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


بسم الله الرحمن الرحيم

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 نسخه‌برداری‌شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. درصورت 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 مش 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 مش 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 مش، اپلیکیشن موبایل، همسایگی

فهرست مطالب

فهرست تصاویر

فهرست جداول

فهرست الگوریتم‌ها

فهرست علائم اختصاری

شتاب گرانش ..... a ($\mathrm{m/s^2}$)

نیرو ..... F (N)

فصل ۱

مقدمه

یکی از نیازهای اساسی انسان‌ها نیاز به برقراری ارتباط با دیگران و اجتماعی شدن است. پایه ای‌ترین 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 بی سابقه شبکه های اجتماعی [1] در بین مردم هستیم. به گونه ای که اکنون شبکه های اجتماعی جزئی از زندگی روزمره مردم شده است.

شبکه های اجتماعی به کاربرانش این فرصت را می‌دهد تا آراء و نظرات خود را با دیگران مطرح کنند و از نظرات مختلف مطلع شوند. در سایت‌های خرید وارد شوند و کالای موردنظر خود را جستجو کنند و نظرات خریداران قبلی را بخوانند و سفارش خود را ثبت کنند. به‌راحتی 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 فعالیت های این چنینی که ذاتی اجتماعی دارند با کمک اینترنت و ظهور شبکه های اجتماعی برای کاربران آسان شده‌اند.

در بین همه ی این قابلیت ها که شبکه های اجتماعی کنونی در اختیار ما قرار می دهند، شرایطی وجود دارد که این شبکه های اجتماعی پاسخگوی نیاز های ما نیستند. مثلا بخواهید خبر گم شدن حیوان خانگیتان را در محله ی خود اعلام کنید و یا بخواهید یک نردبان قرض بگیرید و یا به دنبال یک دندان پزشکی 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 توجه ای 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 فرزندتان 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

[2] Facebook
[3] Twitter
[4] Bluetooth
[5] Peer to Peer WiFi Connnection
[6] Internet
[7] Wireless Mesh Network

محلی اعلان کنید تا اگر پیدا شد دیگران به شما در این شبکه ی محلی خبر بدهند یا مثلا 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. مثلا امروز در مناطق جنوبی شهر آب گرفتگی داریم یا امروز هوا در ساعات ۲ تا ۳ بعد از ظهر بارانی است؛ لطفا 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.

۱ـ۱ کار های مربوطه

۱ـ۱ـ۱ 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.

۱ـ۱ـ۲ 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.

۱ـ۱ـ۳ 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

[8] https://www.opengarden.com
[9] https://nextdoor.com/
[10] https://www.opengarden.com/firechat/
[11] Android
[12] iphone

ارسال مناسب عکس و غیره اشاره نمود.

۱ـ۲ جمع‌بندی

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 ، to 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 ـ 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.

