Dynamic Control for Failure Recovery and Flow Reconfiguration in SDN

Mathematical and Algorithmic Sciences Lab

France Research Center

www.huawei.com

Stefano Paris, Georgios Paschos, Jérémie Leguay

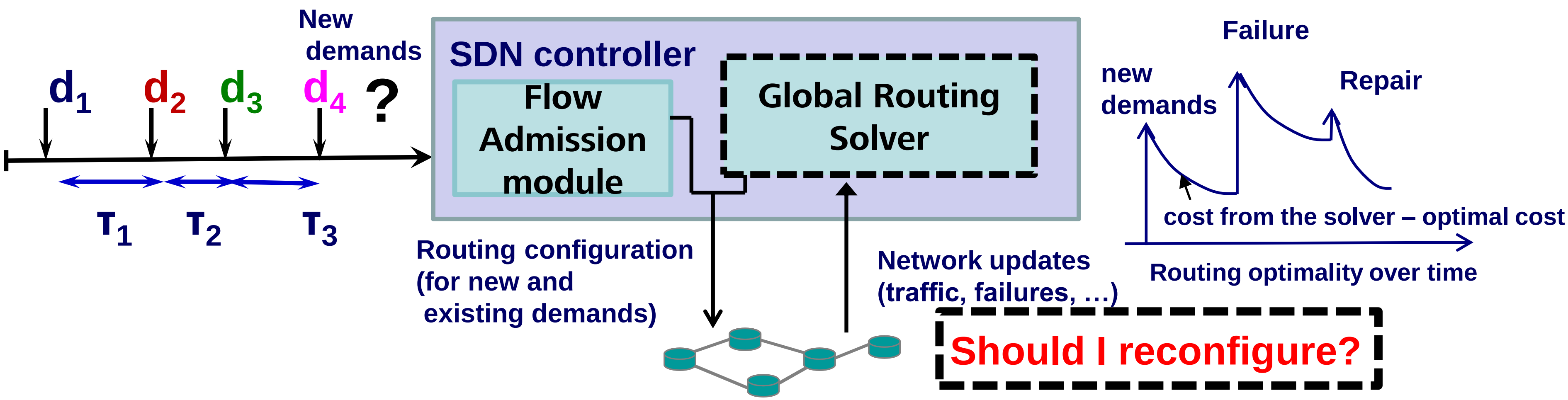
Outline

Introduction

- **Solving an evolving instance of an optimization problem**
  - Over time new events (demands arrive/depart, congestion/failures) change the network status.
  - Limited time to compute a feasible solution as the situation evolves.
  - *Trading off the optimality* and network stability, as operators may want to allocate **a limited reconfiguration budget** to ensure stability.

Which demands to accept and how?

Is there a better configuration?

