



دانشکده مهندسی کامپیوتر

طراحی و پیاده‌سازی شبکه اجتماعی محلی و بی‌سیم بدون اینترنت

پروژه برای دریافت درجه کارشناسی در رشته مهندسی کامپیوتر
گرایش هوش مصنوعی و رباتیک

مریم سادات هاشمی

استاد راهنما

سید صالح اعتمادی

خرداد ۱۳۹۸



تأییدیه‌ی صحت و اصالت نتایج

باسمه تعالی

اینجانب مریم سادات هاشمی به شماره دانشجویی ۹۴۵۲۳۲۵۲ دانشجوی رشته مهندسی کامپیوتر مقطع تحصیلی کارشناسی تأیید می‌نمایم که کلیه‌ی نتایج این پروژه حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه‌برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. در صورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض درخصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می‌نمایم. در ضمن، مسؤولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذیصلاح (اعم از اداری و قضایی) به عهده‌ی اینجانب خواهد بود و دانشگاه هیچ‌گونه مسؤولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی: مریم سادات هاشمی

تاریخ و امضا:

مجوز بهره‌برداری از پایان‌نامه

بهره‌برداری از این پایان‌نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین می‌شود، بلامانع است:

- ☐ بهره‌برداری از این پایان‌نامه برای همگان بلامانع است.
- ☐ بهره‌برداری از این پایان‌نامه با اخذ مجوز از استاد راهنما، بلامانع است.
- ☐ بهره‌برداری از این پایان‌نامه تا تاریخ ممنوع است.

استاد راهنما: سید صالح اعتمادی

تاریخ:

امضا:

قدردانی

سپاس خداوندگار حکیم را که با لطف بی‌کران خود، آدمی را زیور عقل آراست. در آغاز وظیفه خود می‌دانم از زحمات بی‌دریغ استاد راهنمای خود، جناب آقای دکتر سید صالح اعتمادی، صمیمانه تشکر و قدردانی کنم که قطعاً بدون راهنمایی‌های ارزنده ایشان، این مجموعه به انجام نمی‌رسید. از جناب آقای دکتر ... که زحمت مطالعه و مشاوره این رساله را تقبل فرمودند و در آماده سازی این رساله، به نحو احسن اینجانب را مورد راهنمایی قرار دادند، کمال امتنان را دارم. همچنین لازم می‌دانم از پدید آورندگان بسته زی‌پرشین، مخصوصاً جناب آقای وفا خلیقی، که این پایان‌نامه با استفاده از این بسته، آماده شده است و همه دوستانمان در گروه پارسی‌لاتک کمال قدردانی را داشته باشم. در پایان، بوسه می‌زنم بر دستان خداوندگاران مهر و مهربانی، پدر و مادر عزیزم و بعد از خدا، ستایش می‌کنم وجود مقدس‌شان را و تشکر می‌کنم از خانواده عزیزم به پاس عاطفه سرشار و گرمای امیدبخش وجودشان، که بهترین پشتیبان من بودند.

مریم سادات هاشمی

خرداد ۱۳۹۸

چکیده

هدف این پروژه اضافه کردن لایه خدمات شبکه اجتماعی به یک شبکه ارتباطی بی سیم می باشد. در این پروژه ابتدا یک اپلیکیشن موبایل برای ارائه سرویس همسایگی پیاده سازی می شود. سپس خدمات لازم از لایه خدمات اجتماعی برای این سرویس شناسایی می شوند. سپس خدمات این لایه برای ارائه روی شبکه ارتباطی بی سیم می باشد باز مهندسی، طراحی و پیاده سازی می شوند. در نهایت با استفاده از این اپلیکیشن می توانیم شبکه ی اجتماعی را راه اندازی کنیم که حتی بدون اتصال به اینترنت قادر به ارائه خدمات اجتماعی برخط مثل ارسال پیغام های شخصی یا عمومی می باشد. برای مثال شما می توانید با همسایگانتان که در یک ساختمان هستید، بدون اینترنت ارتباط برقرار کنید و یک وسیله همچون نردبان را از آن ها قرض بگیرید.

واژگان کلیدی: شبکه اجتماعی، بی سیم، شبکه می، اپلیکیشن موبایل، همسایگی

فهرست مطالب

خ	فهرست تصاویر
د	فهرست جداول
ذ	فهرست الگوریتم‌ها
ر	فهرست علائم اختصاری
۱	فصل ۱: مقدمه
۴	۱-۱ کارهای مربوطه
۴	۱-۱-۱ Open Garden
۴	۱-۱-۲ NextDoor
۴	۱-۱-۳ Fire Chat
۵	۱-۲ جمع‌بندی
۶	فصل ۲: انتخاب API مناسب برای ایجاد ارتباط Peer to Peer
۷	۲-۱ فناوری WiFi Direct
۷	۲-۲ بررسی فنی
۷	۲-۳ معماری
۸	۲-۴ تشکیل گروه
۹	۲-۴-۱ استاندارد
۹	۲-۴-۲ مستقل

