

מבוא לרשתות תקשורת | 67594

הרצאות | פרופ' מיכאל שפירא

כתיבה | נמרוד רק

תשפ"ג סמסטר א'

תוכן העניינים

I מבוא

3	הרצאה
3	אפליקציות באינטרנט
3	מהירות מעבר מידע
3	האינטרנט ועניינים נוספים
4	רשת הטלפוניה
4	למה האינטרנט לא נראה כמו הטלפוניה
5	רשתות ברודקאסט

II האינטרנט ועקרונות יסוד

5	הרצאה
5	פקטות
6	פרצים ו-Statistical-Multiplexing
7	Circuit-switched vs Packet-
7	מודולריות ומודל השכביות
8	מודל השכבות
9	תרגול

III מודל השכבות ותקשורת ב-LAN

11	הרצאה
11	פרוטוקולים
12	אפיון חמשת השכבות
14	Multiple-Access-Protocol

שבוע II | מבוא

הרצאה

נעסוק בעקרונות מרכזיים ברשתות תקשורת, ובראשם האינטרנט ואתגרים שיש להתמודד איתם בקשר אליו. האינטרנט 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 שבקרב נלמד. חווית הצפייה 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.

בקרב נדבר על מודל 7 השכבות של אבל לעתה נכוץ אותו ל-3 שכבות: החומרה (ראוטרים, מחשבים, סוויצ'ים וכו'), פרוטוקולים (הדרך שבה מדברות החומרות ביניהם, פרוטוקולים 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 סיקלים 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 בסלול 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. בטלפוניה נשמור

את רוחב הפס המקסימלי הנדרש לקיום 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.

בטלפוניה 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) מיד.

כדי להתגבר על כל הבעיות האלה, ב-64' הוצע השימוש בפקטות במקום קו שמור, ביזור הרשת והעברת הפקטות דרך הקודקודים ברשת ובהמשך גם ניתוחים סטטיסטיים 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 ולכן 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 מנותבת 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 דיילי של זמן עיבוד בקודקוד (וזה יכול להצטבר במרחקים ארוכים או קודקודים חלשים).

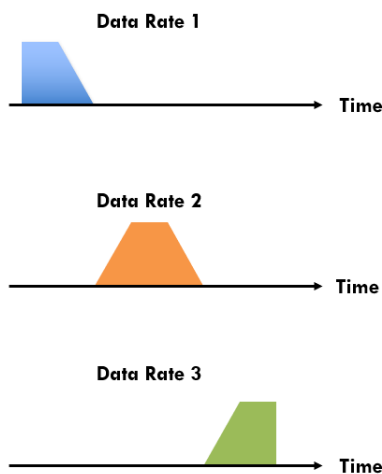
פרצים

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 באיזון 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 לווח 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 לסלטים). בכל פריים נייצר לכל היותר P פקטות ובמוצע 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 Packet ל-Circuit

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 הקווים 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 כתובות 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.

- Physical Layer - העברת ביטים (ואלקטורנים) 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 $P(A|B) = \frac{P(A \cap B)}{P(B)}$ עבור $P(B) > 0$

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

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

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

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

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 $T=t$? מנסחת ההסת' השלמה התשובה היא $pe^{-t} + (1-p)e^{-1000t}$

2. בהינתן שהודעה שרדה t שניות, מה ההסת' שהיא פקטה טובה? מבייס,

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

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 שההודעה התגיע. מתקיים $X \sim Geo((1-p)^n)$ 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.

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

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 $M \sim Pois(50t)$ (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

$$P_{k=2}(t = \frac{15}{60 \cdot 60})$$

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

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

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 $M \sim Pois(\lambda T)$, 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, ומספר האירועים בכל אינטרוול באורך t מתפלג לפי מ"מ פואסון עם פרמטר λt .

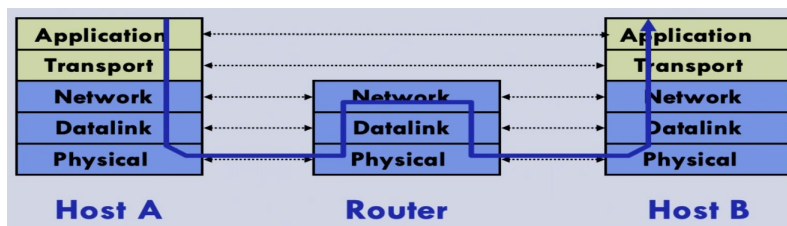
דוגמה מספר האנשים שעולים על אוטובוס הוא לא תהליך פואסוני, כי מספר האנשים תלוי בזמן (ולכן המ"מ שמייצג אינטרוול לא יכול להיות תלוי רק באורך הקטע אלא גם במיקום שלו על ציר הזמן).

שבוע IIII | 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 הפרוטוקולים 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

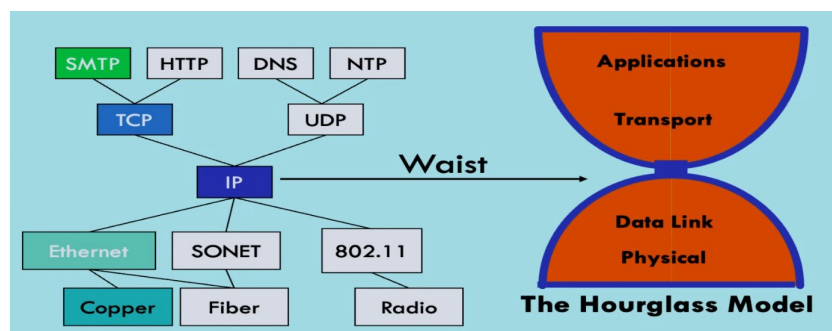
- שירות - תקשורת בין תהליכים, שמבחינה בין שיחות שונות שמתקיימות במקביל בין משתמשים, מספקת אולי אמינות בשליחת הודעות, וויסות קצב הודעות.
- ממשק - שליחת הודעות לתהליך ספציפי ביעד מסוים, העברת ההודעה המתקבלת לתהליך בתוך המכונה.
- פרוטוקול - הגדרת אמינות, פאקטיזציה של הודעות ארוכות, בקרת רצפים (שיגיעו לפי הסדר).
- דוגמאות - TCP, UDP אבל גם T/TCP, SCTP 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-פי), 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 בהדרגה 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' ממין 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 כ"ל הוא Slotted ALOHA. נניח לשם פשטות (1) שכל הפריימים באותו הגודל; (2) שהזמן מחולק לסלטים 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.

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

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

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

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 $p(1-p)^{N-1}$ (הוא שולח וכל השאר לא). ההסת' שקודקוד כלשהו יצליח הוא $Np(1-p)^{N-1}$ כי המאורעות שאנחנו מאחדים זרים. בשביל יעילות מקסימלית נמצא p^* שממקסם את ה"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 סלטים במציאות כי אין סינכרון כאמור, לשם כך יש את Pure ALOHA - כל קודקוד מתנהג כאילו אנחנו כן ב-slotted ALOHA כאשר ייתכן שהסלטים של הקודקודים השונים לא מסונכרנים. במקרה כזה, ההסת' להתנגשות עולה כי לא נקבל רק התנגשויות בתוך פריימים, אלא גם בין פריימים.