



## تمرین برنامه‌نویسی سوم<sup>۱</sup>

### شبکه‌های کامپیوتری

بهار ۱۳۸۹

مدرس: مهدی خرازی

در این تمرین شما پروتکل‌هایی برای ارتباط بی‌سیم (روی شبکه‌های Ethernet) طراحی و پیاده‌سازی خواهید کرد.

#### مقدمه

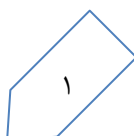
همانطور که در کلاس گفته شد، یکی از روش‌هایی که برای Host Mobility استفاده می‌شود، Mobile IP است. به طور دقیق‌تر در Host Mobility هدف آن است که یک کاربر انتهایی بتواند مکان خود را تغییر دهد و در عین حال ارتباط آن با محیط بیرون قطع نشود. اما مسیریابی در شبکه‌ای همچون اینترنت بر اساس IP صورت می‌گیرد و لذا برای برقرار ماندن ارتباط ولو با تغییر مکان کاربر، باید IP کاربر ثابت بماند. اما مشکل اینجاست که IP وابسته به مکان و شبکه‌ای که کاربر در آن قرار دارد نیز هست. پس ثابت ماندن IP امکان‌پذیر نیست اما می‌توان مکانیزمی طراحی کرد که IP کاربر انتهایی از دید محیط بیرونی ثابت و همان IP اولیه‌ی آن باشد. در حقیقت Mobile IP چنین مکانیزمی است.

شما باید در این تمرین این مکانیزم را پیاده‌سازی کنید. البته Mobile IP واقعی پیچیدگی‌هایی برای بهبود کارایی، برقراری امنیت و ... نیز دارد که نیازی به پیاده‌سازی آنها نیست. بنابراین بهتر است پروتکل‌هایی را که در این مکانیزم استفاده می‌شوند، به ساده‌ترین شکل ممکن، طراحی و پیاده‌سازی کنید. البته این مکانیزم عموماً برای ارتباطات بی‌سیم به کار می‌رود، اما به دلیل سختی کار با بسته‌های چنین ارتباطاتی (در لایه‌ی دو) و مشکلات پیاده‌سازی و فیزیکی، شما باید این مکانیزم را در محیط Ethernet طراحی و پیاده‌سازی کنید و اساساً حرکت کاربر انتهایی در این محیط، برای تست برنامه‌ی شما، شبیه‌سازی می‌شود؛ درست مانند اینکه ارتباط به صورت بی‌سیم باشد اما فقط سرآیند<sup>۲</sup> قاب<sup>۳</sup>‌های لایه‌ی دو، سرآیند Ethernet است.

<sup>۱</sup> با تشکر از صادق دری، شایان پویا، بهنام مؤمنی، حسن اسلامی و امیر شیخها

