Learning how to segment flows in the dark

F. De Pellegrini[◇], L. Maggi[⋆], A. Massaro[◇],
J. Leguay[⋆], D. Sauchez[*] and E. Altman[*]

[◇]Fondazione Bruno Kessler [◇] Huawei [*]INRIA

PGMO Days 2017
November 14
EDF Lab Paris Saclay

Motivation

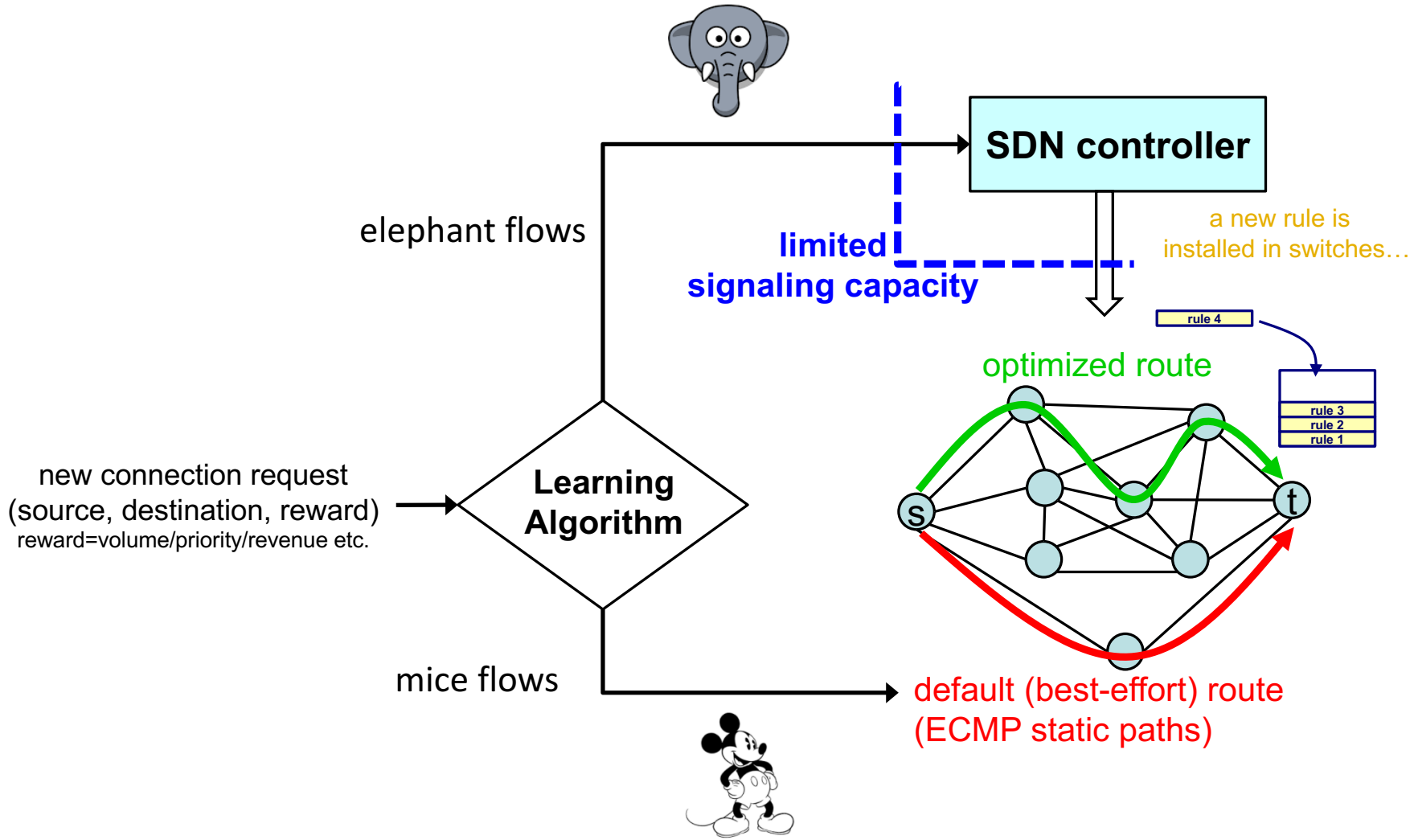
- **SDN controllers:** de-facto production tool for routing traffic in datacenters
  - peaks of several millions of `packet_in` events per second [1]
  - challenging the **control channel capacity**
- **Traffic sources:** flows originate from racks attached to origin switches
- **Datacenter routing:** destination in a different rack, attached to a destination leaf switch
- **Fat-Tree topologies:** equal-cost multi-path (ECMP) routing, hash-based flow load balancing
- **Wildcards:** no `packet_in` control message sent to the controller