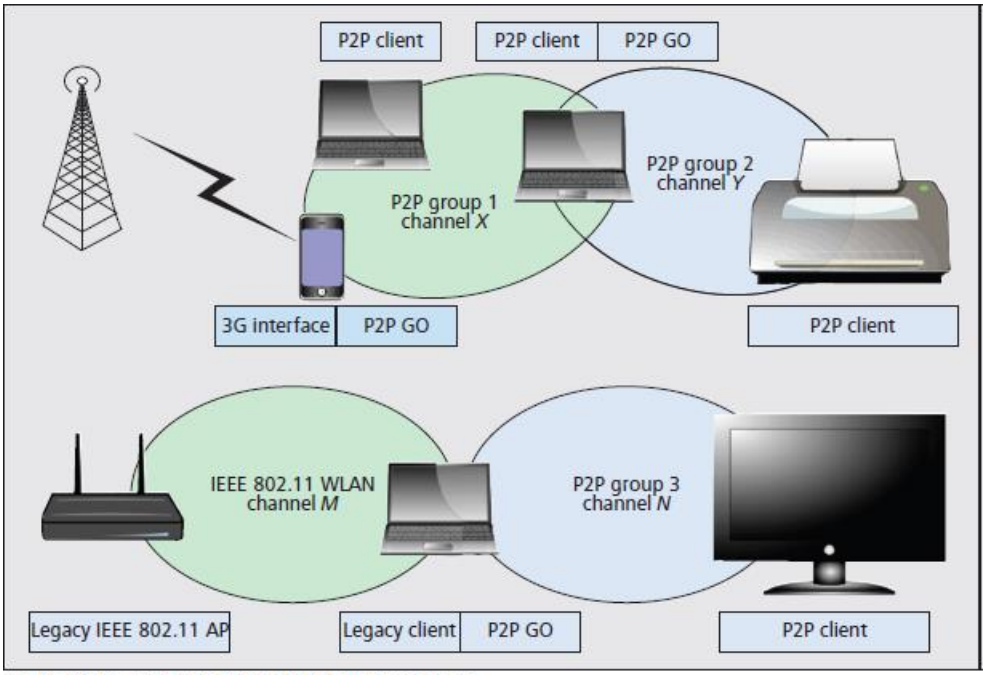
۲ ـ ۲ بررسی فنی

در یک شبکه معمولی WiFi مشتری [۳] اسکن می کند وعضو 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 ـ 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 ی 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 ـ ۱)

[1] WiFi alliance
[2] wireless access point
[3] client
[4] dynamic
[5] Time sharing



Wi-Fi direct supported topologies and use cases.

شکل ۲ ـ 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 ـ ۱) این ارتباط باید در لایه ی شبکه [۶] اتفاق بیفتد و معمولا 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 ـ ۴ تشکیل گروه

سه نوع روش برای تشکیل گروه در فناوری WiFi Direct وجود دارد که عبارتند از استاندار [۸] ،مستقل [۹] و پایدار [۱۰] .

[6] Network Layer
[7] Network Address Translation
[8] Standard
[9] Autonomus
[10] 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.

۲ـ۴ـ۱ 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.

۲ـ۴ـ۲ 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 ۲ـ۴ـ۱، مرحله پیدا کردن در این مورد ساده تر است، زیرا مرحله مذاکره برای مالک گروه شدن حذف شده است.

[11] Credentials
[12] Configuartion
[13] Discovery

۲ـ۴ـ۳ 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.

۲ـ۵ 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 ازاحراز هویت [۱۷] جدید دوباره ارتباط برقرار می کند.

۲ـ۶ ذخیره انرژی

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

۱. پروتکل Opportunistic Power Save

۲. پروتکل Notice of Absence

[14] Flag
[15] Invitation
[16] WiFi Protected Setup
[17] Authentication

۲ـ۶ـ۱ 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.

۲ـ۶ـ۲ 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

۲ـ۷ 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.

[18] Sleep

۲ـ۸ 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 پیاده‌سازی

۳ـ۱ اتصال دستگاه‌ها در لایه‌ی فیزیکی با استفاده از فناوری WifiDirect

برای اجرای خدمات لایه‌ی شبکه‌ی اجتماعی، ابتدا لازم است که دستگاه‌ها در لایه‌ی فیزیکی به یکدیگر متصل شوند و امکان ارسال و دریافت اطلاعات در لایه‌ی فیزیکی فراهم شود. به همین جهت از فناوری WiFi Direct استفاده خواهیم کرد تا شرایط لازم در لایه‌ی فیزیکی را برای ما برقرارکند. در ادامه‌ی این بخش توضیح خواهیم که سه مورد Discovery ، Connect و Create Group برای برقراری ارتباط در لایه‌ی فیزیکی چگونه در WiFi Direct انجام می‌شود.