۱۰	۲-۴-۳ پایدار
۱۰	۲-۵ امنیت
۱۰	۲-۶ ذخیره انرژی
۱۱	۲-۶-۱ Opportunistic Power Save protocol
۱۱	۲-۶-۲ Notice of Absence (NOA) protocol
۱۱	۲-۷ فواید
۱۲	۲-۸ جمع بندی
۱۳	فصل ۳: روش حل مسئله و پیاده سازی
۱۳	۳-۱ اتصال دستگاه ها در لایه ی فیزیکی با استفاده از فناوری WifiDirect
۱۳	۳-۱-۱ Discovery
۱۴	۳-۱-۲ Connect
۱۴	۳-۱-۳ Create Group
۱۴	۳-۲ دریافت و ارسال پیام
۱۵	۳-۲-۱ Multi thread یا Synchronous
۱۷	۳-۲-۲ Asynchronous
۱۹	۳-۳ جمع بندی
۲۰	مراجع
۲۱	پیوست آ: مدیریت مراجع در لاتک
۲۲	پیوست ب: جدول، نمودار و الگوریتم در لاتک
۲۳	واژه نامه فارسی به انگلیسی
۲۴	واژه نامه انگلیسی به فارسی

فهرست تصاویر

۸ ۱-۲ معماری Wifi Direct

فهرست جداول

فهرست الگوریتم‌ها

فهرست علائم اختصاری

$a \text{ (m/s}^2\text{)}$	شتاب گرانش
$F \text{ (N)}$	نیرو

فصل ۱

مقدمه

یکی از نیازهای اساسی انسان‌ها نیاز به برقراری ارتباط با دیگران و اجتماعی شدن است. پایه‌ای‌ترین ارتباطات اجتماعی را می‌توان خانواده، بستگان، دوستان و همکاران دانست که روابط معناداری بین افراد در این گروه‌ها برقرار است. این در حالی است که در سال‌های اخیر فناوری دیجیتال و اینترنت شکل جدیدی از ارتباط را تعریف کرده‌اند و در حال حاضر شاهد گسترش بی‌سابقه شبکه‌های اجتماعی^۱ در بین مردم هستیم. به گونه‌ای که اکنون شبکه‌های اجتماعی جزئی از زندگی روزمره مردم شده است.

شبکه‌های اجتماعی به کاربران این فرصت را می‌دهد تا آراء و نظرات خود را با دیگران مطرح کنند و از نظرات مختلف مطلع شوند. در سایت‌های خرید و فروش و کالای موردنظر خود را جستجو کنند و نظرات خریداران قبلی را بخوانند و سفارش خود را ثبت کنند. به راحتی اطلاعات مقصد و هر آنچه برای یک سفر لازم است را از اینترنت دریافت کرده و از نظرات و راهنمایی‌های افرادی که قبلاً این سفر را تجربه کرده‌اند استفاده کنند و با طیب خاطر به سفر بروند. در فعالیت‌های گروهی و تشکلهای مردم‌نهاد مشارکت کنند. بسیاری از فعالیت‌های این چنینی که ذاتی اجتماعی دارند با کمک اینترنت و ظهور شبکه‌های اجتماعی برای کاربران آسان شده‌اند.

در بین همه‌ی این قابلیت‌ها که شبکه‌های اجتماعی کنونی در اختیار ما قرار می‌دهند، شرایطی وجود دارد که این شبکه‌های اجتماعی پاسخگوی نیازهای ما نیستند. مثلاً بخواهید خبر گم شدن حیوان خانگی‌تان را در محله‌ی خود اعلام کنید و یا بخواهید یک نردبان قرض بگیرید و یا به دنبال یک دندان‌پزشکی یا لوله

¹ Social Networks

کش خوب در نزدیکی محل زندگی تان باشید. تمامی این مثال ها نیازمند این است که شما با همسایگانتان و افرادی که در نزدیکی شما و محله تان زندگی می کنند؛ در ارتباط باشید. موضوعی که امروزه در زندگی ما کمرنگ شده است. از طرفی روند شبکه های اجتماعی مثل فیسبوک^۲، توییتر^۳ و ... به گونه ای است که ما را با دوستانی که ۲۰ سال گذشته با آن ها ارتباط داشتیم، متصل می کند اما با افرادی که در نزدیکی ما زندگی می کنند و نیاز بیشتری به ارتباط با آن ها داریم، متصل نمی کند.

بنابراین در این پروژه ما قصد داریم که یک شبکه ی اجتماعی محلی را پیاده سازی کنیم که امکان برقراری ارتباط با همسایگانمان را برای ما ایجاد کند. از طرفی چون هدف ما در این شبکه ی اجتماعی اتصال به افرادی است که از نظر بعد جغرافیایی به ما نزدیک هستند، می توانیم با استفاده از تکنولوژی هایی همچون بلوتوث^۴، اتصال نقطه به نقطه وای فای^۵ و ... این ارتباط را حاصل کنیم بدون این که بخواهیم از شبکه ی جهانی اینترنت^۶ استفاده کنیم. برای رسیدن به این منظور از شبکه ی بی سیم مش^۷ نیز استفاده خواهیم کرد. البته لازم به ذکر است که در حالت ایده آل اگر تعداد کاربران این شبکه ی اجتماعی مقدار قابل توجهی در سراسر جهان باشد، می توان با کاربران دور دست نیز ارتباط برقرار کرد.

سایر کاربرد های این شبکه ی اجتماعی به صورت زیر خواهد بود:

۱. فرض کنید که شما در نزدیکی دانشکده ی مهندسی کامپیوتر قرار دارید و یک رویداد در دانشکده ی مهندسی کامپیوتر در حال برگزاری است. بنابراین اپلیکیشن به صورت یک اعلان بر روی گوشی شما، برگزاری رویداد را به شما اطلاع رسانی می کند.

۲. فرض کنید شما برای روز چهارشنبه، نیاز به کمک کسی دارید که از فرزندان نگهداری کند. شما می توانید درخواست یک پرستار بچه را در این شبکه محلی به اشتراک بگذارید.

