

תרגיל תכנות: מודול ה-ARP (חלק שני)

0512.4491 מעבדה מתקדמת
בתקשורת מחשבים

מטרה. בתרגיל זה נמשיך את המימוש מהתרגיל הקודם ע"י מימוש מודול ה-ARP. גם כעת, תשתמשו בכרטיס רשת וירטואלי NIC (המשמש כשכבה 1) שאנו נספק, שכבה 3 שכתבתם בתרגיל הקודם, ובמודול נוסף שאנו נספק, המהווה את שכבות 4+5.

מבנה. הסבר מפורט על מבנה התוכנית, המתודות העומדות לרשותכם והמשתנים הסטטיים העומדים לרשותכם מפורט באתר <http://www.eng.tau.ac.il/~tom>. בסוף המסמך מצורף תיאור זרימת המידע בתוכנית ומבנה חבילות ETHERNET וחבילת ARP.

במטלה הקודמת השתמשנו במימוש ארעי להמרת כתובות IP לכתובות MAC (טבלה סטטית). כעת, אנו נממש מודול ARP ייעודי אשר יבצע את התרגום בהתאם לפרוטוקול RFC 826 אשר יפורט בהמשך מסמך זה (בנוסף, תוכלו למצוא אותו ב-Moodle תחת לשונית HW06). בתרגיל הקודם יכולתם לבצע את ההמרה $IP \rightarrow MAC$ בפונקציה `sendToL2()` או בפונקציה אחרת כראות עינכם. בתרגיל הנוכחי אתם נדרשים לבצע את ההמרה בפונקציה `arpresolve()` שתשויך למודול `L2_arp`.

מימוש ARP. הפרוטוקול מפורט במסמך RFC 826, והוא עוסק בתרגום כתובות IP לכתובות MAC. מימוש המודול חייב להיות בהתאם לדרישות IETF המפורטות ב RFC 1122 (מסמך המגדיר דרישות לממשי שכבות ברשת) בסעיף 2.3.2 אשר הועלה לאתר, הכולל בין היתר:

1. מימוש המטמון. המטמון הינו מבנה נתונים המכיל רשומות הכוללות בין היתר (תוספות ייתכנו בהתאם למימושכם): כתובת IP לתרגום וכתובת MAC לפענוח. רשומות במטמון הופכות ל `Invalid` לאחר `Expiration Timeout` זאת על מנת לשמור על עדכניות הטבלה. משמעות הדבר הינה שבנוסף לבחירת מבנה נתונים מתאים, באחריות השכבה `L2_arp` לדאוג לתחזוקת המטמון לאורך כל ריצת התוכנית ע"פ מנגנון לבחירתכם.
- שימו לב: אתם לא נדרשים לממש ARP Proxy ולכן אנו מגדירים את פרמטר ה `Timeout` להיות 100 שניות, וזאת כדי להימנע מ `overhead`, ע"פ המלצות המסמך.
2. מימוש תור לחבילות אשר מיועדות לכתובת IP שטרם תורגמה. פרטים ב RFC 1122.
3. הימנעות מ ARP Flooding. הדבר יבוצע כדלהלן:
 - א. אסור לשלוח יותר מ ARP Request אחד בשנייה.
 - ב. אם לא התקבלה תשובה לאחר 5 בקשות (כלומר, 5 שניות) יש להמתין 20 שניות לפני שליחת בקשה נוספת (באופן זה, השכבה מונעת שליחת הודעות ליעד הנתון למשך פרק זמן זה).
 - ג. אחרי 20 שניות אלו (באם לא התקבלה תשובה כמובן) יש לשלוח שוב בקשות בהתאם לסעיף b וחוזר חלילה.

מודל ה `L2_arp` יממש את פונקציית התרגום `arpresolve` האחראית לתרגום. התרגום נעשה ראשית ע"י קריאה ל `arplookup`. אם הערך המבוקש נמצא בטבלה, הערך יוחזר (לפונקציה הקוראת `SendToL2`) והחבילה תשלח. אחרת הפונקציה מחזירה "" (string ריק). במקרה זה `arpresolve` מאחסן את החבילה בתור למשלוח עתידי, וקורא לפונקציה `arprequest()`. הפונקציה `arprequest()` חבילת `ARP request` ושולחת אותה לכולם (broadcast) ע"י קריאה ל `SendToL2`. כאשר תגיע חבילת `ARP reply`, היא נחשבת כתשובה ולכן `recvFromL2()` תעביר אותה למודול ה ARP ע"י קריאה ל `in_arpinput()`, אשר תמלא את הערך מהתשובה

בטבלה. בתום העדכון, `in_arpinput()` תקרא ל `SendToL2` עם חבילה שכתובת יעדה תורגמה (החבילה מאוחסנת כתור): עתה הקריאה ל `arpresolve` תצליח והחבילה תישלח.