Flow segmentation

- **Elephant flows:** less than 10% carry more than 80% of the entire traffic
  - E.g. elephant flows generated by *Map-Reduce* [2] or *Spark* [3].
- **Flow segmentation:** routing flows using
  - either an *optimized* route: send a `packet_in` signal to the controller
  - or a *default* route: use a wildcard and resort to default routes
- **Customary solution:** default ECMP + optimized routing + a static flow-size threshold (elephant/mice)

Tradeoff: serve each and every flow via optimized routes (control channel constraints) vs all default routes (mice vs elephants)

Flow segmentation



System model

- **Network:** datacenter fabric with a set of leaf origin/destination switches + one controller
- **Flow sizes:** $\mathscr{R}$ sorted in decreasing order $r_1 > r_2 > \cdots > r_N$
- **Traffic model:** flows of size r_j are Poisson with intensity λ_j;

$$\lambda = \sum_{j \in \mathscr{R}} \lambda_j, \qquad p_j = \frac{\lambda_j}{\lambda}$$

- **Flow classifier:** associates a flow size to the tagged flow [4, 5, 6, 7];
  - ideal versus not ideal classifiers

Control Channel Constraint

Maximum `packet_in` rate: maximum `packet_in` events per second

1. control channel capacity (switches to controller)
2. new path calculation
3. routes installation on all switches along the route

$$\Longrightarrow c \cdot \lambda \texttt{ packet-in} \text{ requests per second; } c \in [0; 1].$$

Note: c may depend on the overall flow arrival rate, as well as on the congestion and computation capabilities of the controller

Optimization Problem

- **Events:** at $t^k{}_{k\in\mathbb{N}}$ origin switch $i^k \in \mathcal{S}$ detects a new flow arrival with size $r^k \in \mathscr{R}$.
- **Segmentation action:** stochastic control to admit or reject the `packet_in` generation, $u^k \in \{0, 1\}$

$$\max_{u} \limsup_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[u^k\, r^k\right] \tag{1}$$

- **Control channel constraint:**

$$\limsup_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[u^k\right] \leq c. \tag{2}$$

MDP and Fractional Knapsack

- **Constrained Markov decision process:** state is the current flow requesting admission
- **Stationary solution:** a stationary optimal policy exist randomized in at most one tap
- **Segmentation policy** u**:** probability that a switch interrogates the controller when flow of size r_j is detected

