Uncertainty-driven Ensemble Forecasting of QoS in Software Defined Networks

Kostas Kolomvatsos[1], Christos Anagnostopoulos[2], Angelos Marnerides[3], Qiang Ni[3], Stathes Hadjiefhtymiades[4], Dimitrios Pezaros[2]

[1]University of Thessaly, [2]University of Glasgow, [3]University of Lancaster, [4]University of Athens

22nd IEEE Symposium on Computers and Communications

03-06 July 2017, Heraclion, Crete, Greece

Outline

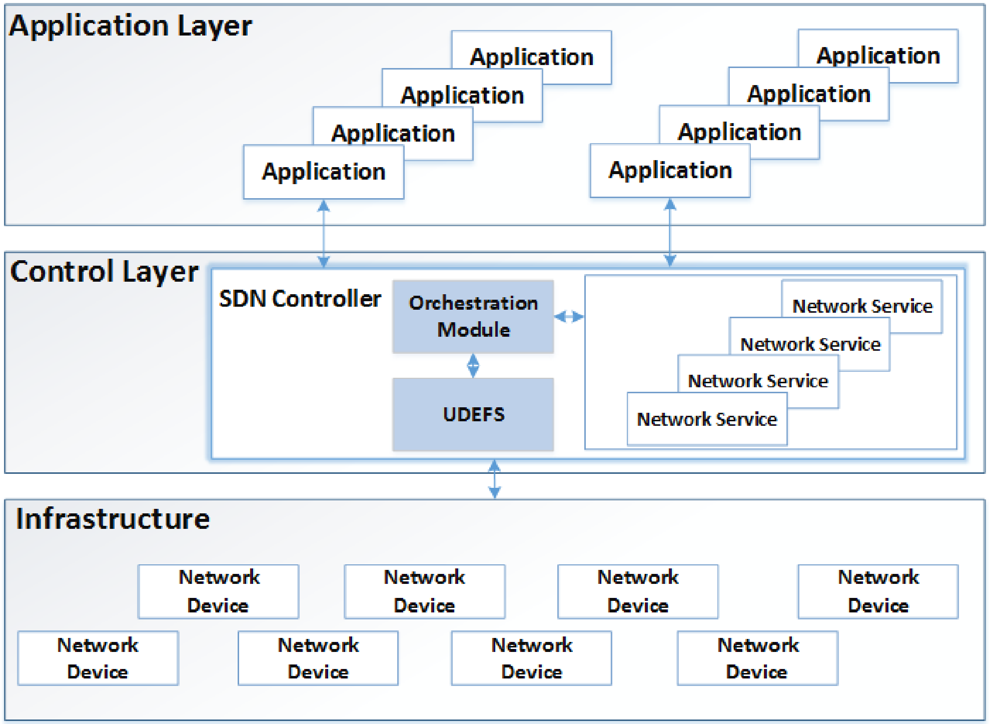
- SDNs Management
- The proposed framework
- The ensemble forecasting scheme
- The uncertainty management mechanism
- Experimental evaluation

SDNs Management (1/2)

- SDN controllers are responsible for performing various network tasks
- They are connected to applications through the northbound interface and to devices through the southbound interface
- They incorporate monitoring functionalities to collect time-series network performance data
- Example metrics: latency, link utilization, switch buffer occupancy, etc

SDNs Management (2/2)

- During the functioning the amount of data becomes huge
- We want to derive predictive analytics on top of the data
- Our aim is to use the analytics to secure the QoS



The proposed framework

- We manage various ***Network Performance Parameters* (NPPs)**
- We propose a module that supports real time decision making
- Our module involves:
  - A Type-2 Fuzzy Logic System (T2FLS) for uncertainty management (it derives alerts about the presence of QoS violation events)
  - The T2FLS derives the ***Potential of Violation* (PoV)**
  - A combination of responses derived through large-scale predictive analytics
  - A combination of multiple aggregated time series forecasting results
- Our goal:
  - Provide forecasting analytics to the SDN controller
  - For each NPP we provide an aggregated value
  - The T2FLS informs the SDN controller for the presence of an event

The ensemble forecasting scheme (1/3)

- We consider a set of estimators
- Examples: auto-regressive estimator, double and triple exponential smoothing, weighted and cumulative moving average, etc
- We adopt 28 estimators
- The ensemble scheme involves the aggregation of multiple estimators for each NPP
- The final aggregated value is derived through an aggregation function on top of historical values

$$\hat{e} = f(e_1, \ldots, e_{|\mathcal{E}|})$$

The ensemble forecasting scheme (2/3)

- We adopt a linear aggregation function

$$\hat{e} = \sum_{i=1}^{|\mathcal{E}|} w_i e_i \qquad \sum_i w_i = 1$$

- Each estimator has a specific weight
- Our model produces a vector of aggregated estimations (a value for each NPP)
- Weights are defined based on estimators' performance
- The performance is affected by the estimation error $|\hat{e}_i^t - r_k^t|$
- $\hat{e}_i^t$ is the estimation and r_k^t is the real observation for the k-th NPP

The ensemble forecasting scheme (3/3)

