



دانشگاه آزاد اسلامی

واحد علوم و تحقیقات

دانشکده فنی و مهندسی، گروه کامپیوتر

پایان نامه ی کارشناسی ارشد مهندسی کامپیوتر – نرم افزار (MSc)

عنوان سمینار

ارزیابی کارایی رایانش مه برای اجرای بازی های رایانه ای

استاد راهنما

جناب آقای دکتر سید مهدی جامعی

نگارنده

کوروش خانزادی

بهار و تابستان ۱۳۹۸

فهرست مطالب

شماره صفحه

عنوان

چکیده

فصل اول: کلیات تحقیق

۱-۱- مقدمه	۶
۱-۲- بیان مسئله	۷
۱-۳- اهمیت و ضرورت انجام تحقیق	۷
۱-۴- جنبه جدید بودن و نوآوری در تحقیق	۷
۱-۵- اهداف مشخص تحقیق	۸
۱-۶- سوالات تحقیق	۸
۱-۷- فرضیه های تحقیق	۸
۱-۸- متغیر های تحقیق	۸
۱-۹- ساختار پایان نامه	۹

فصل دوم: مروری بر ادبیات و پیشینه تحقیق

۱-۲- ادبیات تحقیق	۱۱
۲-۲- فناوری هایی که رایانش مه را به لبه شبکه آوردند	۱۱
۲-۳- ضوابط ارزیابی معماری ها و الگوریتم های رایانش مه	۱۱
۲-۴- معماری های رایانش مه	۱۴
۲-۴-۱- معماری End-user application provisioning	۱۵
۲-۴-۲- معماری Resource management	۱۶
۲-۴-۳- معماری Communication	۱۷
۲-۵- الگوریتم های رایانش مه	۱۸
۲-۵-۱- الگوریتم Resource sharing	۱۸
۲-۵-۲- الگوریتم Task scheduling	۱۹
۲-۵-۳- الگوریتم Offloading and load distribution	۱۹
۲-۵-۴- الگوریتم Content storage and distribution	۲۰
۲-۵-۵- الگوریتم برای سایر کاربرد ها	۲۱

۲۲.....	۶-۲- چالش ها و جهات تحقیق در حوزه های معماری رایانش مه
۲۳.....	۷-۲- چالش ها و جهات تحقیق در حوزه های الگوریتم رایانش مه
۲۵.....	۸-۲- تعاریف Cloud gaming
۲۵.....	۹-۲- پلتفرم های بازی ابری
۲۸.....	۱۰-۲- بهینه سازی پلتفرمهای ابری
۲۸.....	۱-۱۰-۲- Resource allocation : زیر ساختی
۲۸.....	۲-۱۰-۲- Distributed architectures : زیر ساختی
۲۹.....	۳-۱۰-۲- ارتباطات
۲۹.....	۴-۱۰-۲- فشرده سازی ویدئو
۳۰.....	۵-۱۰-۲- فشرده سازی گرافیک
۳۰.....	۶-۱۰-۲- فشرده سازی هایبرید
۳۱.....	۱۱-۲- سرویسهای بازی ابری تجاری

فهرست شکل ها

عنوان شکل ها	شماره صفحه
شکل (۱-۲) معماری رایانش مه	۱۴.....

چکیده

در این پژوهش به بررسی اجرای بازی‌های رایانه‌ای با استفاده از رایانش مه پرداخته شده است. جهت بررسی این موضوع، ابتدا به بررسی رایانش مه در سطح معماری و الگوریتم پرداخته شده است و سپس به بررسی اجرای بازی‌های رایانه‌ای با استفاده از رایانش ابری پرداخته شده است. نتایج بررسی مطالعاتی نشان می‌دهد در اجرای بازی‌های رایانه‌ای مبتنی بر رایانش ابری تاخیر زمانی و عدم اطمینان از درستی در ارسال و دریافت داده‌های کنترلی کاربر وجود دارد. از این رو، برای حل این مسئله در فصل یک راهکاری مبتنی بر رایانش مه جهت کاهش تاخیر زمانی و اتکاپذیری در ارسال و دریافت داده‌های کنترلی کاربر برای اجرای بازی‌های رایانه‌ای ارائه شده است.

کلمات کلیدی: رایانش مه، بازی‌های رایانه‌ای، ارزیابی کارایی، اتکاپذیری

فصل اول

کلیات تحقیق

سرگرمی و بازی همواره نقش بسیار مهمی در جوامع انسانی در دوره‌های گوناگون داشته است و امروزه با در دسترس بودن رایانه‌ها و اینترنت، بازی‌های رایانه‌ای نقش مهمی نه تنها در سرگرمی بلکه در زمینه‌هایی همچون آموزش، شبیه‌سازی و ایجاد شغل داشته است. بنابراین امروزه بازی‌های رایانه‌ای همانند صنایع دیگر نیازمند بررسی و ارائه راهکارهایی جهت توسعه و افزایش بهره‌وری در جهت ایجاد رضایت کاربران می‌باشد. بازی‌های رایانه‌ای در ابتدا بر روی رایانه‌هایی کوچک و با پشتیبانی از یک بازیکن و چند مرحله ارائه گردید و سپس با پیشرفت رایانه‌ها و در دسترس قرار گرفتن شبکه‌ی اینترنت در اختیار عموم بازی‌های رایانه‌ای نیز با این پیشرفت‌ها همراه شد و بازی‌های تحت شبکه که به صورت چند بازیکن که همزمان به یک بازی می‌پردازند ارائه گردید. کمی بعد با افزایش قدرت پردازنده‌های گرافیکی و افزایش قدرت گرافیکی بازی‌های رایانه‌ای، هزینه‌ی سخت‌افزار برای کاربران افزایش یافت. از این رو، برای کاهش قیمت تمام‌شده برای کاربران، استفاده از رایانش ابری برای بازی‌های رایانه‌ای مطرح گردید. رویکرد اصلی در این زمینه مطرح شد: ۱- بازی‌های آنلاینی که چندین بازیکن به صورت همزمان به یک ابر متصل شده و رایانش در سمت کاربران و ابر انجام می‌گیرد. در این رویکرد رایانش گرافیکی سمت کاربران است و رایانشی که نیاز به برقراری ارتباط میان بازی و کاربران است در سمت ابر انجام می‌شود و ۲- بازی‌ها بر روی ابر پردازش شده و سپس تصویر خروجی آن برای کاربران ارسال می‌گردد و کاربران نیز می‌توانند با آن از طریق کنترلرهای فیزیکی خود مانند صفحه کلید تعامل نمایند و اطلاعات کنترلی را به سمت ابر ارسال نمایند. در این رویکرد کاربر نیاز به سخت‌افزار قدرتمند برای پردازش گرافیکی نداشته و تمامی این پردازش سمت ابر صورت می‌گیرد. همچنین این رویکرد تحت سرویس‌های پخش بازی‌های ویدیویی توسط شرکت‌هایی همچون SONY ارائه می‌گردد. بنابراین با در دسترس قرار گرفتن رایانش ابری جهت اجرای بازی‌های رایانه‌ای، نیاز به بررسی و ارائه راهکارهایی در سطح الگوریتم و معماری نرم‌افزاری و سخت‌افزاری جهت افزایش کیفیت و کارایی اجرای بازی‌های رایانه‌ای در محیط‌های غیر محلی به معنای آنکه بخشی از اجرای بازی‌ها در محیط‌هایی به غیر از کامپیوتر کاربران صورت می‌گیرد، بوجود می‌آید.

۲-۱- بیان مسئله

در بازی‌های رایانه‌ای که مبتنی بر رایانش ابری اجرا می‌شوند، ارتباط بین کاربر و ابر از طریق شبکه ای همچون اینترنت صورت می‌گیرد. سه ویژگی بسیار مهم در این دسته از بازی‌ها: ۱- سرعت قابل قبول در دریافت داده، ۲- اطمینان از ارسال و دریافت درست فرمان‌های بازی توسط کاربر در زمانی قابل قبول است و ۳- اجرای درست بازی با توجه به فرمان‌های دریافت شده در زمان مناسب می‌باشد. بنابراین، سرعت یا به معنایی دیگر عدم یا وجود تاخیر بسیار کم در اجرای درست بازی بسیار مهم می‌باشد. در رایانش ابری به دلیل وجود محدودیت‌هایی شامل: ۱- وجود تاخیر در رایانش ابری به دلیل فرآیند رایانشی در چندین بخش مختلف محاسبه و ۲- عدم وجود مرکز داده در مکان‌های فیزیکی جهت ارائه سرویس، در اجرای بازی‌های رایانه‌ای تاخیرهای زمانی و عدم اطمینان از ارسال و دریافت داده به صورت درست بوجود می‌آید. اگر بخش‌های از بازی که حساس به تاخیر زمانی باشند، به جای آنکه در فضای ابر اجرا شوند، در لبه‌ی شبکه و توسط گره‌ها اجرا شوند و بخش‌هایی که فقط نیاز پردازشی بالایی دارند در فضای ابر اجرا شوند، آنگاه ویژگی مطرح شده در اجرای بازی‌های رایانه‌ای در محیطی غیر محلی می‌تواند ارائه گردند.