۳ـ۱ـ۱ Discovery

برای برقرای ارتباط، باید بررسی کنیم که چه افرادی در نزدیکی ما حضور دارند. به این کار Discovery می‌گوییم. برای انجام Discovery در WiFi Direct کافی است از متد ()discoverPeers استفاده کنیم. با صدا زدن این متد، Discovery آغاز می‌شود و در صورت موفقیت‌آمیز بودن، با صدا زدن متد ()requestPeers می‌توانیم به لیست افرادی که در نزدیکی ما هستند؛ دست پیدا کنیم و تمامی افرادی که آمادگی لازم برای برقراری ارتباط دارند را مشاهده کنیم. توجه داشته باشید که در این روش تنها افرادی را می‌توان مشاهده کرد که آن‌ها هم عملیات Discovery را انجام دهند. در غیر این صورت شما نمی‌توانید آن‌ها را در لیست خود مشاهده کنید.

۳ـ۱ـ۲ Connect

بعد از پیدا کردن افراد نزدیک به خودمان، باید بتوانیم به فرد مورد نظر متصل شویم. برای این کار با دانستن اطلاعات پیکربندی دستگاه مورد نظر و با استفاده از متد ()connect عملیات اتصال را آغاز می‌کنیم. اگر متد ()connect موفقیت آمیز باشد، با استفاده از رابط ConnectionInfoListener بررسی می‌کنیم که آیا وضعیت اتصال دستگاه فعلی تغییر کرده است یا خیر. با استفاده از متد ()onConnectionInfoAvailable که رابط ConnectionInfoListener در اختیار ما قرار می‌دهد، چک می‌کنیم که کدام یک از این دستگاه‌ها مالک گروه هست و کدام یک مشتری گروه هست. بنابراین با توجه به نقش هر دستگاه در این ارتباط نقش Server یا Client را برای آن‌ها انتخاب خواهیم کرد. توجه شود که اطلاعات پیکربندی دستگاه شامل مواردی مانند آدرس MAC [۱] دستگاه و اطلاعات WPS [۲] می‌باشد.

۳ـ۱ـ۳ Create Group

در بخش قبل توضیح دادیم که چگونه می‌توانیم 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 موفقیت‌آمیز اجرا شد؛ با فراخوانی متد ()requestGroupInfo می‌توانیم جزئیات بیشتری در مورد دستگاهی‌هایی که در این گروه قرار دارند مانند نام این دستگاه‌ها و وضعیت اتصال آن‌ها را درخواست بدهیم. سپس با استفاده از متد ()onGroupInfoAvailable که رابط GroupInfoListener در اختیار ما قرار می‌دهد؛ می‌توانیم اطلاعات مرتبط با گروه را مشاهده کنیم.

۳ـ۲ دریافت و ارسال پیام

در بخش قبل دیدیم که چگونه می‌توان با استفاده از WiFi Direct دستگاه‌ها را در لایه‌ی فیزیکی به یکدیگر متصل کرد. در این بخش می‌خواهیم با استفاده از برنامه‌نویسی socket و از طریق دو روش Multi thread و Asynchronous این امکان را ایجاد کنیم که دستگاه‌ها با یکدیگر پیام رد و بدل کنند. بنابراین در یک ارتباط،

[1] Media Access Control
[2] Wi-Fi Protected Setup

لازم است که یکی از دستگاه‌ها 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 می‌گیریم. در ادامه‌ی این بخش وظیفه‌ی server و client را در هر دو روش Multi thread و Asynchronous بیان می‌کنیم و توضیح خواهیم داد که هر کدام از چه متدهایی استفاده خواهند کرد.

