

מה זה תקשורת?

תקשורת זה העברת מידע בין מחשבים. כלומר, לא בתוך המחשב עצמו.

תקשורת אלחוטית

נקודת ציון ראשונה זה המצאת הטלגרף (1837) שימוש במורס עדיין קיים בימינו

לא הרבה אחרי- המצאת הטלפון

התחילו ממסרים שלקח ימים ושבועות להעביר, המשכנו עם שעות, היום אנחנו מדברים על זמן אמת.

הטלפון הומצא ב-1876. ב-1915 היה כבר קו טרנסאטלנטי.

1984 - המצאת הטלפון הסלולרי

בתחילת המאה ה-20 התחילו לדבר על רגולציות בעולם הטלפון. תקנים נעשו בשתי רמות- ברמת החומרה הציד צריך להיות מתוקן לתקן מסויים שצויד אחד יכול לדבר עם צויד אחר. ומצד שני, מעבר לצויד, גם ברמת הצורה, צריך שיהיה תקן כדי שאפשר יהיה לדבר האחד עם השני (בהקשר של תדר ודברים אחרים).

כל מכשיר תקשורת צריך לעמוד בתקן מסויים.

ברמת האינפורמציה, בתחילת שנות ה-50 הופיעו המחשבים הראשונים. בשנות ה-90 כבר קמו רשתות של מחשבים והתחילו להעביר אינפורמציה בין מחשבים.

המייילים הראשונים - תחילת שנות ה-90.

כלומר, עולם התקשורת יחסית עולם צעיר.

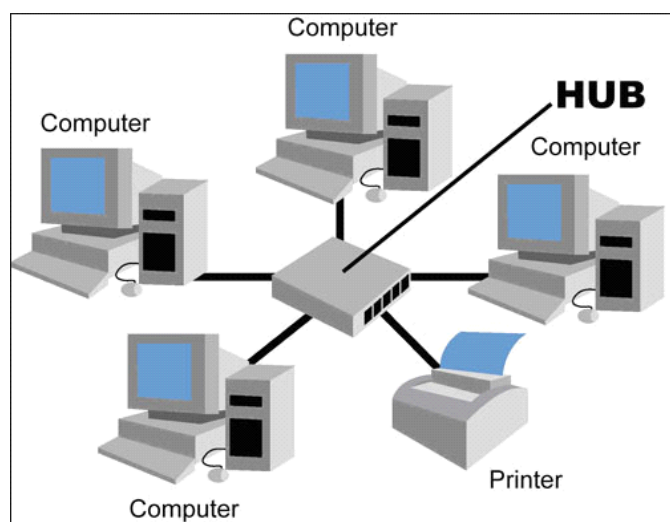
בערך בשנות ה-2000 איחדו את שני הערוצים של טלפניה ודאטא. זה היה נפרד לחלוטין לפני כן.

קו טלפניה בהגדרה חייב ערוץ קבוע שלא מתחלק עם משהו אחר. כשמרימים טלפון, מקבלים ערוץ שקיבל את הזמן שלו כדי שאפשר יהיה לשמוע.

בעבר לא היה הכרחי שיהיה קו כשמרימים את הטלפון. כי היה מספר ערוצים מוגבל. ואז כשמקבלים קו (ערוץ), הערוץ שלך עד שמנתקים.

בשונה מכך, דאטא מועבר בחבילות שרצות. והחבילות הן לא בגודל קבוע, להבדיל מקול. בשלב מסויים הצליחו להעביר voice על גבי דאטא.

האיחוד איפשר את פיתוח כל האפליקציות של שיחות טלפון דרך הרשת.



מבנה של רשת בסיסית:

דבר ראשון יש את המחשבים שהם clients.

באמצע יש Hub. זה אישזהו צויד שמחבר את כל הרשת - כולם

מתחברים אליו.

יש ראوتر שמחבר אחורה

ויש סרברים.

כדי למנוע צביר בקבוק מול המדפסת, יש איזשהו סרבר שמקשר

בינינו לבין המדפסת והוא לא מותנה בקצב ההדפסה, הוא מחזיק

תור להדפסה (זה בא לידי ביטוי באיור של איציק, לא פה).

מושגים

הוגדרו שלוש רמות של רשתות:

LAN - Local Area Networks

רשת מקומית. בד"כ רשת של אתר/ בניין/ חברה. בד"כ מוגבלת בגודל מסויימת (כמה מאות מטרים בערך). למשל אקספרס יושבת על רשת LAN משלה.

רשת בד"כ מסמנים בצורה של ענן.

המשמעות של רשת מסוג LAN היא שכל החברים ברשת מחוברים יחד (יש חיבור פיזי כלשהו). כלומר אפשר להגיע לכל client בצורה

ישירה, בלי לעבור דרך מתווך.

Wan - Wide Area Networks

רשתות הענק שנמצאות בבסיס עולם האינטרנט. בד"כ רשתות של מדינות.

