

بسمه تعالی
گزارش و تحلیل مقاله

An SDN architecture for time sensitive industrial IoT

تعریف مسئله و هدف اصلی مقاله

برنامه‌های اینترنت اشیا صنعتی (IIoT)، نیازهای متعددی در ارتباطات شبکه‌ای دارند. برنامه‌های مهمی مانند، رله هشدار اضطراری، خروج اضطراری صنعتی و سیستم‌های نظارتی نیازمند بروزرسانی‌های متعدد و تازه جهت نگه داشتن بسته‌های جدید هستند. بنابراین نیاز است بر اساس رخدادها، شبکه مجدد تنظیم شده و عکس‌العمل مناسب را انجام دهد.

روش‌های زیادی برای انتقال اطلاعات با دقت بالا و تاخیر محدود ارائه شده‌اند. یکی از راه‌حل‌های برجسته در این زمینه شبکه‌های نرم‌افزارمحور (SDN) هستند که مشکلات مربوط به تاخیر مثل ازدحام را در انتقال اطلاعات در کارخانه، حل و فصل می‌کنند. رویکردهای موجود راه‌حل‌های آفلاین که با محاسبات فشرده به دست می‌آیند را ارائه می‌کنند. در این مقاله روشی برپایه استراتژی‌های آنلاین و ساده با استفاده از کنترلر SDN، ارائه شده است.

به طور دقیق‌تر در این مقاله برپایه استاندارد IEEE Time Sensitive Networking (TSN):

- یک چارچوب سیاست کنترل که TSNU نامیده است، در راستای تخصیص فرجه زمانی¹ برای ترافیک‌های زمانبندی شده همزمان با کاهش ازدحام شبکه، طراحی کرده می‌کند.
- رویکردی برای بهینه‌سازی همزمان زمانبندی، مسیریابی و کنترل جریان ورودی با تضمین پایداری شبکه و حداکثر کردن جریان ورودی شبکه، گسترش داده است.
- تحلیل نظری و شبیه‌سازی گسترده‌ای در زمینه ارزیابی TSNU انجام داده است که بیانگر بهبود قابل ملاحظه در مقایسه با دیگر روش‌ها است.

مطالعات پیشین

با ظهور Industry 4.0 و IIoT، چالش‌های مدیریت ترافیک حساس به زمان به وجود آمده است. در این زمینه گروه کاری TSN، استاندارد IEEE 802.1Qbv را توسعه داده است. این استاندارد شامل Time Aware Shaper (TAS) است که به چرخه‌های زمانی هماهنگ و بازه‌های زمانی دروازه² برای ترافیک برنامه‌ریزی شده با اولویت بالا، متکی است. TAS تلاش می‌کند تا بسته‌ها را به موقع تحویل دهد.

¹ Time-slot

² Gate

مطالعات موجود در زمینه TSN به موضوع این مقاله نزدیک هستند که به بررسی آن‌ها می‌پردازیم. جهت بررسی و مقایسه بهتر، در ... کارهای مرتبط به نمایش در آمده‌اند.

