח כרגיל הוא $Np\left(1-p\right)^{N-1}$, והיא מקסימלית כאשר $p=\frac{1}{N}$, נסמן הסת' מקסימלית זו ב-S. זה לא מספיק לנו כדי לשדר כי אנחנו צריכים להתחשב בזמן שמתבזבז כשאנחנו במצב מתיחות.

השאלה שלנו כאן היא מה ההסת' לאינטרוול מתיחות באורך $T-1$ סלוטים, ולאחריו שידור מוצלח? לכן בתוחלת, הזמן הנדרש לשידור מוצלח הוא $2T_{prop}\cdot\frac{1}{S}$ בתוחלת (מספר הסלוטים כפול תוחלת המ"מ למספר הסלוטים עד לשידור מוצלח), ולכן הזמן המבוזבז בתוחלת הוא $2T_{prop}\left(\frac{1}{S}-1\right)$.

מכאן שה-goodput הוא

$$\begin{aligned}\eta &= \frac{T_{trans}}{T_{trans}+\text{זמן מבוזבז}}\\ &= \frac{T_{trans}}{T_{trans}+2T_{prop}\left(\frac{1}{S}-1\right)} \xrightarrow[n\to\infty]{} \frac{1}{1+\frac{2T_{prop}}{T_{trans}}\left(e-1\right)}\end{aligned}$$

וכאשר $T_{trans}\gg T_{prop}$ נקבל goodput שואף לאחד.

הערה היינו יכולים לעשות את אותו הניתוח גם במקרה של slotted ALOHA, כשאז היינו מציבים $T_{trans}=2T_{prop}$ כי זמן השידור של הודעה הוא בדיוק גודל הסלוט.

שאלה נתונה רשת בפרוטוקול CSMA/CD עם רוחב פס 10 Mbps וקצב פעפוע ביט של $2\cdot10^8\frac{m}{s}$. נרצה להוסיף עוד רשת דומה, מרוחקת מרחק של 20 ק"מ מזו. הרשתות מחוברות באותו התווך כנ"ל והמרחק המקסימלי בין קודקודים הוא 20 ק"מ.

- האם ניתן להריץ CSMA/CD כאשר גודל ההודעה הוא 64 בתים?

לא! נצטרך ש-$T_{trans}>2T_{prop}$ אבל כאן יש לנו זמן פעפוע של ביט אחד לאורך הרשת של

$$\tau=\frac{20km}{2\cdot10^8\frac{m}{s}}=10^{-4}s=100\mu sec$$

ואילו זמן השידור של 64 בתים הוא

$$\frac{8\cdot64}{10Mbps}=\frac{8\cdot64}{10^7\frac{b}{s}}=51.2\mu sec$$

שזה פחות מ-$2\times$ הזמן לשידור ביט לאורך הרשת. אם נגדיל את גודל ההודעה פי 4, אז זמן השידור יגדל פי 4 ואז זה יהיה חוקי.

- מה יהיה השינוי אם נגדיל את רוחב הפס ל-$100Mbps$?

זמן השידור יקטן פי 10 ולכן נגדיל את ההודעה פי 10.

שאלה נתונים שלושה קודקודים $A-B-C$ (בסדר הזה) עם 5 ק"מ מרחק בין כל שתי התחנות. נניח רוחב פס $B=3Mbps$ ומהירות פעפוע $v_{prop}=6\cdot 10^7\frac{m}{s}$.

- כדי ש-CSMA/CD יעבוד כראוי, מה גודל הפקטה המינימלי כדי ש-A יוכל לשלוח הודעות ל-C?

$$T_{prop}=\frac{10}{6\cdot 10^4}s=\frac{1}{6}10^{-3}s$$
$$T_{trans}=\frac{x}{B}\geq 2T_{prop}\iff x\geq 2\frac{1}{6\cdot 10^4}\cdot 3\cdot 10^7=10^3bit$$

- כדי ש-CSMA/CD יעבוד כראוי, מה גודל הפריים המינימלי כדי ש-A יוכל שלוח ל-B הודעות כראוי?