MAN - Metropolitan Area Network

יושב באמצע. בד"כ כולל הרבה רשות LAN שמקשרות באמצעות אמצעי ניתוב כזה או אחר

אמרנו שיש הרבה אמצעי תקינה

חלקם ספציפיים לתחום התקשורת וחלקם לא
 ISO
 ITU-T
 ANSI
 IEEE (I triple E)
 IETF

כשרוצים לשדר, יש כמה צורות לתקשורת.
 התקשורת הכי בסיסית - unicast
 צד אחד מדבר עם הצד השני

broadcast - אחד משדר לכולם
 מה זה כולם? כל מי שנמצא באותו דומיין.
 למשל, כל מי שיושב על אותו הלאן, מקבל את ההודעה של הברודקאסט

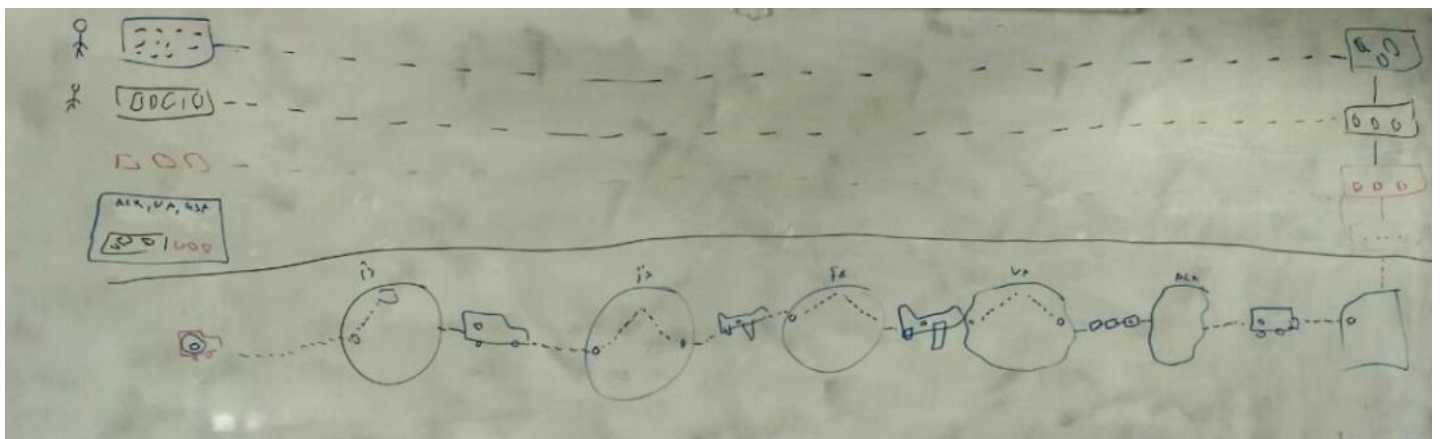
Multicast
 זה משהו בין לבין. זה many to many, אבל בצורה מובחנת. רק מי שנרשם לשירות מסויים נכלל. למשל שיחת ועידה. כולם שומעים את כולם, אבל רק מי שמראש נרשם. זה לא אקראי.

קצב תעבורה
 הממשעות שנגזרת מהקצב - מה אפשר להעביר ומה אי אפשר להעביר.
 רוחב פס
 בעולם האנלוגי רוחב הפס נמדד בהרצים (תדר של השעון). היום הכל עובר לעולם הדיגיטלי והחישוב הוא ביט פר שניה (לא בייט). העולם הדיגיטלי יותר אמין, פחות חשוף לרעשים.

כשהתחילו לחשוב על המודל של עולם התקשורת, ניסו לחשוב איך למדל.
 ניסו לחשוב אם אנחנו מכירים צורת העברת מידע שבאמצעותה אפשר למדל.

מודל שבע השכבות הוגדר ב-1978

מאוד דומה לתהליך רישום פטנטים:



כל הזכויות שמורות ל-X (השם המלא שמור במערכת)

1. השכבה הפיזיקלית - Physical
 פיזית - הכבלים עצמם (אמצעי התעבורה). דבר נוסף שקשור לרמה הפיזיקלית - צריך שיהיה משדר ומקלט בשני הצדדים. ברמה הפיזיקלית מדובר אך ורק ברמת הביטים - 0-1. אין שום מבנה שחורג מרמת הביטים (בין אם זה אנלוגי או דיגיטלי).
 המקבליה לרמה הפיזיקלית בתמונה למעלה - כלי הרכב. מכונות, משאיות, מטוסים, אוניות, לא משנה - זו הרמה הפיזיקלית. זה קורה גם ברשתות תקשורת - יש קצבים שונים, טכנולוגיות שונות, תדרים שונים.
2. data link
 תפקידה העיקרי - להעביר איזשהי חבילה מנקודה לנקודה שמחוברים אליה פיזית. רוצים להעביר מפה חבילה לניו זילנד. אני הרי לא מחוברת פיזית

לאותה נקודה. כדי להגיע לשם, אני צריכה להעביר לנקודה מסויית שבה אני כן מחוברת ומשם זה יעובר הלאה. ברמה הזו לא מחליטים מה הנקודה הבאה - קובעים מבחוץ (מעל) מה היעד הבא. כאן זה הביצוע. האנלוגיה הכי טובה- הנהג. הוא יודע שהוא צריך להעביר מנקודה לנקודה. ברמה הזו המטרה היא אחת- להעביר לנקודה הבאה.