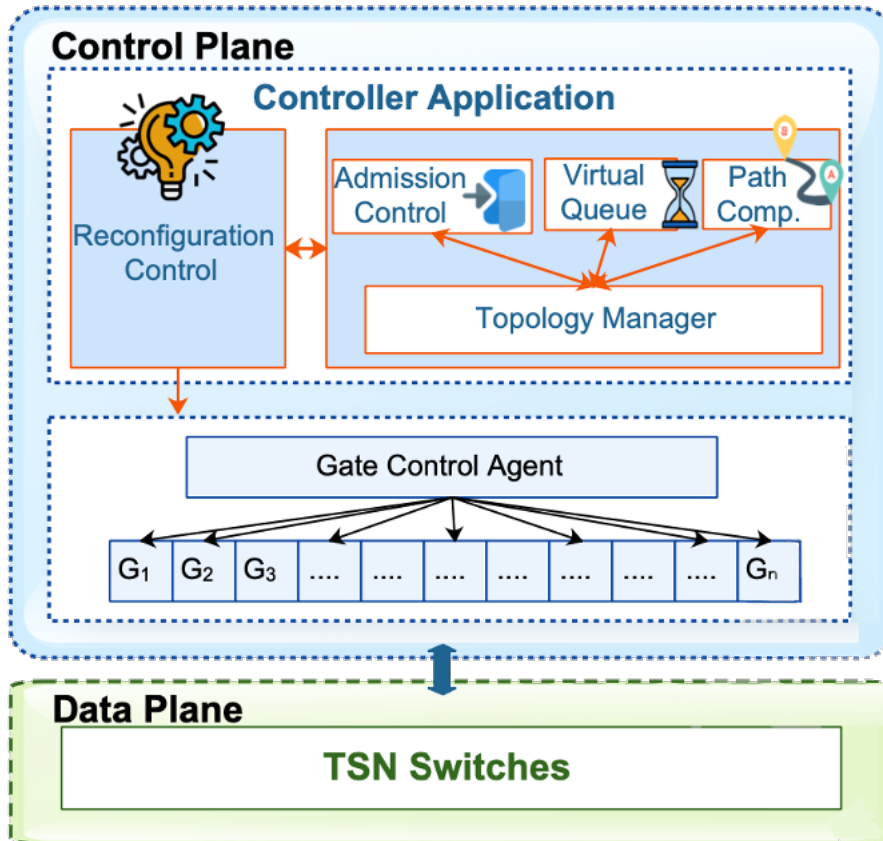
کار مرتبط	شرح، محدودیت‌های اصلی و تفاوت با کار این مقاله
Farzaneh et al. [1,2]	<ul style="list-style-type: none"> - ارائه یک مدل شبیه‌سازی برای خودکارسازی برنامه‌ریزی TSN - محدودیت‌های دروازه کلیدی در نظر گرفته نشده است.
Stanton et al. [3]	<ul style="list-style-type: none"> - روال‌های رایج برای چگونگی استفاده از اشتراک زمانی در سناریوهای Industry 4.0 را نمایش می‌دهد. - راه‌کار آنلاینی برای پیکربندی مجدد آنلاین ارائه نمی‌کند.
Zhao et al. [4]	<ul style="list-style-type: none"> - حداکثر تاخیر برای جریان‌های حساس یا فعال شونده با زمان را محاسبه کرده و روش بهینه‌ای برای مدیریت آن‌ها ارائه می‌کند. - پیچیدگی و میزان عملیات برای رسیدن به Gate control list (GCL) زیاد است.
Boehm et al. [5]	<ul style="list-style-type: none"> - کنترلر یکتا برای شبکه‌های حساس به زمان و SDN طراحی می‌کند. - با سویچ‌های SDN، همگام‌سازی زمانی انجام نمی‌دهد که به معنای عدم مدیریت بر پارامترهای (GC) Gate control است.
Said et al. [6]	<ul style="list-style-type: none"> - مدل IEEE 802.1Qcc را به صورت ثابت با رویکرد SDN را ارائه کرده است. - بهینه‌سازی کنترل جریان‌های ورودی، مسیریابی و برنامه‌ریزی، نادیده گرفته شده است.
Nayak et al. [7]	<ul style="list-style-type: none"> - ارائه معماری Time Sensitive Software Defined Network (TSSDN)، که نزدیک‌ترین راه‌کار به این مقاله است. - استفاده از دید کلی کنترلر برای مسیریابی و برنامه‌ریزی با پیش‌فرض ورودی‌ها

در این مقاله، روش پیشنهادی TSNU با روش TSSDN، مقایسه شده است.

راه‌حل پیشنهادی مقاله برای مسئله

طراحی

معماری ارائه شده در تصویر- ۱ به نمایش در آمده است. این معماری از شش مولفه شکل گرفته است که در ادامه به توضیح آن‌ها می‌پردازیم.



تصویر- ۱

- **Reconfiguration Control**: این مولفه اطلاعات حداکثر تعداد هاپ مورد نیاز جهت دستیابی به یک تنظیم مجدد را از **Topology manager** و فرجه زمانی‌ای که در آن رخ می‌دهد را از **Gate control engine** دریافت کرده و بر اساس آن مسیر جایگزین را در مواقعی مانند از دست رفتن یک مسیر، اجرایی می‌کند.
- **Topology Manager**: این مولفه مدیریت سویچ‌ها و کاربران نهایی را برعهده دارد.
- **Admission Control**: این مولفه بر اساس اطلاعات و متغیرهای لازم، جریان ورودی را رد یا تایید می‌کند.
- **Virtual Queue Database**: این مولفه پارامترهای لازم جهت برنامه‌ریزی انتقال اطلاعات را دارد که تمامی منابع را به صورت صف‌ها در می‌آورد.
- **Path Computation Element and Gate Control Engine**: این دو مولفه با هم در سناریوهای پویا و ایستا، نقش اساسی دارند. پس از پذیرش یک جریان ورودی، **Path computation element** مسیرهای هم وزن را در زمان یک شکست یا ازدحام، محاسبه می‌کند. فرجه‌های زمانی لازم توسط **Gate control agent** از میان ورودی‌های **Gate control** که به عنوان G_1, \dots, G_n تعریف شده‌اند،

انتخاب می‌شوند با در نظر گرفتن این‌که کل مسیری که برای یک جریان رزرو می‌شود نباید با یک جریان ترافیک فعال شدنی با زمان، تداخل داشته باشد.