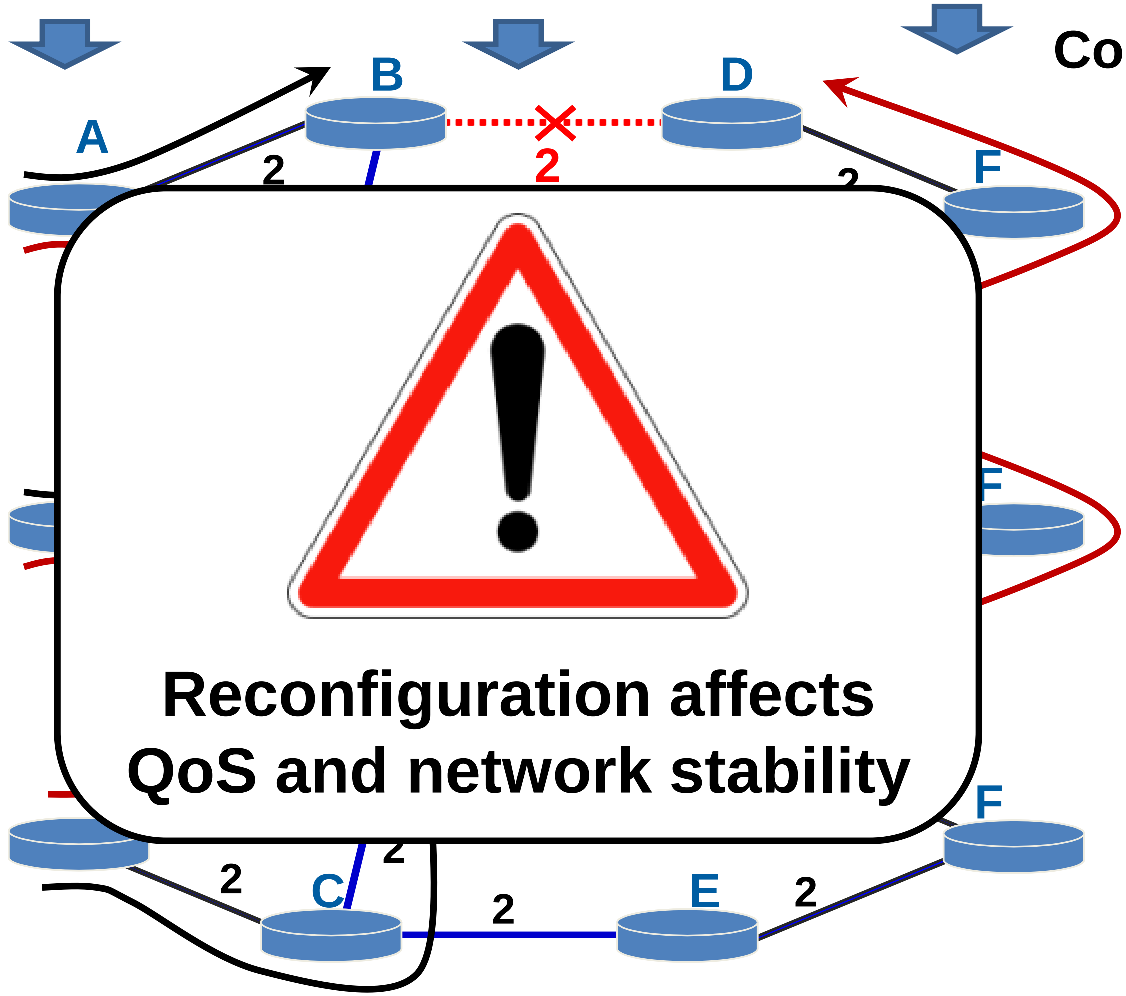


Connections:
1. A → B (1 Mb/s)
2. A → D (1 Mb/s)
Links: 1 Mb/s

SDN controller

OpenFlow | Standard protocols (IETF) | Vendor specifics

Events		**Cost evolution**
1. Reconfiguration after failure of link BD. **FCS/FR**		2+8=10
2. Link repaired.		2+8=10
3. Re-optimization **GC**		4+4=8 (-20%)



