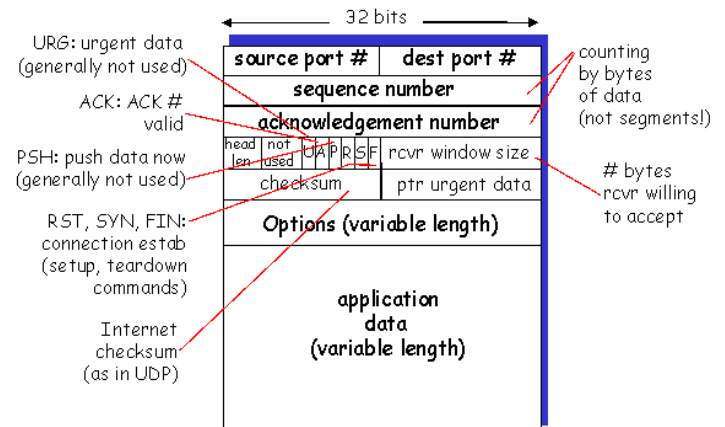


שכבת התעבורה L4

תקשורת לוגית בין תהליכי יישומים בין מארחים שונים

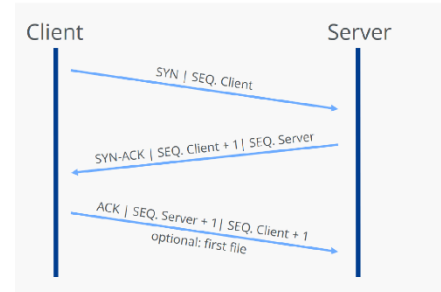
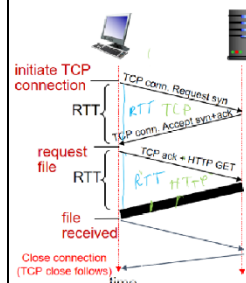
TCP (אמין)

מבנה



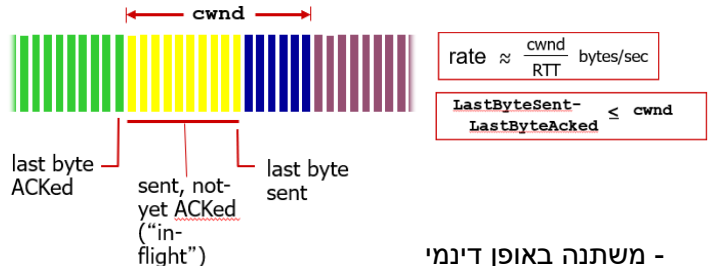
בקשת קובץ

לחיצת יד



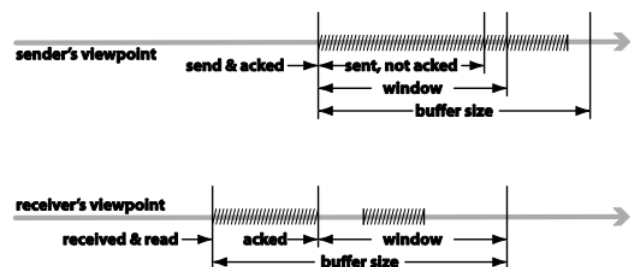
חוצצים (חלון בקרת עומס cwnd)

sender sequence number space



משתנה באופן דינמי

החוצץ בצד המקבל מאפשר לשלוח כמות גדולה של מידע לפני שהוא מקבל אישור למנה כלשהי - מאפשר לקבל הודעות גם כשהתהליך המקבל עסוק - מאפשר בצד השולח מאפשר לתהליך לרוץ בשוטף ולנצל את הרשת