فرموله‌سازی مسئله بهینه‌سازی

در این بخش، مشکل پیکربندی مجدد TSN به صورت یک مشکل بهینه‌سازی سودمندی شبکه، فرموله می‌شود. هدف اصلی طراحی در این بخش، به عنوان طراحی یک کنترلر سیاست پایدار برای چارچوب صف‌های مجازی شرح داده می‌شود.

مقدمات

محدودیت‌های مسئله مجموعه‌ای محدود و بسته است و تابع هدف یک تابع مقعر^۱ است. به طور خاص، در ظرفیت لینک‌های ارتباطی و طول صف‌ها محدودیت وجود دارد. مشتق دوم تابع هدف منفی است و بنابراین به صورت قطعی یک تابع مقعر است. در این شرایط باید یک بیشینه^۲ انحصاری برای تابع هدف پیاده شود. نمودار شبکه به صورت یک گراف $G(J,E)$ که J مجموعه گره‌ها و E مجموعه یال‌های آن است، در نظر گرفته می‌شود. به صورت خلاصه، تعاریف اولیه در جدول ۱ آمده است.

جدول - ۱

نشانه	شرح
ساختار شبکه	
J	مجموعه گره‌های شبکه (سویچ‌های TSN)
e	yal شبکه (ارتباط بین دو سویچ TSN)
E	مجموعه yal‌های شبکه
R	مجموعه مسیرهای موجود در گراف شبکه
ساختار زمان‌بندی	
t	یک فرجه زمانی متناظر با مدت زمان ورودی gate control که لینک روشن یا خاموش دارد
T	یک افق زمانی که به دوره‌های پایه (TSN cycle times) که بعداً به فرجه‌های زمانی تقسیم می‌شود.
مدل‌سازی ترافیک	
c	کلاس ترافیک که برای ST مقدار ۱ و BE مقدار صفر دارد.
γ	نمونه‌ای از ترافیک $\gamma = c$ برای یک جریان داده شده
$\tau(t)$	وضعیت لینک، ۱ روشن و ۰ خاموش در یک فرجه زمانی
$\hat{v}_e(t)$	اندازه صف مجازی برای لینک e و فرجه زمانی t
$v_e(t)$	اندازه صف فیزیکی، تعداد بسته‌های در انتظار برای گذر از yal e
مدل‌سازی صف و سرویس TSN	
$\mathcal{F}(t)$	تابع Gating (packets/time-slot) برای یک جریان در فرجه زمانی t
$C_e(t1, t2)$	تخصیص سرویس (Gbit/s) به yal e بین فرجه زمانی t1 و t2
$\hat{Q}_e(t)$	مجموع بسته‌های تایید شده توسط کنترلر در yal e در فرجه زمانی t

¹ Concave

² Maximum

$\nabla(T)$ $\tau(t)p_e a_e$ Z	نرخ (packets/time-slots) تعداد بسته‌های دریافت شده در مقصد داده شده در انتهای زمان T نرخ سرویس برای صف مجازی لینک e در فرجه زمانی t تابع کارایی برای شبکه TSN
--	--

مسئله بیشینه‌سازی کارایی

اگر Π سیاست برای پذیرش داشته باشیم، مسئله بیشینه‌سازی کارایی، سیاست $\pi \in \Pi$ را پیدا می‌کند که حداکثر ترافیک هر کلاس را در کنار پایدار نگه داشتن نرخ ارسال در شبکه، خواهد داشت:

$$\max_{\pi} \mathbb{E} \left[\sum_{\gamma} Z_{\gamma}(r_{\gamma}) \right]$$

با دو قید زیر:

$$\lim_{T \rightarrow \infty} \frac{\nabla(T)}{T} = r_{\gamma}$$

$$\lim_{T \rightarrow \infty} \sum_e v_e(T)/T = 0 \quad \forall e \in E$$

قید اول به معنای این‌که چقدر سریع باید یک انتقال انجام شده و قید دوم به معنای نرخ سرویس‌دهی صف است. به این ترتیب هرچه ازدحام کم‌تر باشد، تاخیر نیز کم‌تر است.