Introduction

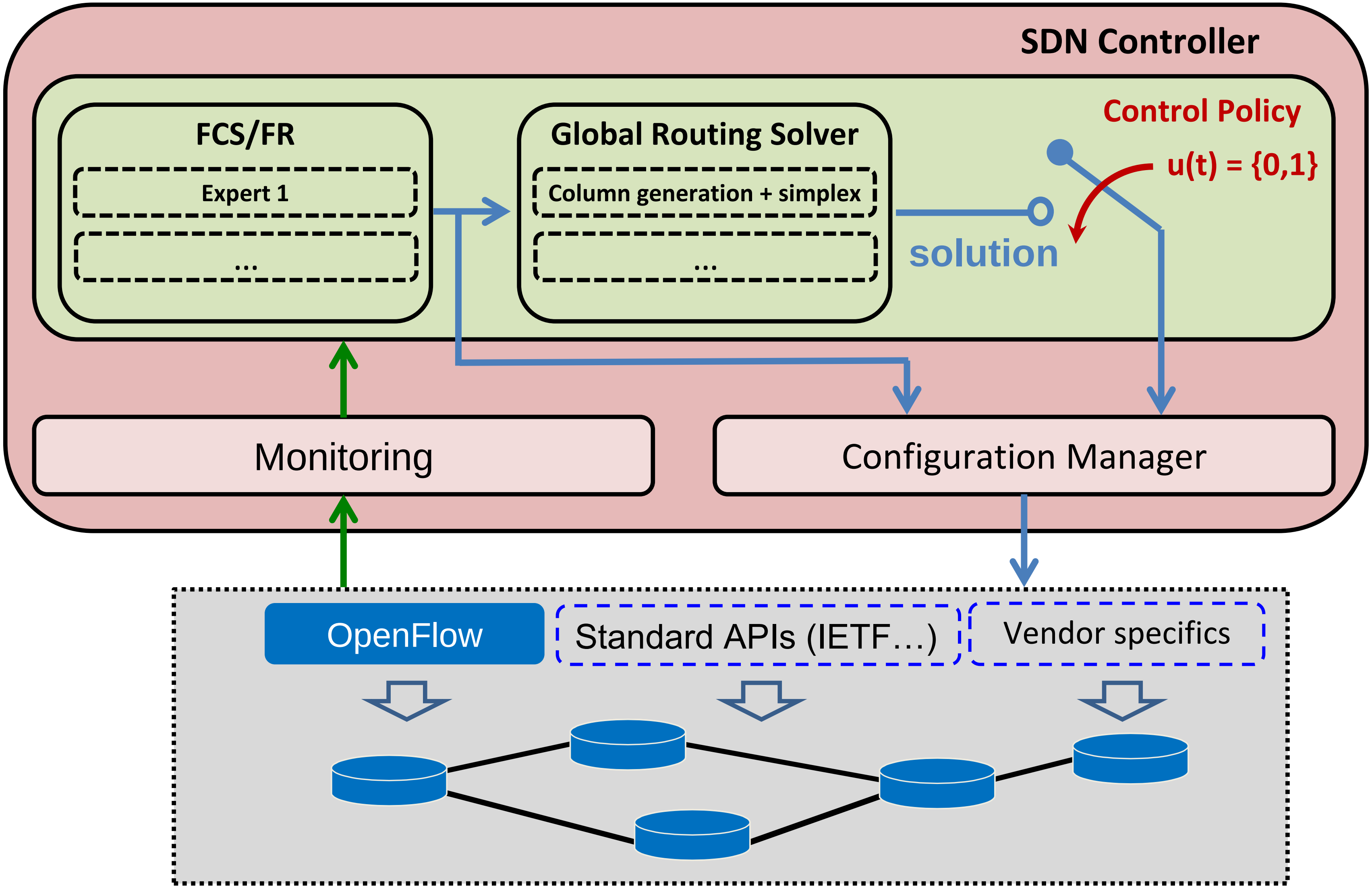
- **Fast connection setup / Fast Recovery (FCS/FR)**
  - Single commodity minimum cost path
  - Fast but suboptimal

- **Garbage collector (GC) ➔ Global routing solver**
  - Min cost multi-commodity flow
  - Optimal but slow
  - Iterative algorithm

Contribution

- **Dynamic control of the SDN controller**
  - When reconfiguring the network
  - Reconfiguration budget

Control Framework For SDN Controllers



Garbage Collector – Global Routing Solver

- **Garbage Collector: an iterative solver to find optimal routing**
  - Min cost multi commodity flow problem (splittable flow)
  - Iterative methods like column generation + simplex
  - Sequence of feasible solutions with diminishing returns

$$C^{OPT} = \min \sum_{e \in E} c_e y_e \quad \longleftarrow \textbf{Cost}$$

$$s.t \quad \sum_{p \in P_k} x_p = 1 \qquad \forall k \in K \quad \longleftarrow \textbf{Demand}$$

$$\sum_{p \in P_e} \sum_{k : p \in P_k} d_k x_p \le b_e y_e \qquad \forall e \in E \quad \longleftarrow \textbf{Capacity}$$

$$0 \le x_p \le 1 \qquad \forall p \in P$$

$$0 \le y_e \le 1 \qquad \forall e \in E$$

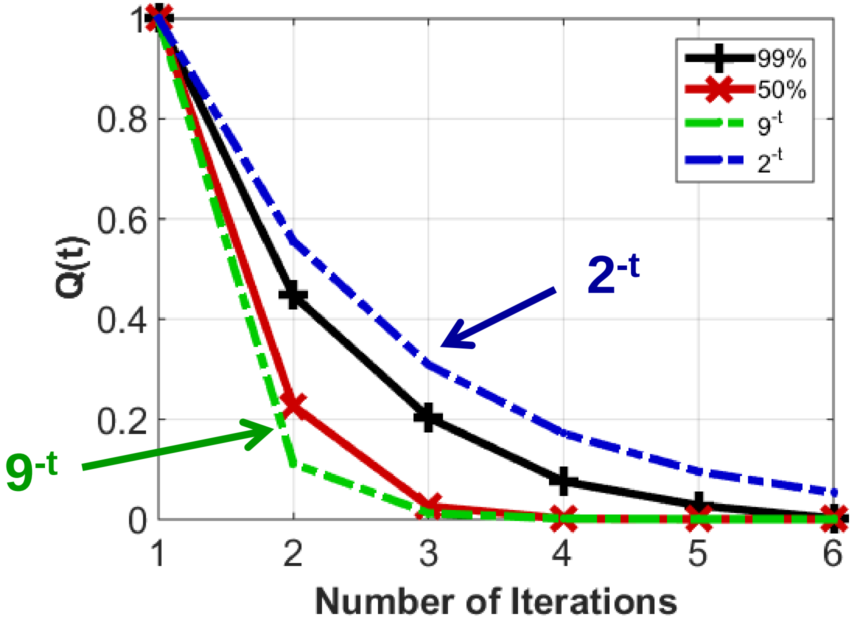
Modeling Optimality Gap

- **Optimality Gap (static scenario):**

$$Q(t) = C(t) - C^{OPT} \geq 0$$

Cost at iteration t → $C(t)$ **Optimal cost** → C^{OPT}

- **Curve fitting obtained using Geant network and 500 trials**



Modeling Optimality Gap

- **Optimality Gap (dynamic scenario):**
  - **Arrival** and **departure** of demands ➔ K(t): dynamic set of demands
  - Network **failures** and **recovery** ➔ G(t) = (N(t),E(t)) dynamic topology
- **Auto-regressive process of 1st kind**

$$Q(t+1) = (1-\eta)Q(t) + e \cdot A(t)$$

Improvement of the iterative solver ($(1-\eta)$)

Jump due to sub-opt. of FCS/FR (e)

Arrival process: ($A(t)$)
- **Demands arrival/departure**
- **Network failure/recovery**

$$E[A(t)] = \lambda$$

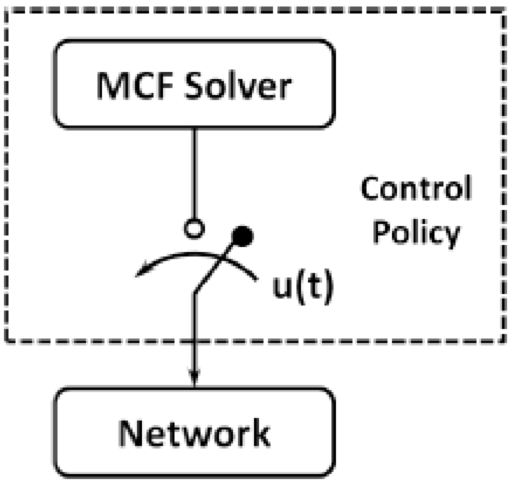
$$Var[A(t)] < \infty$$

- **Stability:**

$$\limsup_{T\to\infty} \frac{\sum_{t=0}^{T} E\left[Q(t)\right]}{T} = \frac{e\lambda}{\eta} < \infty$$

Control Policies

- **What if we do not want to reconfigure the network flows?**
  - **R_{max}**: maximum reconfiguration frequency
  - Compute a solution, but do not apply



- **Control:**
  - **u(t) = 1 (apply)**
  - **u(t) = 0 (do not apply)**

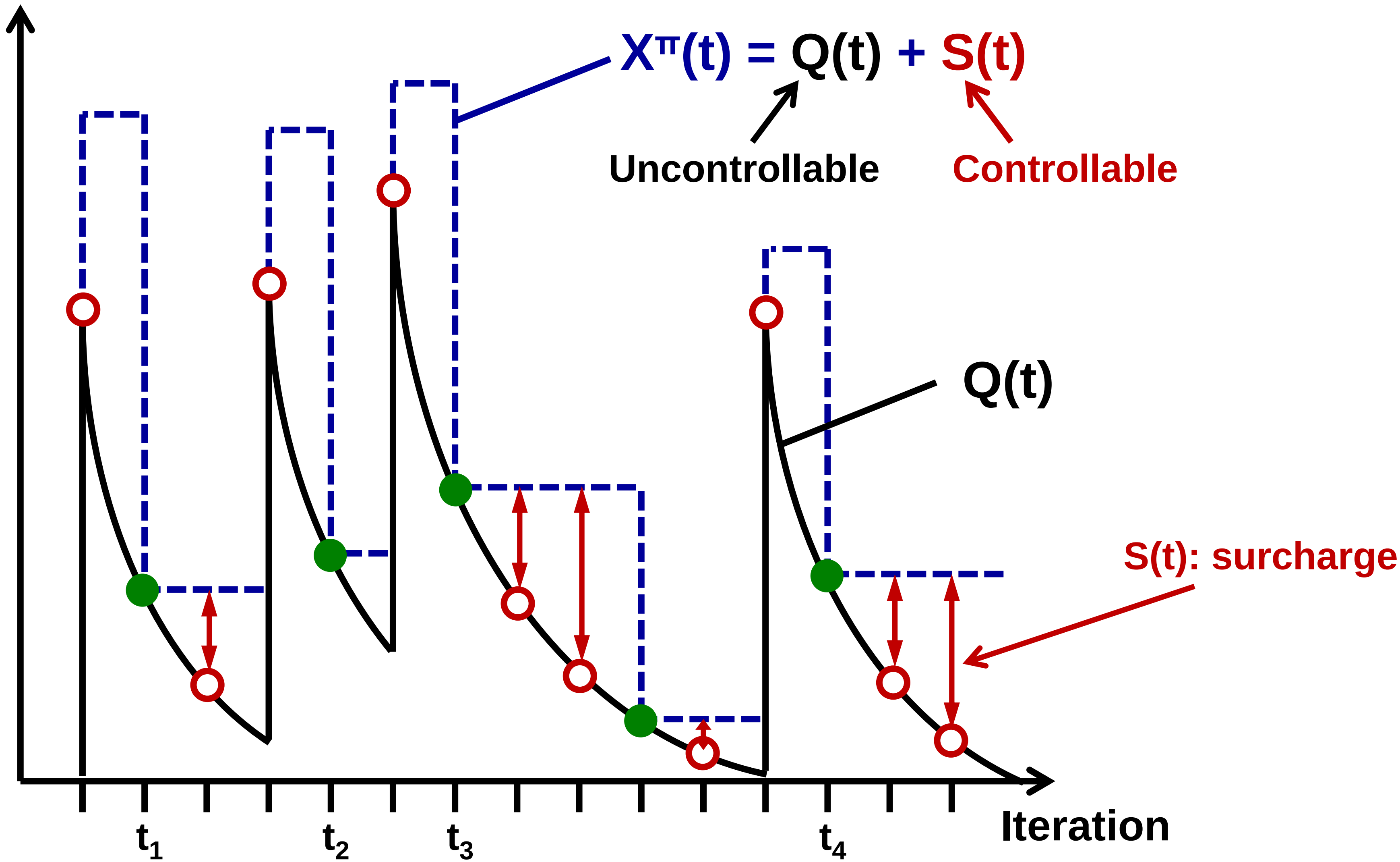
- **Operational cost of the control policy π**

$$X^{\pi}(t+1) = \underbrace{[(1-\eta)Q(t)]\cdot u^{\pi}(t)} + \underbrace{X^{\pi}(t)\cdot(1-u^{\pi}(t))} + e\cdot A(t)$$

Apply the flow configuration

Do not apply the flow configuration
➔ Operational cost accumulates

Control Policies – Operational Cost



Control Policies – Stochastic Optimization

- **Number of reconfigurations at time t:**

$$N_R^{\pi}(t) = \sum_{\tau=1}^{t} u^{\pi}(\tau)$$

- **Frequency of reconfigurations:**

$$\bar{R}^{\pi} = \limsup_{t\to\infty} \frac{E[N_R^{\pi}(t)]}{t} = \limsup_{t\to\infty} \frac{E[\sum_{\tau=1}^{t} u^{\pi}(\tau)]}{t}$$