3. רשת - Network

חייבת להביא את החבילה ליעד הסופי ולהחליט בכל שלב מה היעד הבא כדי להגיע בסופו של דבר ליעד הסופי. שלושת השכבות האלה הן מקבילה לכל מה שקורה למטה בתמונה - כל מה שקורה בחלק בתעבורתי

בארבע השכבות העליונות יש כבר דו שיח בין קצה לקצה.

4. העברה - transport

אחראי על טיב ההעברה. כלומר צריך לוודא שהחבילה באמת הגיעה. יש גם בקרה על קצב ההעברה ועד בקורות נוספות. דבר נוסף שיש בשכבה הזו: עד רמה של רמה 3 אנחנו מביאים ליעד הסופי אבל אנחנו לא יודעים מי צריך אותה ולמי צריך להעביר אותה בתוך היעד. כאן כבר לא מספיק שזה הגיע לכתובת, צריך מעין מדוייק וזה נעשה בשכבה 4.

שלוש השכבות העליונות - היום מתייחסים אליהם כשכבה אחת שנקראת שכבת האפליקציה.

בכל מערכת שבה עובדים עם נטוורק, שכבות 2, 3, ו-4 ממומשות על ידי מערכת ההפעלה. אחד הבלוקים המשמעותיים במערכת ההפעלה זה בלוק network.

מעל זה יש את שכבת ההאפליקציה. כלומר היום הנטיה היא לומר שיש 5 שכבות.

5. session

השכבה שבודקת האם אנחנו רשאים לעלות לאפליקציה. יכול להיות שהגענו למען הנכון, אבל אנחנו לא רשאים. למשל, אפליקציות מסויימות דורשות יוזר וסיסמא, אם אין לנו, אנחנו לא רשאים לעלות. הסשן זה מה שמאפשר לשתי אפליקציות לדבר באמת. לפעמים אין מה לממש את השכבה הזו, אם אין שום הגבלה להרשאה.

6. presentation

הפורמט שבו יושב הדאטא.

כל נושא ההצפנה, דחיסה. הרעיון הוא שאם בצד אחד עשינו דחיסה, השכבה הזו צריכה לדעת מה לשעות את זה כדי להחזיר את האינפורמציה כמו שהיא. השכבה הזו דואגת לפורמט של הדאטא.

7. האפליקציה עצמה.

גם ברמת האפליקציה יש פרוטוקולים. נגיד שאני רוצה לבנות משחק, צריך לבנות פרוטוקול. פרוטוקול זה איזושהי תקן שכל הצדדים מכירים אותו וככה אפשר לדבר ולהבין אחד את השני.

בכל שכבה יש פרוטוקולים שמממשים את השכבה ותמיד שני הצדדים צריכים להכיר ולהבין.

גם השכבות צריכות לדעת איך לתקשר ביניהן. כששכבה 3 קובעת מה היעד הבא, היא צריכה לתקשר בצורה כזו ששכבה 2 תבין.

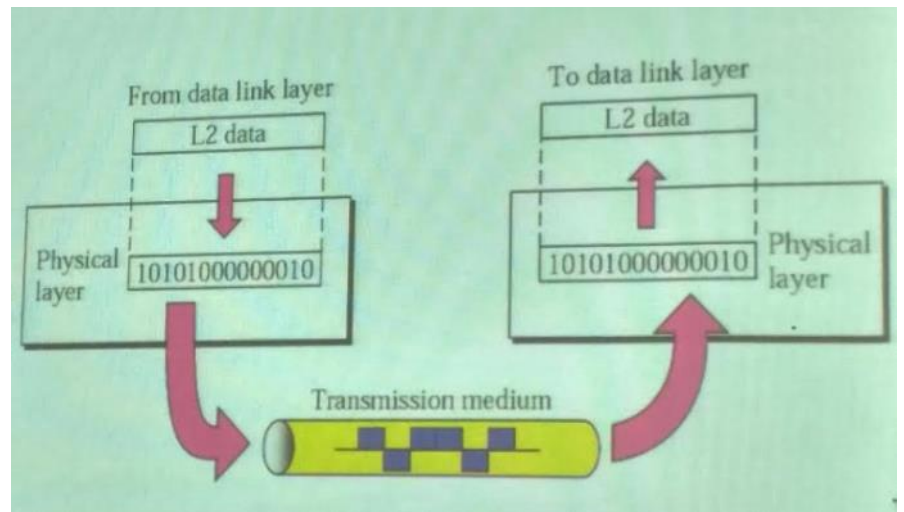
בפועל, כשאנו כותבים תוכנה שמשתמשת בתקשורת, אנחנו נממש רק את רמת האפליקציה (שזה כאמור שלוש השכבות העליונות במודל).

