Learning the Optimal Synchronization Rates in Distributed SDN Control Architectures

Konstantinos Poularakis (Yale), **Qiaofeng Qin** (Yale), **Liang Ma** (IBM US), **Sastry Kompella** (NRL), **Kin K. Leung** (Imperial), **Leandros Tassiulas** (Yale).

Objectives

Software Defined Networking (SDN) is an attractive approach for applying to Tactical Edge Networks due to its new capabilities:

- Programmability facilitating "on the fly" deployment of new services
- (Logically-) centralized control facilitating implementation of end-to-end network policies.

But, often, ..

- Physical distribution of the control plane (multiple controllers) brings new issues:
- How to coordinate the management decisions of controllers?
- Event-driven or periodic synchronization message dissemination?
- How often (at what rate) should the controllers synchronize?

Technical Challenges

- Incomplete synchronization among controllers can affect the performance of network applications.
- Temporal inconsistency among controllers can cause routing over failed paths or hinder the effective load balancing of traffic.

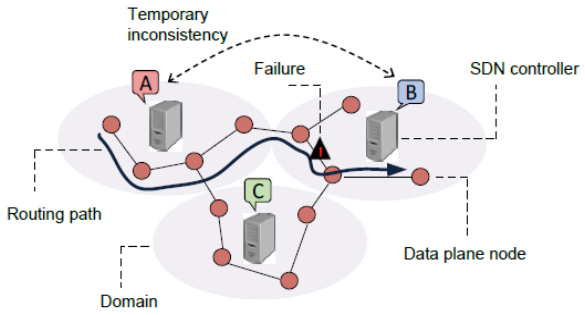


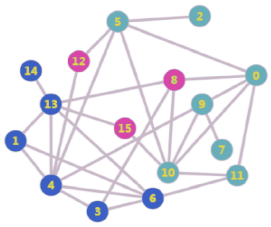
Figure 1: Synchronization vs routing performance example.

Military & Coalition Relevance

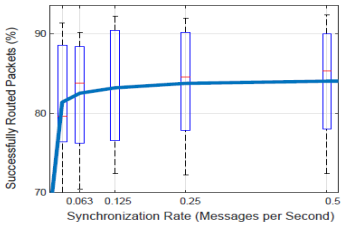
- The proposed methods of synchronizing distributed SDN controllers provide solutions to military scenarios, e.g., in tactical networks that are highly dynamic and therefore the synchronization needs are higher and the impact of synchronization decisions is more critical.

Emulations in Mininet

Despite randomness, the average performance increases with the synchronization rate and saturates eventually showing that a diminishing return rule applies.



(a)



(b)

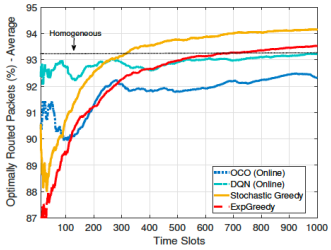
SDN Synchronization Problem

- Cast it as a **learning problem**.
- Do not know in advance how exactly the synchronization rate x will affect application performance $\Psi(x)$.
- Try-out different synchronization rates $\{ x_t \}$ to estimate the unknown function $\Psi(.)$ based on the observed values $\{ \Psi(x_t) \}$
- Challenges of uncertainty of observations and high dimensionality of solution space

Learning algorithm & Evaluation

Algorithm 1: Stochastic Greedy with (σ, τ) input

Initialize $\widehat{x} = 0$;
Try out $x^t = \widehat{x}$ and observe $\Psi_t(x^t)$ $\forall t \in \{1, \ldots, \tau\}$;
Estimate $\widehat{\Psi}(\widehat{x}) = \frac{1}{\tau}\sum_{t=1}^{\tau} \Psi_t(x^t)$;
for *each iteration* k *from* 1 *to* B **do**
  Pick σ random controller pairs p for which $\widehat{x}_p < R$;
  for *each picked pair* p *from* 1 *to* σ **do**
    Set $\widehat{x}' = \widehat{x}$ where $\widehat{x}'_p = \widehat{x}_p + 1$;
    Try out $x^t = \widehat{x}'$ and observe $\Psi_t(x^t)$ $\forall t \in \{(k-1)\sigma\tau + p\tau + 1, \ldots, (k-1)\sigma\tau + p\tau + \tau\}$;
    Estimate $\widehat{\Psi}(\widehat{x}') = \frac{1}{\tau}\sum_{t=(k-1)\sigma\tau+p\tau+\tau}^{(k-1)\sigma\tau+p\tau+\tau} \Psi_t(x^t)$;
    Set $D(\widehat{x}, \widehat{x}') = \widehat{\Psi}(\widehat{x}') - \widehat{\Psi}(\widehat{x})$;
  end
  Update $\widehat{x} = \text{argmax}_{\widehat{x}'} D(\widehat{x}, \widehat{x}')$;
end
Output: $\widehat{x}$;



✓ **Superior performance compared to several popular state-of-the-art learning methods.**

Summary & Future Work

- We studied the problem of finding the optimal synchronization rates among controllers in a distributed eventually-consistent SDN system. We will continue to explore the issues related to the multi-controller scenario and different consistency models in our future works.

Publication(s) & Impact

- K. Poularakis, Q. Qin, L. Ma, S. Kompella, K.K. Leung, L. Tassiulas, "Learning the Optimal Synchronization Rates in Distributed SDN Control Architectures", IEEE Infocom 2019.