چارچوب صف مجازی TSN

در این مقاله یک صف مجازی بر اساس سیاست بهینه پایدار ایجاد شده که به صف فیزیکی نگاشت می‌شود. میزان رشد صف مجازی به این صورت است:

$$\hat{v}_e(t+1) = \hat{v}_e(t) + \hat{Q}_e(t) - \tau(t)p_e a_e$$

بر اساس تابع drift-plus-penalty، یک تابع لیاپونوف که صف‌های مجازی را به طول صف فیزیکی نگاشت می‌کند، شکل می‌دهیم:

$$L(\hat{v}(t)) = \sum_e \hat{v}_e^2(t)$$

با یک انتقال شرطی:

$$\Delta(\hat{v}(t), a(t)) = \mathbb{E}[L(\hat{v}(t+1)) - L(\hat{v}(t)) | \hat{v}(t), a(t)]$$

به این شرط که:

$$\Delta(\hat{v}(t), a(t)) \leq \mathbb{E}[\hat{Q}_e^2(t) + p_e^2 + 2\hat{v}_e(t)(\hat{Q}_e(t) - \tau(t)p_e a_e(t)) | \hat{v}(t), a(t)]$$

در سمت راست معادله قبل، p_e^2 و $\hat{Q}_e^2(t)$ ثابت‌های محدودی هستند که حداکثر مقدار آن محدود می‌شود به $Q_{max}^2(t) + p_{max}^2$ که مقدار آن را به H نشان می‌دهیم، با تجمیع سمت راست معادله به این رابطه می‌رسیم:

$$= H + 2\mathbb{E} \left[\sum_{\gamma} \hat{Q}_{\gamma}(t) \sum_e \hat{v}_e(t) 1_{\{e \in \psi_c(t)\}} \mid \hat{v}(t), a(t) \right] \\ - 2\mathbb{E} \left[\sum_e \hat{v}_e(t) \tau(t) p_e a_e(t) \mid \hat{v}(t), a(t) \right]$$

در این رابطه مقدار $1_{\{e\}}$ در صورتی آرگومان True باشد مقدار ۱ و در غیر آن ۰ است. عبارت کنترلی را به این صورت بازنویسی می‌کنیم:

$$\hat{Q}_e(t) = \sum_{\gamma} \hat{Q}_{\gamma}(t) 1_{\{e \in \psi_c(t)\}}$$

بدون در نظر گرفتن مقدار ثابت H، هدف اصلی کمینه کردن این تابع است:

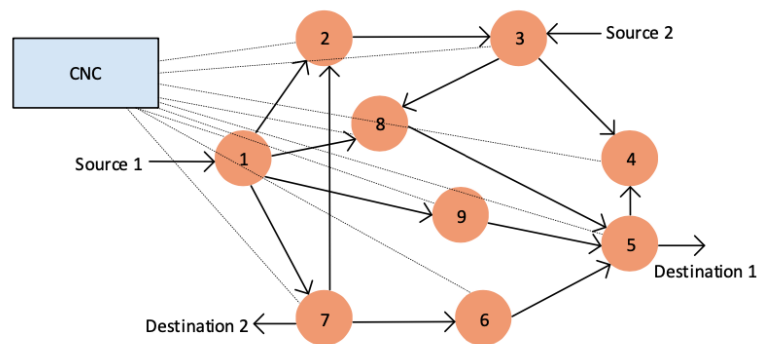
$$2\mathbb{E} \left[\sum_{\gamma} \hat{Q}_{\gamma}(t) \sum_e \hat{v}_e(t) 1_{\{e \in \psi_c(t)\}} \mid \hat{v}(t), a(t) \right] - 2\mathbb{E} \left[\sum_e \hat{v}_e(t) \tau(t) p_e a_e(t) \mid \hat{v}(t), a(t) \right] \\ - 2J \sum_{\gamma} Z_{\gamma}(\hat{Q}_{\gamma}(t))$$

با کمینه سازی این معادله فوق، قانون TSNU مورد نظر به دست می‌آید در حالی که ضمن در نظر گرفتن مدیریت ورودی‌ها و مسیریابی، نرخ پایدار وجود دارد.

ارزیابی راه‌حل پیشنهادی

تنظیمات کلی ارزیابی

در این مقاله، با استفاده از Python 3.6 شبیه‌ساز و کنترلر سفارشی برای تکرار محیط استاندارد کنترل QCC ساخته شده است. برای استاندارد QCC ساختار صف مشخصی وجود ندارد بنابراین صف خاص این مقاله طراحی شده است. لینک‌های ارتباطی دارای ظرفیت 1Gbps هستند که از یک نمودار جهت‌دار پیروی می‌کنند که نمونه آن در تصویر ۲- به نمایش درآمده است. زمان پایه (TSN Cycle Time) ۱ میلی ثانیه خواهد بود که به ۷۰ فرجه زمانی تقسیم شده است.



تصویر- ۲

سیاست TSNU با یک تابع $\ln(\cdot)$ اجرا می‌شود و در نظر گرفته می‌شود $Z_r(r_r) = \ln(1 + (r_r))$. به طور مثال در تصویر-۲، از مبدا ۱ به مقصد ۱، ۳ مسیر داریم که $r_{\gamma 1} = 3$ و از مبدا ۲ به مقصد ۲ هم ۱ مسیر داریم که $r_{\gamma 2} = 1$ بنابراین $Z = \ln(4) + \ln(2)$ خواهد شد. هرچه تعداد مسیرها بیشتر شوند، تعداد جریان‌های برنامه‌ریزی شده هم افزایش پیدا می‌کنند. جریان‌ها پس از یک دوره پایه منقضی می‌شوند و فریم‌هایی به اندازه ۱۰۰ بایت تولید می‌کنند تا بار ترافیکی نسبی λ را ایجاد کنند. در این مقاله برای هر سناریوی ارزیابی، ۱۰۰ تکرار مستقل از شبیه‌سازی انجام می‌شود.

