

مقایسه ای بر پروتکل های سیگنالینگ hard-state و soft-state

پینگ جی، زیهوی ژ، جیم کروز، و دان توسلی
دپارتمان علوم کامپیوتر، دانشگاه ماساچوست در آمهرست

فهرست مطالب

۲	چکیده
۲	۱. مقدمه
۴	۲. Soft-state، hard-state، و پروتکل های میانی
۷	۳. مدلسازی و تحلیل روش های سیگنالینگ
۷	۳،۱ سیگنالینگ در یک سیستم تک-پرشی
۱۱	۳،۱،۲ حل مدل و محاسبات عملکرد
۱۲	۳،۱،۳ ارزیابی مدل
۱۸	۳،۲ سیگنالینگ در یک سیستم چندپرشی
۱۸	۳،۲،۱ مدل سیستم های چندپرشی
۲۰	۳،۲،۲ راه حل و نتایج مدل چند-پرشی
۲۱	۴. پژوهش های مربوطه
۲۲	۵. نتیجه گیری و پژوهش های آتی
۲۲	مراجع

چکیده

یکی از اجزای کلیدی زیرساختی در همه شبکه های ارتباطی، از شبکه تلفن گرفته تا شبکه های داده مبتنی بر VC، تا اینترنت، عبارت است از سیستم سیگنالینگ آن. دو رویکرد در رابطه با سیگنالینگ می توان عنوان کرد: رویکردهای **hard-state** و **soft-state**. علیرغم اهمیت مبنایی سیگنالینگ، درک ما از این رویکردها - مزایا و معایب آنها و شرایطی که در بهترین حالت می توان آنها را مورد استفاده قرار داد - اکثراً تخیلی (و گاهی مذهبی) است. در این مقاله، ما انواعی از رویکردهای سیگنالینگ را از **soft state** محض گرفته تا رویکردهای **soft state** تقویت شده با حذف وضعیت صریح و/یا سیگنالینگ قابل اطمینان، تا روش **hard-state** محض مورد مقایسه قرار می دهیم. ما یک مدل تحلیلی طراحی می کنیم که به ما امکان می دهد ناسازگاری وضعیت در حالت های تک-پرشی و چند-پرشی، و هزینه (هم از لحاظ سربار سیگنالینگ، و هزینه های وابسته به اپلیکیشن ناشی از ناسازگاری حالت) مرتبط با یک رویکرد سیگنالینگ خاص و پارامترهای آن (مانند تایمرهای **refresh** و **removal** وضعیت) را اندازه گیری کنیم. در گروه رویکردهای **soft-state**، پی می بریم که یک رویکرد **soft-state** به همراه حذف صریح درجه سازگاری وضعیت را به میزان زیادی بهبود می بخشد. اضافه کردن راه اندازی/بروزرسانی/حذف به رویکرد **soft-state** اجازه می دهد به سازگاری قابل قیاس (و گاهی بهتر) در مقایسه با رویکرد **soft-state** برسد.

دسته بندی ها و توصیف کننده های موضوعی

C.2.2 [شبکه های کامپیوتری-مخابراتی]

واژه های عمومی

عملکرد، طراحی

کلمات کلیدی

سیگنالینگ، **soft-state**، **hard-state**

۱. مقدمه

یکی از اجزای کلیدی در همه شبکه های مخابراتی، از شبکه تلفن گرفته تا شبکه های داده ای مبتنی بر VC و اینترنت عبارت است از سیستم سیگنالینگ آن. دو دسته وسیع از رویکردهای سیگنالینگ را می توان مورد بررسی قرار داد: رویکردهای **hard-state** و **soft-state**. بین این دو محدوده، روش هایی برای سیگنالینگ وجود دارد که در عمل مکانیزم های مختلفی از هر کدام عاریه می گیرند. علیرغم اهمیت اساسی سیگنالینگ، درک ما از این دو رویکرد - مزایا و معایب آنها و شرایطی که بهتر می توان از آنها استفاده کرد هنوز بخوبی فهمیده نشده است.

به بیان عمومی، ما واژه **soft-state** را با روش های سیگنالینگ ارتباط می دهیم که در آن وضعیت نصب شده «تایم اوت» (و حذف) می شود مگر اینکه به صورت دوره ای با دریافت یک پیام سیگنالینگ **refresh** گردد (که پیام مذکور معمولاً از موجودیتی می آید که در ابتدا وضعیت را نصب کرده است) که این نشان می دهد که وضعیت باید همچنان نصب باقی بماند. از آنجا که وضعیت **refresh** نشده نهایتاً تایم اوت خواهد شد، سیگنالینگ **soft-state** نه نیاز به حذف وضعیت صریح دارد و نه رویه ای برای حذف وضعیت یتیم شده^۱ در صورت وقوع اختلال در نصب کننده وضعیت. به طور مشابه، از آنجا که پیام های نصب و **refresh** وضعیت قبل از پیام های **refresh** دوره ای بعدی ارسال می شوند، سیگنالینگ قابل اطمینان مورد نیاز نمی باشد. واژه **soft-state** توسط کلارک [۳] ابداع شد، که مفهوم پیام های **refresh** حالت دوره ای ارسال شده توسط یک سیستم انتهایی را توصیف کرد، و پیشنهاد کرد که با چنین پیام های **refresh**، ممکن است وضعیت در یک اختلاف گم شود و سپس به وسیله پیام های **refresh** بعدی به حالت قبل بازگردانده شود - کلیه این قضایا برای سیستم انتهایی، شفاف است، بدون اینکه مستلزم هیچ گونه رویه بازبازی اختلال صریحی باشد:

^۱ Orphaned state

بین محدوده های یک رویکرد **hard-state** و یک رویکرد **soft-state** محض پروتکل های زیادی وجود دارند که از عناصر هر روش استفاده کرده اند. در حقیقت، پروتکل هایی که در ابتدا به عنوان پروتکل های **soft-state** شناخته می شد از تعدادی از مکانیزم های **hard-state** (اغلب به صورت افزونه هایی) در طی زمان استفاده کرده اند. برای مثال، در **IGMPv1** [۴]، تایم اوت **soft-state** در یک مسیریاب برای تشخیص جدا شدن هاست های قبلاً ثبت شده به کار رفت؛ **IGMPv2/v3** [۸، ۲] بعداً یک پیام خداحافظی^۲ صریح اضافه کرد تا به یک هاست اجازه دهد به صورت صریح مسیریاب دارنده وضعیت را از رفتن خود آگاه کند. در **RSVP** اصلی [۲۰]، پیام های نصب وضعیت **PATH** و **RESV** به صورت بهترین-تلاش با این فرض منتقل می شد که از دست رفتن یک پیام سیگنالینگ از پیام **refresh** بعدی بازبایی می شود؛ سیگنالینگ مبتنی بر **ACK** قابل اطمینان به صورت افزونه ای به **RSVP** در [۱] معرفی شد و نیز [۱۳] پیشنهاد گردید. **RSVP** همچنین حذف صریح (هرچند اختیاری) ویژگی های فیلتر را از زمان معرفی خود ارائه کرده است. در پروتکل سیگنالینگ **hard-state** از نوع **ST-II**، پیام های دوره ای **HELLO** به فرستنده **HELLO** اطلاع می دهد که اوضاع همسایه خوب است، و اینکه وضعیت خودش که مبتنی بر یک همسایه بخصوص است هنوز معتبر می باشد – **refresh** کردن صریح وضعیت خود. با توجه به تمایزهای مبهم بین روش های **hard-state** و **soft-state** و این موضوع که پروتکل های موجود در یک دسته اغلب از مکانیزم هایی استفاده می کنند که مرتبط با دیگری است، ما بر این باور هستیم که موضوع مهم این نیست که مطلقاً رویکرد **hard-state** یا **soft-state** به شکل مطلق بهتر است. بلکه، به عقیده ما سؤال مهم تر فهمیدن این موضوع است که چگونه مکانیزم هایی که در پروتکل های مختلف **hard-state** و **soft-state** به کار

«... اطلاعات وضعیت ممکن است در نگهداری نوع مطلوب سرویس مرتبط با جریان، حیاتی نباشند. در عوض، این نوع سرویس توسط نقاط انتهایی مشخص خواهد شد، که به صورت دوره ای پیام هایی ارسال می کنند تا اطمینان یابند که نوع درست سرویس با جریان ارتباط داده شده است. به این شکل، اطلاعات وضعیت ارتباط داده شده با جریان در یک اختلال گم می شوند بدون اینکه ویژگی های سرویس مورد استفاده به طور دائم دچار اختلال شود. من این مفهوم را «**soft state**» می خوانم، و شاید بخوبی به ما این امکان را بدهد که به اهداف اصلی خودمان از لحاظ بقاءپذیری و انعطاف پذیری برسیم...»

به طور کلی، آنگاه، اساس روش **soft-state** استفاده از نصب وضعیت/**refresh** دوره ای بهترین-تلاش توسط نصب کننده وضعیت و حذف وضعیت با تایم اوت در دارنده وضعیت می باشد. رویکردهای **soft-state** در پروتکل های بیشماری از جمله **RSVP** [۲۰]، **SRM** [۹]، **PIM** [۷، ۵، ۶]، **SIP** [۱۰] و **IGMP** [۴] به کار رفته اند. سیگنالینگ **hard-state** رویکرد برعکس نسبت به **soft state** اتخاذ می کند – وضعیت نصب شده همچنان باقی می ماند مگر اینکه صریحاً توسط گیرنده یک پیام انقطاع وضعیت از سوی نصب کننده وضعیت حذف شود. از آنجا که اگر وضعیت صریحاً حذف نشود همچنان به صورت نصب شده باقی می ماند، سیگنالینگ **hard-state** به مکانیزمی نیاز دارد تا وضعیت یتیم را که بعد از اختلال در نصب کننده یا جدا شدن آن بدون حذف حالت باقی می ماند حذف کند. به طور مشابه، از آنجا که نصب وضعیت و حذف آن تنها یک مرتبه انجام می شود (بدون **refresh** وضعیت یا تایم اوت وضعیت)، مهم است که نصب کننده وضعیت بداند چه وقت وضعیت نصب یا حذف شده است. بنابراین پروتکل های سیگنالینگ قابل اطمینان (به جای بهترین-تلاش) معمولاً با پروتکل های **hard-state** ارتباط داده می شوند. به بیان کلی، آنگاه، اساس یک رویکرد **hard-state** نصب و حذف صریح اطلاعات وضعیت است. رویکردهای **hard-state** در پروتکل هایی مانند **ST-II** [۱۴، ۱۸] و **Q.2931b** [۱۵] به کار رفته است.

^۲ Leave message

^۱ State-teardown message

رفته اند می توانند به بهترین حالت در موقعیت های بخصوص به کار روند، و چرا.

بر عملکرد تأثیر خواهند گذاشت. بخش ۳،۱ یک مدل تحلیلی برای بررسی عملکرد این روش ها در حالت تک-پریش ارائه کرده و به مقایسه عملکرد آنها می پردازد. بخش ۳،۲ به بررسی حالت چند-پریشی می پردازد. بخش ۴ کارهای مربوطه را مورد بحث قرار می دهد. نهایتاً، بخش ۵ به جمع بندی مقاله و بحث راجع به پژوهش های آتی می پردازد.

۲. **Soft-state, hard-state** و پروتکل

های میانی

در این بخش، ما به توصیف عملکرد پنج پروتکل سیگنالینگ انتزاعی مختلف می پردازیم. این پروتکل ها در نحوه نصب، نگهداری، و حذف وضعیت و همین طور اینکه آیا پیام های سیگنالینگ انتخابی به صورت بهترین-تلاش انتقال یابند یا به صورت قابل اطمینان، متفاوت می باشند. ما یک گره تک را (که از این به بعد آن را «فرستنده سیگنالینگ» می خوانیم) بررسی خواهیم کرد که می خواهد وضعیت را در یک گره دوردست (که آن را «دریافت کننده سیگنالینگ» خواهیم خواند) نصب، نگهداری، و نهایتاً حذف کند (یا وادار به حذف نماید). ما مثالی ساده، ولی گویا از یک فرستنده سیگنالینگ بررسی می کنیم که دارای یک مقدار وضعیت محلی است که می خواهد در یک یا چند دریافت کننده سیگنالینگ در مسیر سیگنالینگ نصب کند. وقتی مقدار وضعیت فرستنده سیگنالینگ برابر مقدار وضعیت گیرنده (های) سیگنالینگ باشد، می گوئیم که مقادیر سازگار^۱ [۱۶] هستند؛ در غیر اینصورت مقادیر وضعیت فرستنده و گیرنده (ها) سازگار می باشند. هدف ما در اینجا آن نیست که یک پروتکل سیگنالینگ بخصوص مانند RSVP یا Q2931b مدلسازی کنیم، بلکه هدف آن است که جنبه های اساسی رویکردهای پروتکل های مختلف در رابطه با سیگنالینگ را بیان کنیم. بعد از توصیف پروتکل ها، ما سپس به بررسی معیارهای عملکردی می پردازیم که به وسیله آن می توان

در این مقاله، از این رو، ما به مقایسه انواعی از روش های سیگنالینگ از روش soft-state محض گرفته تا soft-state مجهز به حذف وضعیت از راه دور صریح و/یا سیگنالینگ قابل اطمینان، تا روش hard-state محض می پردازیم. ما دسته ای از پروتکل های عمومی تعریف می کنیم که در راستای این طیف قرار دارند، و یک مدل تحلیلی پارامتری تعریف می کنیم که به ما امکان می دهد یک معیار عملکرد کلیدی را با یک پروتکل سیگنالینگ بخصوص - کسری از زمان که وضعیت نصب کننده حالت و دارنده حالت ناسازگار هستند [۱۶] - اندازه بگیریم. ما همچنین هزینه را (هم از لحاظ سربار سیگنالینگ، و هم هزینه های وابسته به اپلیکیشن ناشی از ناسازگاری وضعیت) مرتبط با یک روش سیگنالینگ بخصوص و مقادیر پارامتر آن (مانند فواصل زمانی refresh حالت و تایم اوت حذف) می سنجیم. در میان گروه رویکردهای soft-state، در می یابیم که اضافه کردن حذف صریح در یک رویکرد soft-state به میزان زیادی سازگاری وضعیت را بهبود می بخشد، و در عین حال سربار پیام سیگنالینگ پایینی دخیل می کند. اضافه کردن راه اندازی/بروزرسانی/حذف قابل اطمینان همچنین به روش soft-state امکان می دهد به سازگاری قابل قیاس (و گاهی بهتری) در مقایسه با وضعیت hard-state برسد.