<sup>۲</sup> header

<sup>۳</sup> frame

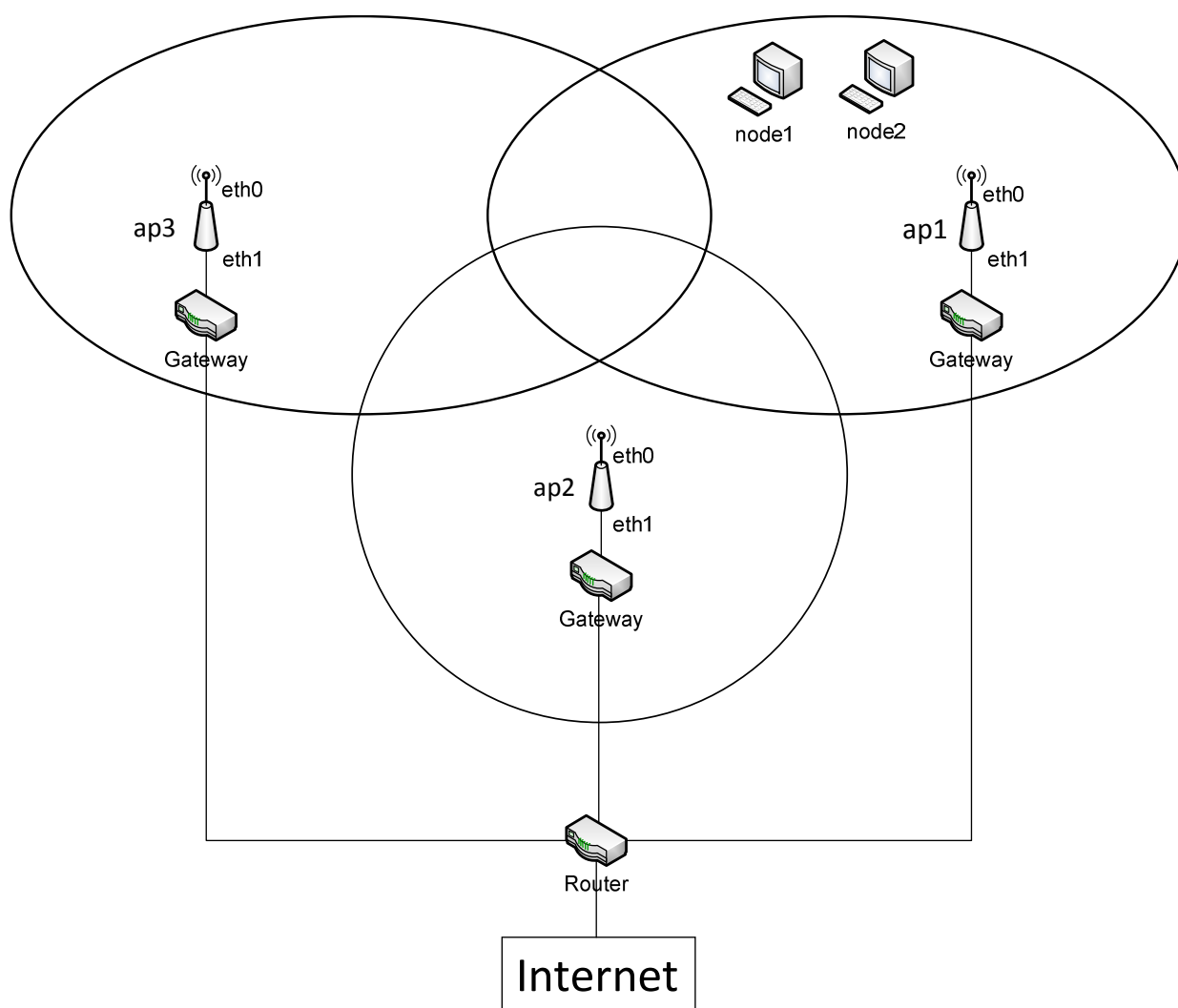


## محیط

اساس کار این مکانیزم در لایه‌های دو و سه و چهار است. به همین خاطر در این تمرین نیز، مانند تمرین قبل، از محیط پرتو<sup>۴</sup> برای اجرای برنامه‌ی شما استفاده خواهد شد. برای این منظور لازم است برنامه‌های خود را بر روی «چارچوب کاربر»<sup>۵</sup> پرتو پیاده‌سازی کنید. برای اطلاعات بیشتر در مورد این چارچوب به مستند «راهنمای چارچوب کاربر» مراجعه کنید.

## توپولوژی

توپولوژی اولیه که در اختیار هر دانشجو قرار می‌گیرد مانند شکل ۱ است. البته برنامه‌ی شما به هیچ وجه نباید وابسته به توپولوژی باشد و یا فرض خاصی در مورد توپولوژی انجام دهد.



شکل ۱. توپولوژی نمونه

<sup>4</sup> Portable And Reliable Tool for Virtualization (PARTOV)

<sup>5</sup> Client Framework (CF)

هر توپولوژی شامل تعدادی Node (کاربران انتهایی) و تعدادی Access Point (به اختصار AP) است. هر AP دامنه‌ی دیدی دارد که پیام‌های ارسالی تمام Node های داخل آن را از طریق interface شماره صفر خود دریافت می‌کند. البته ممکن است دامنه‌ی دید چند AP با یکدیگر اشتراک نیز داشته باشد که در این صورت پیام‌های ارسالی هر Node که در ناحیه‌ی اشتراک قرار گیرد به تمام آن AP ها خواهد رسید. همچنین پیام‌هایی که یک AP از طریق interface شماره صفر خود ارسال می‌کند به تمام Node هایی که در دامنه‌ی دید آن AP قرار دارند خواهد رسید. بعلاوه، یک AP از طریق interface شماره یک خود به Gateway (و در نتیجه به محیط اینترنت) متصل است. هر Node نیز از طریق interface شماره صفر خود پیام‌ها را ارسال و دریافت می‌کند.