مقایسه TSNU با non-TSNU و صف مجازی پایه

در تصویر-۳ طول Gate Control Queue (GCQ) به عنوان تابعی از تعداد گره‌ها به نمایش درآمده است. همان‌طور که مشاهده می‌شود روش non-TSNU صف‌های کمتری نسبت به دو روش دیگر اشغال می‌کند و این روش ازدحام را به طور دقیق ضبط نمی‌کند در حالی که دو روش دیگر با در نظر گرفتن یک صف مجازی مشترک برای همه پورت‌های خروجی یک سویچ TSNU، ازدحام را با دقت ضبط می‌کنند.

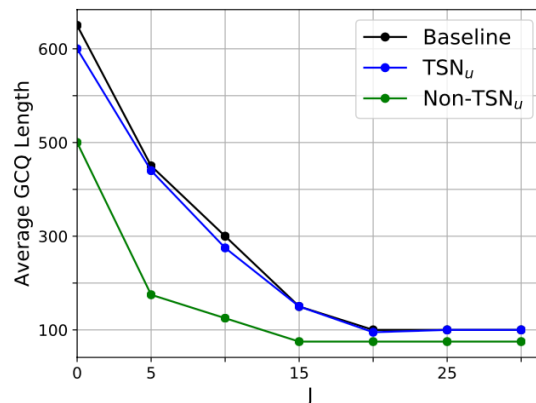


Figure 3: TSN queue length (in bytes) as a function of number J of nodes; traffic load $\lambda = 0.6$, fixed.

تصویر-۳

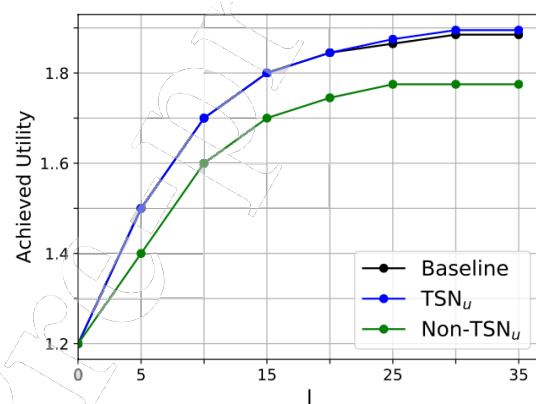


Figure 4: Achieved utility as a function of number J of nodes; traffic load $\lambda = 0.6$, fixed.

تصویر-۴

تصویر- ۴ کارایی را به عنوان تابعی از تعداد گره‌ها در توپولوژی نشان می‌دهد. در این شکل افزایش کارایی با افزایش تعداد گره‌ها، مشهود است. این ارزیابی نشانگر خروجی بهینه سیاست‌های TSNU است. در تصویر- ۵ میانگین تاخیر بسته از ترافیک برنامه‌ریزی شده به عنوان تابعی از تعداد سویچ‌های TSN در ترافیک ثابت به نمایش درآمده است. تاثیر بیشتر non-TSNU به چرخ بسته‌ها در شبکه نسبت داده می‌شود.

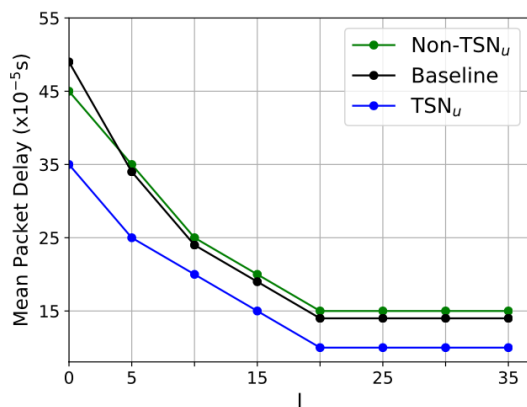


Figure 5: Mean packet delay with respect to number of switching nodes J ; for fixed traffic load $\lambda = 0.6$.

تصویر- ۵

نمودار تصویر- ۶ مقایسه میانگین Gate control queue را به عنوان تابعی از میزان ترافیک ورودی نشان می‌دهد. در مقایسه، روش non-TSNU طول صف کمتری نسبت به دیگر روش‌ها، مصرف می‌کند چون که به ازای هر پورت خروجی سویچ TSN یک صف فیزیکی در نظر می‌گیرد در حالی که دو روش دیگر به ازای هر سویچ TSN یک صف مجازی در نظر می‌گیرند. نکته مهم این است که با مسیریابی در مسیرهای غیرحلقوی بهینه، TSNU تاخیر بسته کمتری نسبت به روش non-TSNU به دست می‌آورد.