- **Control Problem:**

$$\min \quad \limsup_{t\to\infty} \frac{\sum_{\tau=1}^{t} E[S^{\pi}(\tau)]}{t}$$ ⟵ **Operational cost**

$$s.t. \quad \bar{R}^{\pi} \le R_{\max}$$ ⟵ **Reconfiguration budget**

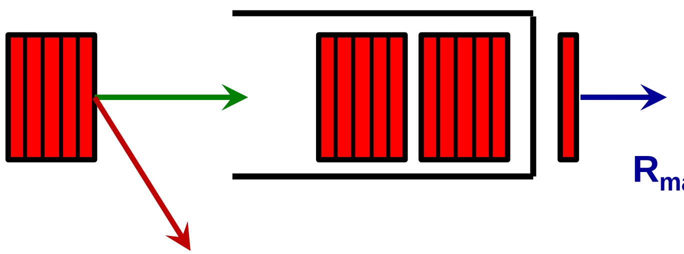
Control Policies – Virtual Queue

- **Virtual queue evolution (reconfiguration constraint)**

$$V(t+1) = \max\{V(t) - \underline{R_{\max}}, 0\} + \underline{u^{\pi}(t)}$$

Reconfiguration budget **Decision variable (0 or 1)**

u(t) = 1 ➔ add 1 big coin if apply



R_{max} ➔ always serve 1 small coin

u(t) = 0 ➔ do not add if not apply

- **Stability of the virtual queue ➔ reconfig. constraint is satisfied**
  - Drift definition:

$$\Delta(t) = E[V^2(t+1) - V^2(t) \mid X^{\pi}(t), Q(t), V(t)]$$

Control Policies – Drift-Plus-Penalty (DPP)

- **Minimization of the operational cost**
  - Penalty definition:

$$\delta(t) = E\,[\underline{X^{\pi}(t+1) - Q(t+1)} \mid X^{\pi}(t), Q(t), V(t)]$$

Surcharge S(t)

- **Minimization of the drift-plus-penalty metric**

$$\Delta(t) + K\delta(t) \;\leq\; u^{\pi}(t)\cdot[1 + 2V(t) - K(X^{\pi}(t) - Q(t+1))] + f(R_{\max}, V(t), X^{\pi}(t), Q(t))$$

  - Minimizing the drift ➔ Satisfying reconfiguration constraint
  - Minimizing the penalty ➔ Minimizing the surcharge (operational cost)
  - Guarantee the stability of the virtual queue

Drift-Plus-Penalty (DPP) Policy

- **At each iteration t, select u(t) as follows:**

$$u^{DPP}(t) = \begin{cases} 1 & V(t) < \dfrac{K\,(X^{DPP}(t) - Q(t+1)) - 1}{2} \\ 0 & \mathit{otherwise} \end{cases}$$

- **Simple policy**
  - Evolution of the cost/optimality gap (estimation)
  - Evolution of the surcharge
  - Counter for the number of times u(t) = 1

- **Initialization**
  - K set arbitrarily large
  - V(0) = 2K

Numerical Results – Simulator

- **Fast Connection Setup / Fast Recovery (FCS/FR):**
  - Shortest path on the network with residual capacity.
- **Iterative solver:**
  - Column generation + simplex (splittable MCF)
- **Three control policies:**
  - Drift Plus Penalty (DPP)
  - Random (RND) ➔ $u^{RND}(t) = 1 \ \textit{with probability} \ R_{\max}$
  - Periodic (Per) ➔ $u^{Per}(t) = 1 \ \textit{every} \ {}^{1}\!/_{R_{\max}} \ \textit{time slots (iterations)}$

Numerical Results – Settings

- **Geant as network topology:**
  - 22 nodes
  - 36 links (40 Gbps)

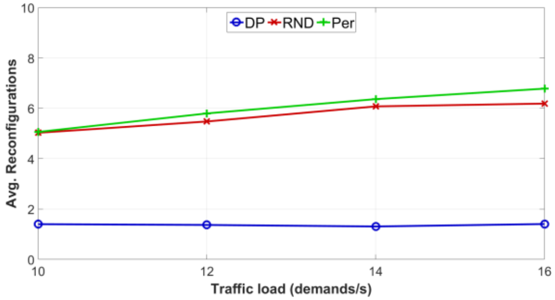
- **Events (demands and failures):**

	Demands	Failures
Arrival process	Poisson $\lambda = \{10,12,14,16\}$ dmd/s	Poisson $\lambda = \{1,1.2,1.4,1.6\}$ fail/s
Duration	10 s	5 s
Size	[5;8] GB/s	1 link