۳-۱- اهمیت و ضرورت انجام تحقیق

وجود تاخیر زمانی و عدم اطمینان از ارسال و دریافت درست داده در اجرای بازی‌های رایانه تحت رایانش ابری، باعث عدم رضایتمندی کاربران در استفاده‌ی با کیفیت از این سیستم می‌گردد. همچنین اگر ویژگی‌های مطرح شده ایجاد نگردند، امکان توسعه‌ی بازی‌های رایانه‌ای با کیفیت گرافیکی بالاتر و وجود کاربران بیشتر در یک بازی آنلاین بدلیل افزایش تاخیرهای زمانی در اجرا پدید نمی‌آید. از این رو مطالعه‌ی موضوع برای ارائه‌ی راهکارهایی در جهت کاهش تاخیر زمانی و اطمینان حاصل پیدا کردن از ارسال و دریافت درست داده در اجرای این دسته از بازی‌ها دارای اهمیت می‌باشد.

۴-۱- جنبه جدید بودن و نوآوری در تحقیق

- بررسی نیازمندی‌ها در اجرای بازی‌های رایانه با استفاده از محاسبات مه
- بررسی معماری‌های نرم افزاری جهت اجرای بازی‌های رایانه با استفاده از رایانش مه
- بررسی الگوریتم‌های رایانشی برای کاهش تاخیر زمانی در رایانش مه جهت اجرای بازی‌های رایانه‌ای

۵-۱- اهداف مشخص تحقیق

- ارائه سند نیازمندی ها برای اجرای بازی های رایانه با استفاده از محاسبات مه
- ارائه معماری نرم افزاری جهت اجرای بازی های رایانه با استفاده از محاسبات مه
- ارائه الگوریتم رایانشی جهت کاهش تاخیر زمانی در اجرای بازی در رایانش مه
- ارائه الگوریتم رایانشی جهت اطمینان از ارسال و دریافت درست داده های کنترلی کاربر

۶-۱- سوالات تحقیق

- آیا امکان پیاده سازی بازی های رایانه ای با استفاده از رایانش مه وجود دارد ؟
- چگونه می توان بازی های رایانه ای با استفاده از رایانش مه اجرا کرد؟
- آیا امکان کاهش تاخیر زمانی در اجرای بازی های رایانه ای با استفاده از رایانش مه توسط معماری نرم افزاری و یا الگوریتم محاسباتی وجود دارد ؟
- چگونه می توان با ارائه الگوریتمی از ارسال و دریافت داده در اجرای بازی رایانه ای در محیطی غیر محلی اطمینان حاصل کرد؟

۷-۱- فرضیه های تحقیق

- در اجرای بازی های رایانه ای در محیط ابری اگر در سطح معماری و الگوریتم محاسبات مه افزوده شود آنگاه تاخیر زمانی در اجرای بازی های رایانه ای در این محیط کاهش پیدا می کند.
- در اجرای بازی های رایانه ای در محیط ابری اگر در سطح معماری و الگوریتم محاسبات مه افزوده شود ، آنگاه با استفاده از الگوریتمی می توان از وجود ویژگی اتکاپذیری در ارسال و دریافت داده های کنترلی کاربر اطمینان حاصل کرد.

۸-۱- متغیر های تحقیق

- تاخیر زمانی در اجرای بازی های رایانه با استفاده از محاسبات مه
- اتکاپذیری داده های کنترلی کاربر در اجرای بازی های رایانه ای با استفاده از محاسبات مه

۹-۱- ساختار پایان نامه

در این پایان نامه به مطالعه و بررسی امکان اجرای بازی‌های رایانه‌ای با استفاده از محاسبات مه در جهت کاهش تاخیر زمانی و اطمینان حاصل پیدا کردن از درستی در دریافت و ارسال داده‌های کنترلی کاربر پرداخته شده است. ساختار پایان نامه بدین شرح می‌باشد: در فصل دوم به مرور ادبیات و تعاریف پایه‌ای در مورد محاسبات ابری و محاسبات مه در مورد اجرای بازی‌های رایانه‌ای پرداخته شده است.

فصل دوم

مروری بر ادبیات و پیشینه تحقیق

۱-۲- ادبیات تحقیق

رایانش مه به این دلیل مطرح شد که ارتباط بین کاربر و ابر چون شامل اینترنت نیز می شود، برای برنامه هایی که نسبت به تاخیر حساسیت دارند، تنها وجود فضای ابری کافی نیست. از طرفی برنامه های موجود در فضای ابر، ممکن است از چندین جزء تشکیل شده باشند و ارتباط بین همین اجزاء در فضای ابر خود تاخیر ایجاد می کند. به علاوه این که سرویس دهنده ی فضای ابر ممکن است در مکان هایی که باید سرویس ارائه کند، هیچ مرکز داده ای نداشته باشد. به دلیل این محدودیت های مطرح شده، بحث رایانش مه مطرح شد. رایانش مه [۱] می تواند محدودیت های مطرح شده را جبران کند. به این طریق که اگر اجزاء برنامه ای حساس به تاخیر باشند، به جای آنکه در فضای ابر اجرا شوند، می توانند در لبه ی شبکه و توسط Fog node اجرا شوند و آن اجزائی که فقط نیاز پردازشی بالایی دارند، می توانند در فضای ابر اجرا شوند. هم چنین رایانش مه این امکان را به وجود می آورد تا بتوان رایانش را در مکان های مشخص انجام داد.

۲-۲- فناوری هایی که رایانش مه را به لبه شبکه آوردند

در کنار محاسبات مه یک سری مفاهیم پایه ای نیز مطرح شده اند که تا بتوان محاسبات را در لبه ی شبکه انجام داد. اولین مفهوم به نام Cyber foraging اولین بار در مقاله ی [۲] مطرح شد. در این مفهوم دستگاه های متحرک یا Mobile device ها که از لحاظ سخت افزاری با محدودیت هایی روبرو هستند، می توانند از سرور هایی که به آن ها نزدیک هستند و به اینترنت پهن باند نیز مجهز هستند، استفاده کنند تا توان خویش را افزایش دهند. به این سرور ها Surrogate یا جایگزین گفته می شود که دو وظیفه با نام های محاسبات و Data staging دارند. Data staging به معنی پیش واکشی داده های راه دور به سرور های جایگزین و نزدیک می باشد. دومین مفهوم Cloudlet نام دارد و در مقاله ی [۳] مطرح شده است. Cloudlet ها، سرور ها یا خوشه هایی از سرور ها با منابعی غنی هستند که تنها در فاصله ی یک Hop از Mobile device ها قرار می گیرند. در هر سرور می تواند تا چندین ماشین مجازی ایجاد شود و Mobile device ها می توانند اجزائی از برنامه هایشان که نیاز به پردازش دارند را در این ماشین های مجازی به اصطلاح Offload کنند. از جمله ی ویژگی دیگر Cloudlet ها می توان به قابلیت گسترش پذیری به صورت پویا اشاره کرد که این ویژگی را از ماشین های مجازی به ارث گرفته اند. اما ویژگی مهم تر Cloudlet ها آن است که می توانند به صورت Standalone و با کمک ماشین های مجازی نقش فضای ابری را ایفا کنند.

سومین مفهوم MEC یا Multi-access Edge computing نام دارد. MEC یک ابتکار صنعتی و تحت حمایت موسسه استانداردهای مخابراتی اروپا است. در ابتدا موضوعیت آن فقط شبکه های موبایلی و استفاده از VM به عنوان ماشین مجازی بود و در نهایت موضوعیت آن به شبکه های غیر موبایلی و استفاده از فناوری های ماشین مجازی غیر از VM، گسترش پیدا کرد. در مقاله [۴] و قبل از این گسترش هدف آن بود تا محاسبات ابری در شبکه RAN یا Radio Access Network به لبه ی شبکه های موبایلی آورده شود و در این حالت سرور های محاسباتی Mobile-edge در مکان های قرار گیری ایستگاه های LTE و 3G، مستقر می شدند. از کاربرد های این فناوری می توان به انجام محاسبات برای واقعیت افزوده نام برد. هدف نهایی این موسسه از به وجود آوردن این فناوری، استاندارد سازی API ها بین پلتفرم های Mobile-edge و برنامه های کاربردی و در یک محیط Open بود. چهارمین مفهوم بحث خود Fog computing یا محاسبات مه می باشد که در ابتدا و در سال 2012 توسط CISCO مطرح شد. محاسبات مه همان محاسبات ابری است که به لبه ی شبکه ی کاربران و دستگاه های IOT منتقل شده است. Fog به طور کلی به فضای ابری وابسته است و نمی تواند به صورت Standalone عمل کند. بر خلاف MEC و Cloudlet که وابسته به Cloud نبودند.

۳-۲- ضوابط ارزیابی معماری ها و الگوریتم های رایانش مه

قبل از اینکه مقاله های ارائه شده در حوزه ی معماری Fog بررسی شوند، در مقاله ی مروری [۵]، یک سری ضوابط ارزیابی از C1 تا C6 مطرح شده اند که ابتدا به توضیح این ضوابط پرداخته خواهد شد.

▪ ضابطه C1: Heterogeneity

گروه ها در Fog و Cloud، از لحاظ امور رایانشی با یکدیگر ناهمگن می باشند که سیستم Fog می بایستی با این ناهمگنی خود را تطبیق دهد.

▪ ضابطه C2: QoS management

به دلیل نزدیکی گروه های Fog به End-user device ها، ویژگی Real-time بودن برخی برنامه های کاربردی با به حداقل رساندن تاخیر، بایستی فراهم شود. کنترل تاخیر یا latency خود زیر مجموعه QoS می باشد، پس QoS در سیستم Fog، باید مدیریت شود.

▪ ضابطه C3: Scalability

هر سیستم Fog خود شامل تعداد زیادی Fog domain است و هر Fog domain خود شامل تعدادی Fog node می باشد. هم چنین این سیستم می تواند شامل میلیون ها IOT/end-user device شود و هر کدام می توانند از تعداد زیادی برنامه ی کاربردی را شامل شوند. واضح است که با توجه به این حجم از موارد مطرح شده، سیستم Fog باید خاصیت کشسانی و گسترش پذیری داشته باشد.

▪ ضابطه C4: Mobility

Fog node ها و IOT/end-user device ها می توانند Mobile یا متحرک باشند. پس سیستم Fog باید از این ویژگی پشتیبانی کند.

▪ ضابطه C5: Federation

سیستم Fog می تواند به صورت جغرافیایی توزیع شده باشد و ممکن است هر Fog domain توسط یک provider خاص ارائه شده باشد. هم چنین خود Cloud نیز می تواند توسط چندین provider ارائه شده باشد. حال برنامه های کاربردی که در این سیستم ها ارائه شده است ممکن است دارای چندین Component باشد و هر Component توسط چندین provider میزبانی شده باشد. پس باید بین این provider ها یک اجماع یعنی یک فدراسیون موجود باشد.

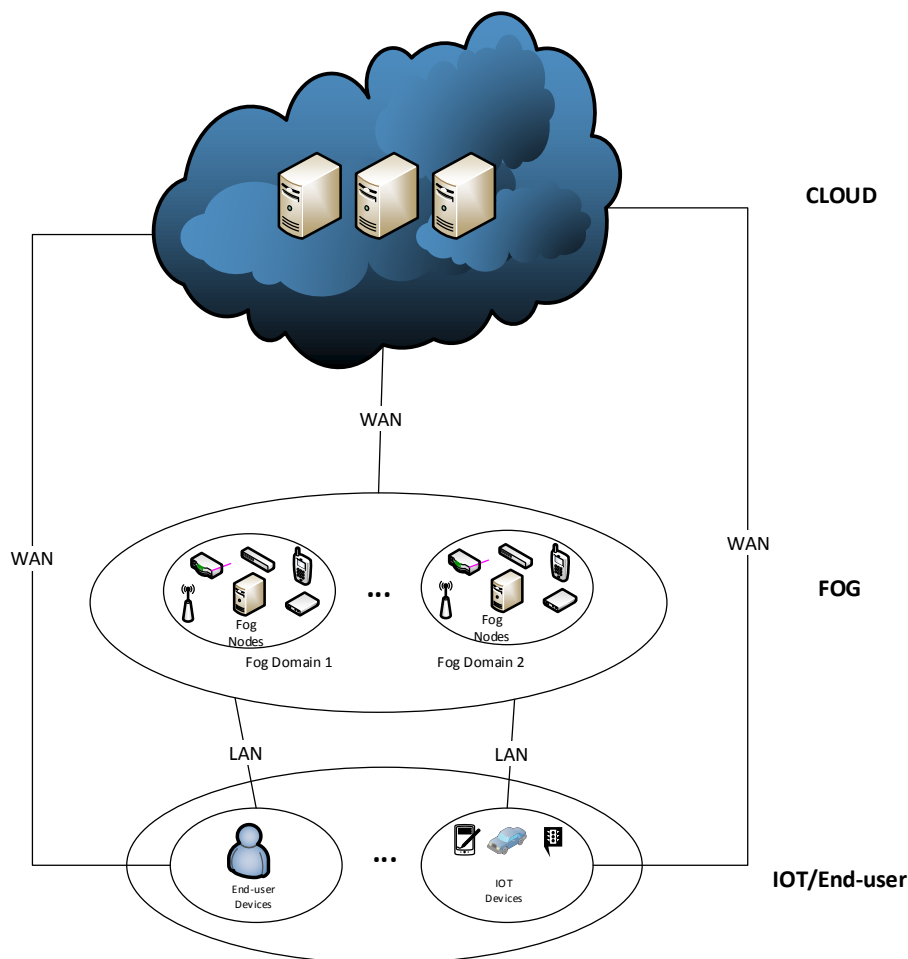
▪ ضابطه C6: Interoperability

به دلیل آنکه Component های برنامه های کاربردی بین چندین Provider ممکن است توزیع شده باشد، برای اجرای صحیح این برنامه ها باید یک همکاری در سطح provider ها در سیستم Fog وجود داشته باشد.

حال با توجه به ضوابط بیان شده، معماری های سیستم Fog بررسی خواهند شد.

۴-۲- معماری های رایانش مه

شکل ۱-۲، نمایانگر معماری Fog می باشد که شامل سه لایه است. لایه اول Cloud و لایه ی دوم Fog و لایه ی سوم IOT/end-user، نام دارد. لایه ی Fog می تواند شامل یک چند Fog domain باشد و می تواند توسط یک یا چند provider ارائه شوند. هر کدام از این Fog domain ها شامل چندین Fog node می باشد که شامل Router و Switch و PC و تلفن های همراه هوشمند و غیره می باشند. لایه ی IOT/end-user خود شامل دو Domain با نام های IOT devices و End-user devices می باشد. اتصال لایه ی IOT/end-user و لایه ی Fog از طریق LAN می باشد و اتصال لایه ی IOT/end-user و لایه ی Cloud از طریق WAN، با گذر از Fog یا بدون گذر از آن می تواند انجام شود.



شکل (۱-۲) معماری رایانش مه [۵]

این معماری برای Fog به صورت کلی مطرح شد و اما به صورت مشخص تر، دو دسته معماری برای Fog مطرح می شود که شامل موارد زیر است:

▪ **Application agnostic architectures :**

این نوع معماری به طور مشخص روی جنبه های معماری شامل end-user application provisioning و Resource management و Communication issues و Fog and cloud federation تمرکز دارد که در ادامه معماری های مطرح شده در این جنبه ها مورد بررسی قرار خواهند گرفت و هدف آن برنامه های کاربردی برای End-user می باشد.

▪ **Application specific architectures :**

بیشتر تمرکز این نوع معماری در بخش Healthcare می باشد.

۲-۴-۱- معماری End-user application provisioning

در این قسمت دو نوع معماری کد نویسی بیان شده است. اولین نوع Mobile-fog نام دارد که در مقاله ی [۶] مطرح شده است. در این نوع معماری می توانند برنامه هایی نوشته شوند و به هر گره و در هر سطح از معماری Fog اعمال شود. در طول اجرا، برنامه ی کاربردی اطلاعاتی را از منابع و توانایی های سطحی از معماری Fog که به آن تعلق دارد را جمع آوری می کند و حال بر مبنای همین اطلاعات دریافتی، برنامه ی کاربردی تصمیم می گیرد کدام دستور العمل ها را اجرا کند. این معماری هم چنین یک سری API ارائه می دهد تا اجزای مختلف و توزیع شده ی برنامه ی کاربردی بتوانند با هم ارتباط برقرار کنند. به دلیل ارائه ی API، که باعث شد اجزاء مختلف برنامه ی کاربردی در هر سطحی بتوانند با هم ارتباط برقرار کنند، پس ضابطه ی C1 رعایت شده است. ضابطه ی C2 به دلیل عدم ارائه ی راهکار برای QoS management، رعایت نشده است. معماری Mobile-fog قابلیت گسترش پذیری دارد. پس ضابطه ی C3 رعایت شده است. معماری Mobile-fog دارای مدل هایی برای پیش بینی موقعیت های احتمالی آینده ی دستگاه های

mobile است. پس ضابطه ی C4 رعایت شده است. در این معماری حرفی از هماهنگی برای اجرا بین گره های توزیع شده بیان نشده است. پس ضابطه ی C5 رعایت نشده است. در این معماری به دلیل فراهم آوردن Interface های داده و کنترل، ضابطه ی C6 رعایت شده است. دومین نوع معماری کد نویسی در رابطه با Distributed Data Flow یا DDF می باشد که در مقاله ی [۷] مطرح شده است. در مفهوم DDF، Topology در یک برنامه ی کاربردی به شکل یک گراف جهت دار یا Flow بیان می شود و هر گره در این گراف بیان کننده ی هر جزء از برنامه ی کاربردی می باشد. توسط DDF می شود برنامه های کاربردی سمت end-user را طوری که هم Cloud و هم Fog را پوشش دهد، را توسعه داد. گره ها در DDF نقش مهمی دارند. به این شکل که هر گره چه در Fog و چه در Cloud، می تواند یک یا چند Component از برنامه های کاربردی را اجرا کنند. اما اجرای این اجزاء، با توجه به Topology گراف صورت می گیرد. هم چنین این گره ها دارای ماژول هایی برای برقراری ارتباط با گره های دیگر هستند. هم چنین این ماژول ها، زمان ممکن برای گسترش را تصمیم گیری می کنند. در این معماری کد نویسی برای گره های نا همگن ممکن است. پس ضابطه ی C1 رعایت شده است. هم چنین تاخیر با توجه به Topology در نظر گرفته شده است. پس ضابطه ی C2 رعایت شده است. به دلیل وجود ماژول هایی برای تصمیم گیری زمان برای گسترش، ضابطه ی C3 نیز رعایت شده است. در این معماری یک Component می تواند در گراف تکثیر یابد و به موقعیت مکانی که IOT device حرکت کرده است، نقل مکان کند. پس ضابطه ی C4 رعایت شده است. ضابطه ی C5 رعایت نشده است، چرا که در حالت توزیع شدگی اجرای برنامه ی کاربردی در نظر گرفته نشده است. به دلیل وجود Interface برای Control و Data، ضابطه ی C6 رعایت شده است.

۲-۴-۲- معماری Resource management

در مقاله [۸]، یک معماری ارائه شده است که توسط آن می توان بار کاری را بین فضای ابر و مه توزیع کرد. این معماری دارای سه لایه با نام های Client و Fog و Cloud می باشد که هر کدام شامل مواردی است. لایه ی Client که در طبقه ی IOT/end-user در معماری Fog، مطرح شده است شامل Mobile device ها و Sensor ها و خود Client ها می شود. لایه ی Fog که در طبقه ی Fog و در معماری Fog، مطرح شده است، شامل یک Module با نام Fog server manager می شود و وظیفه ی

آن بررسی این موضوع است که آیا منابع پردازشی کافی برای میزبانی از اجزاء برنامه ی کاربردی وجود دارد یا خیر. بر حسب شرایط یا همه ی اجزاء با هم اجرا می شوند یا قسمتی اجرا و قسمتی دیگر با تاخیر اجرا می شوند و یا اگر نتوانست اجرایشان کند، به لایه ی Cloud ارجاع داده خواهند شد. لایه ی Cloud که در طبقه ی Cloud در معماری Fog، مطرح شده است، دارای Module برای دریافت اجزاء برنامه ی کاربردی از Fog server manager است. در این مقاله با در نظر گرفتن مازول با نام Fog server manager، که وظیفه ی تطبیق ظرفیت پردازشی Fog node ها و نیازهای پردازشی اجزاء برنامه ی کاربردی را بر عهده دارد، پس ضابطه ی C1 رعایت شده است. در این مقاله هیچ مازولی برای QoS management در نظر گرفته نشده است پس ضابطه ی C2 رعایت نشده است. هم چنین با در نظر نگرفتن تعداد End user های پشتیبانی شده، ضابطه ی C3 رعایت نشده است. در این مقاله در مورد قابلیت متحرک بودن User و یا Fog node سخنی به میان نیامده است. پس ضابطه ی C4 رعایت نشده است. هم چنین از همکاری بین Fog provider و Cloud provider برای اجرای توزیع شده ی اجزاء برنامه ی کاربردی صحبتی نشده است. پس ضابطه ی C5 رعایت نشده است. برای ارتباط بین Fog و Cloud از Interface های Data و Control صحبتی نشده است. پس ضابطه ی C6 رعایت نشده است.

۲-۴-۳- معماری Communication

در محیط Fog، دو نوع ارتباط به صورت ارتباط های درون لایه ای و ارتباط های برون لایه ای مطرح است. در مقاله [۹]، به ارتباط های درون لایه ای و در لایه ی IOT/end-user پرداخته شده است. در این مقاله سه نوع ارتباط بین Device ها با انواع ارتباط مستقیم و ارتباط از طریق لایه ی Fog و ارتباط از طریق لایه ی Cloud، در لایه ی IOT/end-user، در نظر گرفته شده است. برای هر کدام از انواع ارتباط بیان شده، تاخیر ارسال داده بین Device ها با سه نوع پروتکل ارتباطی با نام های CoAP و SNMP و NETCONF، اندازه گرفته شده است. در نهایت با توجه به نتایج، ارتباط مستقیم به عنوان سریع ترین ارتباط مشخص شد. در این مقاله با در نظر نگرفتن ظرفیت های گره های ارسال کننده داده به مانند تاخیر شبکه و یا تاخیر پردازشی، می توان گفت که ضابطه ی C1 رعایت نشده است. در این مقاله تاخیر ارتباطی بین Fog و Cloud در نظر گرفته شده است که به معنای در نظر گرفتن QoS است. پس ضابطه ی C2 رعایت شده

است. گسترش پذیری دستگاه های IOT و گره های Fog در نظر گرفته نشده است. پس ضابطه ی C3 رعایت نشده است. هم چنین درباره ی Mobility دستگاه های IOT سخنی بیان نشده است. پس ضابطه ی C4 نیز رعایت نشده است. هم چنین ضابطه های C5 و C6 به ترتیب، به دلیل در نظر نگرفتن همکاری بین Fog provider و Cloud provider و در نظر نگرفتن Interface بین Fog و Cloud، رعایت نشده اند. تا این جا ارتباط درون لایه ای مطرح شد. برای ارتباطات برون لایه ای در مقاله [۱۰]، یک معماری بیان شده است که این اجازه را به User device می دهد تا تصمیم بگیرد یک جزء برنامه ی کاربردی در Fog اجرا شود یا در Cloud. در این مقاله هیچ صحبتی از ناهمگن بودن گره های میزبان نشده است و هم چنین QoS نیز در نظر گرفته نشده است. پس به ترتیب C1 و C2 رعایت نشده است. از گسترش پذیری و قابلیت متحرک بودن گره ها نیز صحبتی نشده است. پس به ترتیب C3 و C4 رعایت نشده است. هم چنین از هماهنگی بین Fog و Cloud برای پردازش گره ها سخنی به میان نیامده است. پس ضابطه ی C5 نیز رعایت نشده است. ضابطه ی C6 به دلیل در نظر نگرفتن Interface بین Fog و Cloud رعایت نشده است.

۲-۵- الگوریتم های رایانش مه

۲-۵-۱- الگوریتم Resource sharing

در مقاله ی [۱۱]، اشتراک منابع بین گره ها برای انجام امور رایانشی در محیط Fog، مورد بررسی قرار گرفته است. در شبکه های Cellular، کلمه ی Cell به معنای یک منطقه ی جغرافیایی است که مورد پوشش شبکه ی Cellular قرار گرفته است. حال در این مقاله خوشه هایی از این Cell ها را ایجاد کرده اند که هر Cell در این خوشه ها می تواند منابع را برای انجام عمل Offloading بار محاسبتی دستگاه های موبایلی، به اشتراک بگذارد. هدف در این مقاله آن بود تا به صورت هم زمان بتوان هم این خوشه ها را تشکیل داد و هم منابع رایانشی و ارتباطی را با توجه به محدودیت های تاخیری کاربران شبکه، تخصیص داد. در این مقاله به دلیل در نظر گرفتن محدودیت های رایانشی منابع و در نظر گرفتن تاخیر به ترتیب ضابطه های C1 و C2 رعایت شده است. گسترش پذیری یعنی ضابطه ی C3 و تحرک پذیری یعنی ضابطه ی C4، رعایت نشده اند. هم چنین به دلیل در نظر نگرفتن وجود چندین Provider، ضابطه ی C5 نیز رعایت نشده است.

۲-۵-۲- Task scheduling الگوریتم

در مقاله ی [۸] ، الگوریتمی ارائه شده است تا بتوان بار کاری را روی Cloud و Fog توزیع کرد. همان طور که در شکل ۱-۲ مطرح شد، سطح Fog می تواند دارای چندین Fog domain باشد و هر Fog domain شامل چندین Fog node باشد. حال در این مقاله و در Fog domain با قرار دادن یک Fog server manager، درخواست های کاربر دریافت می شود و نیاز های وی با توجه به منابع موجود در محیط Fog، تطبیق داده می شود. بر مبنای این درخواست، Fog server manager، بررسی می کند که آیا منابع رایانشی کافی در محیط Fog موجود است یا خیر. حال بر حسب منابع موجود یا تمام کار انجام می شود یا اگر منابع کافی نبود این کار ها به محیط ابر فرستاده می شود. نتیجه در این مقاله به حداقل رساندن زمان پاسخ گویی و زمان رایانش بود. در این مقاله به دلیل در نظر گرفتن محدودیت منابع موجود برای انجام امور رایانشی و با به حداقل رساندن ناهمگنی بین این دو، ضابطه ی C1 رعایت شده است. همچنین تاخیر در سمت کاربر نیز در نظر گرفته شده است. پس ضابطه ی C2 رعایت شده است. اما ضوابط C3 و C4 و C5 به ترتیب به دلیل در نظر نگرفتن گسترش پذیری و تحرک پذیری و عدم تخصیص چندین Provider، رعایت نشده اند.

۲-۵-۳- Offloading and load distribution الگوریتم

همان طور که در الگوریتم های Task scheduling که توضیح داده شد، بار کاری تنها بین گره های رایانشی توزیع می شدند و نا متوازن بودن بار کاری بین این گره های رایانشی در نظر گرفته نمی شد. در مقاله [۱۲]، هدف آن بود تا بار کاری دستگاه های موبایلی به گره های فضای Fog، به اصطلاح Offload شود. نتیجه آن شد که واگذاری بار کاری به گره های فضای مه، از هر لحاظ نسبت به طبقات دیگر معماری Fog، مثلاً Cloud، بهینه تر عمل می کرد. تا این جا ثابت می شود که عمل Offloading به گره های رایانشی در فضای مه، بهتر از بقیه ی حالات عمل میکرد. اما در مقاله [۱۳]، به حل مشکل Load distribution پرداخته شد. در این مقاله یک چارچوب Coding طراحی شد که این اجازه را می داد تا کار ها بتوانند در فضای مه باز توزیع شوند و یا کار های اضافه بتوانند به فضای مه تزریق شوند. این چارچوب یک توازن بین

بار ارتباطی و تاخیر رایانشی ایجاد می کرد. در این چارچوب دو مورد نوع Coding مطرح شد که بسته به نیازمندی های سیستم، یکی از آن دو اجرا می شود. نوع اول هدف خود را بر به حداقل رساندن پهنای باند مورد استفاده، بنا نهاد. به این طریق که در این نوع، بار کاری بیشتر به گره هایی اعمال می شد که تبادل اطلاعاتی کمتری با بقیه ی گره ها داشتند. پس به این شکل بار ارتباطی کاهش پیدا می کرد. اما نوع دوم، هدفش به حداقل رساندن تاخیر بود. در این نوع با تزریق بار رایانشی اضافه به گره هایی که باعث کاهش زمان محاسبه، در شرایطی که گره های دیگر ممکن بود کندتر و یا مسدود شده باشند، می شدند، این عمل را انجام می داد. در این مقاله ضابطه ی C1، به دلیل رعایت کردن و در نظر گرفتن بحث ناهمگنی بین گره ها، صحت دارد. هم چنین در این مقاله چارچوب پیشنهادی این اجازه را می داد تا نوع Coding با توجه به تاخیر رایانشی انتخاب شود. پس QoS یعنی ضابطه ی C2 رعایت شده است. ضوابط C3 و C4 به ترتیب و به دلیل عدم در نظر گرفتن گسترش پذیری و قابلیت متحرک بودن، رعایت نشده اند. C5 به دلیل در نظر گرفتن تنها یک Provider، نیز رعایت نشده است.

۲-۵-۴- الگوریتم Content storage and distribution

برای بحث Content storage، در ابتدا موضوع F-RAN یا Fog Radio Access Network ها مطرح شد. F-RAN یک Extension بر C-RAN یا Cloud-RAN می باشد. C-RAN در اصل عملکرد های شبکه های سلولار را به سمت فضای ابر سوق می داد. اما هنوز بحث تاخیر در فضای ابر مطرح بود. به همین دلیل با استفاده از F-RAN برخی عملکرد های مهم این شبکه به نزدیک کاربر سوق داده شد و هم چنین باعث شد تا با Cache کردن Content نزدیک کاربر، تاخیر را به حداقل برسانیم [۱۴]. اما مقاله [۱۵]، سیستم F-RAN را تکمیل تر کرد. در این مقاله هدف آن شد تا بتوانند تشخیص دهند که کدام Content در Cloud و کدام Content در Fog، Cache شود. نتیجه آن شد که Content با حجم بالا باید در فضای ابر و Content با حجم کم باید در فضای مه و در نزدیکی کاربر مستقر شود. در این مقاله ناهمگن بودن گره ها در نظر گرفته نشد. پس C1 رعایت نشد. تاخیر در نظر گرفته شد. پس ضابطه ی C2 رعایت شد. ضوابط C3 و C4 به ترتیب به دلیل در نظر نگرفتن گسترش پذیری و تحرک پذیری رعایت نشده اند. هم چنین به دلیل در نظر گرفتن تنها یک شبکه ی Cellular، ضابطه ی C5 رعایت نشد.

در مقاله [۱۶]، یک سرویس پخش زنده ویدئو برای سایر کاربران ارائه شده است. در این حالت یک تبدیل کننده ی دیجیتالی یعنی Transcoder، برای تبدیل یک Encoding خاص به Encoding دیگر، با توجه به ظرفیت های دریافت کنندگان پخش و ویژگی های شبکه، احتیاج بود. برای این کار در همین مقاله یک Transcoding framework ارائه شد که از رایانش مه استفاده می کرد. به این گونه که چندین مرکز داده به صورت منطقه ای قرار می گرفتند که هر کدام از آن ها مسئول مدیریت یک منطقه با گره های فضای مه می بود. اعمال رایانشی Transcoding به گره های فضای مه فرستاده می شد. هم چنین در این مقاله یک الگوریتم برای انتخاب تعداد کافی از گره های فضای مه برای انجام امور رایانشی Transcoding ارائه شد. هدف این الگوریتم به حداقل رساندن واگذاری مجدد Transcoding task برای رایانش آن، چه به صورت درون منطقه ای و چه به صورت برون منطقه ای بود. در این مقاله به دلیل در نظر گرفتن ظرفیت های هر گره، ضابطه ی C1 رعایت شده است. ضابطه ی C2 با در نظر گرفتن QoS برای پخش زنده، نیز رعایت شده است. به دلیل در نظر گرفتن مقیاس به صورت کوچک، می توان گفت که ضابطه ی C3 یعنی گسترش پذیری، رعایت نشده است. هم چنین قابلیت متحرک بودن گره ها در نظر گرفته نشده است. پس ضابطه ی C4 رعایت نشده است. ضابطه ی C5 به دلیل در نظر گرفتن کاربرد خاص، غیر قابل بررسی است. مورد دیگر از کاربرد های Video streaming، در مقاله [۱۷]، مطرح شده است. در این مقاله همکاری بین دستگاه های در حال پخش ویدئو برای ارتقاء QoE و در شبکه های Cellular، مورد مطالعه قرار گرفته است. مدل پیشنهادی با نام Crowdsourcing Cooperation Model، عمل Pooling را روی ظرفیت های download هر کاربر اعمال می کرد تا با آن بتواند تغییرات وضعیت کانال ارتباطی را کنترل کند و هم چنین از ظرفیت های شبکه به طور بهینه استفاده کند. در این مقاله به دلیل در نظر گرفتن ظرفیت های شبکه ی Cellular برای هر کاربر، ضابطه ی C1 رعایت شده است. ضابطه ی C2 یعنی QoS در نظر گرفته شده است. ضابطه ی C3 یعنی گسترش پذیری به دلیل در نظر گرفتن تعداد کاربران کم، رعایت نشده است. ضابطه ی C4 نیز به دلیل در نظر نگرفتن قابلیت تحرک پذیری، رعایت نشده است. ضابطه ی C5 به دلیل در نظر گرفتن کاربرد خاص، غیر قابل بررسی است.

تا بدین جا معماری ها برای Fog system بیان شده است. اما فضای Fog علاوه بر این معماری ها، دارای الگوریتم هایی در حوزه های Resource sharing و Task scheduling و Offloading and load distribution و Content storage and distribution و Energy consumption و حوزه های دیگر نیز می باشد که در ادامه به این دسته از موارد در مقالات ارائه شده پرداخته خواهد شد.

۶-۲- چالش ها و جهات تحقیق در حوزه های معماری رایانش مه

▪ Heterogeneity

راه حل های بیان شده برای مسئله ی ناهمگونی یا همان heterogeneity، هنوز بر مبنای رویکرد معنایی یا Semantic-based، نیستند. این رویکرد می تواند عبارات بیان شده در هر حوزه و ارتباطشان را به صورت ریاضی بیان کند. در فضای مشخص Fog، با تعریف درست این رویکرد می توان انواع مختلف گره ها را در هر سطح معماری Fog، یعنی Cloud و Fog و IOT، را پوشش داد تا در نهایت بتوان به تعاریفی همگن از نحوه ی اجرا برنامه های کاربردی در این گره ها رسید. در اصل تمام Provider ها در هر سطح از معماری Fog، باید درک مشترکی از منابع برای اجرای برنامه های کاربردی روی این گره ها داشته باشند. پس سمت و سوی تحقیق در این حوزه باید ایجاد سمانتیک های جامع باشد تا بتواند این ناهمگونی را پوشش دهد.

▪ QoS management

در بحث مدیریت کیفیت سرویس، سمت و سوی تحقیق با عنوان مدیریت SLA یا Service Layer Agreement مطرح می شود. در اصل در محیط پویای Fog، که می تواند چندین Provider و Domain داشته باشد و هر Provider یک Business model خاص خود را در اختیار دارد، مدیریت SLA نیز برای هر Provider، متفاوت می شود. در نهایت سمت و سوی تحقیق در این حوزه، طراحی تکنیک های مدیریت SLA به شکلی که چندین Business model ها را پوشش دهد، می باشد.

▪ Scalability

گسترش پذیری به صورت جامع یعنی به صورتی که برای تمام سطوح معماری Fog مطرح شده باشد، هنوز تعریف نشده است. پس سمت و سوی تحقیق در این حوزه، می تواند به علاوه بر در نظر گرفتن

گسترش پذیری در منابع سطح Cloud، باید گسترش پذیری در منابع سطوح دیگر یعنی Fog و IOT/end-user نیز در نظر گرفته شود.

▪ Mobility

در مقالات و در حوزه ی تحرک پذیری گره ها، تنها این تحرک پذیری بر روی گره های یک سطح از معماری Fog در نظر گرفته شده است. به عنوان مثال در بحث Healthcare فقط بر روی تحرک پذیری گره های IOT/end-user کار می شود. پس سمت و سوی تحقیق در این حوزه می بایستی ارائه ی راهکار برای در نظر گرفتن تحرک پذیری در تمام سطوح معماری Fog باشد.

▪ Federation

در این حوزه ممکن است حالتی پیش بیاید که اجزاء یک برنامه ی کاربردی بخواهد روی سطوح مختلف معماری Fog و در انواع Domain ها، اجرا شود. در این حالت سمت و سوی تحقیق، برای اجماع دامنه ها و سطوح Fog، طراحی مکانیزمی برای ترکیب اجزاء مختلف برنامه ی کاربردی، می باشد. در مقاله [۱۸]، به طور کلی دو مکانیزم ترکیب با نام های Orchestration و choreography مطرح می شود. در اولی یک موجودیت مرکزی وظیفه ی ترکیب اجزاء برنامه را بر عهده دارد و در دومی اجزاء برنامه به صورت غیر متمرکز کنترل می شوند.

▪ Interoperability

این لغت به معنای قابلیت یک سیستم در تبادل اطلاعات تعریف می شود. حال برای این کار باید یک Interface در موارد Data و Signaling و Control، بین سطوح Fog و Cloud در یک سیستم ایجاد شود. پس به عنوان سمت و سوی تحقیق در این حوزه باید یک توافق برون دامنه ای ایجاد شود و سپس طراحی استاندارد برای Interface که این توافق را پیاده سازی می کند.

۷-۲- چالش ها و جهات تحقیق در حوزه های الگوریتم رایانش مه

▪ Heterogeneity

ناهمگونی در بخش الگوریتم های مطرح در سیستم Fog، به معنای مدل کردن تفاوت های بین گره ها با توجه به ظرفیت های آن ها می باشد. در این بخش دو چالش مطرح می شود. چالش اول این است که

چقدر می توان از منابع دستگاه های IOT در اطراف کاربر به عنوان Fog node برای انجام درخواست های رایانشی دستگاه های دیگر استفاده کرد. چالش دوم این است که علاوه بر منابع دستگاه های IOT در اطراف کاربر به عنوان Fog node، بتوان گره های رایانشی اضافی را به سیستم اضافه کرد. حال سمت و سوی تحقیق در این حوزه، بحث Optimization در جایگذاری گره های رایانشی اضافی مطرح می شود.

▪ QoS management

در اکثر مقالات تنها یک متریک به نام Latency یا تاخیر به عنوان یک متریک کارایی مطرح می شود و متریک های دیگر سیستم مثلاً پهنای باند در سمت کاربر در نظر گرفته نمی شود که این خود یک چالش است.

▪ Scalability

اگر یک الگوریتم در سیستم Fog مطرح می شود، باید توانایی این را داشته باشد تا در مقیاس های وسیع قابلیت پیاده سازی داشته باشد. که این خود یک چالش است. اما سمت و سوی تحقیق در این حوزه می تواند ارزیابی این الگوریتم ها در آزمایش ها به صورت واقعی یا Real world باشد که بسیار هزینه بر است.

▪ Mobility

اگر یک Device بخواهد بین چندین Fog domain حرکت کند، ادامه دار بودن سرویس ارائه شده به آن می بایستی تضمین شود. در این حالت سمت و سوی تحقیق می تواند بدست آوردن مدل های تحرک پذیری به صورت Realistic باشد.

▪ Federation

اجماع بین اپراتورها یا Provider ها، هنوز به عنوان یک چالش مطرح است. برای مثال در الگوریتم های Resource sharing، تنها یک اپراتور در نظر گرفته می شود و اشتراک منابع بین این اپراتورها هنوز در نظر گرفته نشده اند. هم چنین اگر یک سیستم اجماع شده یا Federated systems وجود داشته باشد، اینکه الگوریتم های مدیریت منابع چگونه درون این گونه سیستم ها می توانند انجام شوند، هنوز یک چالش است. از جمله چالش های دیگر در این حوزه این است که یک اپراتور چگونه برای وام دادن منابع خود

قیمت گذاری می کند. هم چنین یک اپراتور چه موقع تصمیم به اشتراک گذاشتن منابع خود می کند و چه موقع تصمیم به استفاده از منابع دیگر اپراتور ها در داخل یک Federation می کند.

۸-۲- تعاریف Cloud gaming

در مقاله [۱۹]، نخستین تعریف آکادمیک از Cloud gaming مطرح شد. در این مقاله بحث Gaming را به عنوان یک کاربرد کشنده (به تعبیر نویسنده)، برای رایانش ابری تعریف کردند و بحث تحویل خدمات Gaming که توسط کمپانی AMD مطرح شده بود که می توانست بازی را رندر کند و سپس آن را فشرده کند و در مرحله ی بعد آن را توسط اینترنت به کاربر انتقال دهد، را بررسی کردند. توسط این طرح که توسط AMD مطرح شد، کاربران می توانستند پردازش های لازم برای بازی را به سرور های ابری منتقل کنند و این کار سبب شد تا پردازش های سنگین بازی دیگر به کامپیوتر های کاربران تحمیل نشود. تعریف دیگر از بحث Cloud gaming را می توان در مقاله [۲۰]، پیدا کرد. در این مقاله مورد Cloud gaming system را یک معماری نوین در کامپیوتر بیان کرده اند که توسط این معماری می توان از منابع فضای ابر برای ارتقاء کارایی بازی استفاده کرد. تا اینجا تعاریف پایه ی Cloud gaming مطرح شد. اما در سال 2010 میلادی و یک سال بعد از اولین تعریف آکادمیک Cloud gaming در مقاله [۱۹]، Onlive به عنوان یک سرویس در حوزه ی Cloud gaming معرفی شد. بعد از معرفی این سرویس بود که طراحی یک Business model برای Cloud gaming مطرح شد. طبق تحقیقات به عمل آمده در مقاله [۲۱]، شرکت های بازی سازی کوچک و متوسط به دلیل نبود یک Business model جامع در حوزه ی Cloud gaming، برای استفاده از سرویسها ی ابری از خود تردید نشان می دادند. در سال 2011 شرکت G-cluster در مقاله [۲۲] Business model خود را در حوزه ی Cloud gaming معرفی کرد و طی تحقیقات به عمل آمده طی مرور زمان با ساده تر کردن این Business model، هزینه ی تمام شده برای کاربر کمتر می شد.

۹-۲- پلتفرم های بازی ابری

در این بخش سه مورد با نام های System integration و QoS evaluation و QoE evaluation در حوزه ی پلتفرم های Cloud gaming، بررسی خواهند شد. اما قبل از آن به تعریف انواع پلتفرم های Cloud gaming پرداخته خواهد شد. در کل دو نوع پلتفرم برای Cloud gaming موجود است:

پلتفرم های شفاف: در این نوع پلتفرم ها، بازی ها بدون اصلاحات، برای بازی به اجرا گذاشته می شوند. مزیت آن است که در این نوع، به اصطلاح Deploy کردن بازی راحت تر بوده اما به دلیل انجام عدم اصلاحات در بازی، مثلاً اضافه کردن کد یا کامپایل دوباره بازی به صورت بهینه تر، کارایی در اجرای آن به طور قطع پایین می آید. اولین پلتفرم به صورت Open source و شفاف در مقاله ی [۲۳]، با نام Gaming anywhere مطرح شد.

پلتفرم های غیر شفاف: در این نوع، بازی ها را با اضافه کردن کد یا کامپایل دوباره به صورت بهینه تر، اصلاح می کنند که در این حالت ممکن است بازی به صورت بهینه تر اجرا شود اما بسیار زمان بر و احتمالاً دارای خطاهای زیاد خواهد بود.

تا این جا انواع پلتفرم ها معرفی شد. اما در ادامه به موارد System integration و QoS و QoE در حوزه ی پلتفرم های بیان شده پرداخته خواهد شد.

▪ System integration

این مورد بیان گر آن است که می توان با یکپارچه سازی یک سری ویژگی جدید به پلتفرم های Cloud gaming، تجربه ی بازی در این پلتفرم ها را بهبود بخشید. اما به دلیل طبیعت ناهمگن و توزیع شده ی فضای ابر، این کار مشکل خواهد بود. به صورت کلی یک رابطه ی مستقیم بین بهینه بودن و پیچیدگی وجود دارد. به این صورت که هر چه از پیچیدگی کاسته شود، فضای کمتری برای بهینه سازی ایجاد می شود که بیان گر پلتفرم های شفاف است. عکس این موضوع نیز صادق است. به این صورت که با در نظر گرفتن پیچیدگی بیشتر برای سیستم، فضا برای بهینه سازی بیشتر می شود که بیان گر پلتفرم غیر شفاف می باشد.

▪ QoS evaluation

وجود این ویژگی برای اندازه گیری کارایی ضروری است. چنان چه این ویژگی به صورت Real time در سیستم اعمال شود، آنگاه به صورت پویا می توان پلتفرم های بیان شده در بالا را بهتر کرد. به صورت کلی مقالاتی که در حوزه ی Cloud gaming QoS مطرح می شوند در دو مورد می گنجند. مورد اول بحث مصرف انرژی می باشد. در حوزه ی Cloud gaming، وقتی یک بازی در سرور های فضای ابر، Render یا پرداخت می شود، تصاویر پرداخت شده در سمت سرور به یک فرمت ویدئویی مثلاً MP4، به

اصطلاح Encode می شود. حال این تصویر که به گیرنده می رسد، باید Decode شود تا نمایش داده شود. بحث مصرف انرژی، در decoder ها مطرح می شود. به این صورت که هرچه Bitrate و Resolution تصویر بالاتر باشد، هزینه و مصرف انرژی Decoding نیز بیشتر می شود. این موضوع در مقاله [۲۴]، بررسی شده است. اما مورد دوم بحث Network metric می باشد. از آن جا که در بازی هایی که در فضای ابر پرداخته می شوند، کیفیت تجربه ی کاربر از بازی به کیفیت شبکه مربوط است، پس در نظر گرفتن این موضوع در حوزه ی Cloud gaming امری اجتناب ناپذیر است. در مقاله [۲۵]، آزمایشاتی برای ارزیابی تأخیر حس شده توسط کار در اجرای بازی چه به صورت Local و چه به صورت Online، انجام شده است. در این آزمایش از یک ابزار برای اندازه گیری با نام GALAMETO.KOM استفاده شد و بیان گر این موضوع بود که تأخیری که اجرای بازی در فضای ابر ایجاد می کرد نسبت به تأخیر اجرای بازی به صورت محلی، بین ۸۵٪ تا ۸۰۰٪، بیشتر بود. آزمایش آن ها هم چنین بیان گر این موضوع بود که مکان جغرافیایی قرار گرفتن Cloud data center در زمان رفت و برگشت از کاربر تا Data center، تأثیر به سزایی دارد.

▪ QoE evaluation

QoE در حوزه ی Cloud gaming به دو مورد با نام های General cloud gaming QoE و evaluation Mobile cloud gaming QoE evaluation، تقسیم می شود. در مقاله [۲۶]، از یک Measure با نام Mean Opinion Score یا MOS، به عنوان یک QoE metric استفاده شده است. نتایج بدست آمده از MOS، بیان گر این موضوع است که مقادیر MOS به پارامتر های QoS یعنی تأخیر شبکه و میزان گم شدگی بسته ها و game context یعنی ژانر بازی یا میزان مهارت کاربر، وابستگی دارد. در مقاله [۲۷]، علاوه بر مقادیر بیان شده برای MOS که در مقاله [۲۶]، مطرح شد، مقادیر دیگری همچون Input sensitivity و Video quality و Audio quality و Overall quality و Complexity و Pleasantness و Perceived value، مطرح شده است. در مقاله [۲۸]، آزمایشاتی در حوزه ی Cloud gaming، هم برای کاربران دستگاه های موبایل و هم برای کاربران کامپیوتر های دسکتاپ صورت گرفته است. نتایج بدست آمده بیانگر این موضوع بود که کاربران دستگاه های موبایل، بیشتر از گرافیک بازی راضی شده اند ولی کاربران کامپیوتر های دسکتاپ، بیشتر به کیفیت کنترل کردن بازی اهمیت داده اند. نتایج دیگر بدست آمده بیانگر این موضوع است که گرافیک بازی بیشتر تحت تأثیر مواردی هم چون Bitrate و

Framerate و تاخیر شبکه قرار می گیرد اما کیفیت کنترل بازی تحت تاثیر نوع کاربر یعنی موبایلی یا دسکتاپ، قرار می گیرد.

۱۰-۲- بهینه سازی پلتفرم های ابری

در این بخش، بحث بهینه سازی در پلتفرم های Cloud gaming، و در حوزه های زیر ساختی و ارتباطات مطرح می شود. در حوزه ی زیر ساخت، دو مورد برای بهینه سازی با نام های Resource allocation و Distributed architectures، مطرح می شوند که در ذیل به آن ها پرداخته خواهد شد.

۱-۱۰-۲- زیر ساختی : Resource allocation

در مورد تخصیص منابع، بهینه سازی به صورت دو مورد تخصیص منابع به صورت هوشمند بین سرور ها و طراحی نو آورانه ی ساختار های توزیع شده، تعریف می شود. در مقاله [۲۹]، با استفاده از روش حریصانه و اکتشافی، اقدام به افزایش بهره وری در تامین منابع و در بازی های MMOG یا Massively Multiplayer Online Games، به صورت تخصیص حداقلی گره های پردازشی به MMOG، کرده اند که دی این صورت گره های دیگر برای کار های آتی آزاد می مانند. هم چنین در مقاله [۳۰]، یک آنالیز روی هم GPU یا Graphical Processing Unit و هم CPU، روی سرور هایی که قرار است بازی ها قرار بگیرند، انجام شده است. به این صورت که قسمتی از بار پردازشی را به سرور های ابری ارسال می کردند و بقیه را به صورت محلی و در کامپیوتر کاربر، اعمال می کردند. نتیجه آن شد که اعمال برخی رایانش پردازشی سمت کاربر ممکن است موجب افزایش QoS شود.

۲-۱۰-۲- Distributed architectures : زیر ساختی

در مقاله [۳۱]، به دلیل آنکه در بازی های MMOG، ممکن است تاخیر موجب کاهش QoS در سمت کاربر شود، روشی ارائه شد که بر مبنای معماری P2P بود که می توانست با این تاخیر مقابله کند. هم چنین در مقاله [۳۲]، این موضوع مورد مطالعه قرار گرفته است که به دلیل مکان جغرافیایی قرار گیری

Cloud data center ها، RTT یا Round Trip Time برای کاربرد هایی حساس به تاخیر از جمله بازی های آنلاین، زیاد می شود. برای حل این مشکل در این مقاله با قرار دادن Edge server در نزدیکی کاربر، اقدام به حل این مشکل کرده اند. این موضوع در مقاله [۳۳]، نیز مورد مطالعه قرار گرفته است. در این مقاله یک معماری توزیع شده در حوزه ی Cloud gaming مطرح شده است که این اجازه را می دهد که سرور ها در نزدیکی کاربر قرار بگیرند. نتایج نشان داد که اگر به اندازه ی کافی این گونه سرور ها مثلا در ISP تامین شود، با افزایش کیفیت بهتر، کاربران بیشتری جذب می شوند.

۲-۱۰-۳- ارتباطات

و اما در حوزه ی ارتباطات، در دو مورد با نام های الگوریتم های فشرده سازی داده برای کاهش ترافیک شبکه و الگوریتم های انتقال تطبیقی برای تطبیق یافتن با حالات پویای شبکه، مطالعاتی که صورت گرفته است. در مورد اول، یعنی فشرده سازی داده، بعد از اینکه بازی در سرور های ابری اجرا شد، می بایستی ضبط شود و سپس عمل فشرده سازی رو آن صورت بگیرد تا بتواند از طریق شبکه برای کاربر ارسال شود. حال آنکه عمل فشرده سازی در سه مورد با نام های Video compression و Graphics compression و Hybrid compression، مورد مطالعه قرار گرفته است.

۲-۱۰-۴- فشرده سازی ویدئو

در مقاله [۳۴]، به این موضوع پرداخته شده است که اگر چند بازیکن در یک Session از بازی نزدیک یک دیگر باشند، احتمال اینکه یک محیط از بازی را ببینند بیشتر می شود. پس برای این هم پوشانی بین دید بازیکن ها، در این مقاله پیش بینی فریم های ویدئو بین بازیکنان، برای افزایش بهره بری در نظر گرفته شده است.

در مقاله [۳۵]، یک Cloud gaming platform، بر مبنای Graphics compression، بیان شده است. پلتفرم بیان شده دارای سه ابزار با نام های Intra frame و Inter frame compression و Caching می باشد. در نهایت این ابزار به ساختار های سه بعدی و بافت های دو بعدی اعمال می شود. مورد Graphics compression برای دستگاه های موبایلی و هم چنین کامپیوتر های شخصی و به دلیل اینکه از لحاظ توان پردازشی، پایین و یکسان نمی باشند، بسیار مهم محسوب می شوند.

در مقاله [۳۶]، هم مورد فشرده سازی ویدئو و هم مورد فشرده سازی گرافیک، با هم در نظر گرفته شده اند. در این مقاله، فشرده سازی گرافیکی روی ساختار های سه بعدی و هم چنین بافت های دو بعدی ساده شده، صورت می گیرد و سپس به کامپیوتر کاربر ارسال می شود. این صحنه های ساده شده سپس روی کامپیوتر های کاربر، رندر گرفته می شود که به آن Base layer گفته می شود. سپس ویدئو با کیفیت کامل و هم چنین Base layer video، در سرور های ابری رندر گرفته می شود و هرآنچه که از ویدئو باقی می ماند، فشرده می شود و سپس به کاربر ارسال می شود که به آن Enhancement layer گفته می شود. حال که Base layer، گرافیک فشرده شده است و Enhancement layer، ویدئو فشرده شده است، به آن Hybrid گفته می شود.

در مورد دوم، یعنی انتقال تطبیقی، برای شرایطی در شبکه استفاده می شود که عوامل ناپایدار کننده و غیر قابل پیش بینی شبکه هم چون Jitter و RTT و Bandwidth، بر QoE تاثیر منفی می گذارند. در مقاله [۳۷]، روشی بررسی شده است تا با آن انتقال ویدئو بازی با پهنای باند موجود، تطبیق داشته باشد. برای این کار یک ماژول به نام Video adaption را در سیستم تعبیه می کنند که می تواند وضعیت شبکه را به صورت Real time از طریق مشاهده ی شبکه برآورد کند و هم چنین پارامتر های کدینگ ویدئو مثل Frame rate، را نیز تنظیم کند تا در نهایت به یک Bit rate تطبیق یافته با واحد مگابیت بر ثانیه، دست یابند. در مقاله [۳۸]، زمان پاسخ گویی سیستم Cloud gaming، به سه زیر جزء با نام های Server delay و

بهینه سازی با نام Rate selection algorithm، توانستند به یک راه حل پویا برای مشخص کردن زمان و راه تعویض Bitrate با توجه به تاخیر شبکه برسند. در مقاله [۳۹]، بحث تطبیق رندرینگ نیز در نظر گرفته شده است. در ابتدا به شناسایی پارامترهای رندرینگ با نام های View distance و Enabling grass و Texture filtering و غیره پرداختند. سپس به آنالیز ویژگی های ارتباطاتی و هزینه های رایانشی این پارامتر ها پرداختند. در نهایت منجر به ارائه ی یک طرح از تطبیق رندرینگ شد که شامل تنظیمات تطبیق رندرینگ بهینه بود.