אותו הדבר כנ"ל, עדיין יכולה להיות התנגשות עם C ולכן אין שינוי.

שאלה נתונה רשת עם slotted ALOHA בגודל סלוט $T_{trans}>2T_{prop}$.

- נניח שנוסיף CSMA (לא CD), האם זה תשפר את ה-goodput?

לא! ה-CSMA יגרום לנו לחכות עד שנראה ערוץ פנוי, אבל כולם יראו ערוץ פנוי בתחילת הסלוט (או מיד אחריו) כי כולם aligned ולכל הפחות לא נקבל goodput יותר גבוה.

- האם הוספת CSMA/CD תשפר את ה-goodput?

גם לא! אנחנו לא מתחילים לשדר עד הסלוט הבא ולכן גם אם נזהה התנגשות ונפסיק לשדר, הסלוט הנוכחי מבוזבז כי לא נתחיל לשדר עד לתחילת הבא.

- האם הוספת CSMA/CD תשפר את ה–goodput במקרה של pure ALOHA עם אותם הנחות על הסלוט?

כן! לכל הפחות לא נהרוס שידורים קיימים ולכן נשפר את ה-goodput.

- האם הוספת CSMA/CD תשפר את ה-goodput מעבר ל-CSMA לרשת נ"ל?

גם כן! נפסיק שידורים שכבר התנגשו בהם ונאפשר לאנשים שעוד לא התחילו לשדר הזדמנות לשידור בסלוט מנקודת מבטם.

מתי כדאי להשתמש ב-CSMA/CD באופן כללי? כש-T_{prop} קטן יחסית ביחס ל-T_{trans}, ואז כשמזהים התנגשות נפסיק את השידור די מהר ולא נבזבז (לפחות לא אנחנו אישית) את הערוץ עם פריימים מושחתים.

שבוע $\mathbb{VI}$ | סוויצ'ים בשכבה 2

הרצאה

נסיים לעסוק בשכבת הלינק, ונדון ב-Ethernet.

ב-Ethernet אחרי התנגשות מחקים זמן מוגרל אחיד על פני $\{0, \ldots, 2^m - 1\}$ כאשר m מספר ההתנגשויות ולכן ככל שנתנגש יותר, כך ההסת' להתנגשות יורדת, כי ההסת' ששנינו חיכינו אותו הזמן הוא די קטן. ה-goodput של Ethernet הוא $\eta = \frac{1}{1+5\frac{t_{prop}}{t_{trans}}}$ שזה כמו ב-CSMA/CD, די טוב, כי אם אנחנו שולטים בגודל הרשת והפקטות, נוכל להשיג goodput לא רע בכלל.

שכבת הלינק ברשתות אלחוטיות

בתקשורת אלחוטית, בגלל שהאות שלנו כל כך הרבה יותר חזק מאותות שמגיעים, לא נוכל לזהות התנגשות בזמן שאנחנו מדברים ולכן במקום לזהות התנגשויות, נצטרך להימנע מהתנגשויות.

בכל רשת תקשורת אלחוטית, יש Access Point שהוא קודוקד מיוחד ברשת שיכול להוות סדרן לשאר הקודקודים (שהם משתמשים ברשת). ה-AP יקבע מי רשאי לשדר, ואז נחלק את הפעולה של כל קודוקד למצב שידור ומצב האזנה.

802.11

האלג' לשידור פריים ברשת אחלוטית בפרוטוקול 802.11 הוא כדילקמן (ומשתמש ב-CSMA/CA שיפורט בהמשך):

1. אם הערוץ פנוי לאיזשהו זמן (DIFS), נשדר את כל הפריים בלי לעצור ונחכה לאישור קבלה. אם לא נקבל, נעבור ל-2.
2. אם הערוץ תפוס, נתחיל טיימר עם זמן אקראי שסופר אחורה כל פעם שיש פרק זמן שקט בערוץ.

   כשהשעון נגמר, נשדר ואם לא נקבל אישור קבלה, נגדיל את זמן ההמתנה ונחזור להתחלה.

