گزارش وضعیت پروژه نهایی

مسیریابی معنایی مقیاس‌پذیر با Gossip در NDN، و سوگیری کانال بازخورد

نگارندگان: محمدمهدی یاری، سجاد تقی‌زاده - استاد راهنما: دکتر محمدرضا شکورنیا

همه اعداد این گزارش از خروجی خام آزمایش‌ها (۲۰ اجرا با seed متفاوت) استخراج شده‌اند.

---

1. مسئله چیست

در شبکه‌های مبتنی بر نام (NDN) درخواست‌ها با نام دقیق مسیریابی می‌شوند، اما دستگاه‌ها و کاربران مختلف یک سرویس را با نام‌های متفاوت صدا می‌زنند؛ مثلا یکی /hospital/temp-sensor/room-101 می‌خواهد و جدول مسیریابی فقط /smart-hospital/building-a/floor-1/temperature/room-101 را می‌شناسد. مسیریابی دقیق چنین درخواستی را رد می‌کند. کار قبلی (Amadeo و همکاران) این مشکل را با اجرای یک مدل امبدینگ متنی روی هر درخواست نامنطبق و مقایسه شباهت کسینوسی حل می‌کند، اما اجرای این مدل روی هر روتر پرهزینه است: یک بار اجرای MiniLM حدود ۷ میلی‌ثانیه طول می‌کشد، در حالی که جست‌وجوی شباهت در جدول مسیریابی هزاران برابر سریع‌تر است. پس تنها کمیتی که ارزش بهینه‌سازی دارد، تعداد دفعات اجرای مدل است.

راه‌حل آن کار برای کاهش این هزینه یک کش محلی است، اما این کش با افزایش تعداد روترها بدتر عمل می‌کند: هرچه ترافیک بین روترهای بیشتری تقسیم شود، سهم هر کش کمتر می‌شود. این پروژه همین نقطه را هدف گرفته و یک نتیجه دوم هم پیدا کرده که در ابتدا دنبالش نبود.

2. چه چیزی ساخته شده است

یک شبیه‌ساز شبکه رویداد-گسسته به‌طور کامل نوشته شده است: روتر با صف سرویس، جدول مسیریابی (FIB)، جدول انتظار درخواست (PIT) و حافظه محتوا (Content Store)، و لایه انتقال با تأخیر لینک و صف‌بندی. روی آن ده راهبرد مسیریابی پیاده‌سازی و مقایسه شده‌اند: NDN معمولی، دو نسخه مسیریابی معنایی (با و بدون کش)، SEF، و راهبردهای پیشنهادی این پروژه به‌همراه نسخه‌های حذفی (Ablation) برای سنجش سهم هر بخش.

درستی مدل انتقال به‌طور مستقل با شبیه‌ساز استاندارد ns-3/ndnSIM مقایسه شده و میانه زمان رفت‌وبرگشت بسته‌ها با اختلاف کمتر از 0.11 میلی‌ثانیه همخوانی دارد. پیاده‌سازی با 51 تست خودکار بررسی می‌شود.

3. نتیجه اول: اشتراک‌گذاری با Gossip

در راهبرد پیشنهادی، یک تطبیق نام فقط زمانی که با برگشت واقعی داده تأیید شود، از طریق یک پروتکل Gossip (anti-entropy) به بقیه شبکه اطلاع داده می‌شود. یعنی یک نام در کل شبکه یک بار نیاز به اجرای مدل دارد، نه یک بار به ازای هر روتر. تعداد اجرای مدل برای بار ترافیکی یکسان، با افزایش روترهای لبه از 1 به 16 (اجرای 60 ثانیه‌ای):

راهبرد	1 روتر	2 روتر	4 روتر	8 روتر	16 روتر	رشد
معنایی بدون کش	2022	2045	2062	2073	2079	+3%
معنایی + کش محلی	833	901	1006	1155	1330	+60%
Gossip (این پروژه)	875	893	877	914	959	+10%

اما این عدد یک قید مهم دارد که در نسخه‌های قبلی گزارش گفته نشده بود و در این دور اندازه‌گیری شد: این یک اجرای 60 ثانیه‌ای است، و مزیت با طولانی‌تر شدن اجرا کم می‌شود. دلیلش ساده است: هزینه کش سرد یک هزینه یک‌باره است که N روتر آن را N بار می‌دهند، ولی فقط یک بار. با اجرای طولانی‌تر این هزینه ثابت روی ترافیک بیشتری تقسیم می‌شود و کش هر روتر گرم می‌شود:

طول اجرا	صرفه‌جویی Gossip	صرفه‌جویی sync کامل	بایت مصرفی
60 ثانیه	26% / 27%	21% / 23%	1.2 برابر
240 ثانیه	16% / 17%	13% / 15%	1.4 برابر
600 ثانیه	7.5% / 9.5%	6.2% / 8.3%	1.6 برابر

(اعداد به‌صورت بیمارستان / شهر، یعنی دو کاتالوگ مستقل.) ترتیب نتیجه هیچ‌وقت در مقیاس بزرگ برعکس نمی‌شود، ولی اندازه مزیت حدود دو سوم کم می‌شود. قید دوم: زیر حدود چهار روتر لبه، Gossip اصلا بازنده است؛ در یک روتر لبه کسی نیست که با او به اشتراک بگذاری و anti-entropy فقط سربار است (حدود 5 درصد بدتر). پس ادعای دقیق این است: اشتراک‌گذاری به شبکه‌ای می‌ارزد که بزرگ است و هنوز کاتالوگش را یاد می‌گیرد، و هرچه آن شبکه بنشیند کمتر می‌ارزد.