۱۱-۲- سرویس های بازی ابری تجاری

در این بخش به تاریخ سرویس های Cloud gaming، پرداخته می شود. در اوایل سال 2000 میلادی بود که یک شرکت با نام [۴۰] G-cluster، سرویسی را برای Cloud gaming ارائه کرد. این شرکت نشان داد که می توان توسط فناوری Wi-Fi، بازی را روی دستگاه های PDA، پخش کرد. سال ها بعد کمپانی های ارائه دهنده ی سرویس های ابری مثل OnLive [۴۱]، اقدام به ارائه ی سرویس های Cloud gaming، به صورت OTT یا Over-The-Top، کردند. OTT به معنای تحویل محتوای چند رسانه ای بر بستر اینترنت و بالای حوزه ی اپراتورهای شبکه می باشد که می تواند QoS را برای بازی های تحت محیط ابری و به صورت همه جایی ارائه دهد. کمپانی OnLive در سال 2009 میلادی خود را معرفی کرد و توسط سرمایه گذارانی هم چون Warner Bros و AT&T و Ubisoft، توانست معروف شود. اما در سال 2012 میلادی و به دلیل مشکلات مالی، فعالیت های خود را متوقف کرد و patent های خود را به Sony [۴۲] فروخت. یک کمپانی دیگر به نام Gaikai [۴۳]، آمد و یک model Business جدیدی را ارائه داد. به این گونه که بدون اینکه کاربران نرم افزاری را روی سیستم خود نصب کنند می توانستند اقدام به بازی کنند. در پایان بازی این اختیار به کاربر داده می شد که اگر از بازی لذت برد، سپس می تواند آن را خریداری کند. این کمپانی نیز توسط Sony [۴۴] و در سال 2012 میلادی خریداری شد و Sony بر مبنای آن سرویس Cloud gaming خود با نام PS Now [۴۵] را در سال 2014 میلادی ارائه کرد. سرویس های Cloud gaming بیان شده به دو دسته تقسیم می شوند. در دسته ی اول سرویس های Cloud gaming بر پایه ی شبکه اصلی یعنی Under laying network عرضه می شود و در دسته ی دوم، به عنوان سرویس های OTT ارائه می

شوند. دسته اول باعث ارائه ی QoS بهتر می شود ولی دسته ی دوم همراه با احتمال تجربه ناپایدار و یا بد کاربر است اما هزینه های Cloud را به همراه ندارد.

منابع

- [1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13-16.
- [2] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Personal communications*, vol. 8, pp. 10-17, 2001.
- [3] M. Satyanarayanan, V. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, 2009.
- [4] .(2017)*To Build an Open Reference Implementation of CORD®*. Available: <https://opencord.org/about/>
- [5] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, pp. 416-464, 2017.
- [6] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, 2013, pp. 15-20.
- [7] N. K. Giang, M. Blackstock, R. Lea, and V. C. Leung, "Developing iot applications in the fog: A distributed dataflow approach," in *2015 5th International Conference on the Internet of Things (IOT)*, 2015, pp. 155-162.
- [8] S. Agarwal, S. Yadav, and A. K. Yadav, "An efficient architecture and algorithm for resource provisioning in fog computing," *International Journal of Information Engineering and Electronic Business*, vol. 8, p. 48, 2016.
- [9] M. Slabicki and K. Grochla, "Performance evaluation of CoAP, SNMP and NETCONF protocols in fog computing architecture," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 1315-1319.
- [10] Y. N. Krishnan, C. N. Bhagwat, and A. P. Utpat, "Fog computing—Network based cloud computing," in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, 2015, pp. 250-251.

- [11] J. Oueis, E. C. Strinati, S. Sardellitti, and S. Barbarossa, "Small cell clustering for efficient distributed fog computing: A multi-user case," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, 2015, pp. 1-5.
- [12] M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, "Help your mobile applications with fog computing," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking-Workshops (SECON Workshops)*, 2015, pp. 1-6.
- [13] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coding for distributed fog computing," *IEEE Communications Magazine*, vol. 55, pp. 34-40, 2017.
- [14] Y.-J. Ku, D.-Y. Lin, C.-F. Lee, P.-J. Hsieh, H.-Y. Wei, C.-T. Chou, *et al.*, "5G radio access network design with the fog paradigm: Confluence of communications and computing," *IEEE Communications Magazine*, vol. 55, pp. 46-52, 2017.
- [15] S.-C. Hung, H. Hsu, S.-Y. Lien, and K.-C. Chen, "Architecture harmonization between cloud radio access networks and fog networks," *IEEE Access*, vol. 3, pp. 3019-3034, 2015.
- [16] Q. He, C. Zhang, X. Ma, and J. Liu, "Fog-based transcoding for crowdsourced video livecast," *IEEE Communications Magazine*, vol. 55, pp. 28-33, 2017.
- [17] M. Tang, L. Gao, H. Pang, J. Huang, and L. Sun, "Optimizations and economics of crowdsourced mobile streaming," *IEEE Communications Magazine*, vol. 55, pp. 21-27, 2017.
- [18] C. Peltz, "Web services orchestration and choreography," *Computer*, pp. 46-52, 2003.
- [19] P. E. Ross, "Cloud computing's killer app: Gaming," *IEEE Spectrum*, vol. 46, pp. 14-14, 2009.
- [20] D. Mishra, M. El Zarki, A. Erbad, C.-H. Hsu, and N. Venkatasubramanian, "Clouds+ games: A multifaceted approach," *IEEE Internet Computing*, vol. 18, pp. 20-27, 2014.
- [21] L. Riungu-Kalliosaari, J. Kasurinen, and K. Smolander, "Cloud services and cloud gaming in game development," in *Proc. the IADIS Game and Entertainment Technologies*, vol. 22, p. 2013, 2013.
- [22] A. Ojala and P. Tyrvaenen, "Developing cloud business models: A case study on cloud gaming," *IEEE software*, vol. 28, pp. 42-47, 2011.
- [23] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "GamingAnywhere: an open cloud gaming system," in *Proceedings of the 4th ACM multimedia systems conference*, 2013, pp. 36-47.

- [24] C.-Y. Huang, P.-H. Chen, Y.-L. Huang, K.-T. Chen, and C.-H. Hsu, "Measuring the client performance and energy consumption in mobile cloud gaming," in *2014 13th Annual Workshop on Network and Systems Support for Games*, 2014, pp. 1-3.
- [25] U. Lampe, Q. Wu, S. Dargutev, R. Hans, A. Miede, and R. Steinmetz, "Assessing latency in cloud gaming," in *International Conference on Cloud Computing and Services Science*, 2013, pp. 52-68.
- [26] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "Gaming in the clouds: QoE and the users' perspective," *Mathematical and Computer Modelling*, vol. 57, pp. 2883-2894, 2013.
- [27] S. Möller, D. Pommer, J. Beyer, and J. Rake-Revelant, "Factors influencing gaming QoE: Lessons learned from the evaluation of cloud gaming services," in *Proceedings of the 4th International Workshop on Perceptual Quality of Systems (PQS 2013)*, 2013, pp. 1-5.
- [28] C.-Y. Huang, C.-H. Hsu, D.-Y. Chen, and K.-T. Chen, "Quantifying user satisfaction in mobile cloud games," in *Proceedings of Workshop on Mobile Video Delivery*, 2014, p. 4.
- [29] M. Marzolla, S. Ferretti, and G. D'angelo, "Dynamic resource provisioning for cloud-based gaming infrastructures," *Computers in Entertainment (CIE)*, vol. 10, p. 4, 2012.
- [30] Z. Zhao, K. Hwang, and J. Villeta, "Game cloud design with virtualized CPU/GPU servers and initial performance results," in *Proceedings of the 3rd workshop on Scientific Cloud Computing*, 2012, pp. 23-30.
- [31] R. Süselbeck, G. Schiele, and C. Becker, "Peer-to-peer support for low-latency massively multiplayer online games in the cloud," in *2009 8th Annual Workshop on Network and Systems Support for Games (NetGames)*, 2009, pp. 1-2.
- [32] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "The brewing storm in cloud gaming: A measurement study on cloud to end-user latency," in *Proceedings of the 11th annual workshop on network and systems support for games*, 2012, p. 2.
- [33] T. Kämäräinen, M. Siekkinen, Y. Xiao, and A. Ylä-Jääski, "Towards pervasive and mobile gaming with distributed cloud infrastructure," in *2014 13th Annual Workshop on Network and Systems Support for Games*, 2014, pp. 1-6.
- [34] W. Cai, V. C. Leung, and L. Hu, "A cloudlet-assisted multiplayer cloud gaming system," *Mobile Networks and Applications*, vol. 19, pp. 144-152, 2014.
- [35] X. Liao, L. Lin, G. Tan, H. Jin, X. Yang, W. Zhang, *et al.*, "Liverender: A cloud gaming system based on compressed graphics streaming," *IEEE/ACM Transactions on Networking (TON)*, vol. 24, pp. 2128-2139, 2016.

- [36] S.-P. Chuah and N.-M. Cheung, "Layered coding for mobile cloud gaming," in *Proceedings of International Workshop on Massively Multiuser Virtual Environments*, 2014, pp. 1-6.
- [37] S. Jarvinen, J.-P. Laulajainen, T. Sutinen, and S. Sallinen, "Qos-aware real-time video encoding how to improve the user experience of a gaming-on-demand service," in *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.*, 2006, pp. 994-997.
- [38] S. Wang and S. Dey, "Addressing response time and video quality in remote server based internet mobile gaming," in *2010 IEEE Wireless Communication and Networking Conference*, 2010, pp. 1-6.
- [39] S. Wang and S. Dey, "Rendering adaptation to address communication and computation constraints in cloud mobile gaming," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, 2010, pp. 1-6.
- [40] (2015)G-cluster. Available: <http://www.gcluster.com/eng>
- [41] .(2015)OnLive. Available: <http://www.onlive.com/>
- [42] N. STATT. (2015). *Sony buys OnLive streaming game service, which will shut down later this month*. Available: <https://www.cnet.com/news/sony-buys-onlive-streaming-game-service-which-will-shut-down-later-this-month/>
- [43] *Gaikai*. Available: <http://www.gaikai.com/>
- [44] .(2012)*Cloud Gaming Adoption is Accelerating...and Fast!* Available: <http://www.nttcom.tv/2012/07/09/>
- [45] .(2015)*Playstation Now*. Available: <http://www.playstation.com/en-us/explore/playstationnow/>