לגבי הרמה הפיזיקלית- פעם כל רמה פיזיקלית הייתה תומכת רק בקצב (תדר) אחד. היום יכול להיות קצבים שונים וזה מסבך כי צריך להסתנכרן. יש תהליך של סנכרון די מורכב. אלא הבעיות הכי בסיסיות- זה הדבר הראשון שצריך לפתור לפני שבכלל אפשר להתחיל לעבוד. הדבר נכון על אחת כמה וכמה היום, כשיש תדרים מאוד גבוהים. ביט אחד שמשלבש גורם ל"חבילה" להעלם.

משכבה 4 ומעלה, יש קשר ישיר בין שני הצדדים. ראינו את זה גם במודל הפטנט.

שכבה 1:

ברמה הזו רק ביטים



שכבה 2

זו הפעם הראשונה שיש כבר חבילה - מקבץ של ביטים שיש להם משמעות. החבילה ברמה הזאת נקראת frame כל יחידת קצה ברמה הזו נקראת node network - שני node או יותר שמחוברים ביניהם.

יש שני חלקים בשכבה הזו- האחד קרוב לשכבה (הפיזית) הראשונה שגם זה חומרה (MAC) Media Access Control - תפקידו לזהות שניתן לשדר ולהעביר את התשדורת לרמה הפיזיקלית וגם לזהות collision. ברמה הזו אין הירארכיה. מי שרוצה לשדר יכול "להתפרץ" ולשדר. זה יכול לגרום למצב שבו שנים ביחד מתפרצים. אחד התפקידים של ה-MAC הוא לזהות שהיה collision, לקחת צעד אחורה ולנסות לשדר שוב. ככל שיש יותר נודים, יש יותר סיכוי לקולוזין. והחלק הלוגי- LLC - logical link control

בכל מחשב יש NIC- network interface card

כשמגיע משהו מהערו וה-MAC אמר לבנות מזה frame, הוא מעלה את זה לחלק הלוגי של שכבה 2. ואז קורה אינטארפט. מערכת ההפעלה מסתכלת ומליטה אם להעלות את זה הלאה. ברגע שמערכת ההפעלה מזהה שזה שייך לאפליקציה מסויימת, היא כבר מכוונת.

מה שמאפיין וייחודי לשכבה השנייה. השכבה מוסיפה trailer. מגיע דאטא ומתווסף header (שזה משהו שקורה בכל השכבות) אבל גם trailer והוא מכיל אישזהו חישוב שנעשה על הדאטא ונשמרה ומטרתה שהצד השני יעשה את אותו החישוב כדי להגיע לאותו הדאטא. המקום שהכי מועד לקלקול דאטא זה הרמה הפיזיקלית, שהיא הכי מושפעת מרעשים. ולכן חשוב לבדוק את זה ברמה השנייה- שכל מה שהגיע הגיע באופן תקין. אם זה לא הגיע באופן תקין, החישוב מאפשר לפעמים למצוא איפה השתבש משהו ולתקן. אם אי אפשר לתקן, החבילה תזרק. השכבה הזו לא תעלה משהו שהוא לא תקין.

כלומר כששולחים משהו זה יורד מרמת האפליקציה עד הרמה הפיזיקלית. שכבה 2 עושה חישוב לפני הרמה הפיזיקלית כדי שהצד השני יוכל לבדוק אם משהו השתבש. אז זה מועבר לשכבה 2 ואז כשזה יוצא בצד השני מרמה 1 לרמה 2 נעשית בדיקה- ששום דבר לא השתבש וזה נעשה על ידי בדיקת ה-trailer.