ستون سوم جدول بالا به مهم‌ترین سؤالی جواب می‌دهد که این طراحی برمی‌انگیزد: NDN از قبل لایه همگام‌سازی دارد (ChronoSync، PSync، SVS)، پس چرا از آن استفاده نشود؟ یک آرم با همگام‌سازی کاملا بی‌قید (هر روتر با همه همسایه‌ها، بدون سقف) اجرا شد و نتیجه این بود که بدتر است: inference کمتری حذف می‌کند و 1.2 تا 1.6 برابر بایت بیشتر می‌فرستد. دلیلش این است که هیچ‌کدام محدود به سرعت انتشار نیستند؛ چیزی که صرفه‌جویی را محدود می‌کند تعداد عبارت‌های متمایزی است که شبکه هنوز ندیده.

4. نتیجه دوم: سوگیری کانال بازخورد

این یافته در ابتدا هدف پروژه نبود و از دل بررسی یک نتیجه منفی بیرون آمد. ایده اولیه این بود که شبکه می‌تواند از بازخورد خود مسیریابی (Data یعنی درست، Nack یعنی غلط) برچسب رایگان برای کالیبراسیون بگیرد و به‌جای آستانه شباهت ثابت، برای هر مسیر یک مرز تصمیم یاد بگیرد.

در بررسی کد مشخص شد که تولیدکننده سرویس دو دلیل کاملا متفاوت رد را در هم می‌ریزد: «این نام اصلا مال من نیست» که یعنی مسیر غلط بوده، و «این نام مال من هست ولی این عبارت را نمی‌شناسم» که یعنی مسیر درست بوده و فقط واژگان اعلام‌نشده بوده است. تولیدکننده تفاوت را می‌داند و روی سیم دور می‌ریخت. نتیجه اینکه هر شکاف در واژگان اعلام‌شده، به‌عنوان شاهدی علیه مسیری که درست بود وارد کالیبراسیون می‌شد و مرز همه مسیرها را بی‌جهت سخت‌گیرتر می‌کرد.

این همان الگویی است که در یادگیری از بازخورد ضمنی با نام MNAR شناخته می‌شود؛ سهم این پروژه پیداکردن آن در یک صفحه داده NDN و ردیابی‌اش به یک شکاف پروتکلی است: Nack در NDN معادلی برای 406 در HTTP یا NODATA در DNS ندارد. راه‌حل افزودن کد دلیل به Nack است. نرخ پاسخ‌گویی بر حسب پوشش اظهارنامه تولیدکننده (بیمارستان):

پوشش اظهارنامه	1.0	0.9	0.7	0.5
واژگان اعلام‌نشده	0%	10%	29%	51%
بدون کد دلیل	0.919	0.864	0.801	0.762
با کد دلیل	0.947	0.924	0.880	0.839

بیشترین سود (حدود 0.08+) دقیقا جایی است که سوگیری بیشترین شدت را دارد، و روی کاتالوگ شهر هم همین الگو تکرار می‌شود. این «رژیم داشتن» خودش شاهدی است بر اینکه توضیح درست است: اگر تصحیح همه‌جا یکسان سود می‌داد، دلیلی نبود باور کنیم مکانیزمش همان چیزی است که ادعا می‌شود. بودجه خطا هم در هر 14 حالت آزمون حفظ شد، یعنی این سود به قیمت شکستن ضمانت به دست نیامده است.

5. آنچه آزموده شد و رد شد

ادعای اولیه پروژه این بود که بودجه خطای یادگیرنده از آستانه ثابت کاراتر است. این ادعا در برابر آستانه‌ای که روی یک کاتالوگ تنظیم و به کاتالوگ دیگر منتقل شده بود آزمایش شد و رد شد: در هیچ‌کدام از 14 مقایسه برتری کامل نداشت. پس از کشف سوگیری بالا، این فرضیه مطرح شد که شاید علت شکست همان سوگیری بوده و تصحیحش ادعا را برگرداند. این هم آزمایش شد و نتیجه منفی بود: با تصحیح، آستانه منتقل‌شده 11 بار غالب شد به‌جای 4 بار قبلی، یعنی مقایسه بدتر شد.