כאשר מגיעה הודעה, כרטיס הרשת (NIC) יטפל בה באופן מתירני (`promiscuous`) או מסונן, עפ"י הדרך בה אותחל (באמצעות משתנה בוליאני בקריאה לבנאי). בטיפול מתירני (בו משתמשת למשל אפליקציית `sniffer`) כל חבילה מועברת לשכבה 2; בטיפול מסונן, רק חבילות שיעדן כרטיס זה מועברות לשכבה 2 (כלומר רק חבילות שכתובת יעדן היא כתובת ה MAC של הכרטיס או הודעות `broadcast`). הפונקציה `lread` קוראת לפונקציה `RecvFromL2` אותה אתם מממשים. הפונקציה בודקת את סוג ההודעה ומטפלת בהתאם. אתם נדרשים לתמוך בשני סוגים:

- **IP:** במקרה זה עליכם לקרוא לפונקציה `RecvFromL3` שמעבירה את ההודעה לשכבה 3, אשר בתורה תעביר את תוכן ההודעה לשכבה 4 ע"י קריאה ל `RecvFromL4`.
- **ARP:** במקרה זה יש להעביר את הטיפול למודול ה ARP ע"י קריאה ל `in_arpinput`.

דיבוג התכנית.

- כדי לעקוב אחרי ריצת התכנית, עליכם להכניס הדפסות של תוכן החבילות בצורה קריאה וברורה לבודק, בתוך תנאי `if` המקבל את המשתנה הגלובלי במחלקה, `debug` (אשר נקבע בעת אתחול המחלקה).
- ניתן לקבוע בתוכנית הראשית את רמת הדיבוג (`debug level`) ע"י הזנת `true` או `false` לבנאי השכבות בהם מעוניינים לראות הדפסות.
- בנוסף, לכרטיס הרשת ניתנה אפשרות להוספת `filter` בפורמט של [Winpcap filtering expression syntax](#) לשימושכם בעת דיבוג התוכנית.

שימו לב: הוספת ה `filter` הינה פיצ'ר שהוכנס במטרה להקל על הליך דיבוג התוכנית בלבד, כלומר, ההגשה הסופית תיבדק כאשר לא מוכנס כלל `filter` וכאשר ה NIC במצב `promiscuous`.

שימו לב: כיוון ששני החוטים עלולים לנסות להדפיס בו זמנית, יש להשיג את המנעול `print_mutex` שנוצר בכרטיס הרשת (שאנו מספקים) לפני הדפסה ולשחררו בסופה. דוגמא כיצד להשתמש בו מוצגת להלן:

```
pthread_mutex_lock(&NIC::print_mutex);  
cout << "Hello World!" << endl;  
pthread_mutex_unlock(&NIC::print_mutex);
```

מה אתם מקבלים. כאשר תורידו את החומר לתרגיל, תמצאו ספריה ובה:

- ספריות (ריקות) בשם `Debug` ו `Release` – כרגיל, לעבודה.
- `WinPcap`: מכילה מודולים בינאריים מועילים שלא תשתמשו בהם ישירות.
- `Pthreads`: מכילה מודולים בינאריים מועילים בהם תשתמשו לתמיכה במקביליות התוכנית.
- `NetlabTAU`: זהו האזור בו עליכם להתמקד. הוא מכיל 3 תת-ספריות:
 - `Debug, release`: מכילים את הספרייה הבינארית איתה תעבדו בתמיכה בגרסת **VS2013**.
 - `Debug10, release10`: מכילים את הספרייה הבינארית איתה תעבדו בתמיכה בגרסת **VS2010**.
 - `Include`: מכילה את קבצי ההגדרות (`*.h`).
- קובץ המקור של הלקוח תחת השם `main.cpp`. הלקוח שולח חבילת ICMP מסוג ECHO ליעד המצוין כארגומנט בשורת הפקודה ומדפיס את תוכן התשובה.