۳. شما دنبال یک کارواش، مکانیکی یا دندان پزشکی خوب در نزدیکی خانه تان هستید. می توانید در مورد این موضوعات از دیگران در این شبکه ی محلی بپرسید.

۴. فرض کنید که شما دوچرخه تان را اطراف خانه تان گم کرده اید. می توانید این موضوع را در شبکه

^۲Facebook

^۳Twitter

^۴Bluetooth

^۵Peer to Peer WiFi Connection

^۶Internet

^۷Wireless Mesh Network

محلی اعلان کنید تا اگر پیدا شد دیگران به شما در این شبکه ی محلی خبر بدهند یا مثلاً می توانید این موضوع را به عنوان جرم در یک منطقه یا امن نبودن یک محله اعلام کنید.

۵. نهاد های دولتی می توانند در این شبکه عضو بشوند و ساکنین یک محله را از اتفاقاتی مثل آتش سوزی، دزدی، خرابی تلفن، قطع آب و گاز و برق و ... مطلع کنند و یا برعکس به این شکل که در صورت اتفاق افتادن هر یک از این حوادث ساکنین آن محله بتوانند در سریع ترین زمان ممکن نهاد دولتی مربوطه را مطلع سازند.

۶. فرض کنید که یک سازمان مثل هواشناسی در این شبکه محلی عضو شود و مردم یک شهر، روستا یا یک استان را از آب و هوای آن روز با خبر کند. مثلاً امروز در مناطق جنوبی شهر آب گرفتگی داریم یا امروز هوا در ساعات ۲ تا ۳ بعد از ظهر بارانی است؛ لطفاً از چتر استفاده کنید.

۷. شما می توانید زمان رفتن به سر کار خود را به بقیه اعلام کنید و اگر کسی از همسایگانتان در مسیر شما قرار داشت، با شما همراه شود. (کاهش ترافیک و آلودگی هوا)

۸. فروشگاه ها و هایپرمارکت های محله می توانند موجودی کالا های خود را اعلام کنند و یا تخفیف های اقلام مختلف را در این شبکه ی محلی قرار بدهند.

۹. می توانید از همسایگانتان در خواست اجاره ی حیاط، پارکینگ یا خانه شان را برای برگزاری جلسه هاو مراسم هایتان و ... داشته باشید.

۱۰. فرض کنید شما نیاز دارید که بسته ای را در نزدیکی محدوده ی زندگی تان ارسال کنید ولی وقت کافی ندارید و می توانید در شبکه درخواست کنید که یک نفر به صورت رایگان این کار را برای شما انجام دهد.

۱۱. می توانید در مورد مسائل مختلف محله تان رای گیری کنید مثل تغییر نام یک کوچه.

۱-۱ کارهای مربوطه

۱-۱-۱ Open Garden

Open Garden^۸ یک سرویس است که این امکان را به مردم می دهد که خدمات اینترنت خود را با افرادی که در نزدیکی آن ها هستند؛ به اشتراک بگذارند. بنابراین با استفاده از این دستگاه اشتراک گذاری اینترنت، هر کس می تواند پهنای باند اضافی اینترنت خود را ارائه دهد و برای آن پول بگیرد یا خدمات اینترنتی را از دیگران خریداری کند.

۱-۱-۲ NextDoor

NextDoor^۹ یک سرویس شبکه اجتماعی خصوصی برای محله است. این شرکت در سال ۲۰۰۸ در سانفرانسیسکو، کالیفرنیا تاسیس شد و در اکتبر ۲۰۱۱ در ایالات متحده راه اندازی شد. کاربران NextDoor نام و نشانی واقعی خود را به وب سایت ارائه می دهند. پیام های منتشر شده در وب سایت، فقط برای سایر اعضای NextDoor در همان محله قابل مشاهده است.

۱-۱-۳ Fire Chat

FireChat^{۱۰} یک برنامه اختصاصی تلفن همراه است که توسط Open Garden توسعه داده شده است. این شبکه ی اجتماعی از شبکه بی سیم مش استفاده می کند تا تلفن های هوشمند بتوانند از طریق بلوتوث، وای فای یا در چارچوب Multipeer شرکت اپل بدون اتصال به اینترنت به یکدیگر متصل شوند. همچنین این برنامه این قابلیت را دارد که از طریق اینترنت نیز اتصال برقرار شود.

FireChat به عنوان یک ابزار ارتباطی در برخی اعتراضات مدنی مورد استفاده قرار گرفت؛ اگر چه که برای چنین اهدافی طراحی نشده است. از جمله دلایلی که این شبکه ی اجتماعی محبوبیت چندانی بین مردم ندارد؛ می توان به ضعف در پیاده سازی، عدم اتصال مناسب بین دو دستگاه اندورید^{۱۱} و آیفون^{۱۲}، اشکال در عدم

^۸<https://www.opengarden.com>

^۹<https://nextdoor.com/>

^{۱۰}<https://www.opengarden.com/firechat/>

^{۱۱} Android

^{۱۲} iphone

ارسال مناسب عکس و غیره اشاره نمود.

۱-۲ جمع بندی

فصل ۲

انتخاب API مناسب برای ایجاد ارتباط Peer to

Peer

اولین گام برای ساخت یک شبکه ی محلی این است که یک API موجود یا فناوری موجود را که ارتباط محلی و Peer to Peer را برای ما بوجود بیاورد را پیدا کنیم و یا این که از ابتدا قابلیت های مورد نیاز را پیاده سازی کنیم. بدین منظور چندین API شامل p2pkit ، Alljoin ، Nearby ، MultipeerConnectivity ، Wi Fi peer ، Hypelabs و Open garden را پیدا کردیم. با این که هر کدام از این API ها قابلیت های مورد نیاز و اولیه ی شبکه های اجتماعی را در برداند، اما مشکل اساسی همه ی این API ها این است که اپلیکیشنی که بر مبنای این ها پیاده سازی شده باشد، باید به اینترنت متصل باشد تا API بتواند به درستی کار کند. بنابراین از آن جایی که شبکه ی اجتماعی ما قرار است بدون اینترنت کار کند، نمی توانیم از این API ها بهره ببریم. در ادامه، با فناوری WiFi Direct آشنا شدیم که بر روی گوشی های اندروید موجود است و تمامی قابلیت هایی که ما برای برقراری ارتباط محلی گوشی ها به یکدیگر را نیاز داریم را در خود دارد؛ با این تفاوت که مشکل API ها را هم ندارد و بدون نیاز به اتصال اینترنت می تواند کار کند. در ادامه ی این فصل به معرفی این فناوری و قابلیت و مزایای آن می پردازیم.

۲-۱ فناوری WiFi Direct

WiFi Direct فناوری جدید تعریف شده توسط اتحادیه WiFi^۱ است که هدف آن ارتقاء ارتباط مستقیم بین دستگاه ها است؛ بدون این که به یک نقطه دسترسی بی سیم^۲ نیاز باشد. WiFi Direct بر روی زیر ساخت موفق IEEE 802.11 بنا شده است و این اجازه را به دستگاه ها می دهد که در یک ارتباط، یک دستگاه نقش نقطه دسترسی بی سیم را ایفا کند و عملکرد آن را انجام دهد. در حال حاضر می توان با استفاده از استاندارد IEEE 802.11 یک ارتباط مستقیم بین دستگاه ها ایجاد کرد. اما اشکالات فراوانی همچون مصرف زیاد انرژی در دستگاه دارد.

۲-۲ بررسی فنی

در یک شبکه معمولی WiFi مشتری^۳ اسکن می کند و عضو یکی از شبکه های بی سیم موجود که توسط نقطه دسترسی بی سیم ایجاد و اعلام شده است؛ می شود. این فرایند در WiFi Direct به صورت پویا^۴ انجام می شود از این رو یک دستگاه WiFi Direct باید هر دو نقش مشتری و نقطه دسترسی بی سیم را به طور همزمان اجرا کند.

۲-۳ معماری

دستگاه های دارای WiFi Direct با ایجاد یک گروه با عنوان P2P Group می توانند با یکدیگر ارتباط برقرار کنند. دستگاهی که عملکردی همچون نقطه دسترسی بی سیم دارد را P2P Group Owner می نامند و دستگاهی که در نقش مشتری است را P2P client گویند. هنگامی که یک P2P Group ایجاد می شود، سایر مشتری ها می توانند با همان روش سنتی شبکه های WiFi به گروه بپیوندند. زمانی که یک دستگاه هم در نقش P2P Client و هم در نقش P2P Group Owner باشد، دستگاه به طور متناوب با استفاده از اشتراک زمانی^۵ بین این دو نقش تغییر می کند. (مثال: لپتاپ ۲ در بالای شکل ۲-۱)

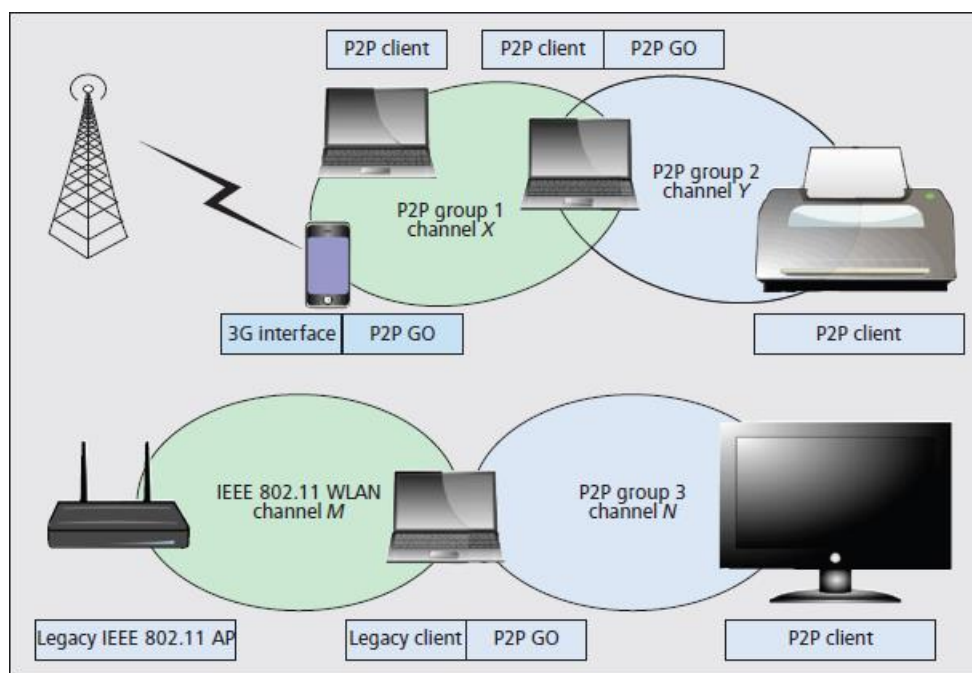
¹WiFi alliance

²wireless access point

³client

⁴dynamic

⁵Time sharing



Wi-Fi direct supported topologies and use cases.

شکل ۲-۱: معماری Wifi Direct

مانند یک نقطه ی دسترسی بی سیم سنتی، یک P2P Group Owner ، خود را از طریق beacons اعلام می کند. تنها دستگاهی که P2P Group Owner است؛ قادر است دستگاه های متصل در گروه خود را به یک شبکه ی خارجی متصل کند. (مثال: موبایل موجود در بالا ی شکل ۲-۱) این ارتباط باید در لایه ی شبکه^۶ اتفاق بیفتد و معمولاً با استفاده از NAT^۷ پیاده سازی می شود. WiFi Direct اجازه نمی دهد که نقش P2P Group Owner به افراد دیگری در گروه انتقال یابد.

۲-۴ تشکیل گروه

سه نوع روش برای تشکیل گروه در فناوری WiFi Direct وجود دارد که عبارتند از استاندارد^۸، مستقل^۹ و پایدار^{۱۰}.

^۶Network Layer

^۷Network Address Translation

^۸Standard

^۹Autonomous

^{۱۰}Persistent

تشکیل گروه شامل دو مرحله است:

۱. تعیین P2P Group Owner

- دو دستگاه با توجه به تمایل یا قابلیت برای P2P Group Owner شدن با یکدیگر مذاکره می کنند.

- در نهایت در این مرحله نقش مالک گروه در سطح اپلیکیشن ایجاد می شود.

۲. تهیه ی P2P Group

- ایجاد session گروه با استفاده از مدارک^{۱۱} معتبر
- استفاده از پیکربندی^{۱۲} ساده Wifi برای تبادل مدارک.

۲-۴-۱ استاندارد

در این حالت دستگاه ها باید یکدیگر را پیدا^{۱۳} کنند و سپس مذاکره کنند که کدام دستگاه به عنوان مالک گروه عمل خواهد کرد. شروع آن با انجام یک اسکن مانند WiFi سنتی است که با استفاده از آن می توانند گروه ها و شبکه های WiFi موجود را پیدا کنند. برای جلوگیری از تضاد، زمانی که دو دستگاه آمادگی خود را برای مالک گروه شدن اعلام می کنند، یک بیت tie-breaker در درخواست قرار می گیرد. هر بار که یک درخواست ارسال می شود، به طور تصادفی این بیت تنظیم می شود.

۲-۴-۲ مستقل

یک دستگاه می تواند به صورت خودکار یک گروه ایجاد کند و بلافاصله این دستگاه مالک گروه می شود. دستگاه های دیگر می توانند گروه های ایجاد شده را با استفاده از مکانیزم های سنتی اسکن پیدا کنند. در مقایسه با بخش ۲-۴-۱، مرحله پیدا کردن در این مورد ساده تر است، زیرا مرحله مذاکره برای مالک گروه شدن حذف شده است.

¹¹Credentials

¹²Configuartion

¹³Discovery

۲-۴-۳ پایدار

در این فرآیند، دستگاه می تواند با استفاده از پرچم ^{۱۴} که به عنوان یک ویژگی در فریم های beacon موجود است یک گروه را به عنوان گروه پایدار اعلام کند. پس از مرحله پیدا کردن، اگر یک دستگاه تشخیص دهد که یک گروه پایدار با همتای مربوطه در گذشته تشکیل داده باشد، هر یک از دو دستگاه می تواند از روش دعوت ^{۱۵} برای استفاده سریع از گروه مجددا استفاده کنند.

۲-۵ امنیت

دستگاه های Wifi Direct از WPS ^{۱۶} پشتیبانی می کنند. WPS یک اتصال امن را با معرفی یک PIN در مشتری یا فشار دادن یک دکمه در دو دستگاه P2P ایجاد می کند. در WPS مالک گروه نقش internal register را اجرا می کند و مشتری نقش Enrollee را اجرا می کند. عمل WPS متشکل از دو بخش است. در بخش اول، Registrar داخلی مسئول ایجاد و صدور مجوزهای شبکه، یعنی کلید های امنیتی، به Enrollee است. در بخش دوم، Enrollee قطع می شود و سپس با استفاده از احراز هویت ^{۱۷} جدید دوباره ارتباط برقرار می کند.

۲-۶ ذخیره انرژی

Wifi Direct دو مکانیسم جدید ذخیره انرژی را به کار می گیرد:

۱. پروتکل Opportunistic Power Save

۲. پروتکل Notice of Absence

¹⁴Flag

¹⁵Invitation

¹⁶WiFi Protected Setup

¹⁷Authentication

۱-۶-۲ Opportunistic Power Save protocol

این پروتکل این اجازه را به مالک گروه می دهد که زمانی که تمامی مشتری های گروه در حالت خواب^{۱۸} هستند؛ انرژی خود را ذخیره کند.

۲-۶-۲ Notice of Absence (NOA) protocol

این پروتکل این اجازه را به مالک گروه می دهد که فواصل زمانی را که به آن ها دوره های زمانی غیابی می گویند را اعلام کند که در این دوره های زمانی، مشتریان مجاز به دسترسی به کانال نیستند. مالک گروه یک برنامه NOA را با استفاده از چهار پارامتر زیر تعریف می کند:

- مدت زمانی که طول هر دوره غیبت را مشخص می کند
- فاصله زمانی که بین دوره های غیبت متوالی وجود دارد
- زمان شروع اولین دوره غیبت پس از فریم beacon کنونی
- تعداد دوره های غیبت برنامه ریزی شده

۷-۲ فواید

در این بخش به فواید و مزایای Wifi Direct می پردازیم.

۱. تحرک و قابلیت حمل: دستگاه هایی که قابلیت Wifi Direct را دارند در هر مکانی و در هر زمانی می توانند به یکدیگر متصل شوند.

۲. سهولت استفاده: دستگاه های داری Wifi Direct ویژگی هایی را دارند که کاربران را قادر می سازد تا قبل از برقراری ارتباط، دستگاه ها و خدمات موجود را شناسایی کنند.

۳. اتصال ساده امن: Wi-Fi Protected setup باعث ساده ساختن ارتباطات محافظت شده بین دستگاه ها می شود. کاربران در بیشتر موارد قادر به اتصال با یک دکمه خواهند بود.

¹⁸Sleep

۲-۸ جمع بندی

در این فصل، ما دلیل انتخاب فناوری Wifi Direct را بیان کردیم و همچنین مروری کامل از ویژگی های فنی موجود در Wifi Direct را ارائه دادیم که شامل مباحث تشکیل گروه و سایر تحلیل های عملکردی مانند ذخیره انرژی و امنیت در این فناوری است. همچنین دانستیم که می توانیم از پروتکل NOA برای مجازی سازی نقش P2P GO / Client در هنگام وجود چندین گروه به صورت همزمان استفاده کنیم.

فصل ۳

روش حل مسئله و پیاده سازی

۳-۱ اتصال دستگاه ها در لایه ی فیزیکی با استفاده از فناوری WifiDirect

برای اجرای خدمات لایه ی شبکه ی اجتماعی، ابتدا لازم است که دستگاه ها در لایه ی فیزیکی به یکدیگر متصل شوند و امکان ارسال و دریافت اطلاعات در لایه ی فیزیکی فراهم شود. به همین جهت از فناوری WiFi Direct استفاده خواهیم کرد تا شرایط لازم در لایه ی فیزیکی را برای ما برقرار کند. در ادامه این بخش توضیح خواهیم که سه مورد Discovery ، Connect و Create Group برای برقراری ارتباط در لایه ی فیزیکی در WiFi Direct چگونه انجام می شود.

۳-۱-۱ Discovery

برای برقراری ارتباط، باید بررسی کنیم که چه افرادی در نزدیکی ما حضور دارند. به این کار Discovery می گوئیم. برای انجام Discovery در WiFi Direct کافی است از متد discoverPeers() استفاده کنیم. با صدا زدن این متد Discovery آغاز می شود و در صورت موفقیت آمیز بودن آن با صدا زدن متد requestPeers می توانیم به لیست افرادی که در نزدیکی ما وجود دارد، دست پیدا کنیم و تمامی افرادی که آمادگی لازم برای برقراری ارتباط را دارند را مشاهده کنیم. توجه داشته باشید که در این روش تنها افرادی را می توان مشاهده کرد که آن ها هم عملیات Discovery را انجام دهند. در غیر این صورت شما نمی توانید آن ها را در لیست خود مشاهده کنید. تصاویر مربوط به هر کدام از این عملیات ها را می توانید در شکل فلان و فلان مشاهده کنید.

۳-۱-۲ Connect

بعد از پیدا کردن افراد نزدیک به خودمان، باید بتوانیم به فرد مورد نظر متصل شویم. برای این کار با دانستن اطلاعات پیکر بندی دستگاه مورد نظر و با استفاده از متد `connect()` عملیات اتصال را آغاز می کنیم. اگر متد `connect()` موفقیت آمیز باشد، با استفاده از رابط `ConnectionInfoListener` بررسی می کنیم که آیا وضعیت اتصال دستگاه فعلی تغییر کرده است یا خیر. با استفاده از متد `onConnectionInfoAvailable()` که رابط `ConnectionInfoListener` در اختیار ما قرار می دهد، چک می کنیم که کدام یک از این دستگاه ها مالک گروه هست و کدام یک مشتری گروه هست. بنابراین با توجه به نقش هر دستگاه در این ارتباط `Server` یا `Client` را برای آن ها صدا خواهیم زد. توجه شود که اطلاعات پیکر بندی دستگاه شامل مواردی مانند آدرس `MAC` دستگاه و اطلاعات `WPS` می باشد.

۳-۱-۳ Create Group

در بخش قبل توضیح دادیم که چگونه می توانیم دو دستگاه را به یکدیگر متصل کنیم. اما اگر بخواهیم چندین دستگاه به یکدیگر متصل شوند؛ باید از متد `createGroup()` استفاده کنیم. در این صورت دستگاهی که این متد را صدا بزند به عنوان مالک گروه خواهد شد و هر دستگاهی که وارد این گروه شود به عنوان مشتری گروه خواهد بود. پس از این که این متد به صورت موفقیت آمیز اجرا شد، با فراخوانی متد `requestGroupInfo()` می توانیم جزئیات بیشتری در مورد دستگاهی هایی که در این گروه قرار دارند شامل نام این دستگاه ها و وضعیت اتصال آن ها را درخواست بدهیم. سپس با استفاده از متد `onGroupInfoAvailable()` که رابط `GroupInfoListener` در اختیار ما قرار می دهد و می توانیم اطلاعات مرتبط با گروه را مشاهده کنیم.

۳-۲ دریافت و ارسال پیام

در بخش قبل دیدیم که چگونه می توان با استفاده از `WiFi Direct` دستگاه ها را در لایه ی فیزیکی به یکدیگر متصل کرد. در این بخش می خواهیم با استفاده از برنامه نویسی `socket` و از طریق دو روش `Multi thread` و `Asynchronous` این امکان را ایجاد کنیم که دستگاه ها با یکدیگر پیام رد و بدل کنند. بنابراین در یک ارتباط، لازم است که یکی از دستگاه ها در نقش `server` و دیگری در نقش `client` باشد. ما در اینجا دستگاهی که به

عنوان مالک گروه باشد را server و دیگری را client در نظر می گیریم. در ادامه ی این بخش وظیفه ی server و client را در هر دو روش Multi thread و Asynchronous بیان می کنیم و توضیح خواهیم داد که هر کدام از چه متد هایی استفاده خواهند کرد.

۳-۲-۱ Multi thread یا Synchronous

همانطور که از اسم این روش پیداست، در این روش، برای هر قسمت از کاری که می خواهیم انجام بدهیم برای مثال اتصال، ارسال و دریافت و ... یک thread اختصاص می دهیم که همه ی آن ها به صورت همزمان در حال انجام وظیفه ی خودشان هستند. در این روش ما منتظر جواب از طرف مقابل هستیم و در واقع تا دریافت جواب، آن متد مسدود یا Block می شود. شبیه زمانی است که شما در حال مکالمه با دوستان در پشت تلفن هستید. وقتی که شما حرفی می زنید، منتظر هستید تا نفر دیگر پاسخ شما را بدهد. بنابراین همین انتظار باعث می شود تا زمان زیادی مصرف شود و منابع ما هدر برود.

همچنین این روش به صورت دنباله ای متوالی انجام می شود. یعنی شما یک کانال برای ارتباط ایجاد می کنید (شماره ی دوستان را می گیرید) و در ادامه یک سری اطلاعات را تبادل می کنید (شما یا دوستان صحبت می کنید و منتظر جواب طرف مقابل هستید) و در آخر هم کانال ارتباطی را می بندید (تلفن را قطع می کنید).

در این روش منابع موجود، بین thread ها به اشتراک گذاشته می شود. این روش برای تعداد اتصالات کم مناسب است. اما وقتی تعداد اتصالات وسیع باشد؛ دیگر نمی توان همه ی اتصالات را به صورت درست مدیریت کرد. در واقع برنامه ریزی thread ها سربار ایجاد خواهد کرد. زیرا تعداد شان زمانی که اتصالات زیاد می شود؛ افزایش پیدا می کند.

برای پیاده سازی این روش از سه کلاس استفاده کرده ایم که در ادامه هر کدام را به صورت مجزا توضیح خواهیم داد.

Server

در کلاس Server دو متد را پیاده سازی کردیم:

۱. Accept(): در این متد server در یک حلقه ی بی نهایت بر روی درگاه ۸۸۸۸ در حال گوش دادن

به درخواست های client است. بعد از آن که server درخواست یک client را قبول می کند، برای این client یک ClientHandler می سازد و آن را به لیست Client هایی که به سرور متصل هستند، اضافه می کند. متد Accept() برای انجام همه ی این مراحل از یک thread استفاده می کند.

۲. Send(): در این متد server یک پیام را برای همه ی client هایی که به آن متصل هستند؛ ارسال می کند. در واقع پیام را بر روی جریان خروجی داده که توسط socket ما بین server و client ایجاد شده است؛ قرار می دهد. برای انجام این کار، از یک thread استفاده می شود.

Client

در کلاس Client سه متد را پیاده سازی کردیم:

۱. Connect(): در این متد client به server ای که در حال گوش دادن بر روی درگاه ۸۸۸۸ هست متصل می شود. به محض این که socket به درستی بین client و server ایجاد شد، client با صدا زدن متد Receive() آماده ی دریافت پیام از سوی server می شود. توجه شود که همه ی این کار ها توسط یک thread انجام می شود.

۲. Send(): در این متد Client پیامی را که می خواهد برای server ارسال کند را بر روی جریان خروجی داده که به کمک socket بین این دو ایجاد شده است، قرار می دهد. از یک thread برای ارسال پیام استفاده می کنیم.

۳. Receive(): در این متد پیامی را که server برای client ارسال کرده است را از روی جریان ورودی داده می خوانیم. همانند ارسال برای دریافت هم از یک thread استفاده می کنیم.

ClientHandler

هدف ما در این کلاس این است که بتوانیم client های متصل به server را مدیریت کنیم. این کلاس تنها یک متد Broadcast() را دارد. در این متد قرار است که server پیام هایی که از client هایش را دریافت می کند را برای سایر client هایش ارسال کند. کاری که دقیقا در این متد انجام می شود؛ این است که پیام را از جریان ورودی داده ی یکی از client هایش می خواند و سپس آن را بر روی جریان خروجی داده سایر client هایش قرار می دهد. دقت شود که همه ی این کار توسط یک thread خاص اجرا می شود.

Asynchronous ۳-۲-۲

در این روش ما کارهای لازم در شبکه را پست می کنیم ولی منتظر جواب نمی مانیم بلکه نتیجه را بعدا بررسی می کنیم. شبیه موقعی است که شما به دوستان پیام متنی ارسال می کنید و منتظر جواب در همان لحظه نمی مانید و جواب دوستان ممکن است الان یا یک ساعت دیگر برسد. بنابراین جواب را بعدا چک می کنید.

