Learning the Optimal Synchronization Rates in Distributed SDN Control Architectures

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

Motivation

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?

Impact of synchronization rate on application performance

Temporal inconsistency among controllers can cause routing over failed paths or hinder the effective load balancing of traffic.

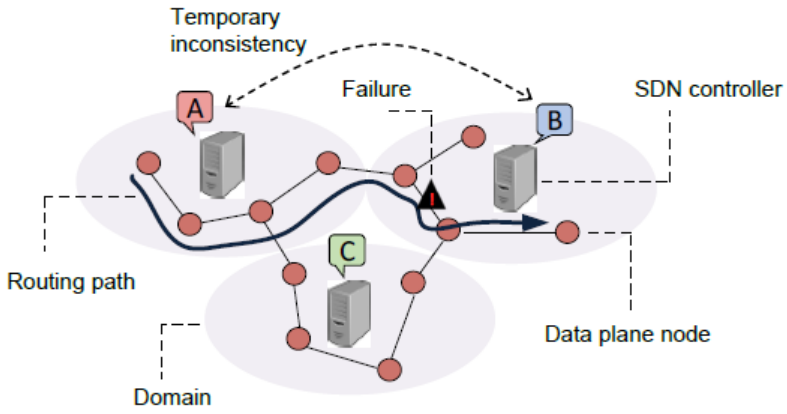
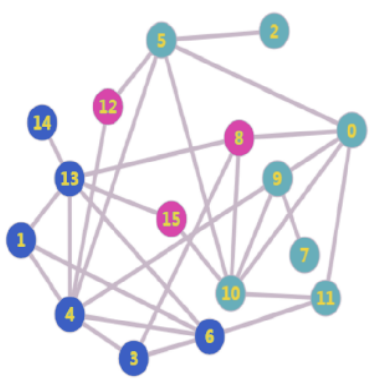
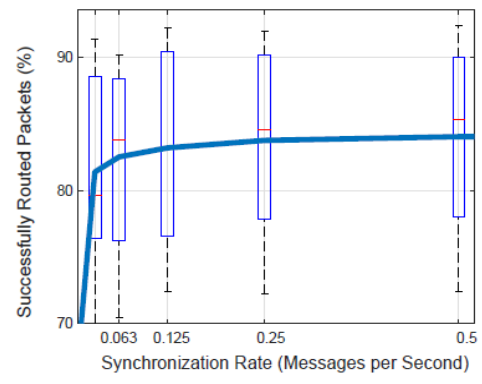


Figure 1: Synchronization vs routing performance example.

Emulations in Mininet

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





(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 $\Psi(x)$.

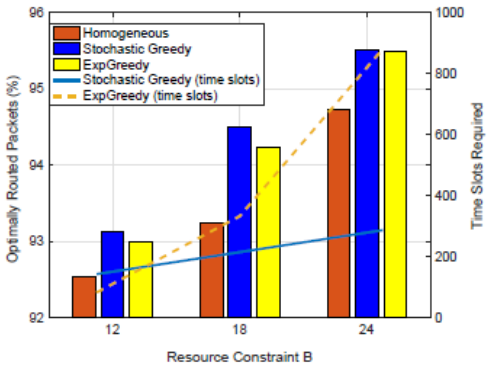
Try-out different synchronization rates $\{ x^t \}$ to *estimate the unknown function* $\Psi(.)$ based on the observed values $\{ \Psi(x^t) \}$

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 $\widehat{\Psi}(\widehat{x}') = \frac{1}{\tau} \sum_{t=(k-1)\sigma\tau+p\tau+1}^{(k-1)\sigma\tau+p\tau+\tau} \Psi_t(x^t)$;
    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: $\widehat{x}$;



- ✓ Better than "all controller pairs equal rate" policy.
- ✓ Shorter training time period than a state of the art method.

Distributed Analytics and Information Science International Technology Alliance
Annual Fall Meeting, September 2018