

# Towards Nearly-Zero-Error Sketching via Compressive Sensing

USENIX

April 2021

Qun Huan (Peking University)

8 citation

اسکچ‌ها الگوریتم‌هایی برای مانیتور جریان‌های شبکه و ذخیره اطلاعات آماری هستند که از دو مزیت استفاده کم از منابع و خطای کراندار<sup>۱</sup> بهره می‌برند. اما تمامی جریان‌ها خطاهایشان کراندار نمی‌باشد و به همین دلیل این الگوریتم‌ها در برخی مسایل خاص مانند heavy hitter ها و یا distinct flow – traffic distribution ها تنها کاربرد دارند. لذا می‌خواهیم اسکچی بر مبنای تکنیک compressive sensing ارایه دهیم که برای تمامی جریان‌ها خطایی نزدیک به صفر<sup>۲</sup> داشته باشد. همانطور که گفته شد ابتدا از این تکنیک به همراه اسکچ برای بالا بردن دقت استفاده می‌کنیم سپس از آن به عنوان یک اسلوب برای بررسی اسکچ‌های طرح‌شده دیگر استفاده می‌کنیم و در نهایت بر اساس نتایج آنالیزها دو الگوریتم NZE می‌سازیم. این دو الگوریتم را در محیط شامل OpenVSwitch و P4 پیاده سازی می‌کنیم. این دو الگوریتم خطای جریان کمتر از ۰.۱٪ برای ۹۹.۷۲٪ جریان‌ها خواهند داشت.

دلایل انتخاب الگوریتم Ccompressive Sensing که برای پردازش سیگنال‌ها می‌باشد: (۱) ریکاوری تقریباً بی نقص سیگنال‌ها با استفاده محدود از منابع (۲) بر پایه خصوصیات ماتریس مثل sparsity که اسکچ‌ها هم از آن اصول ریاضی استفاده می‌کنند، نیز می‌باشد.

دو تا کار را پیش می‌بریم: در ابتدا از compressive sensing با در نظر گرفتن آمارهای جریان به مثابه سیگنال به صورت مستقیم برای مانیتور استفاده می‌کنیم که پس از بررسی‌ها نتیجه می‌شود که این استفاده مستقیم کار آسانی نخواهد بود و در محیط با منابع محدود عملی نخواهد بود. سپس از آنها در کنار اسکچ‌ها برای طراحی الگوریتم‌های جدید استفاده می‌کنیم. به صورت جزیی تر اسکچ‌ها را به صورت ماتریسی مدل می‌کنیم و سپس به صورت کمی میزان مناسب بودن آنها برای compressive sensing را بررسی می‌کنیم (مولفه orthonormality). و سپس از این نتایج برای طراحی الگوریتم‌ها استفاده می‌کنیم که NZE باشند و از آنالیز ماتریسی به جای آنالیز آماری یا تصادم هاش ها استفاده شود.

**Flow monitoring & Sketches**: مانیتورینگ در سطح جریان هر فلو را توالی از بسته‌ها با flowID یکسان در نظر می‌گیرد و flow vlaue را برای هر کدام می‌شمارد. اسکچ‌ها اطلاعات هر بسته را در یک داده ساختار فشرده می‌ریزند. هر الگوریتم اسکچ از یک تعداد شمارنده تشکیل شده است. و دو عملیات کویری و آپدیت را دارا می‌باشد. آپدیت در data-plane انجام می‌شود و برای هر بسته بر اساس مقدار هاش، شمارنده‌های متناظر را آپدیت می‌کند تا تغییرات جریان را بروز دهد. کویری در control-plane انجام میشود و به صورت دوره‌ای داده ساختارهای اسکچ را از نقاط مربوطه مختلف جمع‌آوری می‌کند و می‌تواند به مقادیر flow value هر flow ID دسترسی پیدا کند.

<sup>1</sup> Bounded Error

<sup>2</sup> Nearly Zero Error Sketch (NZE)

در ادامه الگوریتم count-Min را به عنوان مثال مورد بررسی قرار می‌دهد. سپس به صورت خلاصه بقیه الگوریتم‌های شمارنده برای هر فلو<sup>۳</sup> را در دو محیط با رم ۱۰ مگابایت که متدوال سویچ‌هاست و ۱۰۰ مگابایت که تقریباً ایده آل و بدون محدودیت می‌باشد بررسی می‌کند و مشاهده می‌شود که برخی در حافظه ۱۰۰ مگابایت دقت خوبی دارند، اما همانطور که گفته شد، این مقدار حافظه در دنیای واقعی قابل دسترسی نیست.

**Compressive sensing**: روشی برای پردازش سیگنال‌های با ابعاد بالا با میزان استفاده کم از منابع می‌باشد. مدل‌های قدیمی این تکنیک از دورویه تشکیل شده‌اند:

- Sensing procedure: ضبط سیگنال با ضرب آن در یک ماتریس<sup>۴</sup>
- Recovery procedure: بازسازی سیگنال<sup>۵</sup> با یک رویکرد بهینه و با خطای کم<sup>۶</sup> باید گفت که روش‌های مختلفی برای بازسازی سیگنال‌ها وجود دارد اما در این مقاله از رویکرد اصلی که بر مبنای matrix orthonormality هست استفاده می‌کنیم که sound و ساده می‌باشد.

**فرضیات**: ترافیک شبکه پراکندگی<sup>۷</sup> دارد. با اینکه جریان‌های مختلف زیادی می‌تواند وجود داشته باشد اما جریان‌های فعال تعدادشان کمتر است. همچنین فرض می‌کنیم هر الگوریتم اسکچ یک بخش خطی دارد که در آن هر شمارنده به صورت خطی توسط یک بسته بروز می‌شود.

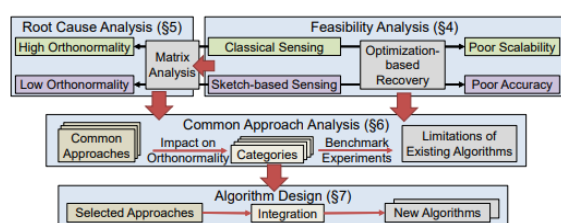


Figure 2: Workflow.

**شیوه کار**: در رویکرد اول دو متد داریم: ابتدا از دورویه CS استفاده می‌کند. ولی در روش دوم از الگوریتم‌های اسکچ برای ضبط اطلاعات هر بسته استفاده می‌کنیم و سپس هر الگوریتم را به صورت یک ماتریکس مدل می‌کنیم (مشاهده می‌شود که این ماتریس‌ها ویژگی متعارف بودن برخوردار نمی‌باشند) و علاوه بر آن از رویکرد ریکاوری بهینه CS کلاسیک بهره می‌گیریم (sketch based sensing+ optimization based recovery)

پس از بررسی‌ها مشخص می‌شود که متود اول مشکل مقیاس پذیری<sup>۸</sup> دارد و متود دوم هم دارای دقت پایینی می‌باشد. به همین دلیل به جزئیات CS وارد شده و یک الگوریتم جدید ارائه می‌دهیم. در ادامه رویکردهای متداول برای بهبود متعارف بودن الگوریتم‌های اسکچ را بررسی می‌کنیم (۴ دسته هستند) و نتیجه می‌گیریم که این رویکردها بهره برداری شان برای به کار بردن همراه Compressive sensing optimization-based recovery کافی نمی‌باشد. در نهایت دو الگوریتم ارائه می‌دهیم که اسکچ‌ها را با CS recovery ترکیب می‌کند و بر اساس نتایج آنالیز روش‌های متداول مرحله قبل و انتخاب روش‌های موثر برای پیاده سازی آنها در data plane این دو الگوریتم استفاده می‌کنیم. و هر کدام نیز به راحتی با ماتریس آنالیز میشوند. در control plane، این دو الگوریتم flow ID ها و flow value را با حل یک مسئله بهینه سازی بازگردانی می‌کنند.

<sup>3</sup> Per-flow counter sketch

<sup>4</sup> Mapping Matrix

<sup>5</sup> Signal Recovery-reconstruction

<sup>6</sup> Optimization-based recovery

<sup>7</sup> sparsity

<sup>8</sup> Scalability