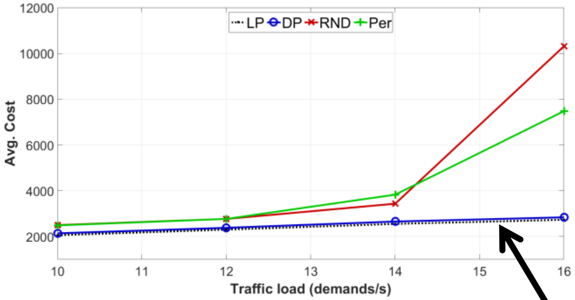
- **Reconfiguration budged ➔ R_{max} = 0.2**

Numerical Results – Without Failures

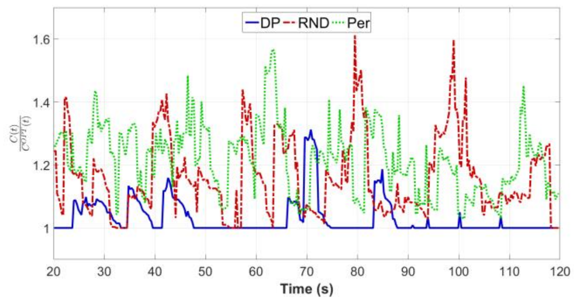
Average reconfigurations (links per demand)



Average cost



Cost evolution

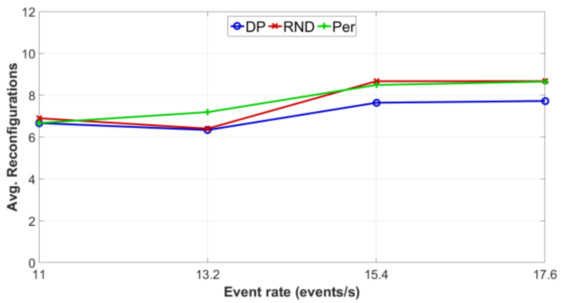


Always reconfigure (all iterations of CG)

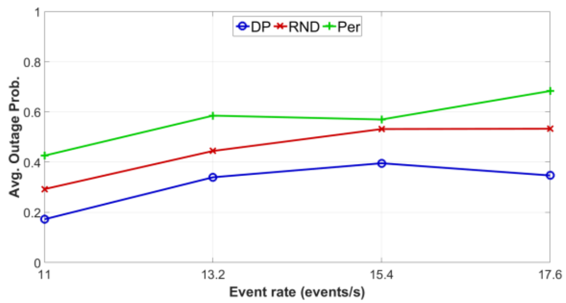
- **DP is very close to always reconfiguring the network**
  - All iterations of CG: u(t) = 1 for all t
- **DP requires less reconfigurations over time**
  - Less changes of the connections used for the demands
- **Cost evolution almost ideal**

Numerical Results – With Failures

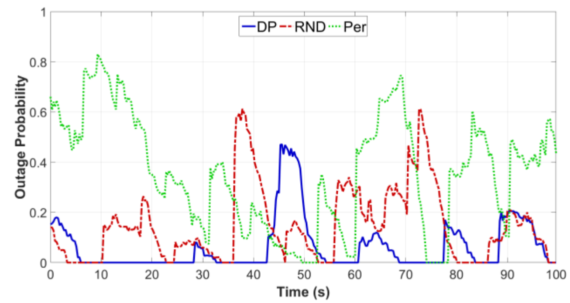
Average reconfigurations (links per demand)



Average outage prob.



Outage prob. evolution



Demands + Failures

- **Outage probability: ratio of dropped demands at every iteration.**
  - If a demand cannot be completely served, it is deactivated (high penalty in the LP).
- **DP uses the reconfiguration budget to increase the number of recovered connections**

HUAWEI

Conclusion

- **Proposition of a control framework for SDN controllers.**
  - Decouple global routing solvers from dynamic control.
  - Flexible.

- **Definition of a Drift-Plus-Penalty policy**
  - ***Robust*** against the relaxation of the model for Q(t) (AR model of the 1st kind).
  - Dynamic threshold on the benefit that pays back a reconfiguration (**simple** yet better than static threshold/periodic/random policies)
  - ***Easy* to extend** (different classes of events).

- **Open questions:**
  - Optimal policy unknown.
  - How to estimate Q(t).

Thank you

www.huawei.com