دلیلش قابل توضیح است: وقتی ردهای واژگانی خطا حساب نشوند، کنترلر سهل‌گیرتر می‌شود و هم نرخ پاسخ‌گویی و هم خطا بالا می‌رود؛ و حریفش قوی است، چون روی این کاتالوگ‌ها منحنی دقت در کل بازه آستانه بالای 0.99 صاف است و یک آستانه سهل‌گیر تقریبا مجانی است. بنابراین پس‌گرفتن آن ادعا سر جایش می‌ماند و در گزارش به‌عنوان مشارکت مطرح نمی‌شود. آنچه باقی می‌ماند محدودتر است: بی‌نیازی از کاتالوگ برچسب‌خورده برای تنظیم، و سود تصحیح سوگیری در رژیمی که بالا مشخص شد.

6. یافته‌های تکمیلی

- در برابر مهاجمی که در هر دور Gossip اطلاعات جعلی تزریق می‌کند، نرخ پاسخ‌گویی از 0.940 به 0.745 در آلودگی 50 درصد افت می‌کند و خطای واقعی به 0.168 می‌رسد. حمله متوقف نمی‌شود؛ فقط خسارتش با این محدود می‌ماند که مسیرهای تأییدشده خود هر روتر بر هر چیزی که به او گفته می‌شود اولویت دارد. مدل تهدید به‌صورت صریح نوشته شد و provenance و reputation به‌عنوان محدودیت شناخته‌شده ذکر شدند.
- وقتی تولیدکننده بدون هیچ رویداد مسیریابی فقط دامنه پاسخ‌گویی خودش را کم کند، تأیید و Gossip مزیت معناداری می‌دهد: 0.0052+ روی بیمارستان (20 از 20 اجرا) و 0.0101+ روی شهر (19 از 20). در جابه‌جایی معمول تولیدکننده‌ها همه راهبردها تقریبا یکسان عمل می‌کنند.
- اگر کنترل ریسک بدون هیچ اکتشافی اجرا شود در نقطه سخت‌گیرانه گیر می‌کند: چون درخواستی قبول نمی‌شود، بازخورد جدیدی هم نمی‌آید. با اختصاص 5 درصد از تصمیم‌های ردشده به جمع‌آوری بازخورد، تعداد مشاهدات از 104 به 598 و نرخ پاسخ‌گویی از 0.738 به 0.818 می‌رسد.

7. دو باگ بازتولیدپذیری که پیدا و رفع شد

هنگام بازآزمون مشخص شد که اجرای یکسان با seed یکسان در دو پروسه متفاوت اعداد کمی متفاوت می‌دهد. دو علت داشت: دایجست لایه Gossip از تابع hash پایتون ساخته می‌شد که نمکش در هر پروسه فرق می‌کند، و جدول PIT چهره‌های ورودی را در یک set نگه می‌داشت که ترتیب پیمایشش ترتیب ارسال پاسخ‌ها و در نتیجه ترتیب رویدادها را عوض می‌کرد.

هیچ‌کدام متریک نتیجه را جابه‌جا نمی‌کرد و انحراف حدود 0.03 درصد و فقط در شمارنده‌های Gossip بود. ولی معنایش این بود که هیچ‌کس نمی‌توانست اعداد گزارش را دقیقا بازتولید کند، که برای یک کار علمی بدتر از یک خطای کوچک است. هر دو رفع شد و تستی اضافه شد که دو پروسه فرزند با مقادیر متفاوت PYTHONHASHSEED اجرا می‌کند؛ صحت تست با برگرداندن هر دو باگ بررسی شد. سپس کل کمپین از یک نسخه کد بازتولید شد و همه ادعاهای کلیدی رقم‌به‌رقم پابرجا ماندند.

8. جمع‌بندی و ارزیابی صادقانه

خروجی این پروژه یک نتیجه مقیاس‌پذیری با دامنه اندازه‌گیری‌شده است (نه یک ادعای مطلق)، به‌علاوه یک یافته تازه درباره سوگیری کانال بازخورد که مکانیزم و رفعش هر دو مشخص شده‌اند، روی زیرساختی که نتایجش بازتولیدپذیر است.

صادقانه درباره نوآوری: مکانیزم Gossip جدید نیست، چون NDN از 2013 همگام‌سازی anti-entropy دارد و NLSR از قبل پیشوندهای نام را روی همان لایه پخش می‌کند. ایده یادگیری مرز تصمیم زیر یک کران خطا هم از قبل در کش معنایی و در ادبیات conformal وجود دارد. چیزی که واقعا نو است، یافته سوگیری کانال بازخورد و شکاف پروتکلی پشت آن است. همه این‌ها در بخش کارهای مرتبط با استناد مشخص شده‌اند.

این نتایج در سطح یک مقاله سنجش با یک یافته مکانیزمی قابل ارائه‌اند، نه یک مقاله مکانیزم جدید. بزرگ‌ترین محدودیت باقی‌مانده این است که هر دو کاتالوگ ساختگی‌اند، و اینکه یک استقرار واقعی چقدر واژگانش را کامل اعلام می‌کند، همان چیزی است که ارزش تصحیح بخش 4 کاملا به آن بستگی دارد. ساختن یک کاتالوگ از داده‌های واقعی، ارزان‌ترین گام بعدی برای تقویت کار است.