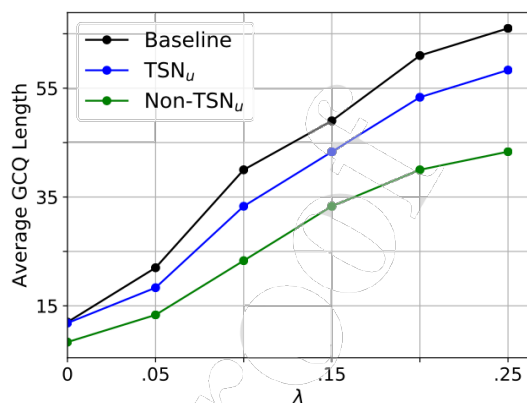


Figure 6: Average GCQ length w.r.t arrival rate λ ; for fixed number of nodes $J = 9$.

تصویر- ۶

تصویر- ۷ رفتار مجموع طول صف‌ها شامل BE و ST را به عنوان تابعی از بار ترافیکی λ نشان می‌دهد. در تصویر- ۸ میانگین تاخیر بسته‌ها در سه سیاست را مشاهده می‌کنیم. حدود $\lambda \approx 1.5$ ، سیاست non-TSNu شروع به اشباع می‌کند.

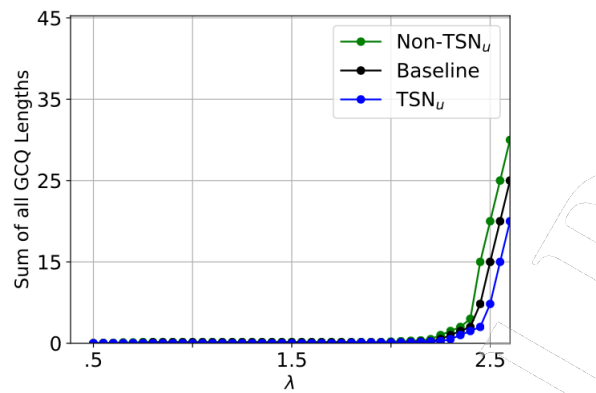


Figure 7: Sum of all GCQ lengths as a function of traffic load λ .

تصویر- ۷

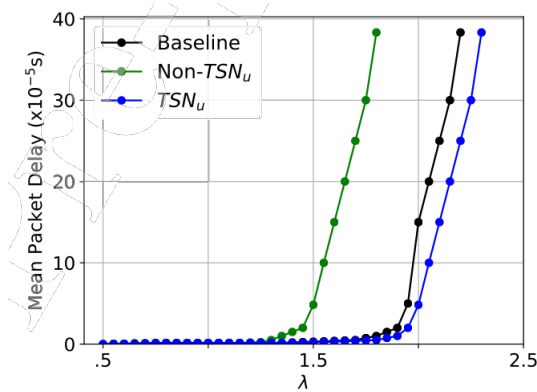


Figure 8: Mean packet delay as a function of traffic load.

تصویر- ۸

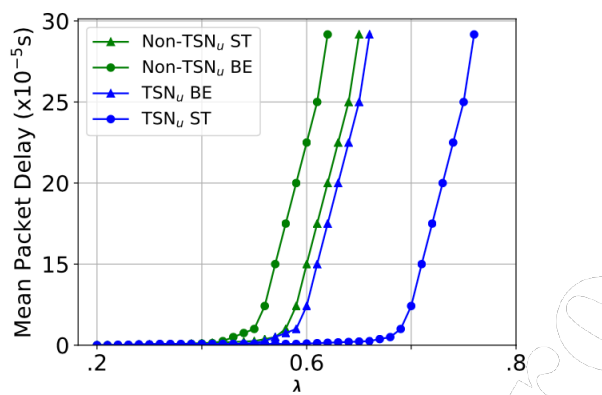


Figure 9: Non-TSN_u vs TSN_u under different load conditions

تصویر- ۹

تصویر- ۹ نشان می‌دهد که چگونه جدایی ترافیک BE و ST کنترل پویا GCQ را امکان پذیر می‌کند. اساساً برای نرخ رسیدن λ بالاتر از ۰.۶، مشاهده می‌کنیم که ترافیک non-TSNU اشباع می‌شود. non-TSNU قادر به پشتیبانی از نرخ ورود بالاتر نیست، عمدتاً به این دلیل که فاقد کنترل بهینه است. در مقابل، سیاست TSNU کنترل بهینه ای را اجرا می‌کند که عملکرد را به میزان قابل توجهی بهبود می‌بخشد.