הגשת התוכנית: עליכם לשלוח את הקובצים L2.cpp, L2_ARP.cpp, L3.cpp וכמו כן כל קובץ אחר אשר הוספתם או שיניתם (למשל קובץ L2_ARP.h) אותם נחליף במקום קובצי השלד שקיבלתם. אי לכך אין לשנות את הגדרות הפרויקט בשום אופן. בנוסף, לפני ההגשה סופית, יש להוריד עותק נקי של התוכנית, להעתיק אליו את הקבצים אותם רציתם להגיש ולהריץ בדיקה נוספת. שימו לב, אין צורך להגיש קובץ שלא שיניתם.

הגשת הדוח המסכם (בפורמט pdf בלבד) הכולל:

1. פירוט מלא של הפונקציות שמומשו. לכל פונקציה אותה ממשתם יש לספק תיאור הפונקציה בקצרה, הסבר על פונקציונליות הפונקציה, הסבר על משתני הקלט, הסבר על מה מחזירה הפונקציה וכד'.
2. פירוט כל אחד מהמודולים שמומשו. לכל מודול יש לספק הסבר מפורט המכיל את מבני הנתונים שנעשה בהם שימוש (לרבות struct, מחלקות עזר ומשתנים וכד'). יש לשים דגש מיוחד על פירוט מסמון מודול ה ARP ועל התור. עבור מבני נתונים אלו, יש לציין מדוע הם נבחרו וכיצד הם עונים על הדרישות.
3. הצגת דוגמת הרצה וניתוחה. יש להציג דוגמת הרצה של המערכת שבניתם. דוגמת ההרצה צריכה לכלול לפחות: שליחה וקבלה של חבילת ICMP לכתובת IP שאינה שייכת ל-subnet בה אתם נמצאים ושליחה וקבלה של חבילת ICMP לכתובת IP ששייכת ל-subnet שאליו אתם שייכים. על כל דוגמא כזאת, יש לספק הסבר מלא של פעילות המערכת שבניתם, החל משליחת ההודעה מתוך התוכנית הראשית ועד לקריאת התשובה על ידי התוכנית הראשית.
4. הנחות, בעיות ידועות והערות נוספות. במידה ויש עוד משהו שלדעתכם חשוב לדעת, למשל באגים שלא הצלחתם לפתור ואת יודעים שקיימים, הנחות שביצעתם והסברים נוספים שברצונכם לספק.

Ethernet ARP המשוורשרת לכותרת Ethernet מבנה חבילת

ARP Structures

Figure 21.7 shows the format of an ARP packet when transmitted on an Ethernet.

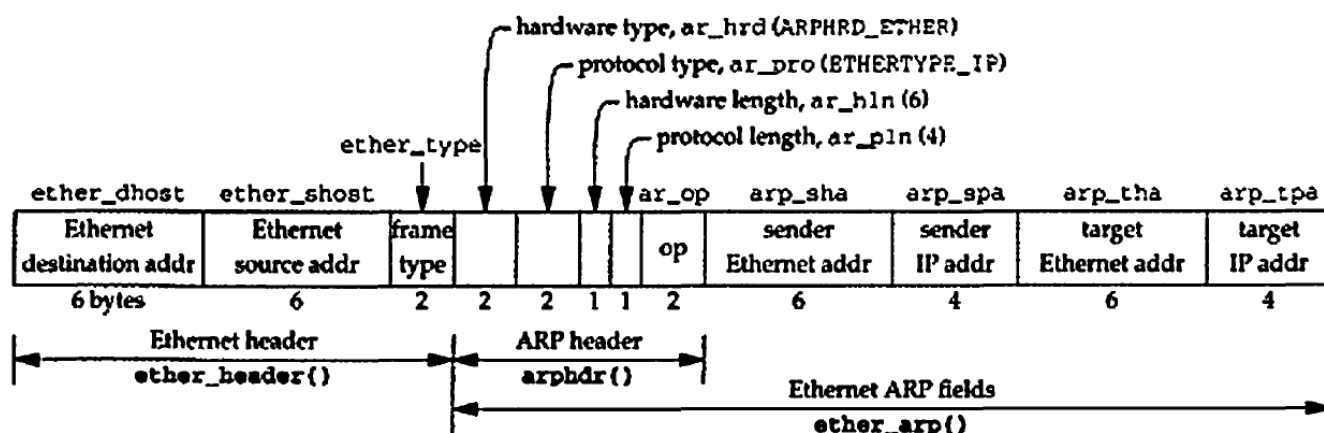


Figure 21.7 Format of an ARP request or reply when used on an Ethernet.

תיאור זרימת המידע בתוכנית:

