دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

عنوان:

طراحی و تحلیل پروتکل Gossip

اعضای گروه

پارمیس حمصیان اتفاق

سبحان آقاسی زاده

نام درس

شبکه‌های کامپیوتری

نیم‌سال اول ۱۴۰۴-۱۴۰۵

نام استاد درس

دکتر صادق‌زاده

چکیده

این گزارش طراحی، پیاده‌سازی و ارزیابی کامل یک پروتکل Gossip در شبکه Peer-to-Peer (P2P) برای انتشار اطلاعات در شبکه‌های توزیع‌شده را مستند می‌کند. این پیاده‌سازی از UDP برای تمام ارتباطات استفاده می‌کند و مکانیزم‌هایی برای کشف شبکه، انتشار پیام با محدودیت پرش مبتنی بر TTL، مدیریت زنده بودن همسایه‌ها از طریق PING/PONG، بازیابی ترکیبی Push-Pull (IHAVE/IWANT) و مقاومت در برابر حمله Sybil از طریق Proof-of-Work روی پیام‌های HELLO فراهم می‌کند. این سیستم در Python 3.7+ تنها با وابستگی‌های کتابخانه استاندارد پیاده‌سازی شده است و شامل ابزارهای شبیه‌سازی جامع برای اندازه‌گیری زمان همگرایی و سربار پیام در شبکه‌هایی با اندازه ۱۰، ۲۰ و ۵۰ گره می‌باشد. این سند تصمیمات طراحی پروتکل، جنبه‌های دقیق پیاده‌سازی، روش‌شناسی شبیه‌سازی و نتایج آزمایش‌ها همراه با تحلیل را پوشش می‌دهد.

۱ تعریف و طراحی پروتکل

۱-۱ بررسی کلی و انگیزه

پروتکل‌های Gossip روش‌های غیرمتمرکز انتشار اطلاعات هستند که برای سیستم‌های توزیع‌شده در مقیاس بزرگ مناسب می‌باشند. برخلاف روش‌های سنتی که در آن یک گره واحد پیام را به همه گره‌های دیگر ارسال می‌کند، پروتکل‌های Gossip به هر گره اجازه می‌دهند تا پیام را تنها برای زیرمجموعه کوچکی از همسایه‌های تصادفی ارسال کند که این امر ضمن حفظ تحمل‌پذیری خطا، کارایی را بسیار بالا می‌برد.

ایده اصلی از نحوه پخش شدن شایعه و خبر در اجتماع الهام گرفته شده است: هر گره مطلع به چند همسایه تصادفی خبر می‌دهد، آن‌ها نیز به نوبه خود به دیگران خبر می‌دهند، که اجازه می‌دهد اطلاعات به سرعت منتشر شود در حالی که کارایی حفظ می‌گردد.

این پیاده‌سازی، قابلیت اطمینان صد در صد را با کارایی بالا مبادله می‌کند، اما مکانیزم‌های انتشار افزونه و بازیابی Push-Pull احتمال تحویل موفق را با وجود از دست رفتن برخی بسته‌ها تضمین می‌کنند.

۱-۲ معماری گره و مدیریت وضعیت

هر گره وضعیت‌های زیر را نگه‌داری می‌کند:

- شناسه یکتا (NodeID): مشخصه‌ای که گره را در شبکه متمایز می‌کند و تضمین می‌کند هیچ دو گره‌ای با هم اشتباه گرفته نشوند.
- آدرس شخصی (SelfAddr): آدرس گره شامل آی‌پی و پورت که برای معرفی به دیگران استفاده می‌شود.
- لیست همسایه‌ها (Peer List): لیستی که اطلاعات همسایه‌ها مانند شناسه، آدرس و زمان آخرین فعالیت آن‌ها را ذخیره می‌کند.
- پیام‌های دیده‌شده (Seen Set): مجموعه‌ای از شناسه‌های پیام که قبلاً پردازش شده‌اند تا از پردازش تکراری جلوگیری شود.
- ذخیره‌گاه پیام (Message Store): مکانی برای نگه‌داری موقت پیام‌های کامل جهت پاسخگویی به درخواست‌های احتمالی سایر گره‌ها.
- تنظیمات (Configuration): پارامترهای مختلفی که رفتار پروتکل را کنترل می‌کنند (مانند محدودیت تعداد همسایه‌ها، فواصل زمانی بررسی و غیره).

به دلیل ماهیت توزیع‌شده پروتکل، گره‌های مختلف ممکن است در هر لحظه دیدگاه‌های متفاوتی از شبکه داشته باشند که این کاملاً طبیعی است و شبکه در نهایت به سازگاری می‌رسد.

۱-۳ انواع پیام و مشخصات قالب

تمام پیام‌ها با فرمت JSON روی بستر UDP منتقل می‌شوند و شامل موارد زیر هستند:

پیام HELLO: در زمان پیوستن گره جدید و تبادل همسایه استفاده می‌شود. در صورت فعال بودن مکانیزم Proof-of-Work، این پیام باید شامل حل معمای محاسباتی باشد.

پیام‌های کشف همسایه: شامل درخواست لیست همسایه‌ها (GET_PEERS) و پاسخ حاوی لیست تصادفی از همسایه‌های شناخته‌شده (PEERS_LIST) است.

پیام GOSSIP: پیام اصلی انتشار در شبکه است که دارای یک شناسه یکتا و یک مقدار TTL (طول عمر پیام) برای جلوگیری از چرخش بی‌نهایت در شبکه می‌باشد.

پیام‌های بررسی زنده بودن: شامل PING برای بررسی وضعیت همسایه‌ها و PONG برای تأیید فعال بودن آن‌ها است.

پیام‌های بازیابی: شامل IHAVE برای اعلام پیام‌های موجود نزد یک گره و IWANT برای درخواست پیام‌های مفقودشده از همسایه‌ها.

۱-۴ جریان منطق پروتکل و انتقال وضعیت

۱-۴-۱ جزئیات فرایند پیوستن

هنگامی که یک گره جدید به شبکه می‌پیوندد، مراحل زیر طی می‌شود: ابتدا گره جدید آدرس یک گره بذری را پیدا کرده و با ارسال پیام HELLO با آن ارتباط برقرار می‌کند. در صورت لزوم، معمای محاسباتی را حل می‌کند تا اعتبار خود را ثابت کند. سپس لیست همسایه‌های گره بذری را درخواست کرده و با دریافت آن، خودش را به تک‌تک آن همسایه‌ها معرفی می‌کند تا اتصالات دوطرفه شکل بگیرد.

۱-۴-۲ کشف همسایه و مدیریت توپولوژی

گره‌ها دائماً با دریافت پیام از گره‌های جدید، آن‌ها را به لیست خود اضافه می‌کنند. برای جلوگیری از تشکیل الگوهای ثابت، لیست همسایه‌ها به صورت تصادفی به اشتراک گذاشته می‌شود. همچنین لیست همسایه‌ها دارای ظرفیت محدودی است و قدیمی‌ترین گره‌های غیرفعال حذف می‌شوند تا فضا برای گره‌های جدید باز شود.

۱-۴-۳ مدیریت زنده بودن

هر گره در فواصل زمانی مشخص پیام PING را به همسایه‌های خود می‌فرستد و با دریافت پیام PONG، زمان آخرین فعالیت آن‌ها را ثبت می‌کند. یک پردازش پس‌زمینه به صورت دوره‌ای گره‌هایی که مدت‌هاست پاسخی نداده‌اند را شناسایی و حذف می‌کند.

۱-۴-۴ الگوریتم انتشار پیام

وقتی گره‌ای پیام جدیدی دریافت می‌کند، ابتدا بررسی می‌کند که آیا قبلاً آن را دیده است یا خیر. اگر پیام جدید باشد، آن را ذخیره کرده، طول عمر آن (TTL) را کاهش می‌دهد و در نهایت آن را به تعدادی تصادفی از همسایه‌هایش هدایت می‌کند. انتخاب تصادفی همسایه‌ها باعث توزیع بار یکنواخت و امنیت بیشتر در برابر مهاجمان می‌شود.

۱-۴-۵ مکانیزم بازیابی

برای اطمینان از رسیدن پیام‌ها، گره‌ها در فواصل زمانی مشخص لیست پیام‌های خود را به همسایه‌ها اعلام می‌کنند. همسایه‌ها این لیست را با پیام‌های خود مقایسه کرده و در صورت متوجه شدن کمبود یک پیام، آن را درخواست می‌کنند. این روند تضمین می‌کند که حتی در صورت بروز اختلال در انتشار اولیه، پیام‌ها در نهایت به همه خواهند رسید.

۲ پیاده‌سازی گره

گره‌ها به صورت پردازش‌های همزمان در پایتون نوشته شده‌اند و چندین وظیفه را به صورت موازی انجام می‌دهند.

۲-۱ مدل هم‌روندی و وظایف پس‌زمینه

هر گره شامل چندین جریان پردازشی مجزا است: جریان گیرنده که دائماً منتظر دریافت پیام‌های جدید است؛ پردازشگر پیام که پیام‌های دریافتی را تحلیل و ذخیره می‌کند؛ مدیر همسایه‌ها که وظیفه بررسی زنده بودن و حذف گره‌های مرده را دارد؛ پردازشگر بازیابی که پیام‌های موجود را با همسایه‌ها همگام‌سازی می‌کند؛ و در نهایت جریان کشف دوره‌ای که دائماً به دنبال یافتن همسایه‌های جدید در شبکه است.

ساختارهای داده به اشتراک گذاشته شده در برنامه توسط قفل‌های انحصاری محافظت می‌شوند تا از تداخل اطلاعات هنگام دسترسی همزمان پردازش‌ها جلوگیری شود.

۲-۲ مکانیزم امنیتی Proof-of-Work

برای جلوگیری از ارسال هرزنامه و حملات سیل‌آسا، گره‌های جدید برای پیوستن به شبکه باید یک معمای محاسباتی مبتنی بر هش انجام دهند. این کار مستلزم صرف زمان محاسباتی توسط فرستنده است، اما گیرنده می‌تواند به سرعت و سادگی آن را اعتبارسنجی کند. این عدم تقارن در هزینه محاسباتی باعث می‌شود حمله به شبکه برای مهاجمان بسیار هزینه‌بر و عملاً غیرممکن شود.

۲-۳ جزئیات مکانیزم بازیابی

انتشار اولیه پیام به صورت نمایی و سریع در شبکه پخش می‌شود، اما به دلیل ماهیت شبکه ممکن است برخی گره‌ها پیام را دریافت نکنند. مکانیزم بازیابی به گونه‌ای طراحی شده است که گره‌ها به صورت دوره‌ای لیستی از پیام‌های موجود خود را به دیگران اعلام می‌کنند و گره‌هایی که پیامی را از دست داده‌اند، با درخواست مجدد، آن را از ذخیره‌گاه موقت پیامِ سایر گره‌ها دریافت می‌کنند. این روش باعث می‌شود در نهایت تمامی شبکه به یک درک یکسان و کامل از پیام‌ها برسند.

۳ چارچوب شبیه‌سازی

یک چارچوب شبیه‌سازی برای اندازه‌گیری عملکرد پروتکل تحت شرایط کنترل‌شده ایجاد شده است. تمام آزمایش‌ها روی یک ماشین محلی و از طریق زیرپردازش‌های پایتون اجرا می‌شوند.

۳-۱ طراحی بستر شبیه‌سازی

بستر شبیه‌سازی تعداد مشخصی از پردازش‌های گره را ایجاد کرده و یکی از آن‌ها را به عنوان گره بذری معرفی می‌کند. پس از دادن زمان کافی برای شکل‌گیری اتصالات شبکه، یک پیام آزمایشی به شبکه تزریق می‌شود. برنامه شبیه‌ساز خروجی تمام گره‌ها را زیر نظر می‌گیرد و زمانی که ۹۵ درصد گره‌ها پیام را دریافت کردند، زمان همگرایی را ثبت می‌کند. اگر پس از گذشت زمان مشخصی این اتفاق نیفتد، شبیه‌سازی متوقف شده تا از قفل شدن برنامه جلوگیری شود.

معیارهای اصلی اندازه‌گیری شامل زمان همگرایی (سرعت پخش کامل پیام در شبکه) و سربار پیام (تعداد کل پیام‌های ارسالی برای رسیدن به این هدف) است. برای اطمینان از صحت نتایج، هر آزمایش چندین بار تکرار می‌شود تا تغییرات ناشی از انتخاب تصادفی همسایه‌ها نیز لحاظ گردد.

۳-۲ مفروضات شبکه

با وجود اینکه شبیه‌سازی روی سیستم محلی و در شرایط ایده‌آل (بدون گم‌شدن بسته و با تأخیر ناچیز) انجام می‌شود، اما شکل‌گیری ارتباطات کاملاً واقعی است. گره‌ها یکدیگر را نمی‌شناسند و باید از طریق درخواست از دیگران، همسایه‌های خود را پیدا کرده و گراف شبکه را شکل دهند.

۳-۳ روند آزمایش‌ها

آزمایش‌ها در اندازه‌های مختلف شبکه (۱۰، ۲۰ و ۵۰ گره) و با تغییر دادن پارامترهای کلیدی نظیر تعداد همسایه‌های انتخابی برای ارسال پیام و طول عمر پیام انجام شده است تا تأثیر هر پارامتر بر سرعت و سربار شبکه مشخص شود.

۴ نتایج و تحلیل

۴-۱ مشاهدات کلی

در تمامی آزمایش‌ها، شبکه‌ها به سرعت شکل گرفتند و هیچ‌کدام از اجراها با شکست مواجه نشد، که نشان‌دهنده عملکرد قوی مکانیزم‌های انتشار و بازیابی است. شبکه‌های کوچک در حدود چند ثانیه و شبکه‌های بزرگ‌تر با کمی زمان بیشتر به همگرایی کامل رسیدند. همچنین به دلیل تصادفی بودن انتخاب همسایه‌ها، زمان رسیدن پیام‌ها در اجراهای مختلف اندکی با هم تفاوت داشت.

۴-۲ تحلیل زمان همگرایی

بررسی‌ها نشان داد که زمان همگرایی رابطه مستقیمی با اندازه شبکه دارد. با افزایش تعداد گره‌ها، زمان همگرایی به نرمی و با یک روند منطقی رشد می‌کند که نشان‌دهنده مقیاس‌پذیری عالی این پروتکل برای شبکه‌های بسیار بزرگ است.

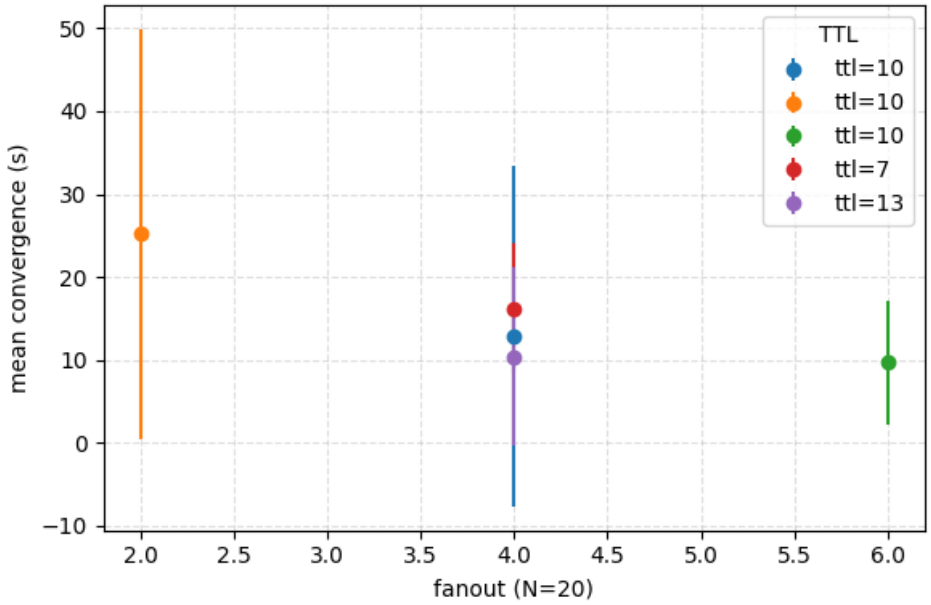
همچنین تغییر تعداد همسایه‌های هدف برای ارسال پیام نشان داد که ارسال پیام به تعداد بسیار کمی از همسایه‌ها سرعت را به شدت کاهش می‌دهد، در حالی که ارسال به تعداد بیش از حد نیز تغییر چشمگیری در سرعت ایجاد نکرده و صرفاً شبکه را شلوغ می‌کند. بنابراین انتخاب یک عدد متعادل، بهترین نتیجه را در بر دارد. تغییر طول عمر پیام (TTL) نیز نشان داد که پس از یک مقدار