این روش بر خلاف روش synchronous به صورت دنباله ای نیست. اما دنبال کردن وضعیت هر یک از کارها کمی مشکل است. بنابراین با توجه به ویژگی هایی که برای این روش ذکر کردیم؛ می توانیم با استفاده از این روش از هدر رفتن منابع جلوگیری کنیم. همچنین می توانیم تعداد اتصالات بیشتری را با همان سخت افزار موجود مدیریت کنیم.

برای پیاده سازی این روش از ۴ کلاس استفاده کرده ایم. که در ادامه هر کدام را به صورت مجزا توضیح خواهیم داد.

WiFiNetService

هدف از این کلاس این است که دستگاه چه در نقش client و چه در نقش server باشد، بتواند ۴ کار ارتباطی زیر را انجام دهد:

۱. ارسال پیام به یک دستگاه خاص:

برای انجام این کار از متد Send() استفاده می کنیم که در آن از متد StartWrite() استفاده شده است که بوسیله ی آن می توان پیام را بر روی بافر دستگاه مورد نظر قرار داد. چون نوشتن بر روی بافر، یک عمل ناهمزمان است از CompletionHandler استفاده می کنیم که کامل شدن یا شکست خوردن را به thread اصلی اعلام کند.

۲. دریافت پیام از یک دستگاه خاص:

برای انجام این کار از متد Receive() استفاده می کنیم که در آن از متد StartRead() استفاده شده است که بوسیله ی آن می توان پیام را از بافر دستگاه مورد نظر خواند. چون خواندن از روی بافر یک عمل ناهمزمان است از CompletionHandler استفاده می کنیم که کامل شدن یا شکست خوردن را به thread اصلی اعلام می کند.

۳. ارسال پیام به تمامی دستگاه هایی که به دستگاه فعلی متصل هستند:

برای انجام این کار از متد Broadcast() استفاده می کنیم که در آن پیام بر روی بافر همه ی دستگاه هایی که به دستگاه فعلی متصل هستند؛ قرار داده می شود. این عمل نیز به صورت ناهمزمان اجرا می شود.

۴. ارسال پیام دریافت شده به تمامی دستگاه هایی که به دستگاه فعلی متصل هستند:

برای انجام این کار از متد ReceiveBroadcast() استفاده می کنیم که پس از دریافت یک پیام آن را بر روی بافر تمامی دستگاه های متصل به دستگاه فعلی قرار می دهیم. این عمل هم به صورت ناهمزمان اجرا می شود.

Device

هدف از ایجاد این کلاس این است که برای هر دستگاه یک کانال نا همزمان socket برای رد و بدل پیام در نظر بگیریم. و همچنین اطلاعات دیگری نظیر اسم دستگاه را نگهداری کنیم.

Server

این کلاس تنها یک متد Start() دارد که با صدا زدن آن server در یک حلقه ی بی نهایت بر روی درگاه ۸۸۸۸ شروع به گوش دادن به درخواست از طرف client ها می کند. اما این کار باعث انسداد یا Blocking نمی شود. بلکه به صورت ناهمزمان انجام می شود. بنابراین از یک CompletionHandler استفاده کرده ایم. که در صورتی که یک client به server متصل شود و عمل کامل شود؛ سرور متد ReceiveBroadcast() را از کلاس WiFiNetService فراخوانی می کند و آماده ی دریافت پیام از یکی از client هایش می شود و سپس آن را برای تمامی دستگاه های متصل، ارسال می کند.

Client

این کلاس همانند کلاس Server تنها یک متد Start() دارد که با فراخوانی آن دستگاه به server ای که بر روی درگاه ۸۸۸۸ در حال گوش دادن هست، متصل می شود. این عمل نیز به صورت ناهمزمان انجام می شود.

بنابراین از یک CompletionHandler استفاده کرده ایم و در صورت کامل شدن اتصال، دستگاه با فراخوانی متد Receive() از کلاس WiFiNetService آماده ی دریافت پیام می شود.

۳-۳ جمع بندی

در این فصل بیان کردیم که چگونه دو دستگاه یا چندین دستگاه می توانند در لایه ی فیزیکی با استفاده از فناوری WiFiDirect همه دیگر را پیدا کنند و بهم متصل شوند. همچنین در ادامه مشاهده کردیم که چگونه دستگاه ها می توانند با استفاده از برنامه نویسی Socket با دو روش Multi thread و Asynchronous پیام ارسال و دریافت کنند. با توجه به ویژگی های هر کدام از این دو روش به این نتیجه رسیده ایم که زمانی که تعداد اتصالات زیاد می شود؛ روش Asynchronous روش بهتری است.

مراجع

پیوست آ

مدیریت مراجع در لاتک

پیوست ب

جدول، نمودار و الگوریتم در لاتک

واژه‌نامه فارسی به انگلیسی

واژه‌نامه انگلیسی به فارسی

Abstract:

The goal of this project is to add a social network service layer to a wireless mesh network. In this project, a mobile application for the neighborhood service is first implemented. Then the necessary services from the social service layer are identified for this service. Then, the services of this layer are engineered, designed and implemented to provide wireless mesh network connectivity. Ultimately, with this app, we can launch a social network that can even provide online social services, such as sending private messages or public ones, even without an internet connection. For example, you can connect with your neighbors in a building without an Internet connection and borrow a Ladder.

Keywords: Social Network, Wireless, Mesh Network, Mobile Application, Neighborhood



**Iran University of Science and Technology
Computer Engineering Department**

Design and implementation of local and wireless social network without Internet

**A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree
of Master of Science in Computer Engineering**

By:

Maryam Sadat Hashemi

Supervisor:

Sayyed Sauleh Eetemadi

July 2019