مقایسه TSNU با TSSDN

به دلیل عدم وجود پیاده‌سازی باز TSSDN در این مقاله یک پیاده‌سازی متناسب انجام شده است. برای مقایسه یک دوره پایه ۱ میلی ثانیه که حداکثر به ۵۰ فرجه زمانی تقسیم می‌شود، در نظر گرفته شده است. تصویر- ۱۰ تعداد جریان‌های برنامه‌ریزی شده پذیرفته شده را به عنوان تابعی از تعداد فرجه‌های زمانی در هر دوره پایه برای یک شبکه با اندازه متوسط تعداد سوئیچ $J=9$ نشان می‌دهد. مشاهده می‌کنیم که تعداد جریان‌های برنامه‌ریزی شده پذیرفته شده با افزایش تعداد فرجه‌های زمانی در هر دوره پایه برای TSSDN و TSNU اشباع می‌شوند. تفاوت کلیدی این است که TSNU حدود ۱۵٪ بیشتر جریان برنامه‌ریزی شده پذیرفته شده برای تعداد زیادی فرجه زمانی در هر دوره پایه نسبت به TSSDN به دست می‌آورد.

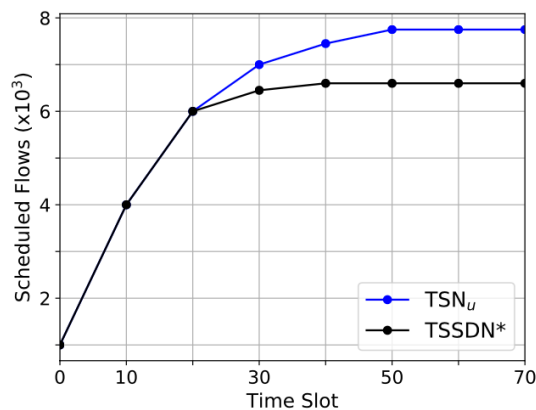


Figure 10: Number of admitted scheduled flows for proposed TSNU and TSSDN* [23] as a function of number of time-slots per scheduling cycle (base period); fixed parameters: $J = 9$ TSN switches, traffic load $\lambda = 0.6$.

تصویر- ۱۰

علاوه بر این، تصویر- ۱۱ نشان می‌دهد که شکاف کارایی از نظر تعداد جریان‌های برنامه‌ریزی شده پذیرفته شده بین TSNU و TSSDN با افزایش تعداد سوئیچ‌های TSN در شبکه به حدود $J = 15$ افزایش یافته و سپس اشباع می‌شود. مزیت کلی TSNU ناشی از بهینه‌سازی مشترک کنترل پذیرش، مسیریابی و برنامه‌ریزی است. تصویر- ۱۲ مقایسه طول صف فیزیکی TSNU و TSSDN برای شبکه‌ای بزرگ با تعداد سوئیچ TSN به میزان $J=40$ را نشان می‌دهد. طول صف‌های کمتر TSNU در مقایسه با TSSDN به تاخیرهای کمتر در صف تبدیل می‌شود.

در مجموع، مزایای عملکرد TSNU به دلیل پایداری نرخ ارائه شده توسط چارچوب TSNU و تمرکز آن بر کنترل مشترک بهینه کنترل پذیرش، مسیریابی و برنامه‌ریزی است. TSSDN تلاش می‌کند تعداد پیوندهایی را که جریان‌ها از طریق آن‌ها برنامه‌ریزی شده است را به حداقل برساند. با این حال، TSNU با کاوش در مسیرهای چرخشی متعددی که در پایگاه داده QCC ذخیره شده‌اند و منجر به طول صف‌های کوتاه می‌شود، تلاش‌های مجدد برای TSSDN را به حداقل می‌رساند. بنابراین، TSNU به دلیل طول صف کمتر در شبکه، به یک کارایی بالاتر دست می‌یابد.

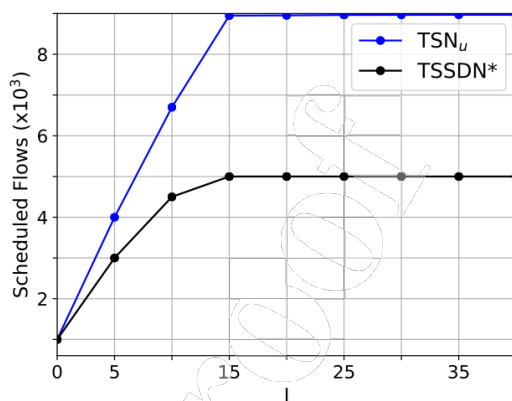


Figure 11: Number of admitted scheduled flows as a function of the number J of nodes (TSN switches); fixed parameters: traffic load $\lambda = 0.9$, 70 time-slots per base period.