$$\max_{u \in [0,1]^N} \sum_{j \in \mathscr{R}} u_j p_j r_j \tag{3}$$

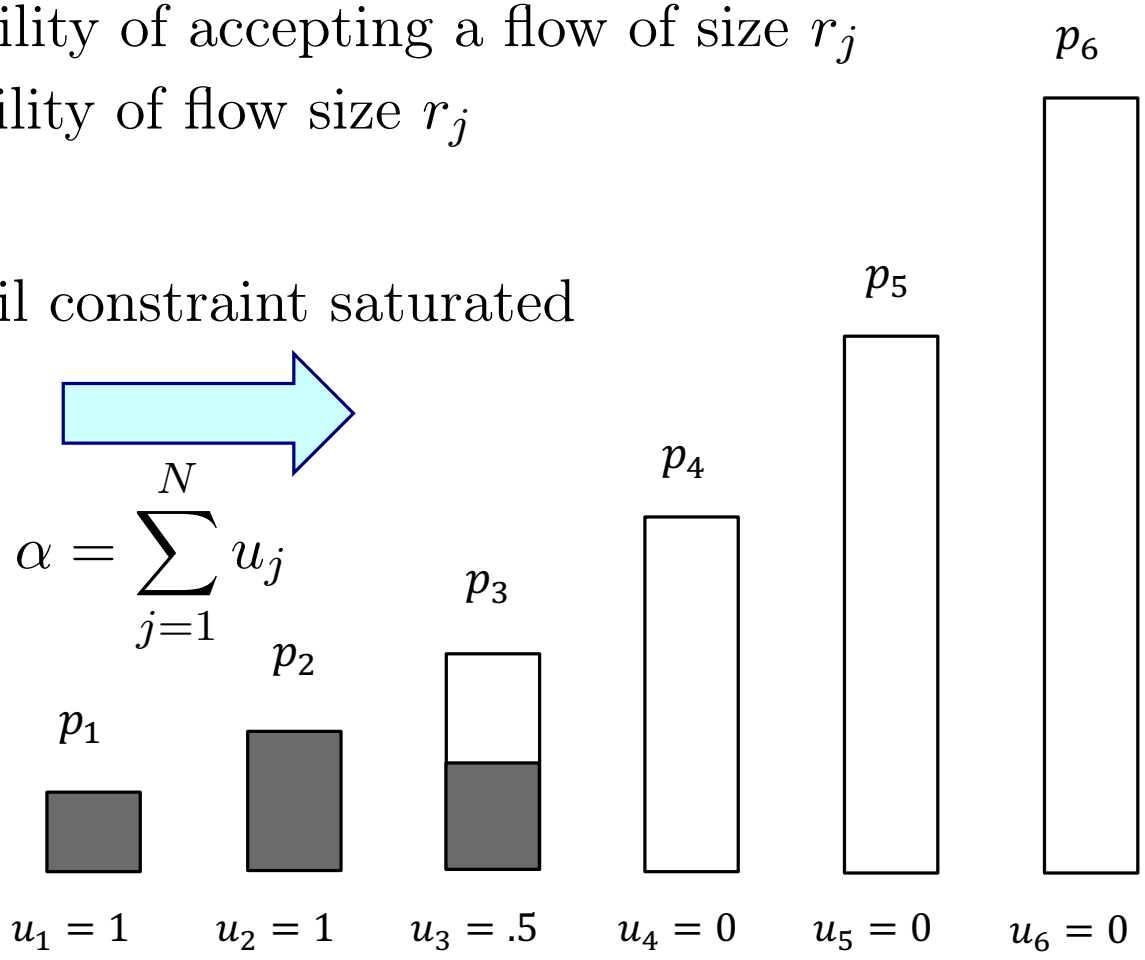
$$\text{s.t.} \sum_{j \in \mathscr{R}} u_j p_j \leq c \tag{4}$$

Offline solution: waterfilling

u_j: probability of accepting a flow of size r_j

p_j: probability of flow size r_j

Fill up until constraint saturated



$$\sum_{j=1}^{N} u_j(\alpha^*)p_j = c$$

Offline solution

Optimal segmentation policy: classic threshold-type solution (Dantzig) depending on a threshold

$$u_j(\alpha^*) = \begin{cases} 1 & j \leq \lfloor \alpha^* \rfloor \\ \alpha^* - \lfloor \alpha^* \rfloor & j = \lfloor \alpha^* \rfloor + 1 \\ 0 & j \geq \lfloor \alpha^* \rfloor + 2 \end{cases} \tag{5}$$

Optimal segmentation policy: α^* solves the equation $\theta(\cdot) = c$, where:

$$\theta(\alpha) := \sum_{j=1}^{\lfloor \alpha \rfloor} p_j + (\alpha - \lfloor \alpha \rfloor) \cdot p_{\lfloor \alpha \rfloor + 1}. \tag{6}$$

Requirements

Blind and online: SDN controller and the switches are agnostic to the flow size distribution $\{p_j\}$

Some viable options:

- **Type-based solution**: send back the histogram per switch, extra traffic
- **Lyapunov technique:** drift-plus-penalty (DPP), requires thresholds at the switches [8]

Proposed solution: stochastic approximation learning algorithm: low signaling traffic, uses the aggregated flow distribution, converges to the optimal flow segmentation policy[1]

[1] F. De Pellegrini, L. Maggi, A. Massaro, J. Leguey, D. Sauchez, and E. Altman, "Blind, adaptive and robust flow segmentation in datacenters," in Proc. of IEEE INFOCOM (to appear.), Honolulu, US, April 2018.