כל עוד לא מנסים לשלוח הודעה, ב-802.11 מאזינים ואם מקבלים פריים שלם טוב מאשרים קבלה.

CSMA/CA

איך נבצע את החלוקה למצב שידור והאזנה של הקודקודים?

- אם קודקוד רוצה לשדר, הוא צריך לשלוח בקשת Request-to-Send ל-AP.
- אם יש אישור, ה-AP שולח Clear-to-Send ברשת וכולם שומעים את זה (עד כדי התנגשויות עם RTS-ים אחרים אבל הם מאוד קצרים אז זה זניח). משם הקודקוד המשדר יכול להתחיל לשדר (לדוגמה באמצעות האלג' של 802.11).

הערה אם יש התנגשות ב-RTS-ים, ה-AP יזהה את זה ויאשר רק אחד מהם.

סיכום ברודקאסטים

למדנו בשכבת הלינק על כמה אלג', הפשוטים ביותר הם ALOHA, S-ALOHA, CSMA. ראינו שבמציאות משתמשים בברודקאסט חוטי ב-Ethernet עם CSMA/CD ובברדוקאסט אלחוטי ב-802.11 עם CSMA/CA.

Switch

סיימנו לעסוק במה שקורה ברשתות ברודקאסט. כיצד נחבר רשתות ברדוקאסט? ניזכר שסוויץ', כמו שהוא שובר רשתות ברודקאסט, הוא גם מחבר אותן. מה נדרוש שסוויץ' יעשה?

- יעביר פריימים של את'רנט ועל בסיס ה-MAC address שמופיע על הפקטה, יעביר הלאה למי שצריך את הפריים, כשהוא מתקשר עם יחידות אחרות באמצעות CSMA/CD.
- שקיפות - משתמשי קצה לא מודעים לקיום שלו וחושבים שהם שולחים ישירות למען שלהם הודעה.
- plug-and-play - לא צריך לקנפג את הסוויץ' בשום דרך והוא ידע כבר בעצמו איך לנתב ברגע שיחובר לרשת.

לכל קודקוד יש חיבור מפורש לסוויץ' (כאשר החוט שמחבר בין השניים הוא רשת הברודקאסט היחידה במקרה הזה), וכך כמה אנשים שונים יכולים לשלוח פקטות למענים שונים בלי התנגשויות.

איך סוויץ' יודע איזו יציאה רלוונטית לאיזו כתובת? הוא שומר switch table שאומרת לו לכל חיבור אילו כתובות MAC מתאימות לו. כל רשומה מכילה את כתובת ה-MAC של משתמש קצה, החיבור שאיתו מדברים איתו ו-Time To Live - משך הזמן שעבורו הרשומה הזו רלוונטית.

איך סוויץ' יוצר רשומות בטבלת הניתוב?

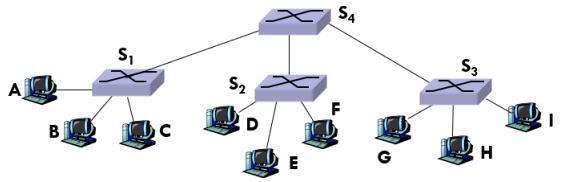
Self-Learning

ללמידה עצמית אין שום קשר ללמידת מכונה. למידה עצמית היא התהליך שבו סוויץ' לומד אילו כתובות יש סביבו.

- כשסוויץ' מקבל פריים, הוא מוסיף (או דורס אם צריך) רשומה לכתובת המקור של הפקטה בטבלת הניתוב שלו לפי החיבור שממנו קיבל את ההודעה.
- מעבירים הלאה את ההודעה:
  - אם אנחנו לא יודעים באיזה חיבור נמצא המען נשלח את ההודעה לכולם ונקווה שמתישהו המען ישלח הודעה בעצמו ונוכל להוסיף אותו כרשומה (לרוב זה יקרה כי השכבות העליונות מחייבות תשובה גם אם היא לא מכילה תוכן חדש).
  - אם אנחנו כן יודעים באיזה חיבור נמצא המען:
    - אם הוא נמצא באותו החיבור כמו הכתובת המקור, נזרוק את הפקטה (לא צריך להעביר לאף אחד).
    - אחרת נעביר אותה לחיבור הרלוונטי.