پژوهش ما در اینجا بر روی ارزیابی عملکرد مکانیزم ها و پروتکل های مختلف سیگنالینگ تمرکز دارد. با این حال، مؤلفه های غیر مبتنی بر عملکرد دیگری وجود دارد که با استفاده از آن می توان روش های سیگنالینگ مختلف (مانند پیچیدگی یا «نیرومندی» اجرای پروتکل) را مقایسه نمود. لیکن، این معیارها فراتر از حوزه این مقاله هستند.

بقیه این مقاله به صورت زیر ساخت یافته است. در بخش ۲، ما پنج پروتکل سیگنالینگ مختلف را شرح می دهیم که از مکانیزم های مختلف hard-state و soft-state بهره می گیرند، و به شکل کمی به بحث فاکتورهایی می پردازیم که

^۱ Consistent

صریح اضافه شده است. وقتی وضعیت در فرستنده سیگنالینگ حذف می شود، فرستنده یک پیام سیگنالینگ بهترین-تلاش (غیر قابل اطمینان) حاوی اطلاعات حذف وضعیت صریح به گیرنده سیگنالینگ می فرستد. پیام های refresh وضعیت و تریگر، و یک تایمر تایم اوت وضعیت همگی همانند حالت SS استفاده می شوند.

Soft-state با تریگر قابل اطمینان (SS+RT): SS+RT مشابه SS است اما با دو افزونه مهم. اول اینکه، پیام های تریگر به صورت قابل اطمینان در SS+RT منتقل می شوند. هر وقت یک پیام تریگر انتقال می یابد، فرستنده یک تایمر باز-ارسال (با مقدار R) شروع می کند. در هنگام دریافت یک پیام تریگر صریح، مقصد نه تنها وضعیت سیگنالینگ را بروزرسانی می کند، بلکه یک تأیید به فرستنده ارسال می کند. اگر هیچ تأیید تریگر قبل از منقضی شدن تایمر بازارسال دریافت نشود، فرستنده سیگنالینگ پیام تریگر را بازارسال می کند. ثانیاً، SS+RT همچنین از یک مکانیزم آگاه سازی استفاده می کند که در آن مقصد سیگنالینگ به فرستنده راجع به حذف وضعیت های ناشی از منقضی شدن تایمر تایم اوت وضعیت اطلاع می دهد. این موضوع به فرستنده سیگنالینگ اجازه می دهد از حذف غلط با ارسال یک پیام تریگر جدید، بازبایی کند.

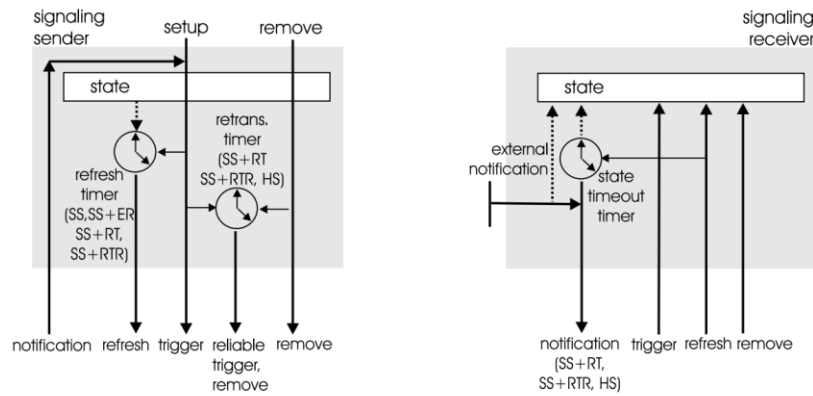
Soft-state با تریگر/حذف قابل اطمینان (SS+RTR): SS+RTR مشابه روش SS+RT است، با این تفاوت که روش SS+RTR از پیام های قابل اطمینان برای کنترل نه تنها راه اندازی/بروزرسانی وضعیت بلکه برای حذف وضعیت استفاده می کند.

روش **Hard-state (HS):** در روش HS، پیام های قابل اطمینان صریح برای راه اندازی، بروزرسانی و حذف حالت در گیرنده سیگنال استفاده می شوند. نه از پیام های refresh استفاده می شود و نه از مکانیزم های حذف تایم اوت soft-state. یک دغدغه مهم در رابطه با هر پروتکل hard-state عبارت است از حذف وضعیت یتیم در گیرنده سیگنالینگ.

این پروتکل ها را مورد ارزیابی قرار داد، و به صورت کیفی فاکتورهایی را که بر عملکرد تأثیر خواهد داشت مورد بحث قرار می دهیم.

پنج رویکرد زیر را مورد بررسی قرار خواهیم داد:
Soft-state محض (SS): در این روش، فرستنده سیگنالینگ یک پیام تریگر [۱] حاوی اطلاعات نصب یا بروزرسانی به گیرنده سیگنالینگ می فرستد، و یک تایمر refresh وضعیت (با مقدار T) شروع می کند. وقتی تایمر refresh وضعیت منقضی می شود، فرستنده سیگنالینگ یک پیام refresh [۲۰] حاوی بروزترین اطلاعات وضعیت سیگنالینگ می فرستد، و تایمر را ریست می کند. پیام های تریگر و refresh به یک شکل بهترین-تلاش (غیر قابل اطمینان) ارسال می شوند. وقتی یک پیام تریگر یا refresh در گیرنده سیگنالینگ دریافت می شود، اطلاعات متناظر وضعیت سیگنالینگ ضبط می شود و یک تایمر تایم اوت وضعیت (با مقدار X) وابسته به این وضعیت به کار انداخته می شود (یا در صورتی که در حال اجرا باشد دوباره راه اندازی می شود). وضعیت سیگنالینگ در گیرنده سیگنالینگ تنها وقتی حذف می شود که تایمر تایم اوت وضعیت آن منقضی شود؛ یعنی، مادامی که گیرنده پیام های refresh قبل از منقضی شدن تایمر تایم اوت وضعیت دریافت می کند، وضعیت حفظ خواهد شد. علت وقوع این تایم اوت می تواند آن باشد که فرستنده سیگنالینگ ارسال پیام های refresh را متوقف کرده است (زیرا وضعیت محلی آن حذف شده است و بنابراین می خواهد وضعیت دوردست در گیرنده سیگنالینگ نیز حذف شود)، یا اینکه پیام های refresh در انتقال گم شده اند، و به یک تایم اوت وضعیت در گیرنده سیگنالینگ انجامیده اند. ما به این حالت دوم حذف غلط وضعیت می گوئیم، زیرا فرستنده سیگنالینگ نمی خواست این وضعیت حذف شود.

Soft-state با حذف صریح (SS+ER): SS+ER مشابه روش SS است، با این فرق که یک پیام حذف وضعیت



شکل ۱: فرستنده و گیرنده سیگنالینگ: پیام ها و مکانیزم ها

کار رفته توسط فرستنده و گیرنده سیگنالینگ در پروتکل های مختلف سیگنالینگ را نشان می دهد.

در بخش زیر، ما یک مدل تحلیلی پارامتری واحد طراحی خواهیم کرد که به ما اجازه می دهد یک معیار کلیدی مرتبط با یک پروتکل سیگنالینگ بخصوص - کسری از زمان که وضعیت نصب کننده وضعیت و دارنده وضعیت سازگار هستند (یعنی مقدار یکسان دارند) را اندازه گیری کنیم. به وضوح، ما می خواهیم این مقدار حتی الامکان به یک نزدیک باشد. علاوه بر اندازه گیری سازگاری، ما همچنین می خواهیم هزینه وابسته به یک روش سیگنالینگ بخصوص و سطح سازگاری که قادر است به دست آورد را اندازه بگیریم. یک جنبه از این هزینه خود نرخ پیام سیگنالینگ است. یک جنبه دوم این هزینه، هزینه بودن در وضعیت ناسازگار است. برای مثال، در IGMP، وقتی یک میزبان انتهایی بدون اطلاع دادن رفتن خود به مسیریاب لبه ای اش جدا می شود، داده های چندپنشی کماکان به سمت گیرنده جریان خواهد داشت (هرچند میزبان گیرنده دیگر در گروه مالیتی کست حضور ندارد) - یک هزینه. در مورد سیستم اشتراک فایل همتا به همتا که در آن مشتری در هنگام پیوستن به شبکه P2P نام فایل هایی را که به اشتراک گذاشته است به یک سرور می فرستد، اما آنگاه شبکه را بدون اطلاع دادن رفتن خود ترک می کند، وضعیت ناسازگار در سرور باعث می شود همتهای دیگر تلاش کنند با همتهای جدا شده تماس بگیرند - دوباره، یک هزینه دیگر. در بخش ۳، ما این هزینه را به صورت مجموع وزنی سربار سیگنالینگ و هزینه های

از آنجا که پروتکل hard-state حذف وضعیت مبتنی بر تایم اوت ارائه نمی کند، باید برای تشخیص اینکه وضعیت یتیم باقی است متکی به یک سیگنال خروجی باشد. این سیگنال را می توان برای نمونه، توسط یک پروتکل ضربان قلب مجزا ایجاد نمود که کار آن این است که تشخیص دهد چه وقت فرستنده سیگنالینگ دچار اختلال می شود و سپس گیرنده سیگنالینگ را از این موضوع آگاه سازد. به روش دیگر، سیگنال خارجی را می توان با آگاه سازی از یک پروتکل پایین تر در گیرنده سیگنالینگ ایجاد کرد که در ارتباط با یک پروتکل لایه پایین در فرستنده سیگنالینگ است و بنابراین تشخیص داد که فرستنده سیگنالینگ شکست خورده است. وقتی چنین آگاه سازی خارجی (سیگنال) دریافت می شود، روش سیگنالینگ hard-state وضعیت سیگنالینگ یتیم وابسته به فرستنده سیگنالینگ را پاکسازی می کند. یک فاکتور پیچیده کننده عبارت است از آگاه سازی غلط - سیگنال خارجی ممکن است به طور غلط یک خرابی در فرستنده سیگنالینگ تشخیص دهد (این اتفاق زمانی روی می دهد که مثلاً یک سری پیام ضربان قلب گم شود، اما فرستنده سیگنالینگ همچنان فعال باشد). همانند حالت SS+RT، آگاه سازی غلط را می توان با داشتن دریافت کننده سیگنالینگ به آگاه سازی فرستنده سیگنالینگ (اگر موجود باشد) از این که وضعیت یتیم شده آن حذف شده است، ترمیم نمود. یک فرستنده سیگنالینگ که وضعیت آن به شکل نادرستی حذف شده است سپس می تواند یک پیام تریگر جدید ارسال نماید. شکل ۱ پیام ها و مکانیزم های به

کشید تا یک پیام به طور قابل اطمینان تحویل داده شود، یا یک پیام refresh بهترین-تلاش تحویل داده شود. در موارد اتلاف و هزینه های بالای ناسازگاری وابسته به اپلیکیشن، پروتکل های با انتقال قابل اطمینان صریح ترجیح داده خواهند شد.

• **تعداد پرش ها:** در برخی پروتکل های سیگنالینگ

بخصوص مانند RSVP و AFSP [19]، یک فرستنده سیگنالینگ باید وضعیت را در چندین گره بین خودش و مقصد نهایی سیگنالینگ نصب نماید. با افزایش تعداد پرش ها، کسر زمانی که گره ها در وضعیت ناسازگار هستند نیز افزایش خواهد یافت.

در بخش های بعدی، ما یک مدل تحلیلی طراحی خواهیم کرد که به ما امکان می دهد به صورت کمی این موضوعات را مورد بررسی قرار دهیم.

۳. مدلسازی و تحلیل روش های سیگنالینگ

ما تحلیل خود را با بررسی مثال ساده یک گره تکی (فرستنده سیگنالینگ) شروع می کنیم که می تواند یک قطعه از اطلاعات وضعیت را در یک گره دوردست (گیرنده سیگنالینگ) نصب، نگهداری، متحول، و نهایتاً حذف کند (یا وادار به حذف نماید). ما در اینجا بر روی یک قطعه (به جای چند قطعه) از وضعیت تمرکز داریم، چرا که به لحاظ مفهومی ساده تر است و مورد دوم (چند قطعه) می تواند عموماً به صورت نمونه هایی از اولی در نظر گرفته شود. نصب، نگهداری، تغییر، و حذف حالت با استفاده از یکی از روش های سیگنالینگ انتزاعی پنج گانه شرح داده شده در بخش قبلی به دست می آید. ما فرض می کنیم که فرستنده و گیرنده بر روی شبکه ای ارتباط برقرار می کنند که می تواند پیام ها را به تأخیر انداخته و گم کند، اما نمی تواند آنها را باز-چینش کند.