Stochastic Online Flow Segmentation Algorithm (SOFIA)

- **Round-based:** rounds with fixed time duration T
- **Switch counters**: switch i counts admitted $A_i^n(t)$ and rejected $R_i^n(t)$ flows in $[nT, nT + t]$
- **Reporting**: report to controller after a random backoff time τ_i^n
- **Aggregation:** controller aggregates the counters sent by switches before the deadline and has

$$\mathbb{E}\left[\bar{Y}^n(\alpha^n)\right] = \theta(\alpha^n) \tag{7}$$

- **Basic iteration:**

$$\alpha^{n+1} = \Pi\left[\alpha^n + \epsilon^n\left(c - \bar{Y}^n\right)\right] \tag{8}$$

with projection $\Pi(x) = \max\{0, \min\{N, x\}\}$

Correctness

Theorem

Let the sequence $\{\epsilon^n\}$ *be such that* $\epsilon^n \geq 0 \; \forall \, n$, $\sum_{n=0}^{+\infty} \epsilon^n = +\infty$ *and* $\sum_{n=0}^{+\infty} (\epsilon^n)^2 < +\infty$. *Then, the policy* $u(\alpha^n)$ *converges to the optimal policy* $u(\alpha^*)$ *with probability* 1.

- Robinson-Monroe type of algorithm [9]
- Convergence proof: ODE method

$$\dot{\alpha} = c - \theta(\alpha) = c - \sum_{j=1}^{\lfloor \alpha \rfloor} p_j + (\alpha - \lfloor \alpha \rfloor) \cdot p_{\lfloor \alpha \rfloor + 1}$$

- Unique asymptotically stable restpoint α^*
- Estimates of `SOFIA`: confined in a small neighborhood of α^*, escaping a finite number of times

Message complexity

Define:

$\epsilon^n := n^{-\gamma}$: standard stepsize where $1/2 < \gamma \leq 1$ and $\eta > 0$
$\Delta^n := \frac{|\theta(\alpha^n) - c|}{c}$: relative error of `SOFIA` at the n-th round
p_f: message error probability

Corollary

Fix $0 < P_\eta < 1$. In order to attain $\mathbb{P}\{\Delta^n > \eta\} < P_\eta$, `SOFIA` generates

$$O\left(\frac{|\mathcal{S}|}{(1-p_f)\nu}\log\left(\frac{\beta}{\zeta P_\eta}\right)\right)$$

message transmissions on the control channel.

Note: the constant appearing in the message complexity depends on the choice of the round duration T

Adaptive solution

- Problem: decreasing stepsize formulation of `SOFIA` (e.g., $\epsilon^n = \epsilon_0 n^{-\gamma}$) against changes in the flow size distribution
- Solution: sacrifice noise-rejection properties, use step size $\epsilon^n = \epsilon$

$$\alpha^{n+1} = \Pi\Big(\alpha^n + \epsilon(c - \bar{Y}^n)\Big) \tag{9}$$

Theorem

For any $\delta > 0$, define by $B_\delta(\alpha^*) = \{x \in \mathbb{R} : |x - \alpha^*| < \delta\}$. As $\epsilon \to 0$, sequence $\{\alpha^n\}_n$ computed as in (9) converges in distribution to elements in $B_\delta(\alpha^*)$. The fraction of time spent by the process in $B_\delta(\alpha^*)$ during $[0, t]$ goes to 1 as t diverges.

Robustness to Misclassification

- **Confusion matrix:** $P_{ij} = \mathbb{P}\{\text{flow classified } r_j \mid \text{actual size } r_i\}$
- Optimization changes accordingly:

$$W^*(P) = \max_{a \in [0;1]^N} \quad \sum_{j=1}^{N} u_j \widehat{p}_j \widehat{r}_j \qquad (10)$$
$$\text{s.t.} \quad \sum_{j=1}^{N} u_j \widehat{p}_j \leq c$$

- **Modified system:** $\widehat{p}_j = \sum_{i=1}^{N} p_i P_{ij}, \quad \widehat{r}_j = \sum_{i=1}^{N} r_i \bar{P}_{ji}$
- **Optimal policy:** still threshold structure as long as we consider a permutation of flows indexes

$$\widehat{r}_{\sigma(i)} > \widehat{r}_{\sigma(j)} \text{ whenever } \sigma(i) < \sigma(j)$$

- As long as the classifier is **order preserving**, `SOFIA` solves (10)

Robustness to Misclassification (cont'ed)

- **Good news:** for $N = 2$, P is order preserving iff $P_{12} + P_{21} \leq 1$
- **Bad news:** for $N \geq 3$, no classifier is order-preserving for all flow size distributions
- **Delayed feedback:** learn the expected misclassified flow sizes $\widehat{r}$ over time

Lemma

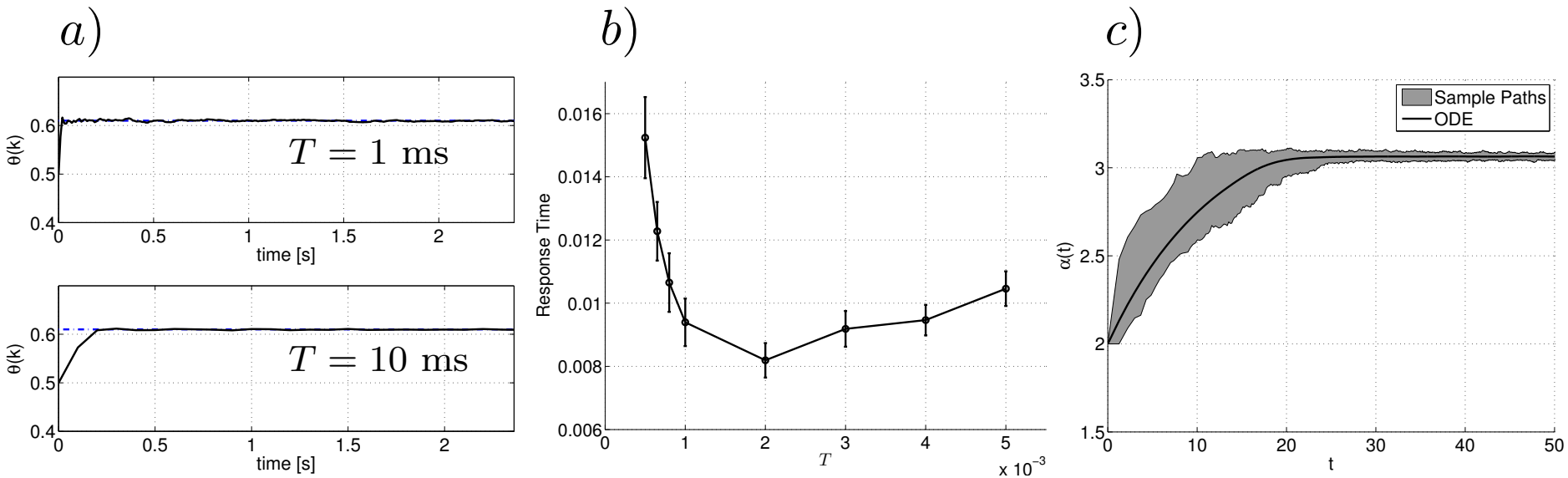
The flow segmentation policy $u(\alpha^n, \sigma^n)$ produced by `SOFIA` *under delayed feedback converges to the optimal policy $u(\widehat{\alpha}, \sigma)$ with probability 1.*