לכל מנה מצורפת תחילית עם השדות הבאים

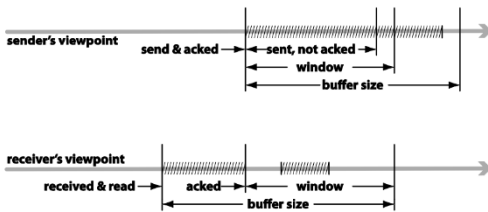
- כמות המידע שהמנה נושאת בבתים (0 הוא ערך מותר)
- מיקום המידע ברצף מודלו 2^{32} (השדה שסודר)
- מיקום הבית האחרון שנקלט ללא חורים על ידי הצד
ששלח את המנה (לא ניתן לאשר מנה שמנות קודמות לה

(חסרות)

- גודל חלון מודיע לצד השני כמה בתים שלא אושרו מהמשך הרצף חוצץ הקבלה יכול להכיל (16 סיביות) Checksum-

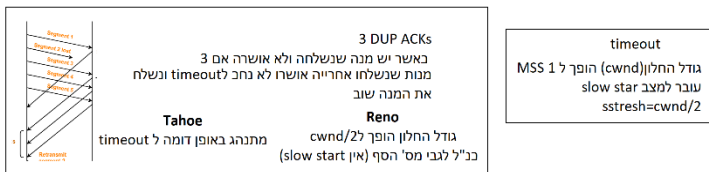
דוגמה לחור אצל המקבל

SN- מאיפה להתחיל את הפעולה
AN- עד איפה המידע רציף

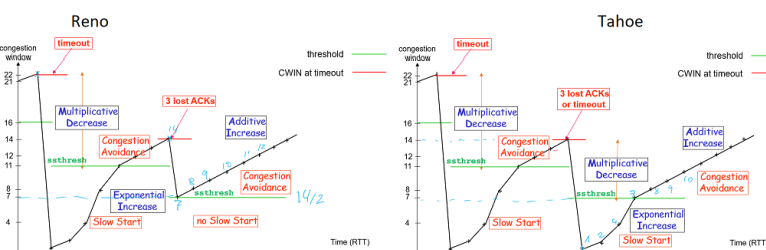


לאחר שהחור מתמלא המקבל שולח אישור לא רק על החור אלא עד סוף הרצף שברשותו אחרי החור והbuffer של השולח יתעדכן בהתאם אם מנות לא אושר לאחר זמן המוגדר בטיימר השולח ישלח את המנה הישנה ביותר שלא אושרה שוב ויפעל את הטיימר מחדש (אם המנה כבר קיימת אצל המקבל ורק האישור שלא לא הגיע הוא ישלח אישור ללא שינוי המנה הקיימת)

slow start(SS) - שיטת אופטימיזציה של גודל החלון כאשר הוא רק נוצר ההגדלה תתבצע במהירות ובכל פעם שנקבל אישור נכפיל את המידע שנשלח עד שנגיע למספר הסף ערך הסף (ssresh) - רק להפסקת slow start כשנגיע למצב בו אנחנו עוברים אותו בכמות החבילות נשווה אליו נסיים את slow start ונעבור לcongestion avoidance (CA) - עבודה רציפה נגדיל בזהירות את החלון וההגדלה תתבצע באופן ליניארי



*תמיד נאפס את הטיימר כשחבילה מגיעה ונפעיל על הבאה שעדיין לא אושרה
* חיסרון- עד שנגיע לשליחת מס' החבילות האופטימלי יתכן שנסיים את העברת הקבצים

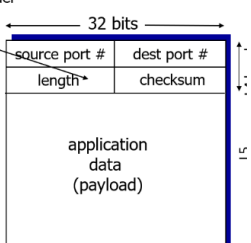


UDP (מהיר אבל לא אמין)

- אין לחיצת ידיים או אישור על קבלת מידע
- אין בקרת עומסים המידע נשלח במלואו
- header קטן

אין מצבים אצל מקבל או שולח מבנה

length, in bytes of UDP segment, including this header (L4 header)



UDP segment format

שכבת הרשת L3

נתב (router) - מעביר מידע בין שני מארחים ברשתות שונות עובד על שכבה 3

מתג (switch) - מעביר מידע בתוך אותה הרשת, שקוף למשתמשים בו על שכבה 2

control plane/routing - קובע את המסלול שהחבילה תיקח מהמקור עד היעד בהתבסס על הסידור הפיזי או הלוגי של הרשת