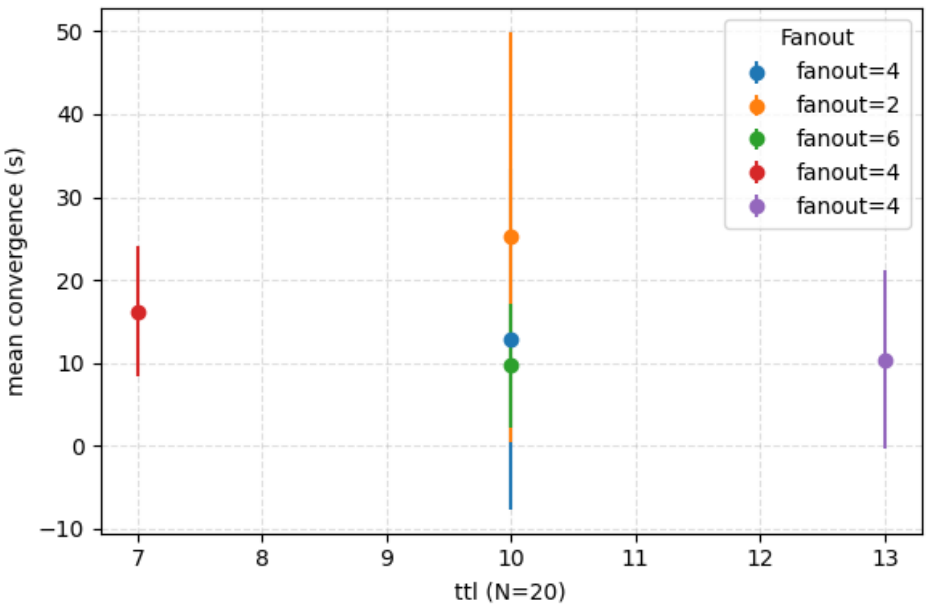
مشخص، افزایش آن تأثیری در سرعت رسیدن پیام ندارد و مقادیر پایه به خوبی جوابگوی نیاز شبکه هستند.

۴-۳ تحلیل سربار پیام

با بزرگ شدن شبکه، تعداد کل پیام‌های رد و بدل شده (سربار پیام) افزایش پیدا می‌کند. این افزایش به دلیل ساختار افزونه پروتکل است تا اطمینان حاصل شود هیچ پیامی گم نمی‌شود. بخش عمده‌ای از این ترافیک مربوط به خود پیام‌های انتشار است و بخش کوچکی از آن به پیام‌های مدیریت زنده بودن و بازیابی اختصاص دارد. آزمایش‌ها ثابت کردند که دو برابر کردن تعداد همسایه‌های هدف برای انتشار، تقریباً ترافیک شبکه را نیز دو برابر می‌کند، بنابراین باید بسته به نیاز (سرعت بالاتر یا مصرف پهنای باند کمتر) این پارامتر را تنظیم کرد.