שכבה 3

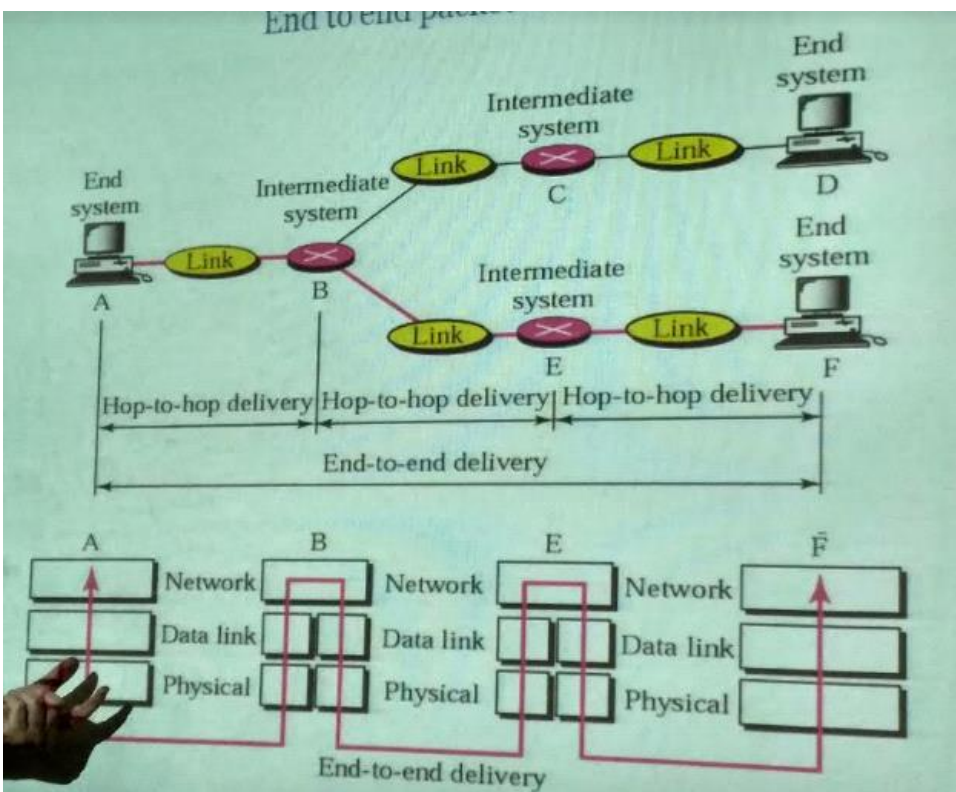
יש פרוטוקול אחד וידוע- IP

מה התפקיד של השכבה הזו?

התפקיד העיקרי- החלטת הניתוב מה היעד הבא של חבילה מסויימת כדי להביא אותה ליעד הסופי.

IP זה connectionless - לא יוצרים קשר עם הצד השני. זה פשוט אומר להעביר למקום מסויים (מחשב מסויים), אבל אם בדיוק סגרו את המחשב, המשמעות היא שזה לא יגיע, אבל אין וידוא שיש משהו שיקבל.

גם שכבה 3 מוסיפה header (כאמור כל שכבה מוסיפה). מופיעה שם כתובת היעד וגם מי שלח.



דוגמא למה קורה בין שלושת השכבות הנמוכות.
אנחנו רוצים לשדר משהו בין מחשב A ל-F.

שכבה 3 במחשב A החליט שכדי להעביר ל-F, היעד הבא זה B. ואז שכבה 3 מעבירה לשכבה 2 את המידע. B במקרה הזה זה ראטר, אבל לעניינינו זה פשוט node בתוך הרשת.

עכשיו זה הגיע לרמה הפיזיקלית ומבחינת שכבה 2 זה הגיע ליעד כשזה B. ואז מעלים שוב לשכבה 3 ובשכבה 3 רואים שזו חבילה שמיועדת ל-F. כיוון ששכבה 3 רואה ש-B זה לא היעד הסופי, היא מודיעה לשכבה 2 לשדר את זה הלאה ל-E. אותו דבר קורה שוב עד שמגיעים ל-F. ב-F שכבה 3 רואה שזה היעד הסופי והיא מעלה את זה לשכבות מעל.

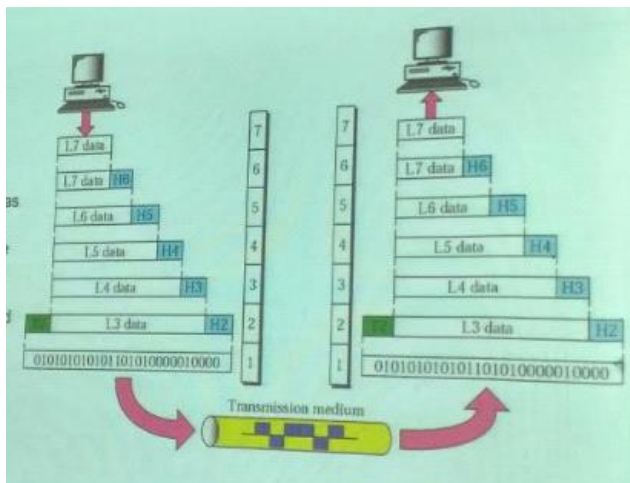
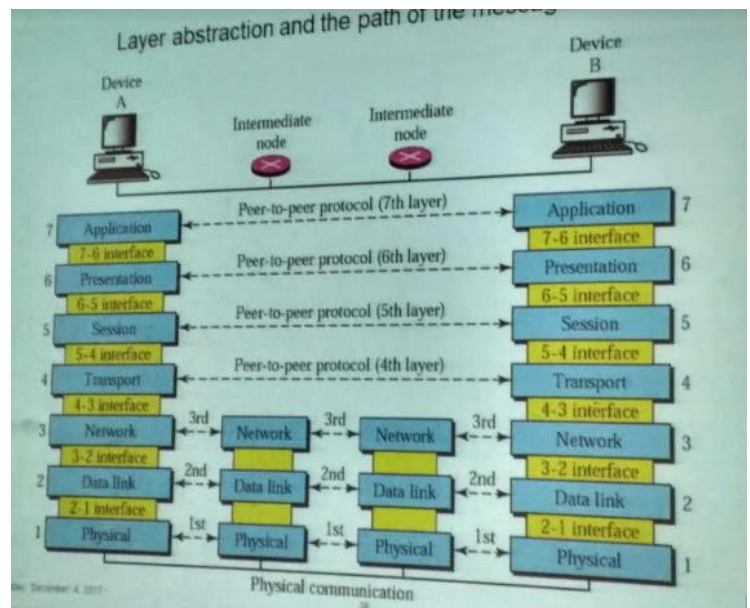
שכבה 4
היחידה כאמור שכבר יוצרת קשר בין שני הצדדים.
וידוא של אמינות ההגעה.
שכבה 3 מוגדרת כ-best effort. אין הבטחה שמהו מגיע. השכבה הראשונה שנותנת אמינות זו שכבה 4. בנוסף, בשכבה הזו מוסיפים עוד רמת מיעון-לאן צריך להעביר את החבילה.
יש שני פרוטוקולים נפוצים עיקריים.
udp, tcp
נלמד את שניהם.
ברמת השכבה הזו, היא מקבלת משכבה 5 דאטא והיא יכולה להחליט לפצל לכמה חבילות. באופן כללי שכבה 5 צריכה להיות שקופה לשכבה 4.
יש גם header ברמה הזו. מה שמופיע שם בין השאר זה אותו מיעון נוסף שלא היה קודם.