*איטי ורץ ברקע, צריך להכיר את סידור האינטרנט, מעדכן את forwarding table בשתי גישות

גישה מסורתית - קיימים פרוטוקולים לכל נתב כך שהם משתפים בניהם מידע

SDN - כל המידע נשמר במקום מרכזי ומקור זה יכול לבצע את חישוב המסלול במצב זה בנתב עדיין קיימת הטבלה אבל העדכון שלהן מתבצע ע"י המקור המרכזי דורש פרוטוקול open flow

מעניק גמישות ומאפשר השתנות מהירה של הרשת

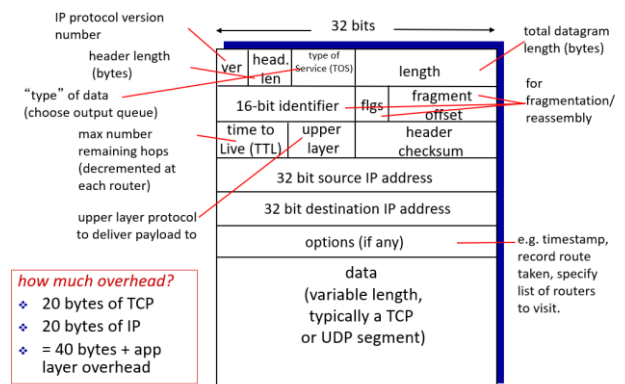
data plane/forwarding - מעביר חבילות בין נתבים בלי לדעת את כל הדרך (גזירה של הנתב)

*מהיר ורץ על כל חבילה, מכיר רק את המעבר הבא, משתמש ב forwarding table

IP-Protocol

- מעביר מנות משולח למקבל
- בצד המקבל מעביר מנות לשכבת התעבורה
- נתב בוחן שדות כותרת בכל חבילות IP העוברות דרכו
- חסר מצב ואופן חיבור

מבנה



LPM - חבילה מגיעה ← נשלף רק את כתובת היעד מהטבלה ← נמצא את היעד בעל התחום הארוך ביותר שמתאים לכתובת ונשלח אליו

קבלת כתובת IP

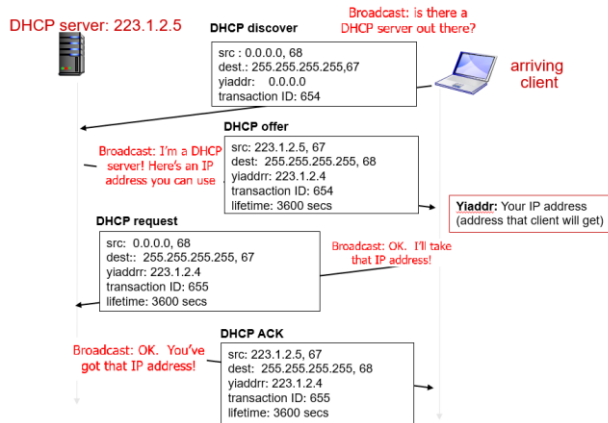
אפשרות 1 - מובנה ע"י מנהל המערכת בקובץ

אפשרות 2 - DHCP

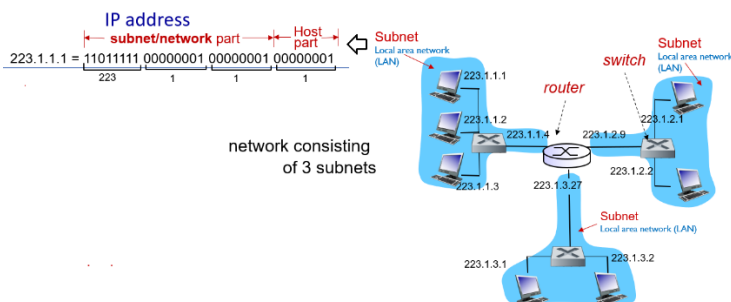
פרוטוקול לקבלת IP משרת DHCP כאשר כל התקשורת בין הלקוח לשרת מתבצעת בbroadcast כדי שאם יש עוד DHCP server ברשת הוא יעודכן בקו הלקוח בתהליך