تصویر- ۱۱

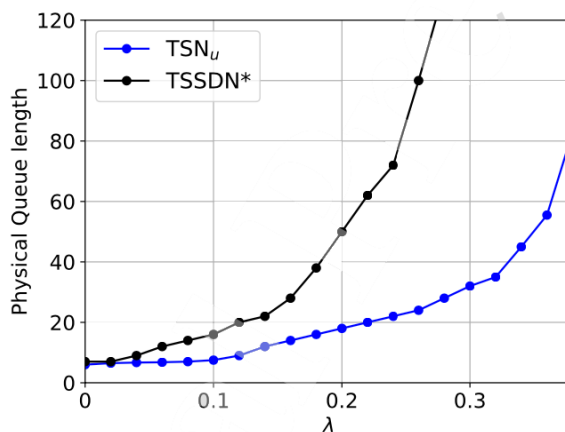


Figure 12: Physical queue length as a function of traffic load ; fixed parameters: $J = 40$ nodes, 70 time-slots per base period.

تصویر- ۱۲

نقاط قوت و ضعف مقاله

نقاط قوت این مقاله:

- بهینه‌سازی مشترک کنترل پذیرش، مسیریابی و برنامه‌ریزی
- بررسی و مقایسه همه جانبه نسبت به روش‌های موجود

نقاط ضعف این مقاله:

- روش‌های بهینه‌سازی ابتدایی و ساده
- پیچیدگی زمانی روش پیشنهادی
- ساده نبودن توسعه و مقیاس‌پذیری روش پیشنهادی

جمع‌بندی و پیشنهادات برای کارهای آتی

در این مقاله، چارچوب TSNU برای بهینه‌سازی مشترک کنترل پذیرش، مسیریابی و برنامه‌ریزی در شبکه‌های حساس به زمان (TSN) ارائه شد. چارچوب TSNU می‌تواند کنترل بهینه را در تنظیمات اینترنت اشیا صنعتی (IIoT) فراهم کند. همچنین برتری و کارایی این روش در مقایسه با دیگر روش‌ها ارزیابی و نشان داده شد. به صورت کار آتی می‌توان این موارد را در نظر گرفت:

- افزودن تکنیک‌های یادگیری ماشین و هوش مصنوعی برای بهینه‌سازی
- ترکیب راه‌کار پیشنهادی با NFV برای ارائه سرویس مقیاس‌پذیر و قابل انعطاف
- کاهش پیچیدگی زمانی روش پیشنهادی

شبیه‌سازی

در این مقاله روش پیشنهادی و همچنین روش‌هایی مانند TSSDN که مرجع مقایسه هستند به صورت باز ارائه نشده‌اند تا بتوان شبیه‌سازی را مجدداً برپا کرد و مقایسه‌ها را اعتبارسنجی نمود.

مشخصات دقیق مقاله

Balasubramanian, Venkatraman, Moayad Aloqaily, and Martin Reisslein, "An SDN architecture for time sensitive industrial IoT", Computer Networks, 2020

منابع

- [1] M.H. Farzaneh, S. Shafaei, A. Knoll, "Formally verifiable modeling of in-vehicle time-sensitive networks (TSN) based on logic programming", Proc. IEEE Vehicular Net. Conf. (VNC), pp. 1–4, 2016
- [2] M.H. Farzaneh, A. Knoll, "Time-sensitive networking (TSN): An experimental setup", Proc. IEEE Vehicular Net. Conf. (VNC), pp. 23–26, 2017
- [3] K. B. Stanton, "Distributing deterministic, accurate time for tightly coordinated network and software applications, IEEE 802.1 AS, the TSN profile of PTP", IEEE Communications Standards Magazine 2 (2), pp. 34-40, 2018
- [4] L. Zhao, P. Pop, S. S. Craciunas, "Worst-case latency analysis for IEEE 802.1 Qbv time sensitive networks using network calculus", IEEE Access 6, pp. 41803–41815, 2018

- [5] M. Boehm, J. Ohms, M. Kumar, O. Gebauer, D. Wermser, "Time-sensitive software-defined networking: A unified control-plane for TSN and SDN", Proc. VDE ITG-Symp. Mobile Commun.-Techno. and Appl., pp. 1–6, 2019
- [6] S. B. H. Said, Q. H. Truong, M. Boc, "Sdn-based configuration solution for IEEE 802.1 time sensitive networking (TSN)", ACM SIGBED Review 16 (1), pp. 27–32, 2019
- [7] N. G. Nayak, F. Durr, K. Rothermel, "Incremental flow scheduling and routing in time-sensitive software-defined networks", IEEE Transactions on Industrial Informatics 14 (5), pp. 2066–2075, 2017