## انتظارات

شما باید دو برنامه بنویسید. یکی برای Node ها و دیگری برای AP ها. بنابراین باید یک کپی از sm.cpp بگیرید و دو برنامه‌ی جدا بنویسید. لذا باید Makefile را نیز تغییر دهید تا هر دو برنامه کامپایل گردند. تغییرات Makefile باید به گونه‌ای باشد که با اجرای فرمان

```
make node
```

یک فایل اجرایی با نام node.out ایجاد گردد (که همان برنامه‌ی Node هاست) و با اجرای فرمان

```
make ap
```

یک فایل اجرایی با نام ap.out ایجاد شود (که همان برنامه‌ی AP هاست). در واقع node.out و ap.out مانند cf.out در تمرین قبلی است که باید آنها را با پارامترهایی مناسب اجرا کنید تا بر روی device های مناسب در توپولوژی قرار گیرند و اجرا گردند. نحوه‌ی اجرا کردن این برنامه‌ها با پارامترهای مناسب در فایل‌هایی به همراه «چارچوب کاربر» به شما داده شده است. در ادامه توضیحات دقیق‌تری در مورد نحوه‌ی کار این دو برنامه داده خواهد شد.

## برنامه‌ها

### برنامه‌ی مربوط به Node ها

یک Node درست مثل یک کاربر انتهایی است که در ابتدای اتصال به شبکه IP ندارد و شاید حتی در دامنه‌ی دید هیچ AP ای نباشد. لذا باید از طریق مکانیزمی یک IP بگیرد. این مکانیزم هم می‌تواند شبیه به DHCP باشد و یک Node، IP خود را از یک AP درخواست کند و هم می‌تواند تخصیص IP از جانب AP ها و با تبلیغ پروتکل IP های در اختیار آنها صورت گیرد. در هر حال طراحی پروتکل در اختیار شماست و حتی می‌توانید پروتکل ابداعی خودتان را پیاده‌سازی کنید. اما:

- برای سادگی فرض کنید دامنه‌ی دیدی که یک Node برای اولین بار وارد آن می‌شود، اشتراک دامنه‌ی دید چند AP نیست و Node تا زمانی که IP دریافت نکرده از شبکه‌ی (دامنه‌ی دید) آن AP خارج نمی‌شود. البته مکانیزم دریافت IP باید حداکثر ۱۰ ثانیه طول بکشد.<sup>۶</sup>
- طبعاً این پروتکل مانند DHCP، باید در لایه‌ی ۳ و روی لایه‌ی Ethernet باشد.
- از آنجا که این پروتکل بین Node ها و AP ها اجرا می‌شود و این ارتباط بی‌سیم است، امکان خرابی یا گم شدن بسته‌ها وجود دارد. بنابراین باید از مکانیزم‌هایی همچون checksum<sup>۷</sup> (برای تشخیص خرابی بسته‌ها) و Timeout و Acknowledge (برای تشخیص گم شدن بسته‌ها) استفاده کنید.
- فرض کنید IP ای که یک Node برای اولین بار از AP می‌گیرد (البته در صورتی که مکانیزم را به درستی پیاده‌سازی کرده باشید) یک IP واقعی در شبکه‌ی اینترنت است و لذا باید بتوانید آن Node را از بیرون ping کنید. برای اینکه بتوانید یک Node را ping کنید، آن Node باید بسته‌های ICMP request را با ICMP reply پاسخ دهد. پس علاوه بر مکانیزم دریافت IP باید این قسمت را نیز برای یک Node پیاده‌سازی کنید. تست کردن برنامه‌ی شما در تحویل حضوری نیز بر اساس ping کردن خواهد بود. شما صرفاً با بسته‌های Echo (Echo Request) و Echo Reply سر و کار دارید. برای اطلاعات بیشتر می‌توانید به منابع زیر مراجعه کنید:

- [http://en.wikipedia.org/wiki/Internet\\_Control\\_Message\\_Protocol](http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol)
- <http://tools.ietf.org/html/rfc792>

## برنامه‌ی مربوط به AP ها

این قسمت قلب اصلی مکانیزم Mobile IP است که همانطور که در ابتدا گفته شد نیازی به پیاده‌سازی مکانیزم پیچیده‌ی واقعی نیست. هر AP سه وظیفه‌ی اصلی دارد:

۱. register کردن و چک کردن وجود Node در شبکه خود:

- اگر یک Node که هیچ IP ندارد وارد دامنه‌ی دید یک AP شد، باید عمل تخصیص IP انجام گیرد. (پروتکل لایه ۳)
- یک AP باید بداند چه Node هایی در شبکه‌اش هستند. این عمل می‌تواند با تبلیغ پروتکل Node یا AP انجام گیرد. (پروتکل لایه ۳)
- وقتی یک Node وارد شبکه‌ی یک AP شد، و آن AP یک foreign agent برای Node باشد، AP باید ورود Node به شبکه‌اش را به home agent آن Node اطلاع دهد. بنابراین باید پروتکلی برای این منظور بین AP ها نیز اجرا شود. اما از آنجا که این اطلاع، اطلاع مهمی است، منطقاً باید ارسال پیام‌ها در این پروتکل reliable باشد. به عبارتی باید از مکانیزم Acknowledge و Timeout استفاده کنید. (پروتکل لایه ۴ و روی IP)

<sup>۶</sup> تمام ثوابت زمانی که در این مستند گفته می‌شوند را به صورت #define در برنامه‌های خود تعریف کنید تا به راحتی قابل تغییر باشند.

<sup>۷</sup> checksum را دو بیتی بگیرید.

۲. forward کردن پیام‌های خارجی به Node ها. اگر AP یک home agent برای یک Node باشد و پیامی از بیرون برای Node دریافت کند، در صورتی که Node در شبکه‌ی AP باشد باید آن را مستقیماً به Node بدهد و الا باید با عمل tunneling پیام را به AP ای ارسال کند که Node در شبکه‌ی آن است و یک foreign agent برای Node محسوب می‌شود (پروتکل tunneling در لایه‌ی ۴ و روی IP). در این صورت foreign agent باید بسته‌ی اصلی را از tunnel خارج کرده و به Node بدهد.

۳. forward کردن بسته‌های داده‌ای Node ها به بیرون (از طریق Gateway ها).

در custom information برنامه‌های AP اطلاعات زیر به ترتیب و در خطوط متوالی خواهد آمد:

- MAC مربوط به Gateway ای که به interface شماره یک AP متصل است،
- IP هایی که AP می‌تواند تخصیص دهد. فرض کنید تعداد این IP ها حداقل به تعداد کل Node هاست.

## نکات ضروری

اولاً IP های واقعی، IP هایی هستند که با 213.233 شروع می‌شوند و فرض کنید Node ها در ابتدا در دامنه‌ی دید AP ای وارد می‌شوند که این IP ها را دارد یا به عبارت دیگر آن AP یک home agent محسوب می‌شود. لذا اگر مکانیزم تخصیص IP به درستی پیاده‌سازی شود، و همچنین ICMP reply هم به درستی تولید شوند، باید بتوان Node ها را ping کرد. البته پیاده‌سازی شما به هیچ وجه نباید مبتنی بر شماره IP باشد و شما نمی‌توانید برنامه‌ی خود را به گونه‌ای بنویسید که مثلاً در آن گفته باشید «اگر فلان IP با 213.233 شروع شده بود فلان کار را بکن».

ثانیاً شما می‌توانید IP و MAC برای هر interface را از طریق دو فیلد public کلاس Interface با نام‌های mac و ip داشته باشید. البته برای یک Node در ابتدای کار فیلد ip مقداری ندارد. همچنین دقت کنید که IP هایی که AP ها از طریق آن با یکدیگر ارتباط برقرار می‌کنند، IP های interface شماره‌ی یک آنهاست.

ثالثاً برنامه‌ی AP ها نیازی به در نظر گرفتن بسته‌های ARP ندارند. این بسته‌ها، که بین AP ها و Gateway ها رد و بدل می‌شوند، توسط محیط پرتو گرفته و بررسی آنها پیاده‌سازی شده است و اساساً برنامه‌ی AP شما چنین بسته‌هایی را دریافت نخواهد کرد. به علاوه، چون شما برنامه‌ی هر دو طرف ارتباط بین یک Node و یک AP را می‌نویسید، تولید یا عدم تولید چنین بسته‌هایی وابسته به پروتکل طراحی شده‌ی شما خواهد داشت و به عبارتی تولید این بسته‌ها می‌تواند جزئی از پروتکل شما باشد یا نباشد.

رابعاً توپولوژی‌ای که شما برنامه‌ی خود را با آن تست می‌کنید همانند شکل ۱ است که در آن نام Node ها node1 و node2 و نام AP ها ap1، ap2 و ap3 است. در این توپولوژی زمان‌بندی حرکت Node ها در جداول ۱ و ۲ آمده است.