- מתבצע וגם מתאושש מבעיות באופן אוטומטי
- שימושי לקונפיגורציה שמשתנה הרבה
- רק מאחריים (host) פעילים מקבלים IP ולכן דורש פחות כתובות
- לכתובות שניתנות יש timeout ואם לא רוענו בזמן זה השרת ייקח את הכתובות בחזרה

הלקוח שולח בקשת חידוש לשרת



Subnet - ממשיקי התקן עם אותו רשת משנה של כתובת IP נקרא גם LAN, network



ממשיקי התקן על אותו subnet יכולים לגשת זה לזה ללא התערבות של הראוטר

המסכה (mask) של subnet -

K most significant bits for network mask = '1'

$32-k$ least significant bits for host = '0'

network mask = 255.255.254.0

11001000 00010111 00010001 10110101 /23

11111111 11111111 11111110 00000000 = 255.255.254.0

11001000 00010111 00010001 10110101

NETWORK HOST

קבלת כתובת IP

כתובות שמורות

broadcast - כתובת הרשת + host#=11...1

משמש להעברת הודעה לכל מי שברשת

כתובת subnet - כתובת הרשת + host#=00...0

מבוצע שימוש רק בטבלאות ניתוב

NAT router

כל החבילות של כל המכשירים שעוזבות רשת מקומית חולקות את אותה הכתובת ומובדלות ע"י port הכתובות שהמכשירים מקבלים לא נתמכות באינטרנט ומשמשות עבור כל חבילה נשנה

(source IP address, port #) → (NAT IP address, new port #)

כל שינוי שעשינו נשמור בטבלה NAT translation table

ועבור כל חבילה נכנסת נשנה

(source IP address, port #) → (NAT IP address, new port #)

*אין צורך במרווח כתובות, יכול לשנות כתובות ברשת בלי לעדכן בחוץ, אפשר להחליף ספק בלי לשנות כתובות, מכשירים ברשת לא גלויים לעולם (אבטחה)

NAT translation table	
WAN side addr	LAN side addr
138.76.29.7. 5001	10.0.0.1. 3345
ip and port allocated	real ip and port

שכבת הערוץ L2 (מימוש ב NIC adapter או צ'יפ)

שגיאות

EDC - ביטים לאיתור ותיקון שגיאות (לעיתים נדירות מפספס

שגיאות)

Parity checking - מזהה שגיאה של ביט אחד מוסיף לכל

שורה ב data ביט parity

even parity	
1	מס' זוגי של 1
0	מס' אי זוגי של 1

odd parity	
0	מס' זוגי של 1
1	מס' אי זוגי של 1

1	0	1	0	1	1	Even parity bit
1	0	1	1	0	0	parity error
0	1	1	1	0	1	
0	0	1	0	1	0	parity error

		row parity	
$d_{1,1}$	\dots	$d_{1,j}$	$d_{1,j+1}$
$d_{2,1}$	\dots	$d_{2,j}$	$d_{2,j+1}$
\vdots	\vdots	\vdots	\vdots
$d_{i,1}$	\dots	$d_{i,j}$	$d_{i,j+1}$
$d_{i+1,1}$	\dots	$d_{i+1,j}$	$d_{i+1,j+1}$

checksum - קיים בשכבת התעבורה, תוצאה של פונקציה

ידועה מראש שמחושבת על המידע המקורי

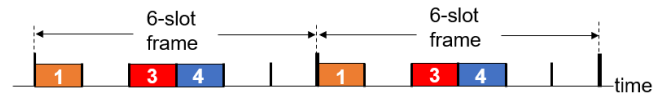
בקרת גישה מרובה (MAC Protocols)