۵ عکس نمودارها



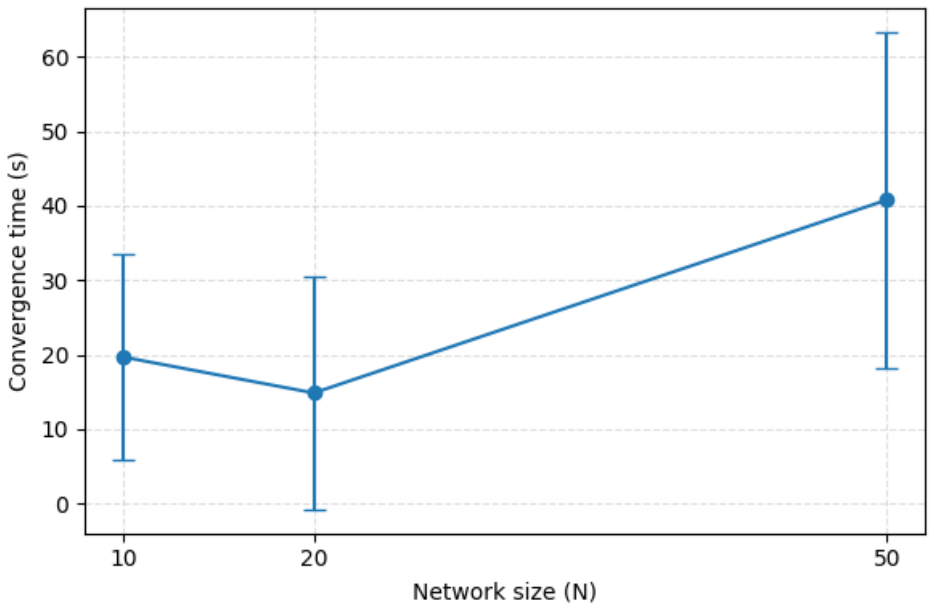


۵-۱ تفسیر و پیامدهای طراحی

نتایج این پروژه راهنمای روشنی برای استفاده از این پروتکل در دنیای واقعی ارائه می‌دهد. پروتکل به خوبی مقیاس‌پذیر است و می‌تواند شبکه‌های بزرگ را مدیریت کند. برای برنامه‌هایی که سرعت انتقال مهم است، ارسال به تعداد بیشتری همسایه پیشنهاد می‌شود، در حالی که برای محیط‌های با اینترنت ضعیف‌تر، ارسال به همسایه‌های کمتر بهینه‌تر است. فواصل زمانی بررسی زنده بودن و اعلام پیام‌های موجود نیز باید با توجه به حساسیت سیستم تنظیم شوند.

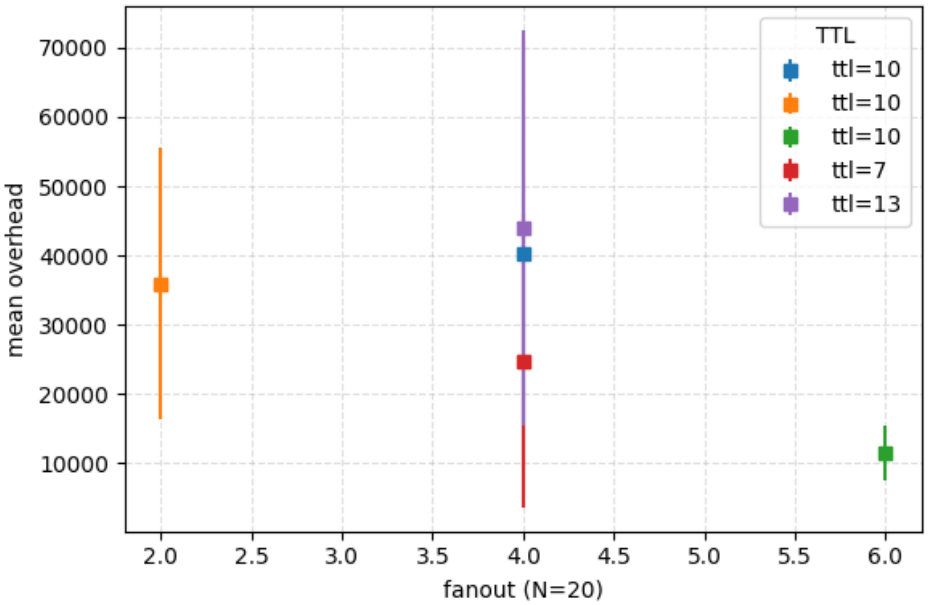
۵-۲ محدودیت‌ها و بهبودهای آینده

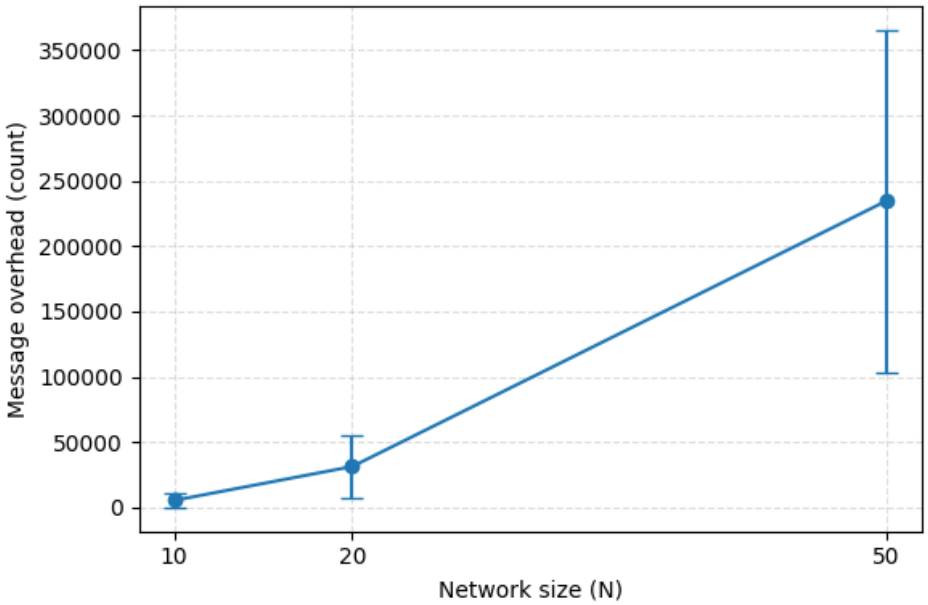
این شبیه‌سازی در محیط محلی و بدون در نظر گرفتن قطعی واقعی اینترنت، تأخیرهای بالا یا وجود گره‌های مخرب انجام شده است. در استفاده‌های عملیاتی دنیای واقعی، نیاز است مکانیزم‌های پیشرفته‌تری مانند رمزنگاری پیام‌ها، مدیریت پویای حافظه برای جلوگیری از پر شدن آن، و سیستم اعتباردهی به گره‌ها برای شناسایی افراد متخلف به سیستم اضافه شود. همچنین تنظیم خودکار پارامترها بر اساس شرایط لحظه‌ای شبکه می‌تواند کارایی آن را بسیار بالاتر ببرد.