۳,۱ سیگنالینگ در یک سیستم تک-پرسی

ما ابتدا یک سیستم تک پرسشی را در نظر می گیریم، که در آن فرستنده و گیرنده سیگنالینگ تنها موجودیت هایی هستند

وابسته به اپلیکیشن (متناظر با هزینه جریان های داده چندپخشی ناخواسته، یا تلاش های اتصال به یک همتای جدا شده در مثال های بالا) مدلسازی می نماییم.

این بخش را با یک بحث کیفی راجع به مؤلفه هایی که بر عملکرد پروتکل های سیگنالینگ تأثیر خواهند گذاشت خاتمه می دهیم:

• **هزینه ناسازگاری وابسته به اپلیکیشن.** همانگونه که

در بالا اشاره شد، این ها هزینه های وابسته به فرستنده و گیرنده سیگنالینگ در وضعیت های ناسازگار می باشند. بوضوح، وقتی این هزینه بالا باشد، فرستنده سیگنالینگ ممکن است راغب باشد سربار سیگنالینگ بالاتری متحمل شود تا وضعیت های فرستنده و گیرنده را تا حد ممکن سازگار نگه دارد.

• **مقدار تایم اوت refresh.** همانگونه که در [۱] اشاره

شد، هرچه مقدار تایم اوت refresh کوچکتر باشد، وضعیت سازگار زودتر در دارنده وضعیت نصب خواهد شد، و در نتیجه هزینه وابسته به اپلیکیشن ناشی از ناسازگاری وضعیت کوچکتر خواهد بود. با این حال، این مزیت به قیمت افزایش نرخ سیگنالینگ تمام می شود. اگر هزینه وابسته به اپلیکیشن وضعیت ناسازگار بالا باشد، این افزایش هزینه سیگنالینگ ممکن است تضمین شود.

• **مقدار تایم اوت soft-state.** از آنجا که این تایمر

قرار است وضعیتی را حذف کند که refresh نمی شود، به لحاظ ایده آل مقدار آن باید تا حد ممکن کوچک باشد تا وضعیت یتیم شده به محض جدا شدن فرستنده سیگنالینگ حذف گردد. با این حال، یک مقدار تایم اوت خیلی کوچک می تواند به حذف وضعیت غلط منجر گردد.

• **اتلاف پیام سیگنالینگ:** با بالا رفتن احتمال اتلاف

پیام، ما انتظار داریم که کسر زمانی که وضعیت های فرستنده و گیرنده سیگنالینگ ناسازگار هستند نیز افزایش یابد، چرا که زمان بیشتری طول خواهد

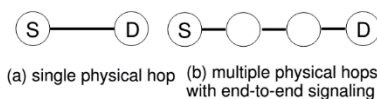
کند. بعد از مقداری تأخیر، پیام سیگنالینگ به گیرنده
دور دست می رسد، و فرستنده و گیرنده را قادر می سازد به
وضعیت سازگار برسند.

بروزرسانی وضعیت سیگنالینگ. فرستنده همچنین می تواند
وضعیت محلی خود را بروزرسانی کند. همانند راه اندازی
وضعیت، فرستنده سپس مقدار وضعیت جدید را در گیرنده
نصب می کند. وقتی یک فرستنده وضعیت محلی خود را
بروزرسانی می کند، وضعیت فرستنده و گیرنده ناسازگار
خواهد بود تا زمانی که بروزرسانی به طور موفق به گیرنده
انتشار یابد.

حذف وضعیت سیگنالینگ. در پایان چرخه حیات، فرستنده
وضعیت خود را حذف می کند. در این نقطه، وضعیت
گیرنده نیز حذف خواهد شد. وقتی فرستنده وضعیت خود
را حذف کرد، وضعیت گیرنده بیات (ناسازگار) است تا
زمانی که حذف گردد. چندین مکانیزم مبتنی بر پروتکل (از
جمله تایم اوت وضعیت، و پیام های حذف صریح) را می
توان برای حذف وضعیت گیرنده مورد استفاده قرار داد.

حذف غلط وضعیت سیگنالینگ. مقصد ممکن است به طور
نادرست وضعیت را حذف کند، هرچند فرستنده هنوز این
وضعیت را نگه داشته است. این می تواند در نتیجه وقایع
مبتنی بر پروتکل مختلفی روی دهد. برای مثال، در
رویکردهای *soft-state*، تایمر تایم اوت وضعیت ممکن
است در گیرنده منقضی شود و وضعیت حذف شود، هر
چند هنوز فرستنده آن وضعیت را نگه داشته است.

که در پروتکل سیگنالینگ درگیر می باشند. همانطور که در
شکل ۲ نشان داده شده است، می توانیم دو موجودیت را
متصل با یک پرش منطقی در نظر بگیریم، که ممکن است
مشکل از یک یا چند پرش فیزیکی باشد. تعدادی از
اپلیکیشن ها و پروتکل های موجود با این مدل تک-پرشی



شکل ۲: سیستم های سیگنالینگ تک-پرشی

سازگاری دارند. برای مثال، سیگنالینگ در پروتکل IGMP
[۴] بین یک سیستم انتهایی و مسیریاب اولین پرش آن روی
می دهد. وقتی سیستم انتهایی به یک گروه چندپخشی ملحق
می شود، وضعیت نشان دهنده این عضویت گروهی باید در
مسیریاب اولین-پرش نصب شود؛ وقتی میزبان انتهایی گروه
چندپخشی را ترک می کند این وضعیت باید از مسیریاب
حذف شود. در اپلیکیشن های اشتراک فایل همتا-به-همتا
بخصوصی مانند Kazaa [۱۲]، یک همتا فایل های اشتراک
گذاشته شده خودش را در یک سرور (یک گره والد^۱ در
مورد Kazaa) ثبت می کند که آنگاه همتهای در پی یک
فایل بخصوص را به گره های همتایی که فایل را دارند
هدایت می کند. ثبت فایل ها در یک گره والد توسط یک
همتا یک فرایند سیگنالینگ تک پرشی است، که در آن همتا
فرستنده سیگنالینگ است، گره والد گیرنده سیگنالینگ است،
و وضعیت سیگنالینگ حاوی شناسه های فایل های به
اشتراک گذاشته و این مطلب که همتا در سیستم حضور دارد
و سرویس فایل می دهد می باشد.

۳،۱،۱ توصیف مدل

قبل از توصیف مدل سیستم خود، ما به اختصار به بحث
وقایعی می پردازیم که در طی چرخه حیات یک زوج
فرستنده/گیرنده سیگنالینگ روی می دهند.

راه اندازی وضعیت سیگنالینگ. وقتی جلسه سیگنالینگ ابتدا
وضعیت محلی خود را نصب می کند (راه اندازی می کند)،
یک پیام سیگنالینگ حاوی وضعیت را به گیرنده ارسال می

^۱ Supernode

- وقتی وضعیت سیگنالینگ هم از فرستنده و هم از گیرنده حذف شود، سیستم وارد یک وضعیت جذب کننده می شود که با وضعیت مارکوف $(-, -)$ نشان داده می شود.

توجه داشته باشید که هر کدام از وضعیت های ناسازگار، $(*, -)$ ، $(-, *)$ ، و \neq همچنین به دو وضعیت مارکوف مجزا تقسیم می شوند که با زیروندهای 1 و 2 نشان داده می شود، و هدف آن بیان جزئیات وابسته به پروتکل است که به زودی شرح خواهیم داد. در شکل ۳، یک پیکان کم رنگ وضعیت اولیه زنجیره مارکوف را نشان می دهد، و وضعیت دو دایره ای $(-, -)$ وضعیت مارکوف جذب کننده می باشد.

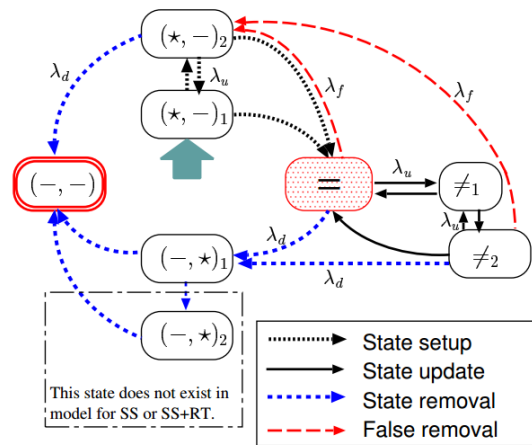
گذرهای بین وضعیت های مارکوف در شکل ۳ با شکل خطهای مختلف نشان داده شده اند که نشانگر وقایع مختلف (راه اندازی وضعیت، بروزرسانی وضعیت، حذف وضعیت و حذف غلط) می باشد که موجب گذرهای وضعیت می شوند. پارامترهای سیستمی بررسی شده در گذرهای وضعیتی عبارتند از:

λ_u : نرخ بروزرسانی وضعیت سیگنالینگ
 λ_d : $1/\lambda_u$ طول عمر میانگین وضعیت سیگنالینگ فرستنده است.

λ_f : نرخ حذف وضعیت نادرست در گیرنده
 D : تأخیر کانال سیگنالینگ

P_l : نرخ اتلاف (گم شدن) کانال سیگنالینگ
 بعلاوه، ما پارامترهای ویژه پروتکل زیر را داریم که قبلاً بحث شده است:

T : مقدار تایمر refresh مربوط به soft-state
 X : مقدار تایمر تایم اوت وضعیت مربوط به soft-state
 R : مقدار تایمر باز-انتقال پیام برای انتقال قابل اطمینان
 ما طول عمر وضعیت سیگنالینگ و فاصله زمانی بین بروزرسانی های وضعیت سیگنالینگ را به صورت متغیرهای تصادفی با توزیع نمایی (به ترتیب برابر $1/\lambda_u$ و $1/\lambda_d$) ، حذف نادرست را به صورت یک فرایند پواسون با نرخ λ_f ، و اتلاف های پیام را به صورت آزمایش های برنولی



شکل ۳: یک مدل زمان پیوسته مارکوف برای سیگنالینگ تک-پرسی

با در نظر گرفتن این وقایع در چرخه زندگی یک فرستنده و گیرنده سیگنالینگ، می توانیم یک مدل مارکوف طراحی کنیم، که در شکل ۳ نشان داده شده است، تا این رفتار را نشان دهیم. وضعیت های مدل مارکوف به صورت زیر تعریف می شود. هر وضعیت از یک زوج مقدار تشکیل شده است، (x_s, x_d) ، که در آن x_s و x_d به ترتیب اشاره به وضعیت های فرستنده و گیرنده سیگنالینگ دارند:

- وضعیت مارکوف $(*, -)$ وضعیت اولیه چرخه حیات را زمانی که وضعیت سیگنالینگ در فرستنده نصب شده است اما در گیرنده نصب نشده است نشان می دهد. این یک وضعیت ناسازگار است، چرا که مقادیر وضعیت فرستنده و گیرنده همخوانی ندارند.

- وضعیت های مارکوف $(-, *)$ متناظر با مواقعی است که فرستنده وضعیت را حذف کرده است، اما گیرنده وضعیت را حذف نکرده است. این وضعیت ها نیز ناسازگار می باشند.

- وقتی فرستنده و گیرنده وضعیت سیگنالینگ ناسازگاری داشته باشند، وضعیت زنجیره مارکوف برابر است با \neq .

- وقتی فرستنده و مقصد وضعیت سیگنالینگ مختلفی داشته باشند (هر دو وضعیت را نصب کرده باشند، اما مقادیر وضعیت مختلف باشند)، زنجیره مارکوف در وضعیت های \neq قرار دارد.

مستقل با پارامتر p_1 مدل‌سازی می‌کنیم. بعلاوه، ما فاصله زمانی refresh وضعیت soft-state، فاصله زمانی تایم اوت وضعیت، فاصله زمانی بازرسال و تأخیر کانال را به صورت متغیرهای تصادفی با توزیع نمایی به ترتیب با میانگین های T, X, R و D تقریب می‌زنیم.

در بخش ۲، ما پنج رویکرد مختلف در رابطه با سیگنالینگ مورد بحث قرار دادیم. هر کدام از این روش‌ها را می‌توان با استفاده از مدل نشان داده شده در شکل ۳، با نرخ های گذر مختلف (و در برخی موارد با گذرهای غیرفعال شده) برای هر کدام از روش‌ها مدل‌سازی نمود. ما در ادامه به توصیف گذرهای مدل برای هر کدام از این روش‌های سیگنالینگ مختلف می‌پردازیم. این گذرها در دیاگرام مدل و یا در جدول ۱ نشان داده شده‌اند.