Channel Access Protocol - חילוק הערוץ

TDMA - גישה לערוץ בסבבים כל תחנה מקבלת זמן קבוע

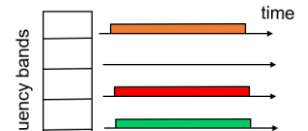
בו היא יכולה לשדר ובזמן זה תקבל את כל רוחב הפס,

משבצות שבאן אין שידור מתבזבזות



FDMA - חילוק רוחב הפס לפי תדרים לכל משתמש יהיה

תדר משלו והמערכת תדע לטפל בו בהתאם



Rando Access Protocols - שניים משדרים יחד נרצה

להגריל לכל אחד זמן המתנה לניסיון שידור חדש (טוב

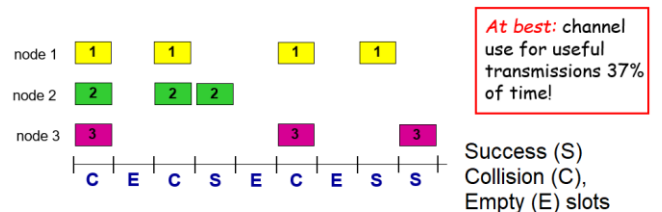
כשיש משתמש אחד)

Slotted ALOHA - מבצע הטלת מטבע האם לנסות לשדר

שוב, הזמן מחולק ל frame בגודל שווה, תחילת שידור

מתבצעת רק בתחילת frame

* לא טוב להרבה משדרים



At best: channel use for useful transmissions 37% of time!

N nodes

each transmits in slot with probability p

prob that given node has success in a slot = $p(1-p)^{N-1}$

prob that any node has a success = $Np(1-p)^{N-1}$

max efficiency: find p^* that maximizes $Np(1-p)^{N-1}$

Pure ALOHA - פשוט יותר, ופחות יעיל כש frame מגיע

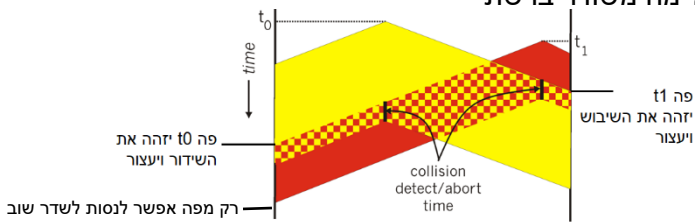
לשכבת הערוץ הוא נשלח ישר

CSMA - נרצה לראות שהערוץ פנוי ורק אז נשדר התנגשויות

יכולות לקרות בזמן הדילי של השליחה יעיל פשוט וזול

CD\CSMA - זיהוי ההתנגשות לפני בזמן העברה ע"י בדיקה

של מה משודר ברשת



זמן ההמתנה יוגרל כך:

after m th collision, NIC chooses K at random from $\{0, 1, 2, \dots, 2^m - 1\}$.