אם יש לנו כמה סוויצ'ים מחוברים, בגלל שהסוויצ'ים שקופים, האלג' עדיין עובד!

דוגמה אם נרצה להעביר מידע מ-A ל-G, כל החיבורים והסוויצ'ים יעבדו מעולה כי S_1 לא מודע לזה שיש סוויצ'ים בדרך ל-G, מבחינתו אם A רוצה לשלוח ל-G הודעה, הוא צריך לשלוח את זה דרך החיבור הימיני עליון שלו וזה יגיע לשם מתישהו (כמובן אחרי שביצע למידה עצמית וגילה שאכן G שם).



איור 6: טופולוגיה מורכבת של סוויצ'ים, הכל עדיין עובד כמו שצריך

הערה כבר כאן אנחנו מבינים שכתובות MAC אינן הפתרון האולטימטיבי לזיהוי משתמשי קצה כי אם כל אחד יצטרך לזכור את כל הכתובות של כולם בלי אבסטרקציה, נצטרך מיליארדי רשומות בכל ראוטר פעוט.

דוגמה מה קורה אם יש לנו לולאה בסוויצ'ים? לדוגמה אם שני סוויצ'ים מחוברים אחד לשני בשני ברודקאסט דומיינים שונים?

אם קודקוד A ישלח הודעה בברודקאסט אחד, היא תגיע לשני הסוויצ'ים, ואז סוויץ' אחד ישלח את ההודעה בברודקאסט האחר, זה יגיע לסוויץ' השני, הוא ידרוס את הרשומה של A אצלו.

לאחר מכן, אם צריך להעביר משהו ל-A זה לא יגיע אליו אף פעם וגם נשלח את הפקטה שוב ושוב בלי שום סיבה.

הערה הבעיה הנ"ל קורת רק כשיש לנו מעגלים בטופולוגיית הסוויצ'ים. אם לא היו לנו מעגלים, לא היה לנו שום redundancy וזה לא טוב.

לכן כן יהיו לנו פיזית מעגלים כלשהם בסוויצ'ים, אבל בפועל נשתמש רק בחיבורים מסוימים כך שיהוו עץ פורש של גרף הסוויצ'ים שנוצר.

Spanning Tree Protocol

כדי לגרום לסוויצ'ים לנוון פורטים (חיבורים) לשם הגעה לעץ פורש, צריך לתאם בין הסוויצ'ים באמצעות פרוטוקול עץ פורש (STP). האלג' שבו משתמשים מכיל שלושה שלבים (שבמציאות לא קורים אחד אחרי השני כי הכל מבוזר).

1. בחירת שורש העץ.

   (א) כל קודקוד יזכור את המספר הסידורי שלו (שהוא ייחודי).

   (ב) בכל פעם שנקבל מספר סידורי מבחוץ, אם הוא נמוך יותר ממה שאנחנו מחזיקים כרגע נבחר אותו.

   (ג) מדי פעם נשלח את המספר הכי נמוך שלנו לכל השכנים שלנו.

   מתישהו כל המערכת תתייצב על איזשהו מספר.

2. חישוב המסלול הקצר ביותר לשורש.

   ברגע שכל קודקוד ידע מה המסלול הקצר ביותר שלו לשורש, הגרף שנוצר מהמסלולים הללו הוא עץ פורש. נשתמש באלג' בלמן-פורד (מבוזר).

   (א) כל קודקוד ישלח לשכנים שלו מדי פעם את המרחק שלו מהשורש.

   (ב) כשקודקוד מקבל הודעה כנ"ל, הוא יעדכן את האב שלו לשכן הכי קרוב שלו לשורש.

   מתישהו התהליך הזה יתכנס וכולם ידעו מה האבא שלהם במסלול (שיהיה מוסכם על כולם בסוף).