שכבה 5
מחסום שמוודא הרשאה כאמור

שכבה 6
אחראי על המבנה של התוכן. אם אנחנו מצפינים, דוחסים וכד. ברמה הזאת, אם צד אחד יוצר פורמט מסוים, הצד השני צריך להחזיר לרמת המידע המקורית. זה יתווסף ל-header בשכבה הזו

שכבה 7
האפליקציה עצמה
כאמור, כל מה שאנחנו נכתוב, אנחנו נכתוב ברמת האפליקציה.

התהליך כולו:

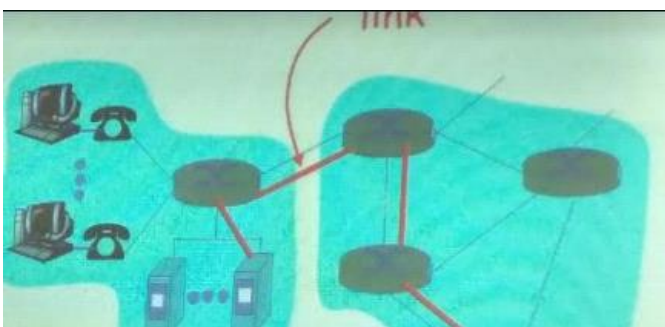


כאן צריך לשים לב שברמת השליחה כל שכבה מוסיפה header. שכבה 7 שולחת את הדאטא. שכבה 6 מוסיפה header ושולחת הלאה. מבחינת שכבה 5, כל מה שהתקבל משכבה 6 זה הדאטא ואליו היא מוסיפה את ה-header שלה וכך זה ממשיך הלאה עד הרמה הפיזיקלית. בקבלה של החילה קורה תהליך הפוך ובו כל שכבה יודעת לפענח ולהוריד את ה-header הרלוונטי.

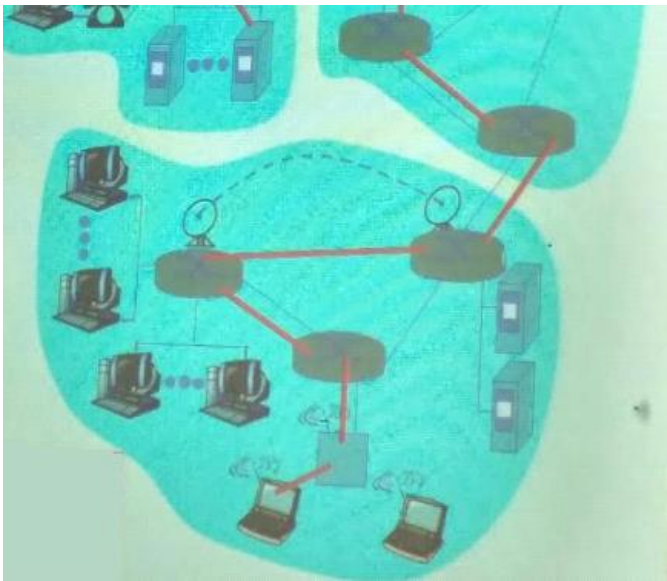
כשמדברים על הרמה הפיזיקלית, הסנכרון נקבע לפי תדר השעון.

איך מזהים שהתחילה חבילה? יש חתימת קוד (בביטים כמובן) ששמורה לתחילת חבילה. פטרן ספציפי של שמונה בתים שאומר שזה תחילת frame. אחכ מתחיל הדאטא.

כבלים רגישים לרעשים. בשביל זה יש שכבת סינון שאמורה להגן. הכבל הכי נפוץ - twisted pair. כבל אחר - copper. יותר קשיח. מוחלף עם הזמן בסיב אופטי.