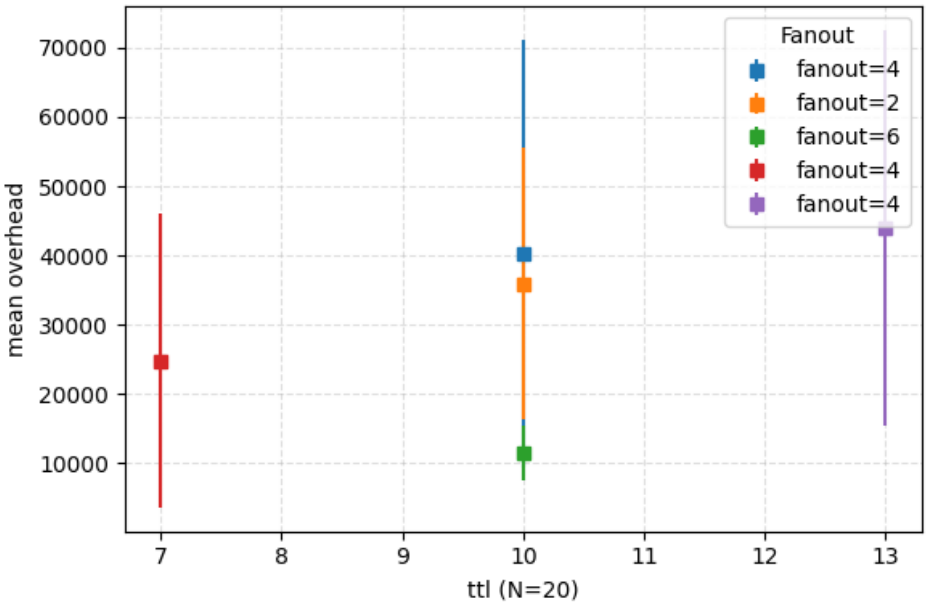


۵-۳ نتیجه‌گیری

این پروژه یک پیاده‌سازی کامل و کارآمد از پروتکل انتشار اطلاعات را به نمایش می‌گذارد. ترکیب روش‌های انتشار سریع و بازیابی دوره‌ای باعث شده تا قابلیت اطمینان سیستم به شدت بالا برود بدون آنکه نیاز به پیچیدگی‌های سنگین هماهنگی بین سرورها باشد. نتایج آزمایش‌ها ثابت کرد که این طراحی برای شبکه‌های توزیع‌شده، جمع‌آوری اطلاعات و هماهنگ‌سازی سیستم‌ها بسیار ایده‌آل و قابل اتکا است.


TTL
ttl=10
ttl=10
ttl=10
ttl=7
ttl=13
mean overhead
70000
60000
50000
40000
30000
20000
10000
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
fanout (N=20)


Message overhead (count)
350000
300000
250000
200000
150000
100000
50000
0
10
20
50
Network size (N)


Fanout
fanout=4
fanout=2
fanout=6
fanout=4
fanout=4
mean overhead
70000
60000
50000
40000
30000
20000
10000
7
8
9
10
11
12
13
ttl (N=20)