مدل (SS) soft-state. وضعیت اولیه مدل، $(-, *)_1$ ، متناظر با ایجاد وضعیت سیگنالینگ جدید در فرستنده می‌باشد. همانطور که قبلاً بحث شد، این به ارسال یک پیام تریگر برای نصب وضعیت در گیرنده می‌انجامد. بعد از یک تأخیر کانال، یکی از دو رویداد اتفاق می‌افتد. اول، پیام تریگر می‌تواند با موفقیت به مقصد برسد. این رویداد با احتمال $(1-p_1)$ روی می‌دهد، و با گذر از وضعیت $(-, *)_1$ به وضعیت $(-, *)_2$ با نرخ $(1-p_1)/D$ مدل‌سازی می‌شود. دومین احتمال آن است که پیام تریگر گم شود. این اتفاق با احتمال p_1 روی می‌دهد، و با گذر از $(-, *)_1$ به $(-, *)_2$ با نرخ p_1/D نشان داده می‌شود. نهایتاً یک پیام refresh به مقصد خواهد رسید. از آنجا که refresh ها به صورت دوره ای با دوره تناوب T ارسال می‌شوند، و هر پیام با احتمال $(1-p_1)$ به مقصد می‌رسد، یک گذر از $(-, *)_2$ به وضعیت $(-, *)_1$ با نرخ $(1-p_1)/T$ وجود دارد. روند بروزرسانی مشابه روند راه اندازی است. وقتی وضعیت سازگار باشد، یعنی زنجیره مارکوف در وضعیت = باشد، بروزرسانی وضعیت باعث می‌شود زنجیره مارکوف از = به وضعیت $(-, *)_1$ با نرخ λ_H گذر کند. پیام تریگر با موفقیت به گیرنده با احتمال $(1-p_1)$ و تأخیر میانگین D می‌رسد، که متناظر است با یک گذر به عقب به = با نرخ

$(1-p_1)/D$. در زمان حضور در \neq_1 ، ائتلاف پیام تریگر باعث می‌شود زنجیره مارکوف به وضعیت \neq_2 با نرخ p_1/D_1 برود. با نرخ $(1-p_1)/T$ ، زنجیره مارکوف از وضعیت \neq_2 دوباره به وضعیت = گذر می‌کند. توجه داشته باشید که یک بروزرسانی همچنین می‌تواند زمانی روی دهد که سیستم در وضعیت $(-, *)_2$ یا وضعیت \neq_2 است، که باعث می‌شود زنجیره مارکوف به ترتیب به وضعیت های $(-, *)_1$ یا \neq_1 با نرخ λ_H گذر نماید. مدل ما وقایع در فرایند سیگنالینگ را سریالیزه می‌کند. برای مثال، مادامی که پیام تریگر در راه رسیدن به گیرنده است اجازه بروزرسانی وضعیت نمی‌دهد. ما فرض می‌کنیم که بروزرسانی یا قبل از ارسال پیام روی می‌دهد یا بعد از اینکه پیام به گیرنده رسیده باشد (یا گم شده باشد).

وضعیت سیگنالینگ فرستنده با نرخ λ_H حذف می‌شود، یعنی یک فرستنده دارای جلسه با طول میانگین $1/\lambda_H$ می‌باشد. اگر وضعیت سیگنالینگ در فرستنده حذف شود قبل از اینکه گیرنده وضعیت را به دست آورده باشد، زنجیره مارکوف به راحتی از $(-, *)_2$ به وضعیت جذب $(-, *)_1$ گذر می‌نماید. با این وجود، اگر گیرنده قبلاً اطلاعات وضعیت را به صورت سازگار یا ناسازگار نصب کرده باشد، یعنی سیستم در وضعیت \neq_2 یا وضعیت = باشد، زنجیره مارکوف به وضعیت $(-, *)_1$ گذر می‌کند. بعد از آن، گیرنده باید منتظر منقضی شدن تایمر تایم اوت وضعیت باشد تا وضعیت یتیم شده را حذف کند. ما این را بدین شکل مدل‌سازی می‌کنیم که اجازه می‌دهیم زنجیره مارکوف از وضعیت $(-, *)_1$ به وضعیت $(-, *)_2$ با نرخ $1/X$ گذر کند. توجه داشته باشید که مدل مارکوف برای SS شامل وضعیت $(-, *)_2$ در شکل ۳ نمی‌باشد.

در نهایت، وضعیت را می‌توان در مقصد به خاطر نبود پیام های refresh قبل از منقضی شدن تایمر تایم اوت وضعیت حذف نمود. این موضوع به وسیله یک گذر زنجیره مارکوف از وضعیت های = ، \neq_2 ، به وضعیت $(-, *)_2$ با نرخ λ_H مدل‌سازی می‌شود. از آنجا که چنین حذف غلطی تنها زمانی روی می‌دهد که همه پیام های refresh در مدت

سیستم منتظر می ماند تا تایمر تایم اوت وضعیت منقضی شود و یا یک بازارسال موفق پیام حذف به وضعیت $(-, -)$ برود. از این رو، نرخ گذر از وضعیت $(-, *)_2$ به وضعیت $(-, -)$ برابر است با $1/X + (1-p_1)/R$.

مدل **hard-state (HS)**: مدل HS مشابه مدل **SS+RTR** است، با این فرق که نرخ های گذر وابسته به پیام های refresh و تایمرهای تایم اوت وضعیت حذف می شوند. علاوه بر این، همانگونه که در بخش ۲ بحث شد، رویکرد HS باید بر سیگنالی خارجی اتکا کند تا از خطای ارسال کننده بازیابی کند. توضیح هزینه مربوطه چنین سیگنال خارجی دشوار است، چرا که وابسته به تدبیر زیربنایی است که تشخیص خطا برای رویکرد **hard-state** را انجام می دهد. برای مثال، یک مکانیزم حسگر لایه پیوند تشخیص خطا برای سیگنالینگ HS فراهم می آورد بدون اینکه سیگنال های اضافی وارد کند؛ در حالی که تشخیص خطای متکی بر یک مکانیزم معاوضه «ضربان قلب» ممکن است یک سربار کلی قابل قیاس با سربار **SS+RTR** داشته باشد. با این وجود، ما به بررسی تشخیص خطا به صورت یک جزء مجزا در معماری سیستم با مکانیزم سیگنالینگ می پردازیم. بنابراین، در مقاله خود این بخش از تحلیل را حذف می کنیم. ولی فرض ما بر این است که سیگنال خارجی را می توان به صورت نادرست با نرخ λ_{∞} تولید کرد، که سبب حذف ناقص یک وضعیت سیگنالینگ در روش HS می شود.

ما گذر وضعیت مبتنی بر پروتکل زنجیره مارکوف برای روش های سیگنالینگ مختلف را در جدول ۱ خلاصه کرده ایم که در آن λ_{ij} نرخ گذر وضعیت از وضعیت مارکوف i به j را نشان می دهد.

۲، ۱، ۳ حل مدل و محاسبات عملکرد

با استفاده از این مدل، می توانیم اکنون عملکرد روش های سیگنالینگ بحث شده در بخش ۲ را مورد مطالعه قرار دهیم. ما به مؤلفه های زیر علاقمند هستیم: نسبت ناسازگاری، δ ، تعریف شده به صورت کسری از زمان که فرستنده و

زمان تایمر تایم اوت وضعیت گم شده باشد، احتمال این رویداد را به صورت $p_1^{\lfloor X/T \rfloor}$ برآورد می کنیم. بنابراین، λ_f را می توان به صورت $\lambda_f = \frac{1}{X} p_1^{\lfloor X/T \rfloor}$ بیان نمود. توجه داشته باشید که این مدل یک گذر وضعیت از \neq_1 به $(-, *)_1$ را به خاطر ملاحظات توالی سازی بیان شده در بالا اجازه نمی دهد.

مدل soft-state با حذف صریح (SS+ER): به یاد داشته باشید که در **SS+ER**، پیام سیگنالینگ حاوی اطلاعات حذف وضعیت صریح (علاوه بر مکانیزم تایم اوت حالت) برای حذف وضعیت سیگنالینگ می باشد. ما این حذف صریح را با تغییر دادن فرایند حذف وضعیت در مدل **SS** به صورت زیر مدل سازی می کنیم. وقتی زنجیره مارکوف وارد وضعیت $(-, *)_1$ در نتیجه حذف وضعیت فرستنده می شود، یک پیام حذف وضعیت صریح آزاد می شود. با احتمال $(1-p_1)$ و بعد از یک تأخیر کانال، این پیام به مقصد رسیده و حذف وضعیت متناظر را تریگر می کند. برای مدل سازی این قضیه اجازه می دهیم زنجیره مارکوف از $(-, *)_1$ به وضعیت جذب کننده $(-, -)$ با نرخ $(1-p_1)/D$ برود. اتلاف پیام های حذف صریح باعث می شود زنجیره مارکوف از $(-, *)_1$ به $(-, *)_2$ گذر کند. از آنجا، سیستم به وضعیت جذب کننده $(-, -)$ با نرخ $1/X$ گذر می کند، که حذف وضعیت ایجاد شده در اثر منقضی شدن تایمر تایم اوت وضعیت را نشان می دهد.

مدل soft-state با پیام های تریگر قابل اطمینان (SS+RT): مدل مارکوف برای **SS+RT** با مدل **SS** این فرق را دارد که وقتی پیام تریگر حاوی اطلاعات راه اندازی/بروزرسانی وضعیت گم می شود، یا یک پیام refresh موفقیت آمیز یا یک بازارسال موفق پیام تریگر می تواند زنجیره مارکوف را از وضعیت \neq یا وضعیت $(-, *)_2$ به وضعیت $(-, -)$ با نرخ $(1-p_1).(1/T + 1/R)$ ببرد.

مدل soft-state با پیام تریگر/حذف قابل اطمینان (SS+RTR): فرق مدل مارکوف برای **SS+RTR** با مدل **SS+RT** آن است که وقتی یک پیام حذف صریح گم شود،

به وضعیت $(*, -)_2$ یا از \neq_1 به \neq_2 می رود. بنابراین نرخ پیام میانگین برای تریگر های صریح، γ_{et} عبارت است از:

$$\gamma_{et} = \pi_{(*,-)_1} \lambda_{(*,-)_1,=} + \pi_{\neq_1} \lambda_{\neq_1,=} + \pi_{(*,-)_1} \lambda_{(*,-)_1,(*,-)_2} + \pi_{\neq_1} \lambda_{\neq_1, \neq_2} \quad (3)$$

به طور مشابه، نرخ پیام میانگین برای حذف صریح، γ_{er} برابر است با

$$\gamma_{er} = \pi_{(-,*)_1} \lambda_{(-,*)_1,(-,-)} + \pi_{(-,*)_1} \lambda_{(-,*)_1,(-,*)_2} \quad (4)$$

پیام های refresh مربوط به soft-state با نرخ میانگین $1/T$ زمانی ایجاد می شوند که زنجیره مارکوف در وضعیت های $(*, -)_2$ ، $=$ ، یا \neq_2 باشد. بنابراین نرخ پیام میانگین برای پیام های refresh، γ_r ، را می توان به صورت زیر بیان نمود

$$\gamma_r = \frac{1}{T} \cdot (\pi_{(*,-)_2} + \pi_{=} + \pi_{\neq_2}) \quad (5)$$

اگر پیام های تریگر به شکل قابل اطمینان منتقل شوند، بازارسال ها با نرخ $1/R$ زمانی ایجاد خواهند شد که زنجیره در وضعیت های $(*, -)_2$ و \neq_2 باشد، و پیام های تأیید برای هر گذر به وضعیت $=$ تولید خواهد شد. بنابراین، نرخ پیام میانگین برای تریگر های مطمئن، γ_{rt} ، را می توان با استفاده از معادله زیر به دست آورد،

$$\gamma_{rt} = \frac{1}{R} (\pi_{(*,-)_2} + \pi_{\neq_2}) + \sum_i \pi_i \lambda_{i,=} + \lambda_f (\pi_{=} + \pi_{\neq_2}) \quad (6)$$

جمله سوم γ_{rt} در اثر حذف نادرست ایجاد می شود، چرا که یک برنامه تریگر قابل اطمینان نیازمند آن است که مقصد سیگنالینگ پیامی به فرستنده سیگنالینگ ارسال کرده و آن را از حذف آگاه سازد. به طور مشابه، برای حذف مطمئن، نرخ میانگین پیام γ_{rr} برابر است با:

$$\gamma_{rr} = \frac{1}{R} \pi_{(-,*)_2} + \pi_{(-,*)_1} \lambda_{(-,*)_1,(-,-)} + \pi_{(-,*)_2} \lambda_{(-,*)_2,(-,-)} \quad (7)$$

به طور خلاصه، نرخ پیام میانگین برای پروتکل های سیگنالینگ مختلف به صورت زیر است:

$$\begin{aligned} \text{SS} &: \gamma = \gamma_{et} + \gamma_r \\ \text{SS+ER} &: \gamma = \gamma_{et} + \gamma_r + \gamma_{er} \\ \text{SS+RT} &: \gamma = \gamma_{et} + \gamma_r + \gamma_{rt} \\ \text{SS+RTR} &: \gamma = \gamma_{et} + \gamma_r + \gamma_{er} + \gamma_{rt} + \gamma_{rr} \\ \text{HS} &: \gamma = \gamma_{et} + \gamma_{rt} + \gamma_{rr} + \gamma_{er} \end{aligned}$$

۳،۱،۳ ارزیابی مدل

ما اکنون به مقایسه عملکرد پنج روش سیگنالینگ مختلف با استفاده از چارچوب مدلینگ خود می پردازیم. برای استفاده

گیرنده سیگنالینگ مقادیر وضعیت سازگار ندارند؛ و نرخ پیام سیگنالینگ میانگین نرمال، γ^* ، تعریف شده به صورت $\gamma^* = \lambda_h \Gamma$ ، که در آن Γ تعداد کل پیام های سیگنالینگ مورد نیاز در طی طور عمر یک جلسه سیگنالینگ (از زمانی که وضعیت سیگنالینگ آغاز می شود تا زمانی که از سیستم حذف می شود)، و $1/\lambda_h$ طول عمر جلسه سیگنالینگ فرستنده است. از آنجا که طول عمر جلسه سیگنالینگ در گیرنده با روش سیگنالینگ تغییر می کند در حالی که $1/\lambda_h$ ثابت است، نرمالیزاسیون یک مقایسه منصفانه بین روش های سیگنالینگ فراهم می آورد.

برای حصول نسبت ناسازگاری، δ ، ما باید کسر زمانی را که سیستم خارج از وضعیت $(=)$ به سر می برد بدانیم، قبل از اینکه نهایتاً به وضعیت جذب کننده $(-,-)$ باز می گردد. این معادل ارزیابی مجموع احتمالات ثابت وضعیت های ناسازگار در مدل مارکوف بازگردنده می باشد که در آن وضعیت جذب $(-,-)$ و وضعیت آغازین $(*, -)_1$ آمیخته می شوند. π_i را احتمال ثابت مدل مارکوف بازگردنده در وضعیت i در نظر بگیرد. بیان زیر را برای δ داریم:

$$\delta = \frac{\pi_{(*,-)_1} + \pi_{(*,-)_2} + \pi_{\neq_1} + \pi_{\neq_2} + \pi_{(-,*)_1} + \pi_{(-,*)_2}}{1 - \pi_{=}} \quad (1)$$

برای به دست آوردن سربار کلی پیام سیگنالینگ، Γ ، می بایست طول عمر میانگین یک وضعیت سیگنالینگ، T ، و نرخ پیام سیگنالینگ میانگین γ را محاسبه کنیم:

$$\Gamma = T \cdot \gamma \quad (2)$$

در اینجا، T از محاسبه زمان میانگین تا جذب برای حالت $(*, -)_1$ در مدل مارکوف گذرا به دست می آید، و γ با در نظر گرفتن اینکه در کدام حالت و با چه نرخ از پیام های سیگنالینگ - پیام های تریگر و حذف صریح، پیام های refresh مربوط به soft-state، پیام های بازارسال و تأیید - در طی فرایند سیگنالینگ ایجاد می شوند به دست می آید. ما به صورت زیر پیش می رویم.

با یک پیام تریگر ارسال شده موفق، زنجیره مارکوف از وضعیت $(*, -)_1$ یا \neq_1 به وضعیت $=$ می رود، و اگر یک پیام تریگر گم شود، زنجیره مارکوف از وضعیت $(*, -)_1$

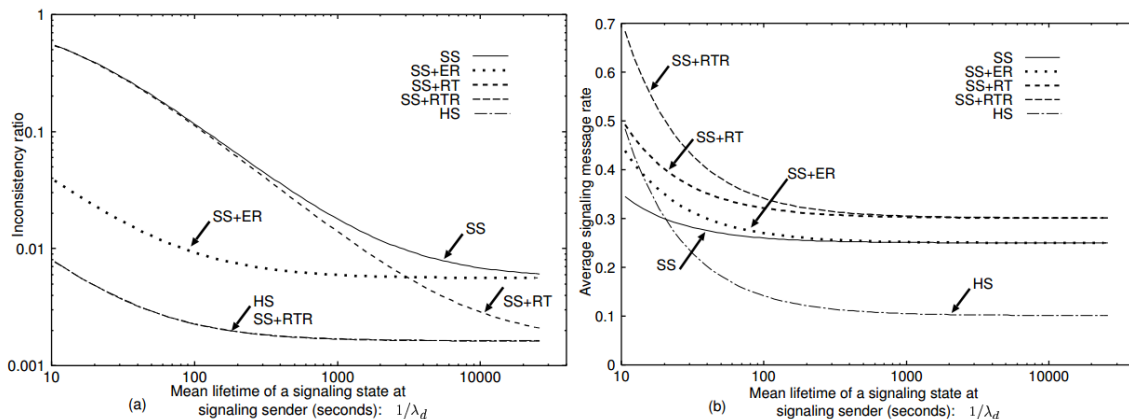
دایرکتوری اشتراک گذاشته اش دانلود شود). وقتی همتا از اپلیکیشن Kazaa خارج می شود، وضعیت همتا که توسط گره والد نگهداری می شود باید حذف شود. اگر این وضعیت در گره والد حذف نشود، یک وضعیت ناسازگار روی خواهد داد. در نتیجه، گره والد ممکن است به غلط به همتهای دیگر پاسخ دهد (مثلاً آنها را به یک همتهای جدا شده هدایت کند؛ این همتهای دیگر نیز آنگاه ممکن است به شکل بی فایده ای با همتهای جدا شده تماس برقرار کنند، که این کار کارایی اپلیکیشن را کاهش می دهد).

تأثیر طول جلسه ($1/\lambda_f$). ما ابتدا به مطالعه عملکرد روش های سیگنالینگ مختلف به صورت تابعی از مدت زمان مورد انتظار که وضعیت سیگنالینگ در فرستنده سیگنالینگ نصب است، ($1/\lambda_f$)، می پردازیم.

از مقادیر پارامتری نمونه، اجازه بدهید به عنوان مثال، فرایند سیگنالینگ بین یک همتهای معمولی Kazaa (از این به بعد برای سادگی آن را فقط همتا می خوانیم) و گره والد آن (طبق توصیف آمده در ابتدای بخش ۳.۱) را در نظر بگیریم. اگر طور دیگری قید نشده باشد، ما از پارامترهای پیشفرض زیر استفاده می کنیم: $p_l = 0.02$ ، $D = 30ms$ ، $1/\lambda_d = 1800s$ ، $1/\lambda_e = 20s$ ، $T = 5s$ ، $R = 4D$ ، $X = 3T$ ، $\lambda_w = 0.001$. این مقادیر پارامتری طوری انتخاب می شوند که رفتار یک جلسه Kazaa را نشان دهند: یک وضعیت سیگنالینگ زمانی اضافه می شود که همتا اپلیکیشن Kazaa را شروع کند، و زمانی بروزرسانی می شود که همتا کلکسیون خود از فایل های اشتراک گذاشته را تغییر دهد (مثلاً یک فایل جدید به

جدول ۱: گذرهای مدل

Transition rates	SS	SS+ER	SS+RT	SS+RTR	HS
$\lambda_{(*,-)_1,(*,-)_2}$ and $\lambda_{\neq 1,\neq 2}$	p_l/D	p_l/D	p_l/D	p_l/D	p_l/D
$\lambda_{(*,-)_1,=}$ and $\lambda_{\neq 1,=}$	$(1-p_l)/D$	$(1-p_l)/D$	$(1-p_l)/D$	$(1-p_l)/D$	$(1-p_l)/D$
$\lambda_{(*,-)_2,=}$ and $\lambda_{\neq 2,=}$	$(1-p_l)/T$	$(1-p_l)/T$	$(1/T + 1/R) \cdot (1-p_l)$	$(1/T + 1/R) \cdot (1-p_l)$	$(1-p_l)/R$
$\lambda_{(-,*)_1,(-,*)_2}$	—	p_l/D	—	p_l/D	p_l/D
$\lambda_{(-,*)_1,(-,-)}$	$1/X$	$(1-p_l)/D$	$1/X$	$(1-p_l)/D$	$(1-p_l)/D$
$\lambda_{(-,*)_2,(-,-)}$	—	$1/X$	—	$1/X + (1-p_l)/R$	$(1-p_l)/R$
λ_f	$p_l^{\lfloor X/T \rfloor}/X$	$p_l^{\lfloor X/T \rfloor}/X$	$p_l^{\lfloor X/T \rfloor}/X$	$p_l^{\lfloor X/T \rfloor}/X$	λ_w



شکل ۴: مقایسه با طول جلسه، $1/\lambda_f$

برای روش های سیگنالینگ مختلف را رسم می کنیم. شکل ۴ اطلاعاتی راجع به سیستم سیگنالینگ تک-پرشی ارائه می کند:

در مثال Kazaa ی ما، این متناظر با یک طول جلسه میانگین همتا است. در شکل ۴ (آ)، نسبت ناسازگاری δ را نشان می دهیم، و در شکل ۴ (ب)، نرخ پیام سیگنالینگ میانگین

- وقتی طول جلسه مورد انتظار افزایش می یابد، هم نسبت ناسازگاری و هم نرخ پیام سیگنالینگ میانگین برای همه روش ها کاهش می یابند. در مورد مثال Kazaa، این یعنی اگر Kazaa بیشتر توسط همتایانی استفاده شود که می خواهند خود را اندکی بعد از شروع خاموش کنند، به جای اینکه زمان طولانی روشن بمانند، (مثلاً همتاها از Kazaa به مدت ۵ دقیقه هر ساعت به جای هر روز دو ساعت استفاده کنند)، احتمال دارد سیستم متحمل سربار سیگنالینگ بیشتری شود، و گره های والد به درخواست ها براساس اطلاعات بیات پاسخ بدهند.

- در مقایسه SS+ER با SS متوجه می شویم که بهبود SS+ER نسبت به SS (با استفاده از نسبت ناسازگاری به عنوان معیار عملکرد) با کاهش طول جلسه میانگین معنادارتر می شود. حتی زمانی که طول جلسه میانگین دارای مرتبه چند هزار ثانیه است، مزیت اضافه کردن حذف صریح هنوز غیر قابل اغماض است. این موضوع به خاطر آن است که حذف وضعیت یتیم نیازمند یک وقفه نسبتاً طولانی است تا تایمر اوت در غیاب حذف صریح منقضی شود. مهم تر آنکه، با بررسی نرخ پیام میانگین در شکل ۴، در می یابیم که وقتی طول جلسه میانگین مرتبه چند هزار ثانیه داشته باشد، اضافه کردن حذف صریح سربار پیام سیگنالینگ قابل اغماضی در مقایسه با روش SS بوجود می آورد. در حالی که هزینه اضافه کردن این قابلیت پایین است، مدل ما نشان می دهد که بسیار مفید است که حذف صریح در سیگنالینگ soft-state در چنین مواردی لحاظ شود. علت این امر آن است که «جریمه» استفاده نکردن از حذف صریح خیلی بالاست.

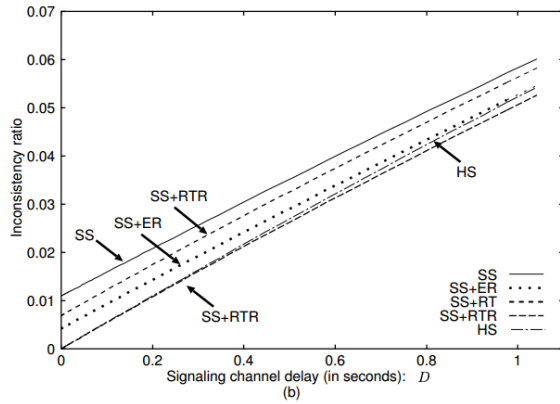
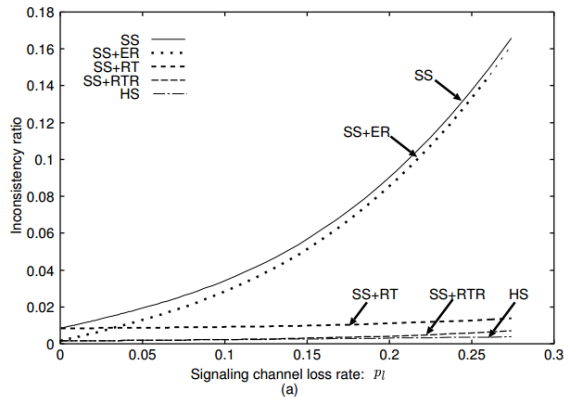
- شکل ۴ (آ) نشان می دهد که بهره عملکرد (از لحاظ کاهش نسبت ناسازگاری) به دست آمده با

مطرح کردن تریگر های قابل اطمینان زمانی معنادار می شود که طول جلسه میانگین همتاها زیاد باشد. شاهد این امر آن است که وقتی طول جلسه میانگین زیاد باشد، روش های پنج گانه براساس اینکه آیا تریگر های قابل اطمینان فراهم می کنند یا نه از هم تمایز داده می شوند. برعکس، وقتی طول جلسه میانگین کوتاه تر باشد (به سمت چپ شکل ۴ (آ)) روش های پنج گانه براساس چگونگی انجام حذف وضعیت گروه بندی می شوند: روش های بدون حذف صریح (SS)، روش های با حذف صریح (SS+RT) و روش های با حذف قابل اطمینان (SS+ER) و روش های با حذف قابل اطمینان (HS, SS+RTR). توجه داریم که برای جلسات طولانی، وقتی اختلاف در قابلیت اطمینان پیام تریگر مشخص تر باشد، نسبت ناسازگاری برای همه روش ها نسبتاً پایین است. همچنین، مطابق شکل ۴(ب)، مزیت محدود SS+RT نسبت به SS با سربار سیگنالینگ اضافی محسوسی همراه است. بنابراین، برای این پارامترهای کاربردی، ارائه پیام های تریگر ظاهراً خیلی ضروری نیست.

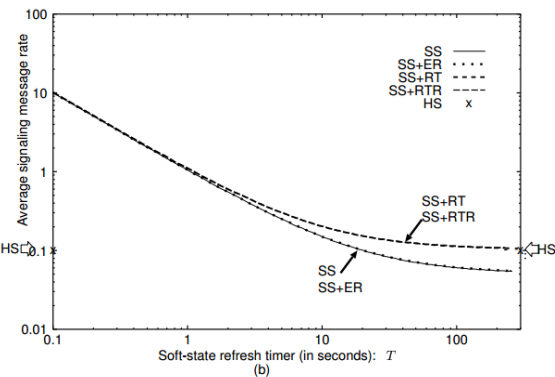
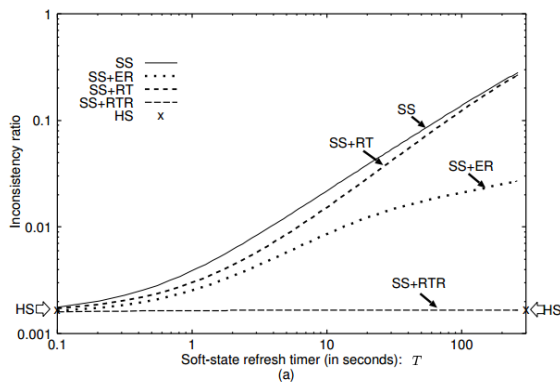
- SS+RTR اساساً همان نسبت ناسازگاری HS را دارد. این یعنی به جز حذف صریح و انتقال مطمئن، هرگونه بهبود برای مکانیزم refresh مربوط به soft-state می تواند تنها بهره های کم (یا صفر) در نسبت ناسازگاری فراهم کند. در واقع، در برخی موارد SS+RTR اندکی بهتر از HS عمل می کند.

تأثیر گم شدن و تأخیر پیام. شکل ۵ نسبت ناسازگاری برای روش های مختلف سیگنالینگ برای نرخ های مختلف اتلاف (آ) و تأخیرها (ب) را بر روی نمودار نشان می دهد. شکل ۵ (آ) نشان می دهد که برای نرخ های اتلاف پایین (مانند ۵ درصد)، انتقال قابل اطمینان به شکل قابل توجهی عملکرد پروتکل های soft-state را بهبود می بخشد. شکل ۵(ب) نرخ ناسازگاری در برابر تأخیر یک طرفه فرستنده به

متناسب با تأخیر کانال می باشد. از این رو، برای بازیابی از اتلاف، روش های با انتقال قابل اطمینان از تأخیرهای طولانی تری در یک محیط با تأخیرهای انتقال طولانی تر رنج می برند، در حالی که روش های *soft-state* که تنها متکی بر یک مکانیزم *refresh* هستند این مشکل را ندارند.



شکل 5: مقایسه در برابر نرخ اتلاف پیوند p_l و تأخیر پیوند D



شکل 6: مقایسه با تایمر *refresh* مربوط به *soft-state* (T)

(برای کاهش نسبت ناسازگاری) و یک تایمر *refresh* بلند (برای پایین نگه داشتن سربار پیام سیگنالینگ) نشان می دهد.

هزینه کلی. همانطور که قبلاً بحث شد، دو مؤلفه هزینه کلی وجود دارد: هزینه پیام سیگنالینگ، و هزینه های وابسته به اپلیکیشن ناشی از وضعیت ناسازگار در فرستنده و گیرنده. برای مثال، قبلاً دیدیم که برای IGMP، این هزینه دوم برابر انتقال داده های چندپخش ناخواسته بود؛ در مورد Kazaа، این هزینه دوم برابر سربار اضافی ایجاد شده توسط گره والد است که به همتها اشاره گرهایی به همتهای قبلاً جدا شده می دهد. برای ارزیابی هزینه های سربار سیگنالینگ و هزینه

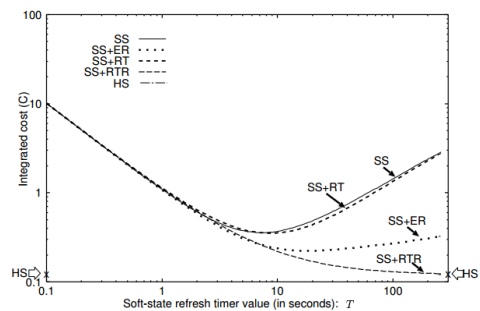
گیرنده را بر روی نمودار نشان می دهد. ما یک افزایش خطی تقریباً خطی در نسبت ناسازگاری تحت همه روش های سیگنالینگ مشاهده می کنیم. با این حال، روش های سیگنالینگ با انتقال قابل اطمینان یک شیب نسبتاً بزرگتر دارند. علت این امر آن است که مقدار تایمر بازارسال عموماً

تأثیر پیکربندی تایمر. سه تایمر مختلف وجود دارد که در روش های سیگنالینگ پنجگانه مورد بررسی ما استفاده شده است: تایمر *refresh* مربوط به *soft-state*، تایمر تایم اوت وضعیت *soft-state* و تایمر بازارسال. شکل 6 به بررسی عملکرد روش های سیگنالینگ *soft-state* تحت تنظیمات مختلف تایمر *refresh* مربوط به *soft-state* می پردازد. از آنجا که HS از یک مکانیزم *refresh* استفاده نمی کند، به صورت یک 'x' روی محور y نشان داده می شود. وقتی زمان تایمر *refresh* عوض می شود، ما تایمر تایم اوت وضعیت را 3 برابر مقدار تایمر *refresh* قرار می دهیم. شکل 6 یک موازنه جالب بین داشتن یک تایمر *refresh* کوتاه

های وابسته به اپلیکیشن ناشی از ناسازگاری وضعیت، ما یک هزینه یکپارچه (C) به صورت زیر تعریف می کنیم

$$C = c \cdot \delta + \gamma^* \quad (8)$$

که در آن c نشانگر وزن نسبی هزینه وابسته به اپلیکیشن ناشی از وضعیت ناسازگار سیگنالینگ می باشد. در Kazaa، برای مثال، c را می توان به صورت تعداد پیام های سیگنالینگ وابسته به درخواست های بی نتیجه تفسیر کرد که در اثر وضعیت اشتراک فایل ناسازگار در گره والد ایجاد می شوند. در بخش بعدی، ما c را برابر ۱۰ (پیام بر ثانیه) قرار می دهیم.



شکل ۷: تایمر refresh مربوط به soft-state (T)

در شکل ۷، هزینه یکپارچه مربوط به روش های سیگنالینگ مختلف در برابر مقدار تایمر refresh متعلق به soft-state، T، را روی نمودار نشان می دهیم. از این شکل، ملاحظه می کنیم که نقاط عملیاتی نسبتاً حساسی برای SS و SS+RT وجود دارد، که بالای آن هزینه ناسازگاری به میزان زیادی افزایش می یابد و پایین آن هزینه سیگنالینگ پیام بسیار افزایش می یابد. چنین نقطه عملیاتی بهینه ای همچنین برای SS+ER وجود دارد، هرچند هزینه یکپارچه خیلی به مقادیر تایمر refresh نسبتاً طولانی تر حساس نمی باشد. در نهایت، برای SS+RTR، یک مقدار تایمر طولانی تر ترجیح داده می شود، و وقتی تایمر به اندازه کافی بزرگ باشد (با مرتبه ۱۰۰ ها ثانیه)، عملکرد قابل قیاسی نسبت به روش hard-state ارائه می کند.

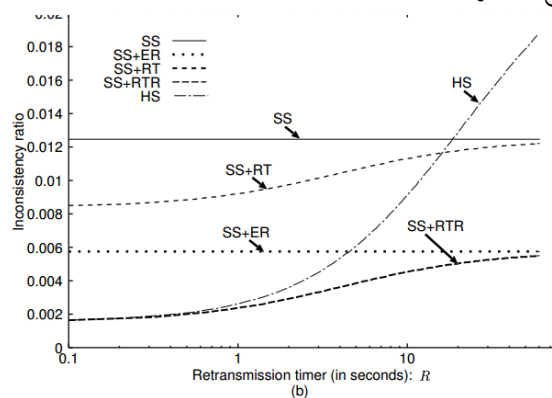
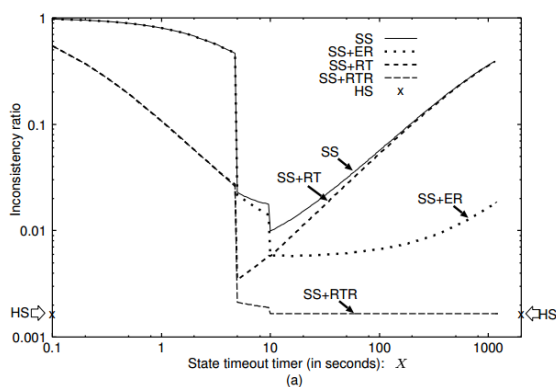
شکل ۸ (آ) به بررسی تأثیر مقادیر تایمر تایم اوت وضعیت مختلف بر روی نسبت ناسازگاری روش های soft-state می پردازد. در اینجا ما تایمر refresh وضعیت را برابر ۵ ثانیه قرار داده و تایمر تایم اوت وضعیت را تغییر می دهیم.

نتایج نشان می دهد وقتی که تایمر تایم اوت وضعیت کوچکتر از تایمر refresh باشد، همه روش های مبتنی بر soft-state عملکرد ضعیفی خواهند داشت، چرا که پیام های refresh آنقدر دیر می رسند که نمی توانند وضعیت سیگنالینگ در گیرنده سیگنالینگ را زنده نگه دارند. وقتی مقدار تایمر تایم اوت وضعیت بزرگتر از مقدار تایمر refresh باشد، روش های مختلف بسیار متفاوت از هم عمل می کنند: SS+RTR به خوبی با مقادیر تایم اوت طولانی کار می کند، چون هر چه تایمر تایم اوت طولانی تر باشد، کمتر احتمال دارد که یک وضعیت به صورت اشتباهی به خاطر گم شدن پیام های refresh حذف گردد. SS و SS+ER زمانی به بهترین شکل عمل می کنند که تایمر تایم اوت وضعیت تقریباً دو برابر مدت طول تایمر refresh باشد، به طوری که احتمال حذف نادرست کاهش یابد. با این حال، از آنجا که تایم های تایم اوت طولانی تر تأخیرهای بزرگتری به حذف وضعیت یتیم شده اضافه می کنند، SS و SS+ER نیز نیاز دارند که تایمر تایم اوت وضعیت به اندازه کافی کوتاه باشد تا از چنین مسائلی جلوگیری گردد. به خاطر بیاورید که SS+RT از یک مکانیزم اعلان استفاده می کند که در آن گیرنده سیگنالینگ فرستنده سیگنالینگ را راجع به حذف های وضعیت آگاه می کند و فرستنده سیگنالینگ از یک حذف نادرست به وسیله ارسال یک پیام تریگر دیگر بازایی می کند. از آنجا که SS+RT بیشتر به فرایند حذف وضعیت یتیم شده حساس است و مکانیزم اعلان آن جریمه حذف نادرست را کاهش می دهد، در بهترین حالت با یک مقدار تایمر تایم اوت کار می کند که فقط اندکی بزرگتر از مقدار متناظر تایمر refresh وضعیت باشد.

شکل ۸ (ب) تأثیر مقادیر تایمر انتقال مختلف بر روی نسبت ناسازگاری روش های سیگنالینگ پنج گانه را مورد بررسی قرار می دهد. از آنجا که HS تنها وابسته به انتقال های قابل اطمینان صریح برای راه اندازی/بروزرسانی/حذف وضعیت است، بیشتر به تغییرات در تایمر انتقال R حساس می باشد.

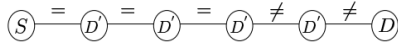
در حالی که مدل ما مقادیر تایمر با توزیع فضایی را فرض می کند، در عمل، پروتکل های سیگنالینگ معمولاً از تایمرهای قطعی استفاده می کنند. برای بررسی تأثیر فرض نمایی ما بر روی مقادیر تایمر، ما شبیه سازی هایی ساختیم که از تایمرهای قطعی تحت تنظیمات سیستمی یکسان استفاده می کنند. نتایج شبیه سازی ما نشان می دهد که استفاده از تایمرهای قطعی تأثیری بر مشاهدات و نتیجه گیری های ما ندارد. برای مثال، در مقایسه با نتایج ارزیابی قبلی ما که در شکل ۴ نشان داده شده است، نسبت ناسازگاری اندکی (کمتر از ۱ درصد) بین نتایج تحلیل و نتایج شبیه سازی با تایمرهای قطعی متفاوت است. برای نرخ پیام سیگنالینگ میانگین، اختلاف بین نتایج تحلیلی و نتایج شبیه سازی بین ۵ تا ۱۵ درصد است، در حالی که رفتار نسبی کیفی در بین پروتکل های سیگنالینگ مختلف بدون تغییر باقی می ماند. برای جزئیات بیشتر به [۱۱] مراجعه کنید.

موازنه بین نسبت ناسازگاری و نرخ میانگین پیام سیگنالینگ. با تغییر دادن تایمر refresh متعلق به soft-state، می توان موازنه ای بین نسبت ناسازگاری و نرخ پیام سیگنالینگ میانگین روش های سیگنالینگ مختلف برقرار نمود. از آنجا که سیگنالینگ محض از تایمر refresh استفاده نمی کند، نه نسبت ناسازگاری و نه نرخ سیگنالینگ میانگین هیچ یک با T تغییر نمی کنند؛ در شکل ۹، hard-state به صورت یک نقطه تک 'x' نشان داده می شود. شکل ۹ همچنین نشان می دهد که نسبت ناسازگاری SS+RTR به نرخ refresh متعلق به soft-state (که با تایمر refresh تعیین می شود) حساس نیست، در حالی که نسبت ناسازگاری دیگر روش های soft-state با سربار سیگنالینگ تغییر می کنند. ما همچنین موازنه هایی بین نسبت ناسازگاری و سربار سیگنالینگ براساس دیگر پارامترهای سیستمی یا طراحی (مانند تأخیر کانال سیگنالینگ D، و غیره) مورد بررسی قرار دادیم. به خاطر محدودیت های فضایی، ما این نتایج را حذف می کنیم. خوانندگان علاقه مند می توانند تحلیل ها و نتایج متناظر را در [۱۱] بیابند.



شکل ۹: تایمر نام اوت وضعیت (X) برای روش های مبتنی بر soft-state و تایمر بازارسال R برای انتقالهای قابل اطمینان

می‌گیریم ($\lambda_i \rightarrow \infty$). ما بروزسانی‌های وضعیت را به صورت فرایند پواسون با نرخ λ_i در نظر می‌گیریم.

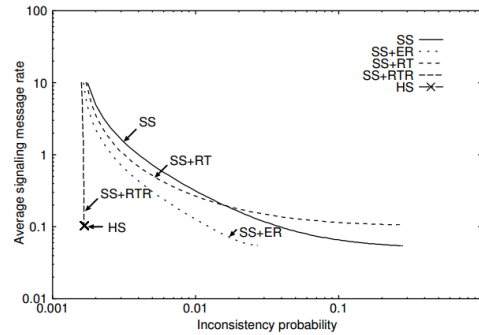


شکل ۱۰: یک سیستم پنج-پرسی با سه پرس سازگار

چارچوب مدل‌سازی ما برای یک سیستم چندپرسی یک افزونه از مدل چندپرسی می‌باشد. N را تعداد کلی پرس‌ها (پیوند‌ها) در یک سیستم چندپرسی و n را تعداد پرس‌های سازگار (پیوند‌ها) در نظر بگیرید. فرض می‌کنیم که این پرس‌ها همگن هستند، یعنی نرخ اتلاف کانال (p_i) همانند و تأخیر کانال میانگین دارند، و فرض می‌کنیم که اتلاف‌های کانال مستقل می‌باشند. در اینجا، یک پرس سازگار را پرس (پیوندی) در نظر می‌گیریم که دارای دو سر با اطلاعات وضعیت سازگار می‌باشد. شکل ۱۰ یک سیستم پنج پرس ($N=5$) با سه پرس سازگار ($n=3$) نشان می‌دهد. ما فضای وضعیت را به صورت $S = \{(n, s)\}$ نشان می‌دهیم، که در آن $0 \leq n \leq N$ تعداد پرس‌های سازگار، و s یک متغیر ویژه است که مقادیر ۰ یا ۱ را می‌گیرد تا نشان دهد آیا زنجیره مارکوف در یک وضعیت مارکوف مسیر سریع ($s=0$) است یا یک وضعیت مارکوف مسیر کند ($s=1$). فرق بین وضعیت مارکوف مسیر سریع و وضعیت مارکوف مسیر کند بعداً روشن خواهد شد. علاوه بر وضعیت‌های (n, s) ، یک وضعیت خاص برای سیگنالینگ HS، Δ ، وجود دارد که زمانی را مدل‌سازی می‌کند که سیستم HS از یک حذف غلط بازیابی می‌کند.

ما در ادامه به شرح گذرهای مدل برای پروتکل soft-state محض (SS)، پروتکل soft-state بهبود یافته با انتقال پرس-به-پرس قابل اطمینان (SS+RT)، و پروتکل hard-state (HS) می‌پردازیم.

مدلسازی گذرهای پروتکل SS تحت SS، بروزسانی‌های وضعیت به وسیله پیام‌های تریگر ارسال شده به گیرنده (های) سیگنالینگ حمل می‌شوند. اگر یک پیام تریگر گم شود، یک پیام refresh حاوی اطلاعات یکسان نهایتاً به گیرنده سیگنالینگ خواهد رسید و وضعیت گیرنده را



شکل ۹: موازنه بین نسبت ناسازگاری و نرخ پیام سیگنالینگ میانگین، به دست آمده با تغییر دادن T

۳,۲ سیگنالینگ در یک سیستم چندپرسی

در این زیربخش، ما موردی را بررسی می‌کنیم که در آن نه تنها سیستم‌های انتهایی (فرستنده و گیرنده سیگنالینگ) بلکه مسیریاب‌های میانی و گره‌های رله نیز باید وضعیت سیگنالینگ را حفظ کنند. این برای مثال در مورد مسیریاب‌های میانی نیاز ندارد برای ارتباط بین سیستم‌های انتهایی عرض باند رزرو کند. در مدل انتزاعی ما از یک سیستم سیگنالینگ چندپرسی، یک فرستنده سیگنالینگ تکی و یک زنجیره از مقصدهای سیگنالینگ وجود دارد. ما الزام می‌کنیم که وضعیت سیگنالینگ ایجاد شده در فرستنده سیگنالینگ در همه گره‌ها در راستای مسیر بین فرستنده سیگنالینگ و مسیر نهایی حفظ شود.

۳,۲,۱ مدل سیستم‌های چندپرسی

در بخش ۳,۱، ما فاکتورهای مختلفی را شناسایی کردیم که بر عملکرد پروتکل‌های سیگنالینگ در حالت چندپرسی تأثیر می‌گذارند. بسیاری از نتایج مستقیماً در اینجا نیز قابل استفاده می‌باشند. در این بخش، ما بر روی موضوعات منحصر به فردی تمرکز می‌کنیم که در مورد پرس‌های چندگانه مطرح می‌شوند. ما بر روی فرایند ثابت در یک سیستم سیگنالینگ چندپرسی تمرکز می‌کنیم، که در آن یک وضعیت در فرستنده سیگنالینگ بروزسانی می‌شود و تغییرات باید به همه گیرندگان انتشار یابد. برای ساده کردن مدل خود، ما طول عمر یک وضعیت را بی‌نهایت در نظر

شوند. فرض کنید که پرش j اولین گرش در زنجیره باشد که در آن یک تایم اوت وضعیت روی می دهد، (در این حالت، تایمر وضعیت های متناظر در $j+1$ امین تا N -امین پرش نیز منقضی خواهد شد). وقتی این روی می دهد، اجازه می دهیم زنجیره مارکوف به وضعیت $(j, 1)$ با نرخ $\lambda_{(i,1),(j,1)}^{(e)}$ گذر کند. نرخ گذر $\lambda_{(i,1),(j,1)}^{(e)}$ را می توان با معادله (۹) محاسبه کرد.

$$\lambda_{(i,1),(j,1)} = \begin{cases} \frac{1}{X} \cdot [1 - (1 - p_i)^{j+1}]^{\lfloor \frac{X}{T} \rfloor} - \frac{1}{X} \cdot [1 - (1 - p_i)^j]^{\lfloor \frac{X}{T} \rfloor}, & 1 \leq j < i \leq N \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

عبارت $(1 - (1 - p_i)^{j+1})^{\lfloor \frac{X}{T} \rfloor} - (1 - (1 - p_i)^j)^{\lfloor \frac{X}{T} \rfloor}$ احتمال اینکه تایم اوت در $j+1$ امین گیرنده سیگنالینگ، ولی نه در هر پرش قبلی، روی دهد را تقریب می زند.

گذرهای مدل برای پروتکل SS+RT. تحت پروتکل SS+RT، وقتی سیستم در یک وضعیت مسیر کند گیر کرده باشد، بازارسال موفق پیام تریگر و پیام refresh موفق می تواند پرش متناظر را سازگار سازد. این موضوع به خاطر آن است که ارسال قابل اطمینان به کار می رود. بنابراین، در SS+RT، نرخ ارسال از وضعیت $(i-1, 1)$ به وضعیت $(i, 0)$ برابر می شود با

$$\lambda_{(i-1,1),(i,0)} = \frac{1}{T}(1 - p_i)^i + \frac{1}{R}(1 - p_i). \quad (10)$$

گذرهای مدل برای پروتکل hard-state (HS). در HS، پیام های تریگر قابل اطمینان (با انتشار مطمئن پرش به پرش) برای بروزرسانی وضعیت در راستای مسیر سیگنالینگ به کار می روند. نه پیام های refresh و نه حذف تایم اوت soft-state هیچ کدام استفاده نمی شوند. بنابراین، یک گذر وضعیت از وضعیت $(i-1, 1)$ به وضعیت $(i, 0)$ تنها به واسطه بازارسال به دست می آید، و نرخ گذر برابر است با

$$\lambda_{(i-1,1),(i,0)} = \frac{1}{R}(1 - p_i). \quad (11)$$

همانند مورد سیستم تک پرشی، حذف های غلط را به عنوان یک فرایند پواسون با نرخ λ_{∞} در هر گیرنده مدلسازی می کنیم. بنابراین، سیستم از یک وضعیت مسیر کند به وضعیت ریکاوری (Δ) با نرخ $N \lambda_{\infty}$ می رود. همانگونه که در

سازگار خواهد ساخت. یک وضعیت در صورتی حذف می شود که تایمر تایم اوت به خاطر اتلاف های همه پیام های refresh فرستاده شده در طی بازه تایم اوت منقضی شود. توجه داشته باشید که اگر یک تایم اوت در یک گره روی دهد، همه گیرندگان به جز این گره در توپولوژی خطی نیز تایم اوت خواهند شد، زیرا آنها نیز پیام های refresh را دریافت نخواهند کرد. ما از مدل مارکوف در شکل ۱۱ برای مدلسازی پروتکل SS در محیط چندپرشی استفاده می کنیم. فرایند به روزرسانی وضعیت را در نظر بگیرید. وقتی وضعیت سیگنالینگ در فرستنده سیگنالینگ بروزرسانی می شود، زنجیره مارکوف به وضعیت $(0, 0)$ از دیگر وضعیت های مارکوف گذر می کند. وضعیت $(0, 0)$ نشانگر حالتی است که در آن هیچ پرشی سازگار نیست ($n=0$) و یک پیام تریگر در راه پرش بعدی است (مدل در وضعیت مسیر موسوم به سریع، $s=0$ می باشد).

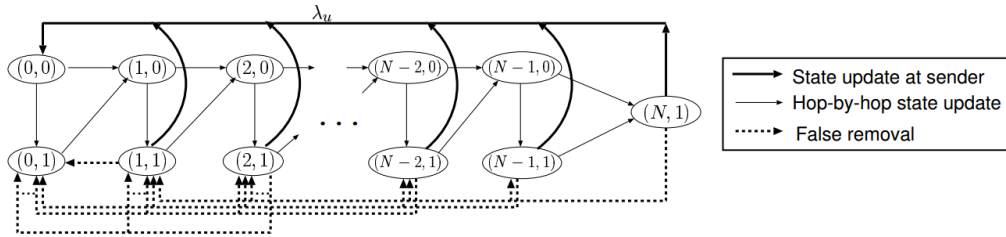
بعد از یک تأخیر کانال یک-پرشی، دو اتفاق ممکن است برای پیام تریگر بیفتد. ابتدا، ممکن است با موفقیت به گیرنده بعدی برسد. بنابراین، سیستم از $(0, 0)$ به $(1, 0)$ ، یا عموماً از وضعیت $(i, 0)$ به $(i+1, 0)$ با نرخ $(1 - p_i) / D$ گذر می کند. دومین احتمال آن است که تریگر گم شود، که در این حالت مدل از $(0, 0)$ به $(0, 1)$ ، یا به شکل عمومی تر از وضعیت $(i, 0)$ به وضعیت $(i, 1)$ ، با نرخ p_i / D گذر می کند و منتظر یک پیام refresh بعدی می شود (یعنی مدل در یک وضعیت مسیر کند، $s=1$ است).

از آنجا که فرض می کنیم بازه های refresh با میانگین T توزیع نمایی دارند، و احتمال آنکه یک پیام refresh تولید شده در فرستنده به $i-1$ امین پرش برسد برابر $(1 - p_i)^i$ می باشد، نرخ گذر از وضعیت $(i-1, 1)$ به وضعیت $(i, 0)$ برابر $(1 - p_i)^i / T$ است. در نهایت، سیستم می تواند به وضعیت $(N, 1)$ از طریق وضعیت مسیر سریع $(N-1, 0)$ یا به وسیله وضعیت مسیر کند $(N-1, 1)$ گذر نماید.

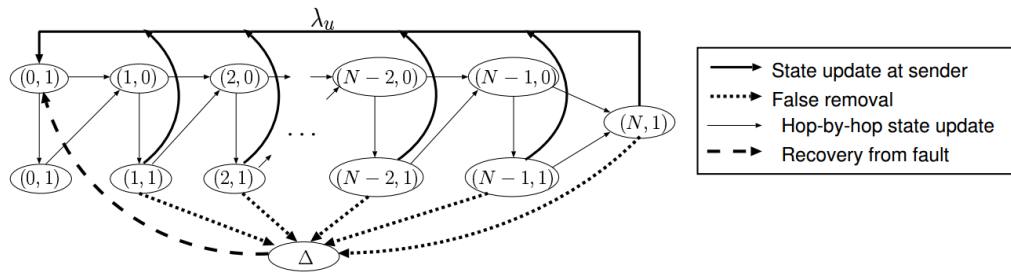
بعلاوه، تایمهای تایم اوت وضعیت در مقاصد سیگنالینگ ممکن است به خاطر تلف شدن پیام های refresh منقضی

بخش ۲ بحث گردید، گیرنده در هنگام وقوع یک خرابی (مانند خرابی پیوند) توسط یک سیگنال خارجی آگاه می گردد. این گیرنده سپس پیام هایی می فرستد تا دیگر گیرندگان و فرستنده را از خرابی آگاه سازد. در هنگام دریافت چنین پیامی، دیگر گیرنده ها وضعیت (های) مربوطه خود را حذف می کنند. اگر گیرنده چنین پیامی دریافت کند، یک پیام سیگنالینگ تریگر می فرستد تا

وضعیت را دوباره نصب کند. ما این را به نحوی مدل سازی می کنیم که به سیستم اجازه می دهیم از وضعیت Δ به وضعیت $(0,0)$ با نرخ $\frac{2}{ND}$ برود، تخمینی که تأخیر مورد انتظار فرستنده برای راه اندازی فرایند بازیابی را نشان می دهد. جزئیات بیشتر مدل چندپرسی را می توان در [۱۱] یافت.