שכבה 2 אחראית על להעביר פריים מנקודה לנקודה (מ-node ל-node באותה רשת). תמונה- כל קו אדום זה לינק שעליו אחראית שכבה 2. כדי להעביר מנקודה לנקודה, אנחנו צריכים איזשהי רמה של כתובת - addressing ברמה של השבה הזו אנחנו מדברים על MAC address. על כל רכיב תקשורת יש MAC צרוב עליו. להבדיל מכתובת IP שהיא כתובת לוגית, ה-MAC הוא פיזי, צרוב. אנלוגיה- לכולנו יש מספר זהות שמלווה אותנו כל חיינו, זה צרוב. מקום המגורים לחילופין יכול להשתנות. כתובת MAC היא ייחודית ברמת כל העולם. כל חברה שמייצרת



להבדיל מכתובת IP שהיא כתובת לוגית, ה-MAC הוא פיזי, צרוב. אנלוגיה- לכולנו יש מספר זהות שמלווה אותנו כל חיינו, זה צרוב. מקום המגורים לחילופין יכול להשתנות. כתובת MAC היא ייחודית ברמת כל העולם. כל חברה שמייצרת רכיבי תקשורת קונה פול של כתובות ואז זה רק שלה.

אפשר לעבוד ברמת half-duplex או full duplex. זה אומר משדרים וקולטים בו זמנית (full) או לפי הסדר (half). ה-MAC היא 48 ביטים (להבדיל מכתובת IP שהיא 32 ביטים).

IP מבוסס במספרים דצימליים, ה-MAC הוא בהקסה (מופיע או אם מקף או עם נקודתיים).

לפעמים צריך לשלוח בקשה לכל מי שיושב על הרשת- broadcast

כתובת הבודקסט - FF-FF-FF-FF-FF-FF
מי שמוכר את כתובות ה-MAC זה IEEE

ARP- address resolution protocol

שכבה 3 אומרת לשכבה 2 לשלוח חבילה ל-IP מסויים. כלומר, שכבה 3 מדברת בשפת IP! איך שכבה 2 תדע מה ה-MAC?

לפעמים אני יודעת. יכול להיות שיש לי טבלה שכבר מקשרת בין IP מסויים לבין MAC. אבל אם אני לא יודעת, אני צריכה לבקש ממי שאמור לקבל שידוע לי מה ה-MAC שלו. אבל איך אני אפנה אליו? אני אשלח broadcast לכל מי שמחובר אלי ברשת - שמי שיש לו IP כזה שישלח לי את ה-MAC שלי. אותו אחד כבר ישלח לי את זה חזרה ב-unicast. דבר ראשון שאני אעשה, אני אשמור את זה ב-ARP table. אבל IP זה לוגי, זה יכול לשהתנות. לכן הטבלה היא זמנית. היא כוללת:

IP

MAC

TTL- time to leave

לרוב הזמן לעזוב זה בדיפולט 20 דקות. ואז זה נמחק. פעם הבאה אני אצטרך שוב לשלוח ברודקסט כדי לבדוק מה ה-MAC של IP.

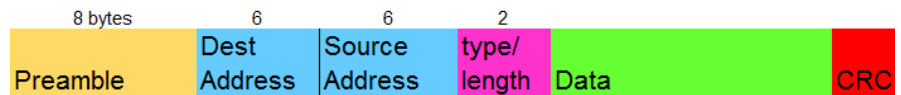
אם אני רוצה לצאת לאינטרנט, מה קורה ברמת שכבה 3. היא מזהה שאני רוצה לצאת מחוץ לרשת והיא מודיעה לשכבה 2 שהיא צריכה לשלוח את החבילה ל-IP ששייך לראוטר.

יש כלי שנקרא strace

הוא משקף את כל ה-system calls למערכת ההפעלה. מפענח את כל הפרמטרים וגם מה מערכת ההפעלה הגיבה למה זה כזה חשוב?

בעולם של תקשורת יוצאים מעולם המחשב שלי וזה קודם כל קורה דרך מערכת ההפעלה. איך אני יכולה לדעת מה עובר ברשת? מה הגיע לרשת? יש כלים שמנתרים מה עובר ברשת. הם גם מפענחים את כל השדות של הפרוטוקולים. כלי כזה הוא Wireshark משקף את מה שקורה ברשת. זה מאפשר לדבג משלב התכנית עד הרשת (או מה שקיבלתי- אם הוא נכון או לא). זה מאפשר לעשות הקלטה של מה שעובר ברשת וגם מאפשר הקלטה מבוקרת- לשים פילטר לפרוטוקול מסויים ואפילו פרוטוקול שמיועד לי. אפשר גם להקליט הכל ואחרי שהקלטתי לפלטר.

פרוטוקול אינטרנט



בהתחלה אמרנו שיש תבנית מסוימת שמזהה תחילה של חבילה אח"כ בא ה-header החצי הראשון הוא ה-destination והשני זה ה-source address (6 בתים ו-6 בתים). למה חשוב הסדר הזה? אמרנו שהשכבה השניה מזהה התחלה ומתחילה לקלוט. קודם כל בודק אם זה מיועד אליו והוא יודע את זה לפי ה-destination אם זה שלו (או שזה ברודקאסט). אם כן- יש לו סיבה להמשיך לבדוק. אם לא הוא לא צריך להמשיך לקלוט דברים נוספים. אח"כ 2 בתים שהם הטייפ או האורך ואז הדאטא עצמו. אח"כ ארבעה בתים של טריילר - crc פריים של אינטרנט מוגבל בגודלו בשני הכיוונים- יש לו גם גודל מינימלי וגם גודל מקסימלי. הגודל המינימלי זה 46 בתים בדאטא. המקסימום- 1500. פלוס הבתים שדירנו על זה- 14 כתובת ועוד 4 של הטרילר.