به این ترتیب اگر برنامه‌تان را درست نوشته باشید باید بتوان به مدت یک دقیقه Node ها را ping کرد (همزمان) و البته در این مدت ممکن است برخی از بسته‌های ping به مقصد نرسند. دقت کنید که عمل ping با IP های واقعی انجام می‌شود و با اینکه

Node ها در شبکه حرکت می کنند، این IP باید از دید ناظر بیرونی (که ping می کند) ثابت باشد. بنابراین بسته های دریافتی ping (ICMP reply ها) نیز باید از طرف IP های واقعی باشند.

جدول ۱. زمان بندی حرکت node1

مدت زمانی که از شروع گذشته (ثانیه)	نواحی دیدی که node1 در آنهاست
۰ تا ۵	-
۵ تا ۱۵	ناحیه دید ap1
۱۵ تا ۲۰	ناحیه دید ap1 و ap2
۲۰ تا ۳۰	ناحیه دید ap2
۳۰ تا ۳۵	ناحیه دید ap2 و ap3
۳۵ تا ۴۵	ناحیه دید ap3
۴۵ تا ۵۰	ناحیه دید ap1 و ap3
۵۰ تا ۶۰	ناحیه دید ap1

جدول ۲. زمان بندی حرکت node2

مدت زمانی که از شروع گذشته (ثانیه)	نواحی دیدی که node2 در آنهاست
۰ تا ۵	-
۵ تا ۱۵	ناحیه دید ap1
۱۵ تا ۲۰	ناحیه دید ap1 و ap3
۲۰ تا ۳۰	ناحیه دید ap3
۳۰ تا ۳۵	ناحیه دید ap2 و ap3
۳۵ تا ۴۵	ناحیه دید ap1 و ap2 و ap3
۴۵ تا ۵۰	ناحیه دید ap1 و ap2
۵۰ تا ۶۰	ناحیه دید ap1

## توصیه های جدی

همانطور که گفته شد مکانیزم Mobile IP واقعی پیچیدگی هایی دارد که نیازی به پیاده سازی آنها نیست. سعی کنید تا آنجا که ممکن است با طراحی یک پروتکل ساده کار پیاده سازی را برای خود ساده کنید. اگر پروتکل هایی که طراحی می کنید ساده نباشند، این تمرین برای شما خیلی سخت خواهد شد. پس وقت کافی برای طراحی پروتکل بگذارید و هر گونه ابهامی را جداً در گروه پستی درس مطرح کنید.

## منابع کمکی

کلیت نحوه کار Mobile IP در اسلایدهای درس آمده است. برای جزئیات بیشتر می توانید از منابع زیر استفاده کنید:

- <http://www.acm.org/crossroads/xrds7-2/mobileip.html>
- <http://www.tml.tkk.fi/Opinnot/Tik-111.550/1999/Esitelmat/MobileIP/Mobip.html>

همچنین می‌توانید به RFC 3344<sup>8</sup> نیز مراجعه کنید (که توصیه نمی‌شود).

## و در نهایت

شما باید یک مستند<sup>9</sup> مختصر (۲ تا ۳ صفحه) بنویسید و در آن تمام پروتکل‌هایی را که طراحی کرده‌اید توضیح دهید. شما صرفاً باید سرآیند پروتکل‌ها و فیلدهایی را که در آن‌ها گرفته‌اید ذکر کنید و بگویید هر فیلد چه معنی‌ای دارد. این فایل را به فرمت PDF و با نام protocol-80123456 (که در آن به جای ۸۰۱۲۳۴۵۶ شماره‌ی دانشجویی خود را می‌آورید) تبدیل کنید.

پس از اتمام تمرین، ابتدا make clean کنید؛ سپس پوشه‌ی کاری خود را به pa3-80123456 تغییر نام دهید که در آن به جای ۸۰۱۲۳۴۵۶ شماره‌ی دانشجویی خود را بیاورید. در این پوشه علاوه بر کدهای خود، فایل PDF گفته شده را نیز بگذارید. این پوشه را فشرده کنید و به آدرس‌های [kharrazi@sharif.edu](mailto:kharrazi@sharif.edu) و [dorri@ce.sharif.edu](mailto:dorri@ce.sharif.edu) بفرستید. موضوع این پست الکترونیکی را نیز دقیقاً همین عبارت قرار دهید.

---

<sup>8</sup> <http://tools.ietf.org/html/rfc3344>

<sup>9</sup> documentation