Note: flow size monitoring can occur at slow timescale

Robust Version R-SOFIA

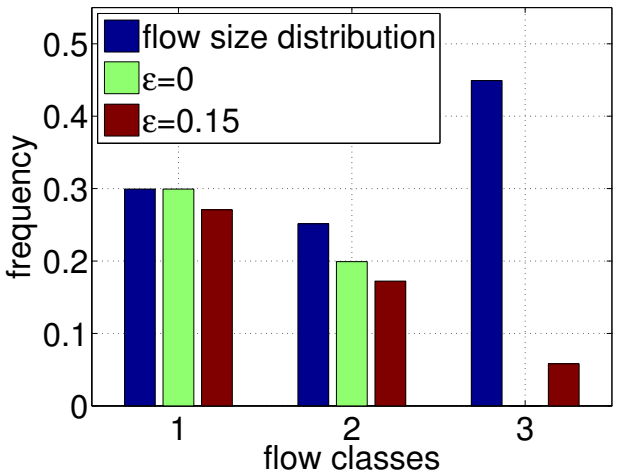
1: **input:** T, $\{\epsilon^n = \epsilon_0 n^{-\gamma}\}_n$, $\epsilon_0 > 0$, $\gamma \in (1/2; 1]$
2: **initialize:** $\alpha_0 \leftarrow 1$, $\sigma^1 \leftarrow \sigma^I$
3: **for** rounds $n = 0, 1, \ldots$ **do**
4: At time nT the controller **broadcasts new threshold** α^n **and flow permutation** σ^n to all switches. Each switch will adopt segmentation policy $u(\alpha^n, \sigma^n)$ during time interval $[nT, (n+1)T)$
5: Each switch $i \in \mathcal{S}$ waits for a random time τ_i^n and then sends to the controller the quantities $A_i^n(\tau_i^n)$ and $R_i^n(\tau_i^n)$
6: At time $(n+1)T$ the controller computes the portion of admitted flows $\bar{Y}^n(\alpha^n)$ during round n
7: Controller computes $\alpha^{n+1} \leftarrow \Pi\Big(\alpha^n + \epsilon^n(c - \bar{Y}^n)\Big)$
8: Controller **monitors flows** in the network and produces the **estimate** $\widehat{r}_j^{n+1}$ **as the average size of flows that are classified as belonging to class** j **up to time** $(n+1)T$. It then computes the **new class permutation** σ^n **that sorts** $\widehat{r}^{n+1}$ **in decreasing order**
9: **end for**

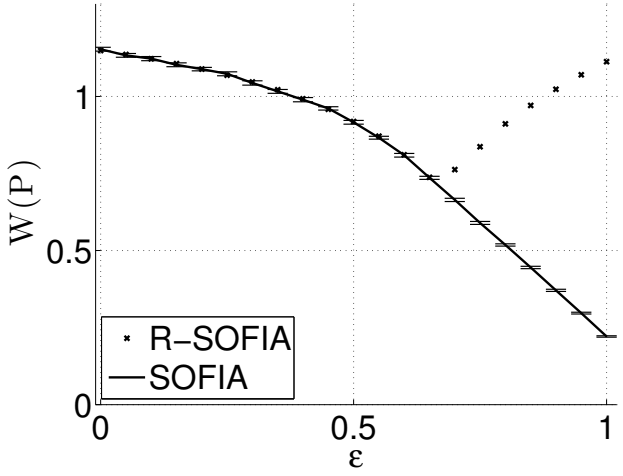
Numerical results

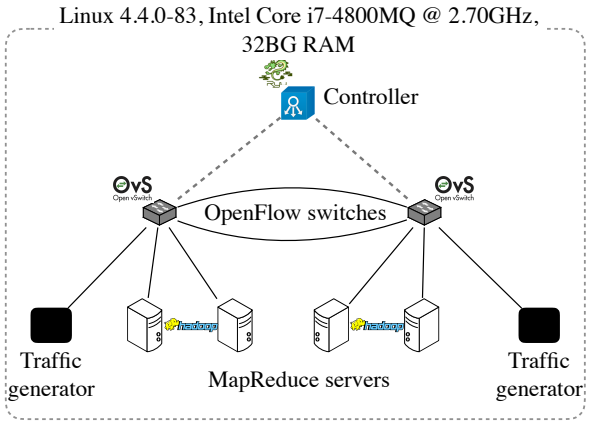


a) Sample paths of `SOFIA` for decreasing step size
b) Convergence time for $\eta = 0.05$
c) Sample paths versus ODE dynamics;