איך יודעים אם זה type או length? הטייפ זה מה הפרוטוקול שהוא מכיל והערך המינימלי שהוא מכיל זה 800 הקסה. אם הגודל המקסימלי של הדאטא הוא 1500 בקיצור- אם יש מספר בין 0 ל-DC5 זה אורך, אם זה 800 ואילך זה הטיפוס מה זה להעלות חבילה משכבה 2 למעלה? למי אני מעבירה? לאיזה פונקציה לקרוא? קוראים לפונקציה של הפרוטוקול. המשמעות של הטיפוס- זה מאפשר לומר לאיזו פונקציה לקרוא. אפשר לראות את זה כמעין ערוצים שיוצאים משכבה 2 וצריך להבין לאיזו פונקציה לקרוא - לאיזה ערוץ. משהו שמאפיין כתובות. אמרנו איך נראית כתובת של ברודקאסט. מה אם אני רוצה multicast? ברור שאני לא יכולה לשלוח לכתובת MAC ספציפית. ואני גם לא יכולה לעשות ברודקאסט. איפיון איך נראית כתובת multicast ב-MAC הבית הראשון (הגבוה) - הביט הנמוך שלו יהיה אחד. כלומר הביט הראשון יהיה אי זוגי. כלומר זה עדיין כתובת ספציפית, אבל זו כתובת לוגית, היא לא צרובה.

נניח שיש לי צינור אחד שעובר בצורת מעגל והרוחב פס שלו זה G1 יש לי הרבה נודים שמחוברים אליו. ככל שיש יותר נודים, למעשה במוצק רוחב הפס שלי יורד. כי בזמן ממוצע אני מתחלקת על הפס

הבעיות בתצורה כזו: collision capacity רשת לא מוגבלת באורכה (עד כמה מאות מטרים). אז יש לי גם הגבלה של אורך הכבל (ערוץ). מה קורה אם יש נתק בכבל? הכל נופל. infra-structure - אם הגדרנו כמות מסוימת של נודים, יש בעיה להוסיף מבחינת התשתית? בעיה נוספת- ארכיטקטורה (לא ברור)

ואז באו ואמרו- במקום שזה יהיה מסודר במעגל, ננסה לסדר בצורת כוכב שמו HUB שיושב באמצע. לוקחים מקטעים של הרשת ומחברים להאב. ככה יצרו תצורת כוכב. מה קורה בתוך ההאב? זהו מוצר חומרה נטו. מה שקורה בפנים, כולם מחוברים בפנים. כמו מפצל שקעים. ההאב עובד בשכבה הראשונה. ההאב לא פותר לא קולידז'ים ולא קאפסיטי. הוא פותר את בעית האורך, בעיית הניתוק וגם את בעיית התשתית

מה עם הבעיות שעוד לא התמודדנו?

ואז אמרו- ננסה להוסיף תוכנה להאב. אין חיבור פיזי, שום דבר לא מחובר. אבל יש תוכנה. כשמיגע פריים, אני אנסה ללמוד כל פעם לאיזה מהפורטים זה שייך. אם הייתי יודעת שהיעד של פריים מסוים נמצא במקום מסוים, אני יכולה לוגית להעביר אותו למקום המסוים ורק הוא יראה, השאר לא יראו. בהאב כולם רואים. זה מוריד את הסיכוי לקולידז' כי "נלחמים" על פורט מסוים ויש הרבה כאלה. זה גם מקטין את בעיית הקאפסיטי. הקאפסיטי למעשה גדל כמספר הפורטים.

אבל איך יודעים לאן החבילה אמור להגיע? אני צריכה ללמוד לשייך תוך כדי תהליך MAC לפורט. אם אני לא יודעת, אני יכולה לשלוח לכולם. זה לא ברודקאסט, פשוט מנתבים לכולם. איך אני יכולה ללמוד? כל פריים שמגיע, לפי ה-source אני יודעת לאיזה פורט הוא שייך. ואז אני צריכה לנהל איזשהי טבלה שמכילה

MAC	פורט	TTL
-----	------	-----

הגישה אל הטבלה הזו צריכה להיות מאוד מאוד מהירה. כי היא צריכה לקבל החלטה על כל פריים שמגיע. לכן הטבלה הזו יושבת בזכרון שהוא מאוד מאוד מהיר. אחרי זמן מסוים שלא הייתה פעילות ב-MAC מסוים, זה נמחק.

ההפצה לכולם נקראת flading

המכשיר הזה שהחליף את ההאב נקרא bridge

ואז באו ואמרו, אפשר לבנות multi-bridge ויהיה בו הרבה יותר פורטים ובכל סלייס יהיה פחות מחשבים, אפילו מחשב לכל פורט. ל- multi
bridge כזה קוראים switch layer 2

פה איבדתי את זה בחיפושים אחר מתכונים טבעוניים לסופגניות...