۳ـ۲ـ۱ Multi thread یا Synchronous

همانطور که از اسم این روش پیداست، در این روش، برای هر قسمت از کاری که می‌خواهیم انجام بدهیم برای مثال اتصال، ارسال پیام و دریافت پیام و ... یک thread اختصاص می‌دهیم که همه‌ی آن‌ها به صورت همزمان در حال انجام وظیفه‌ی خودشان هستند. دراین روش ما منتظر جواب از طرف مقابل هستیم و در واقع تا دریافت جواب، آن متد مسدود یا Block می‌شود. شبیه زمانی است که شما در حال مکالمه با دوستتان در پشت تلفن هستید. وقتی که شما حرفی می‌زنید، منتظر هستید تا نفر دیگر پاسخ شما را بدهد. بنابراین همین انتظار باعث می‌شود تا زمان زیادی مصرف شود و منابع ما هدر برود.

همچینین این روش به صورت دنباله‌ای متوالی انجام می‌شود. یعنی شما یک کانال برای ارتباط ایجاد می‌کنید(شماره‌ی دوستتان را می‌گیرید) و در ادامه یک سری اطلاعات را تبادل می‌کنید(شما یا دوستتان صحبت می‌کنید و منتظر جواب طرف مقابل هستید.) و در آخر هم کانال ارتباطی را می‌بندید(تلفن را قطع می‌کنید).

در این روش منابع موجود، بین thread ها به اشتراک گذاشته می‌شود. این روش برای تعداد اتصالات کم مناسب است. اما وقتی تعداد اتصالات وسیع باشد؛ دیگر نمی‌توان همه‌ی اتصالات را به صورت درست مدیریت کرد. در واقع برنامه‌ریزی [۳] thread ها سربار ایجاد خواهد کرد. زیرا تعدادشان زمانی که اتصالات زیاد می‌شود؛ افزایش پیدا می‌کند.

برای پیاده‌سازی این روش از سه کلاس استفاده کرده‌ایم که در ادامه هر کدام را به صورت مجزا توضیح خواهیم داد.

Server

در کلاس Server دو متد را پیاده‌سازی کردیم:

[3] Scheduling

۱. ()Accept : در این متد server در یک حلقه‌ی بی‌نهایت بر روی درگاه ۸۸۸۸ در حال گوش‌دادن به درخواست‌های client است. بعد از آن که server درخواست یک client را قبول می‌کند، برای این client یک ClientHandler می‌سازد و آن را به لیست Client هایی که به سرور متصل هستند، اضافه می کند. متد ()Accept برای انجام همه‌ی این مراحل از یک thread استفاده می‌کند.

۲. ()Send :در این متد server یک پیام را برای همه‌ی client هایی که به آن متصل هستند؛ ارسال می‌کند. در واقع پیام را بر روی جریان خروجی داده که توسط socket ما بین server و client ایجادشده‌است؛ قرار می‌دهد. برای انجام این کار، از یک thread استفاده می‌شود.

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 آماده‌ی دریافت پیام از سوی server می‌شود. توجه شود که همه‌ی این کارها توسط یک thread انجام می‌شود.

۲. ()Send : در این متد Client پیامی را که می‌خواهد برای server ارسال کند را بر روی جریان خروجی داده که به کمک socket بین این دو ایجادشده‌است، قرار می‌دهد. از یک thread برای ارسال پیام استفاده می‌کنیم.

۳. ()Receive : در این متد پیامی را که server برای client ارسال کرده‌است را از روی جریان ورودی داده می‌خوانیم. همانند ارسال برای دریافت هم از یک thread استفاده می‌کنیم.

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 پیام‌هایی را که از client هایش دریافت می‌کند را برای سایر client هایش ارسال کند. کاری که دقیقا در این متد انجام می‌شود؛ این است که پیام را از جریان

ورودی داده‌ی یکی از client هایش می‌خواند و سپس آن را بر روی جریان خروجی داده سایر client هایش قرار می‌دهد. دقت شود که همه‌ی این کارها توسط یک thread خاص اجرا می‌شود.

۳ـ۲ـ۲ Asynchronous

در این روش ما کارهای لازم در شبکه را پست می‌کنیم ولی منتظر جواب نمی مانیم بلکه نتیجه را بعدا بررسی می‌کنیم. شبیه موقعی است که شما به دوستتان پیام متنی ارسال می‌کنید و منتظر جواب درهمان لحظه نمی‌مانید و جواب دوستتان ممکن است الان یا یک ساعت دیگر برسد. بنابراین جواب را بعدا چک می‌کنید.