   נוכל למשקל במשקולות אי-שליליים חיבורים (לדוגמה המרחק שצריך לעבור בהם) וככה להשיג נקודת מבט אפילו יותר מדויקת על הטופולגיה.

תרגול

נלמד על החצי השני של שכבה 2 - בהינתן כמה רשתות ברודקאסט - איך נחבר ביניהן באמצעות סוויצ'ים ואיך נמנע לולאות בשביל תפקוד איכותי של הרשת.

הסוויץ', ברגע שמתחבר, מתחיל ללמוד מי מחוברים לאן באמצעות אלג' שראינו בהרצאה (בכל הודעה שמקבלים, מעדכנים את הטבלה ושולחים לפורט הרלוונטי או שעושים flooding). מהאלג' הזה נוצרת בעיה ברגע שיש לולאה בטופולוגיית הסוויצ'ים.

כדי להימנע מלולאות, נגרום לסוויצ'ים לנוון חלק מהפורטים שלהם כך שהטופולוגיה תמפה עץ פורש של הסוויצ'ים (אלג' STP). נחלק את הפורטים של כל סוויץ' לשלושה סוגים:

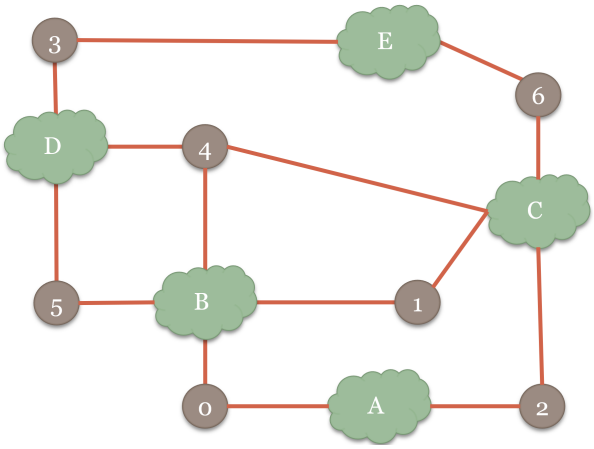
- **R**oot **P**ort - פורט שמחבר את הסוויץ' בדרך לשורש העץ.
- **F**orwarding **P**ort - כל פורט שאינו RP והוא פעיל, מכוון סוויצ'ים אחרים לשורש, וגורם לכך שהסוויץ' אחראי על כל הרשתות שמחוברות דרך הפורט הזה.
- **D**isabled **P**ort - פורטים מכובים, שמקשיבים להודעות "שלום" חדשות (למקרה שקודקוד נופל וצריך לשנות את העץ).

הערה לכל סוויץ' יכול להיות RP אחד וכמה FP-ים ולכל ברודקאסט יש FP אחד וכמה RP-ים.

הערה כשנריץ STP מסומלץ, קודם נמצא את ה-FP-ים מכל הרשתות כך שיכוונו לסוויץ' הכי קרוב לשורש, ואז נמצב את ה-RP-ים בהתאם.

הערה במקרה שב-STP סוויץ' מקבל הודעה משני סוויצ'ים שונים בעלי אותו גובה, הוא מכריע לפי ה-SID (Switch ID) הנמוך מביניהם, ואם זה שווה, הוא מכריע לפי מספר הפורט שממנו קיבל הודעה מכל אחד מהסוויצ'ים (הנמוך מביניהם).

דוגמה נתונה הרשת הבאה, כאשר עננים הם רשתות ברודקאסט וסוויצ'ים מסומנים בריבועים (SID כשם הסוויץ')



איור 7: טופולוגיה של סוויצ'ים

נריץ את האלג' ובסופו של דבר נגיע לכך ש-0 הוא השורש, ואז $1, 2, 4, 5$ כולם בגובה 1, $3, 6$ בגובה 2. ה-RP-ים עתה יהיו 5-B, 1-B, 2-A, 3-D, 4-B ואילו ה-FP יהיו כל מה שאינו RP כאמור או DP, שהם 4-C, 6-C, 6-E.