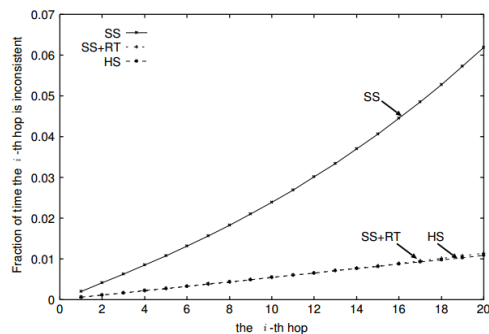


شکل ۱۱: مدل مارکوف برای روش آنها به انتهای soft-state



شکل ۱۲: مدل مارکوف برای روش سیگنالینگ hard-state

عملکرد تمرکز می کنیم. $\lambda_{\omega} = p_l^3$. ما به بررسی تأثیر پرسش های چندگانه بر روی



شکل ۱۳: کسری از زمان که i -امین پرسش ناسازگار است و در آن $1 \leq i \leq N$ و $N=20$

در شکل ۱۳، کسری از زمان که i -امین گره ناسازگار است به نمودار در می آوریم که در آن تعداد کل پرسش ها ۲۰ است. مشاهده می کنیم که با دور شدن پرسش از فرستنده

۳،۲،۲ راه حل و نتایج مدل چند-پرسی
 راه حل مدل چندپرسی مشابه راه حل مدل تک پرسشی است. به خاطر محدودیت های فضایی، ما جزئیات حل را حذف می کنیم؛ خوانندگان علاقمند می توانند به [۱۱] مراجعه نمایند. در عوض بر روی خود نتایج تمرکز می کنیم، و به بررسی نسبت ناسازگاری و سربار پیام سیگنالینگ سه روش سیگنالینگ چند-پرسی می پردازیم. در انتخاب پارامترهای مدل، ما فرآیند رزرو عرض باند روی یک مسیر چندپرسی را به عنوان مثال در نظر می گیریم. اگر طور دیگری مشخص نشده باشد، ما از پارامترهای پیشفرض زیر استفاده می کنیم: $N=20$ ، $p_l = 0.0$ و $D=30ms$ در هر پرسش، $1/\lambda_{\omega} = 60s$ ، $T = 5s$ ، $X=3T$ ، $R=4D$ و

دهد که ناسازگاری روش soft-state محض (SS) نسبت به افزایش در تعداد پرش ها حساس تر است. شکل ۱۴(ب) نشان می دهد که اضافه کردن تریگر به روش soft-state انتها به انتها، و علیرغم بهبود سازگاری، مقداری سربار سیگنالینگ وارد می کند. این مزیت با افزایش تعداد پرش های سیگنالینگ بالا می رود.

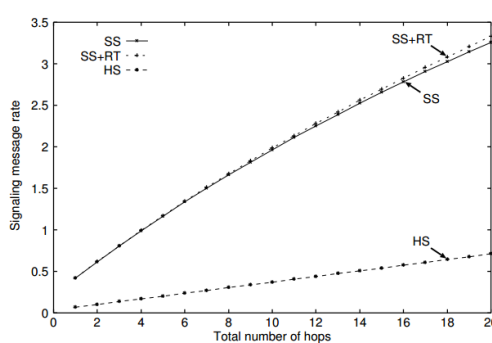
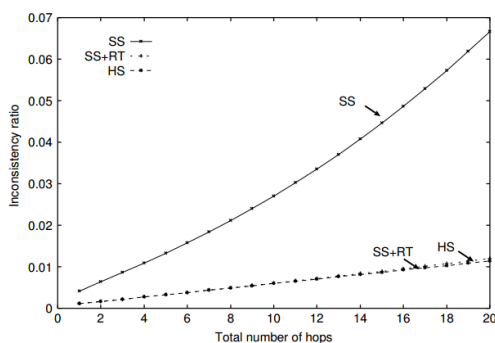
ما همچنین به ارزیابی تأثیر دیگر پارامترها بر روی عملکرد روش های سیگنالینگ چندپرسی پرداختیم. از آنجا که نتایج مشابه نتایج ارزیابی مدل تک-پرسی می باشند، آنها را در اینجا نمی آوریم. برای جزئیات به [۱۱] نگاه کنید.

۴. پژوهش های مربوطه

مرتبط ترین پژوهش به تحقیق جاری ما [۱۶] است، که اولین تلاش برای توسعه مدل های تحلیلی از پروتکل های soft-state می باشد، و همین طور اولین تلاشی است که برای ایجاد درکی اصولی تر از پروتکل های soft-state کوشید. مدل در [۱۶] به بررسی اتلاف لینک و احتمال حذف وضعیت پرداخت، و معیار ناسازگاری را معرفی کرد که ما در اینجا استفاده کرده ایم. چند تفاوت مهم بین کار ما و [۱۶] وجود دارد. دو پروتکل بررسی شده در [۱۶] تناظر

سیگنالینگ ناسازگاری وابسته به یک پرش بر روی مسیر سیگنالینگ افزایش می یابد. این ناسازگاری افزایش یافته در روند تقریباً خطی برای همه روش های سیگنالینگ نشان می دهد. ترکیب تریگرهای قابل اطمینان پرش به پرش با روش soft-state انتها به انتها به شکل قابل توجهی سازگاری را در همه پرش های سیگنالینگ بهبود می بخشد، و در آن سازگاری SS+RT قابل قیاس با سازگاری روش hard-state است. در ارزیابی ما، سازگاری روش hard-state (HS) اندکی بالاتر از SS+RT است. این به خاطر تأثیر حذف نادرست در هنگام انقضای تایمر اوت در روش SS+RT است، و از آنجا که پیام های refresh مربوط به soft-state تنها از سوی فرستنده تولید می شوند، تایم اوت وضعیت با احتمال بیشتری در گیرنده های دور (پرش های بیشتر) از فرستنده روی می دهد.

شکل ۱۴ هم نرخ ناسازگاری (در سمت چپ) و هم نرخ پیام سیگنالینگ (در سمت راست) را به صورت تابعی از تعداد پرش ها در سیستم ندرپرسی نشان می دهد. مشاهده می کنیم که ناسازگاری و سربار پیام سیگنالینگ به شکل یکنواختی با افزایش تعداد پرش ها بالا می روند. شکل ۱۴، مقایسه روش hard-state (HS) و روش تریگر قابل اطمینان (SS+RT)، نشان می



شکل ۱۴: نسبت ناسازگاری و نرخ پیام سیگنالینگ در برابر تعداد کل پرش ها

عمومی تر هدف ما فقط این نیست که پروتکل های soft-state را بفهمیم بلکه آن است انواع روش های سیگنالینگ و مکانیزم های آنها را، از soft state محض گرفته تا روش های soft-state همراه با حذف وضعیت صریح و/یا

نزدیکی با پروتکل های SS و SS+RT ما دارند. در اینجا ما به بررسی بازه وسیع تری از پروتکل ها می پردازیم، از جمله پروتکل هایی که از ویژگی های hard-state استفاده می کنند (از جمله پروتکل های SS+ER و HS). به بیان

سیگنالینگ قابل اطمینان، تا روش `hard-state` محض مقایسه نماییم.

مدل ما همچنین مفصل تر است که به ما امکان می دهد به اندازه گیری تأثیر عملکرد تعدادی از پارامترهای مهم سیستم و پروتکل پردازیم. مدل ما خود، با استفاده از یک وضعیت جذب کننده، از یک نتیجه نسبتاً ضدشهودی در [۱۶] اجتناب می کند - یعنی با بالا رفتن ظرفیت کانال سیگنالینگ درجه ناسازگاری نیز افزایش می یابد. در نهایت، و شاید مهمتر از همه، در حالی که [۱۶] تنها راه اندازی وضعیت را در نظر می گیرد، ما به بررسی نصب وضعیت، بروزرسانی وضعیت، و حذف وضعیت به عنوان بخش های لاینفک یک طیف از پروتکل های سیگنالینگ می پردازیم.

دو تحقیق که به جنبه های ریزتر عملکرد پروتکل `soft-state` عبارتند از [۱۷]، که تکنیک هایی برای مقادیر پویای مقادیر تایمر `soft-state` بررسی کرد، و [۱۳]، که طرحی برای استفاده از تایمرهای مختلف `soft-state` برای پیام های تریگر و `refresh` مورد بررسی قرار داد.

۵. نتیجه گیری و پژوهش های آتی

در این مقاله، ما به مقایسه عملکرد انواع پروتکل های سیگنالینگ از روش `soft-state` محض گرفته تا روش های `soft-state` همراه با حذف وضعیت صریح و/یا سیگنالینگ قابل اطمینان، تا روش `hard-state` محض پرداختیم. هدف ما با این کار آن نبود که بحث کنیم آیا روش `hard-state` به شکلی مطلق بهتر است یا روش `soft-state`. بلکه، با توجه به این موضوع که پروتکل هایی که در یک دسته قرار می گیرند از مکانیزم هایی استفاده می کنند که معمولاً مرتبط با دیگری است، تلاش کردیم بفهمیم چگونه مکانیزم هایی که تا حد لحاظ شدن در پروتکل های مختلف سیگنالینگ `hard-state` و `soft-state` رشد کرده اند، می توانند به بهترین حالت در موقعیت های بخصوص استفاده شوند و چرا. ما دسته ای از پروتکل های عمومی تعریف کردیم که در نقاط مختلف در راستای طیف `hard-state/soft-state` قرار دارند و یک مدل تحلیلی پارامتری تعریف نمودیم که

به ما اجازه می دهد عملکرد آنها را مورد مطالعه قرار دهیم. در واقع، این که یک مدل می تواند بازه ای از پروتکل های سیگنالینگ را نشان دهد (از `soft-state` محض گرفته تا `hard-state`، با انواعی در این میان) نشان می دهد که پروتکل ها آن قدرها که در ابتدا ممکن است به نظر بیاید متفاوت نیستند. نتایج ما نشان می دهد که در میان گروه روش های `soft-state`، یک روش `soft-state` همراه با حذف صریح به میزان زیادی سازگاری وضعیت را بهبود می بخشد، و در عین حال سربار پیام سیگنالینگ اضافی اندکی نیز وارد می کند. اضافه کردن راه اندازی/بروزرسانی/حذف صریح به روش `soft-state` امکان می دهد به سازگاری قابل مقایسه (و گاهی بهتر) در مقایسه با روش `hard-state` دست یابد.

تمرکز ما در این مقاله عمدتاً مبتنی بر کمیت و عملکرد بوده است. ما در حال حاضر در حال بررسی روش های مختلفی برای اندازه گیری پیچیدگی غیر مبتنی بر عملکرد روش های سیگنالینگ مختلف هستیم. در اینجا، موضوعات معماری مانند جفت شدگی و سیگنالینگ با دیگر مؤلفه های سیستم (مانند اینکه پروتکل های `hard-state` نیاز به یک مکانیزم اعلان خارجی یا یک مکانیزم ضربان قلب داخلی دارند)، مهم خواهد بود.

مراجع

- [1] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, and S. Molendini. Rsvp refresh overhead reduction extensions. RFC 2961, April 2001.
- [2] B. Cain, S. Deering, B. Fenner, and A. Thyagarajan. Internet group management protocol, version 3, Oct. 2002. RFC 3376.
- [3] D. D. Clark. The design philosophy of the DARPA internet protocols. In Proceeding of SIGCOMM, Stanford, CA, Aug 1988.
- [4] S. Deering. Host extensions for ip multicasting, August 1989. RFC 1112.
- [5] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei.

- [18] C. Topolcic. Experimental Internet Stream Protocol: Version 2 (ST -II), October 1990. Internet RFC 1190.
- [19] S. Zabele, M.Dorsch, Z. Ge, P . Ji, M. Keaton, J. Kurose, J. Shapiro, and D. Towsley. Sands: Specialized active networking for distributed simulation. In DARP A Active Networks Conference and Exposition (DANCE), San Francisco, California, USA, May 2002.
- [20] L. Zhang, S. Deering, D. Estrin, S. Shenker, , and D. Zappala. RSVP: A new resource reservation protocol. IEEE Network, September 1993.
- The pim architecture for wide-area multicast routing. ACM Transactions on Networks, April 1996.
- [6] S. Deering, D. Estrin, D. Farinacci, V . Jacobson, C. Liu, and L. Wei. The pim architecture for wide area multicasting. IEEE/ACM Tran. Networking, 4(2), Apr. 1996.
- [7] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V . Jacobson, C. Liu, P . Sharma, and L. Wei. Protocol independent multicast-sparse mode (PIM-SM): Protocol specification, June 1998. RFC2362, <http://www.faqs.org/rfcs/rfc2362.html>.
- [8] W. Fenner. Internet group management protocol, version 2, Nov. 1997. RFC 2236.
- [9] S. Floyd, V . Jacobson, C. Liu, S. McCanne, and L. Zhang. A reliable multicast framework for light-weight sessions and application level framing. IEEE/ACM Tran. Networking, 5(6), December 1997.
- [10] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. Sip: Session initiation protocol, Mar. 1999. RFC 2543.
- [11] P . Ji, Z. Ge, J. Kurose, and D. Towsley. A comparison of soft-state versus hard-state signaling. Technical report, University of Massachusetts at Amherst, 2003.
- [12] Kazaa file sharing network, 2002. <http://www.cazaa.com/>.
- [13] P . Pan and H. Schulzrinne. Staged refresh timers for RSVP. In 2nd Global Internet Conference, Phoenix, AZ, 1997.
- [14] C. Partridge and S. Pink. An Implementation of the Revised Internet Stream Protocol (ST -2). Journal of Internetworking: Research and Experience, 3(1), March 1992.
- [15] Q2931. ITU-T Recommendation.
- [16] S. Raman and S. McCanne. A model, analysis, and protocol framework for soft state-based communication. In Proceeding of SICOMM, Boston, MA, 1999.
- [17] P . Sharma, D. Estrin, S. Floyd, and V . Jacobson. Scalable timers for soft state protocols. In Proc. of IEEE INFOCOM, Kobe, Japan, 1997.