Numerical results (cont'd)

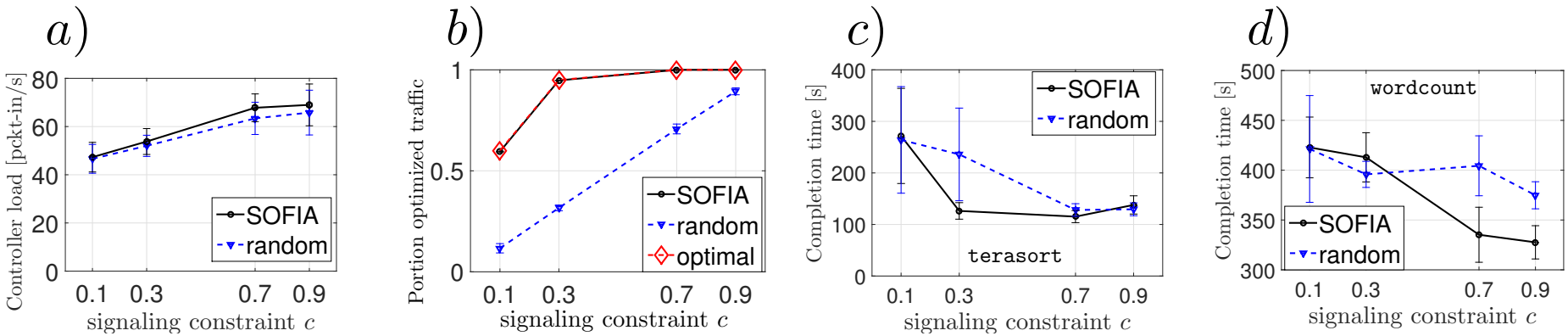






a) Effect of classification errors on the optimal policy;
b) Performance loss for increasing values of classification error ε
c) Cluster used for the network emulation.

Numerical results (cont'ed)



a) Maximum controller load

b) Average portion of optimized background volume

c) Completion time under `terasort` vs. signaling constraint

d) Completion time under `wordcount`

Conclusions

- Flow segmentation: reduces the peak rate of `packet_in` events to match control channel constraints
- Flow distribution unknown, flow sizes estimated using classifiers
- Optimal segmentation: optimal threshold for admission of `packet_in` events (fractional knapsack)
- `SOFIA`: stochastic approximation agnostic of flow size distribution
- `R-SOFIA`: variant provably robust to classification errors
- Current: MDP formulation accounting for memory constraints

Bibliography I

[1] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. ACM IMC*, Melbourne, Australia, November 1-3, 2010.

[2] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Comm. of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[3] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. USENIX HotCloud*, 2010.

[4] M. Moshref, M. Yu, R. Govindan, and A. Vahdat, "DREAM: dynamic resource allocation for software-defined measurement," *ACM SIGCOMM Computer Comm. Review*, vol. 44, no. 4, pp. 419–430, 2015.

[5] M. Malboubi, L. Wang, C. N. Chuah, and P. Sharma, "Intelligent SDN based traffic (de)Aggregation and Measurement Paradigm (iSTAMP)," in *Proc. IEEE INFOCOM*, 2014.

[6] M. Yu, L. Jose, and R. Miao, "Software Defined Traffic Measurement with OpenSketch," in *Proc. USENIX NSDI*, 2013.

[7] B. Claise, "Cisco systems NetFlow services export version 9," 2004.

[8] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Comm. Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[9] H. J. Kushner and G. G. Yin", *Stochastic Approximation and Recursive Algorithms and Applications"*. Springer, 2nd Edition, 2003.