Controlled Access Protocol - שידור בתורות

Polling - קיים Master שתפקידו לעבור אחד אחד וכל אחד

מקבל זמן שידור ששייך רק לו אם הנוכחי לא רוצה לשדר

לאחר המתנה קצרה master יעבור הלאה

Token Ring - מעבירים token בין אחד לשני כאשר מישהו

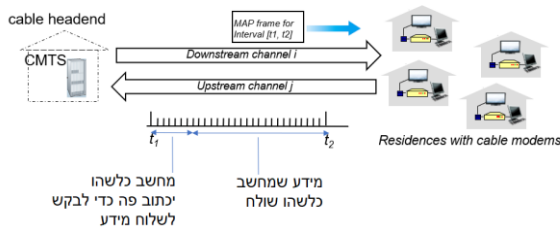
מחזיק את ה token הוא יכול לשדר

* תקלות: מחשב מתנתק עם ה token, token משתבש

שידור ברשת קווית

CMTS=Cable modem termination system

הרשת מחולקת להורדה והעלאה



אם מחשב מבקש לשלוח מידע והדבר אפשרי CMTS

ישלח לו אישור בשדה t_2 להעלות מידע

כתובת MAC - 48 bit לכל מחשב יש כתובת MAC ייחודית

שמשמשת לייחד אותו בתוך הרשת

תקשורת באותה הרשת

שימוש בטבלת ARP שמכילה את כתובת ה MAC שחולקות

איתי את אותה הרשת, כאשר מחשב מחפש מחשב אחר

באותה הרשת שאינו נמצא בטבלה שלו הוא עושה

broadcast ברשת כתובת ה MAC של היעד תהיה FF-FF-FF-FF

כל מי שמקבל את ה broadcast מעדכן את

פרטי השולח, לאחר מכן אם קיים אצלו מידע לגבי כיוון

היעד הוא ישלח בכיוון הנכון ואם לא הוא יעשה גם הוא

broadcast/flood לכל היציאות חוץ מזו ששלחה את

ההודעה

Src MAC	Dest MAC	Type	Source IP	Src MAC	Dest IP	Dest MAC
α	FF-...-FF	Query	a	α	b	?

תקשורת ברשתות שונות

בכל שלב ה MAC היעד יהיה של הנתב הבא או של היעד

ה MAC המקור יהיה של הנתב הקודם או מקור ההתחלה

פרוטוקול האינטרנט

כל רשת שרץ ובכל נקודה כרטיסי רשת יכולים להתחבר

MAC addr	interface	TTL
A	1	60



מחכה frame יכנס במלואו (בעל זיכרון) מסוגל להעביר במקביל חבילות ממקור ויעד שונה אין התנגשויות, משתמש בכתובות MAC, TTL, "לומד" איפה כל מחשב נמצא

bus: coaxial cable



אם מישהו משדר כולם שומעים יכולים להתנגש

שכבת היישום L5

כשתהליך מבקש גישה לרשת הוא פותח Socket (מכניזם של OS שמעביר את החבילות לשכבת התעבורה) שמקושר לפורט לוגי הפורט משמש כדי שה-OS תדע לכוון חבילות נכנסות מתבצע ע"י identifier

Client at 194.4.2.12		Server at 10.12.38.2	
Protocol	TCP	Protocol	TCP
My-IP	194.4.2.12	My-IP	10.12.38.2
My-Port	28732	My-Port	80
Peer-IP	10.12.38.2	Peer-IP	194.4.2.12
Peer-Port	80	Peer-Port	28732

Client process - תהליך שיזום תקשורת

Server process - תהליך שמחכה לתקשורת

Client-Host - אין תקשורת ישירה בין הצדדים

השרת הוא המארח כתובת IP שלו קבועה, יכול להמשיך לעבוד גם כשמתבצע שינוי, הפורט הלוגי שלו ידוע לשניהם, **הלקוח** יוצר קשר עם השרת, יכול להיות מחובר לסירוגין, כתובת IP שלו יכולה להשתנות.

P2P - תקשורת ישירה בין הצדדים

רשת תקשורת בה כל אחד מהקצוות מתפקד הן כלקוח והן שרת, כל אחד מהם יכול ליזום או לסיים התקשרות ולספק או לדרוש שירותים אין הבחנה ברורה בין הצדדים והרשת מתפקדת באופן מבוזר.

מיפוי כתובת - ל-Host יש כתובת URL לפני שאנחנו יכולים לגשת אל Host עליו URL לפרק את כתובת IP ע"י

DNS Resolver



DNS - סידור היררכי של שרתים כל שרת מכיר את כל

הכתובות של הילדים שלו, רמת השורש (root DNS) מכיר את כל שרתי ה-TLD חיפוש כתובת מתבצע באופן רקורסיבי מלמטה למעלה

Local DNS Server (LNS) - לא משתייך לסידור ההיררכי,

מחזיק cache של בקשות אחרונות עם TTL, תומך בהרבה משתמשים (scaling).