این روش بر خلاف روش synchronous به صورت دنباله‌ای نیست. اما دنبال کردن وضعیت هر یک از کارها کمی مشکل است. بنابراین با توجه به ویژگی‌هایی که برای این روش ذکر کردیم؛ می‌توانیم با استفاده از این روش از هدر رفتن منابع جلوگیری کنیم. همچینین می‌توانیم تعداد اتصالات بیشتری را با همان سخت افزار موجود مدیریت کنیم.

برای پیاده سازی این روش از ۴ کلاس استفاده کرده‌ایم. 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 می‌کنیم که در آن از متد ()StartWrite استفاده شده است که بوسیله ی آن می‌توان پیام را بر روی بافر دستگاه مورد نظر قرار داد. چون نوشتن بر روی بافر، یک عمل ناهمزمان است از CompletionHandler استفاده می‌کنیم 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 می‌کنیم که در آن از متد ()StartRead استفاده شده‌است که بوسیله‌ی آن می‌توان پیام را از بافر دستگاه مورد نظر خواند. چون خواندن از روی بافر یک عمل

ناهمزمان است از CompletionHandler استفاده می‌کنیم که کامل شدن یا شکست خوردن را به thread اصلی اعلام می‌کند.

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

برای انجام این کار از متد ()Broadcast استفاده می‌کنیم که در آن پیام بر روی بافر همه‌ی دستگاه‌هایی که به دستگاه فعلی متصل هستند؛ قرارداده می‌شود. این عمل نیز به صورت ناهمزمان اجرا می‌شود.

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

برای انجام این کار از متد ()ReceiveBroadcast استفاده می‌کنیم که پس از دریافت یک پیام آن را بر روی بافر تمامی دستگاه‌های متصل به دستگاه فعلی قرار می‌دهیم. این عمل هم به صورت ناهمزمان اجرا می‌شود.

Device

هدف از ایجاد این کلاس این است که برای هر دستگاه یک کانال ناهمزمان 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 حلقه‌ی بی نهایت بر روی درگاه ۸۸۸۸ شروع به گوش دادن به درخواست از طرف client ها می‌کند. اما این کار باعث انسداد یا Blocking نمی‌شود. بلکه به صورت ناهمزمان انجام می‌شود. بنابراین از یک CompletionHandler استفاده کرده‌ایم. که در صورتی که یک client به server متصل شود و عمل کامل شود؛ سرور متد ()ReceiveBroadcast را از کلاس WiFiNetService فراخوانی می‌کند و آماده‌ی دریافت پیام از یکی از client هایش می‌شود و سپس آن را برای تمامی دستگاه‌های متصل، ارسال می‌کند.

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 می‌شود. این عمل نیز به صورت ناهمزمان انجام می‌شود.

بنابراین از یک CompletionHandler استفاده کرده‌ایم و در صورت کامل شدن اتصال، دستگاه با فراخوانی متد ()Receive از کلاس WiFiNetService آماده‌ی دریافت پیام می‌شود.

۳ـ۳ جمع بندی

در این فصل بیان کردیم که چگونه دو دستگاه یا چندین دستگاه می‌توانند در لایه‌ی فیزیکی با استفاده از فناوری WiFiDirect هم دیگر را پیدا کنند و بهم متصل شوند. همچنین در ادامه مشاهده کردیم که چگونه دستگاه‌ها می‌توانند با استفاده از برنامه‌نویسی Socket با دو روش Multi thread و Asynchronous پیام ارسال و دریافت کنند. با توجه به ویژگی‌های هر کدام از این دو روش به این نتیجه رسیده‌ایم که زمانی که تعداد اتصالات زیاد می‌شود؛ 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