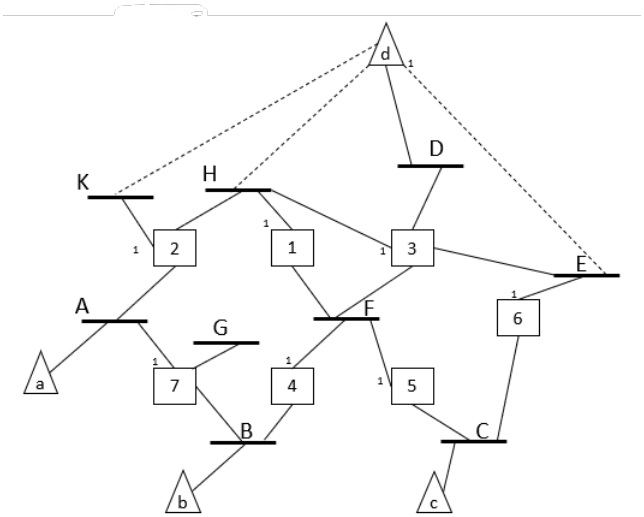
- נפל שורש העץ, איך ה-STP יארגן מחדש את הרשת?

השורש הוא 1. דרגות העצים הן 1 ל-$2, 4, 5, 6$ ו-2 ל-3. עתה ה-FP-ים הם 1-C, 1-B, 4-D, 6-E וה-RP-ים הם 4-B, 3-D, 6-C כאשר כאן 2 ו-5 נפלו כי לא צריך אותם בכלל (הפורטים אליהם מכובים בכל הסוויצ'ים).

- חזר 0 אבל נפל 1, איך נתארגן שוב?

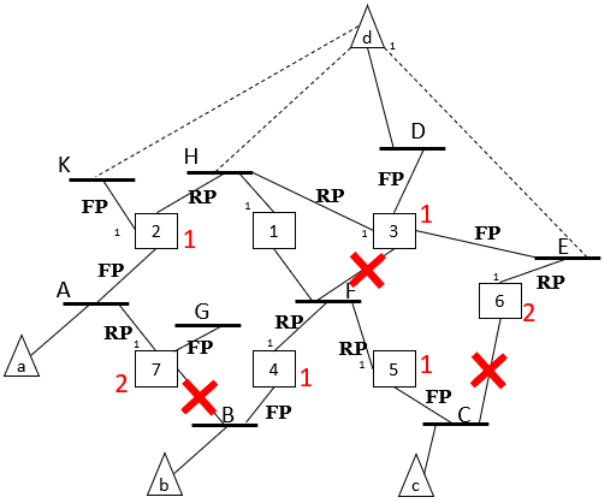
השורש הוא 0. דרגות העצים הן 1 ל-$2, 4, 5$ ו-2 ל-$3, 6$. ה-FP-ים הם 2-A, 4-B, 3-D ו-RP-ים הם 0-A, 2-C, 3-E, 4-D, 0-B ואז $5, 6$ נופלים.

דוגמה נתונה הרשת הבאה (משולשים הם יחידות קצה בתוך רשת, קווים הם רשתות LAN וריבועים הם סוויצ'ים). הפורטים של כל סוויץ' מתחילים מ-1 עם כיוון השעון. נריץ את STP עליו.



איור 8 : טופולוגיה של סוויצ'ים ויחידות קצה

נבחר כשורש את 1 ואז הדרגות יהיו 1 ל-$2, 3, 4, 5$ ו-2 ל-$6, 7$. באלג' יצא העץ הבא



איור 9 : STP שנוצר מהטופולגיה הנ"ל

- כיצד תראה טבלת הסוויצ'ינג של סוויץ' 2 אחרי הרבה הודעות בכל הרשת?

פשוט נסתכל אילו פורטים לא מנוונים מאפשרים הגעה מכל יחידת קצה לסוויץ' (יש מסלול אחד מכל רשת לסוויץ' 2)

כתובת MAC	פורט
a	3
b	2
c	2
d מרשת E	2
d מרשת D	2
d מרשת H	2
d מרשת K	1