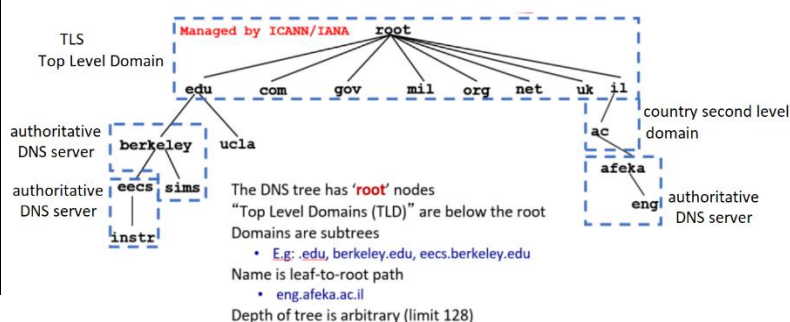
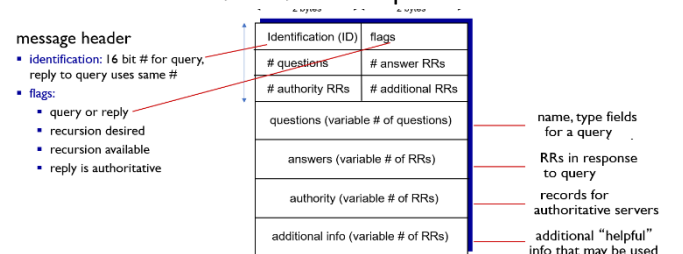
כאשר משתמש קצה מבצע שאילתת DNS השאילתה

נשלחת ל-LNS אם יש לו את התשובה הוא יחזיר אותה

לresolver אחרת הוא ישלח אותה לשרת אחר בפרוטוקול

DNS, לבסוף היישום יקבל את התשובה מהresolver

ויכניס addr. בSocket ip. יעבוד על UDP



Proxy server - שרת שמתווך בין בקשות של לקוח

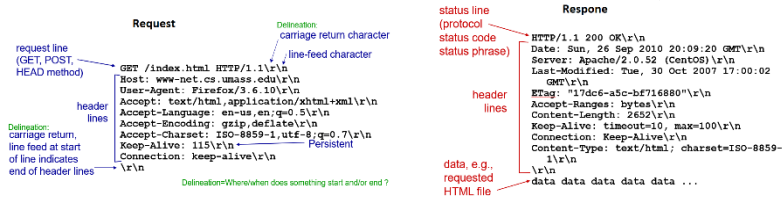
מסרברים, משמש לצורך caching, אבטחה, איזון עומסים, מעקב אחר תוכן..

Http - השרת הוא תוכנה שרצה על מארח שמאכסן דפי

web והתוכנה יודעת לשלוח אותם ללקוח שהו הדפדפן

port=80, משתמש ב-TCP, הורדה מהירה, זמין, מניעת

עומסים, משתלם.



request - conditional get מהשרת לצורך בדיקה האם הוא

עוד עדכני מאז הבקשה האחרונה

Cookies - קבצים המותקנים אצל הלקוח משמשים את

השרת לצורך זיהוי המשתמש, גם לשרת וגם ללקוח יש

גישה לקבצי

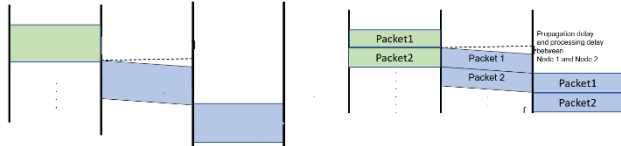
שכבה פיזית L1



transmission + propagation = סיום קליטת מסגרת

זמן התחלה

חבילה חייבת שתגיע במלואה לפני שתוכל להמשיך בדרך יתרון של מס' חבילות לעומת חבילה אחת



ניצול רוחב פס

מס' שלוחות בשנייה * מס מארחים * גודל חבילה (bit) * ratio

רוחב פס

miss http הנתב לשרת

hit בין הנתב proxy



תיעדוף חבילות (L3)

fixed - סדר התיעדוף קבוע

round robin -

בכל פעם התיעדוף משתנה כך שמי שיצא אחרון יהיה בסוף

ומי שאחריו הוא הראשון