- We adopt a sliding window approach
- We determine the weight of an estimator based on its performance in the window
- Weights are based on the average forecasting error μ_i
- We define a convex combination rule for weights definition

$$w_i = \frac{1-\mu_i}{\sum_{j=1}^{|\mathcal{E}|} 1-\mu_j}$$

- The mechanism assigns high weight to estimators with low average error

Uncertainty management (1/2)

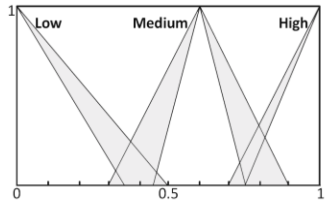
- We want to ‘fire’ the update on the orchestration process of the SDN controller
- Uncertainty is present on how the aggregated estimation depicts a high potential of QoS violation
- We propose the T2FLS for such purposes
- The T2FLS linearly maps the inputs to the outputs
- It adopts as et of rules

$$R_j\text{: If } u_{1j} \text{ is } A_{1j} \text{ and/or } u_{2j} \text{ is } A_{2j} \text{ and/or ... and/or } u_{lj} \text{ is } A_{lj} \text{ Then } v_{1j} \text{ is } B_{1j} \text{ and ... and } v_{zj} \text{ is } B_{zj}$$

- u_{ij} are the inputs, v_{kj} are the outputs and A_{ij} and B_{kj} are membership functions

Uncertainty management (2/2)

- Membership functions in Type-2 FLSs are intervals



- Our T2FLS has |M| inputs; M is the set of NPPs
- Each input corresponds to the aggregated measurement for an NPP
- We consider three linguistic values: Low, Medium, High
- The output is the PoV
- When PoV is over a pre-defined threshold the T2FLS fires an event to the SDN controller

Experimental evaluation (1/3)

- Experimental setup
  - We focus on: link utilization β, average switches buffer size ε and average latency α
  - We adopt two distributions to produce values for each metric: Uniform, Exponential (λ=0.5, λ=2.0)
  - We adopt three decision thresholds: β_T, ε_T, α_T
  - When NPPs violate the thresholds, there is an indication of QoS violation
  - We adopt known performance metrics for our model like:

- Precision

$$\pi = \frac{TP}{TP+FP}$$

- Recall

$$\rho = \frac{TP}{TP+FN}$$

- Accuracy

$$\psi = \frac{TP+TN}{TP+TN+FP+FN}$$

- F-measure

$$\phi = 2\frac{\pi \cdot \rho}{\pi + \rho}$$

TP: true-positive, TN: true-negative, FP: false-positive, FN: false-negative

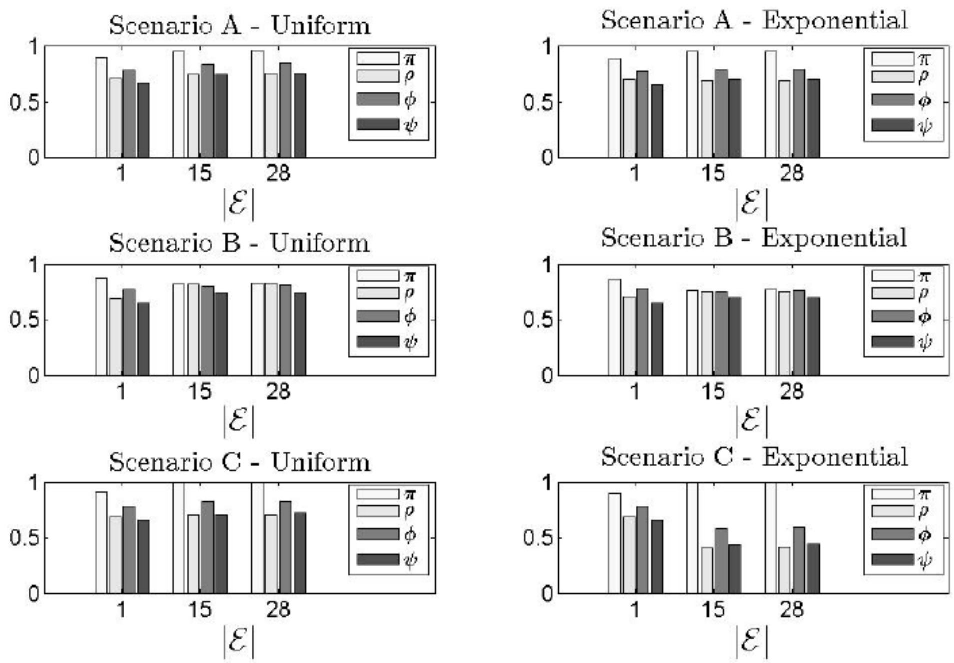
Experimental evaluation (2/3)

- We run 1,000 simulations for 1,000 rounds per simulation
- At t, we collect the realization for NPPs and execute the proposed scheme
- Three scenarios are evaluated:
  - Scenario A: $\beta_T = \varepsilon_T = \alpha_T = 0.5$
  - Scenario B: $\beta_T = 0.3$, $\varepsilon_T = \alpha_T = 0.7$
  - Scenario C: $\beta_T = 0.7$, $\varepsilon_T = \alpha_T = 0.3$

Experimental evaluation (3/3)

- Results ($|\mathcal{E}|$ is the number of estimators)

	Scenario A			Scenario B			Scenario C		
	Uniform	Exponential		Uniform	Exponential		Uniform	Exponential	
		λ=0.5	λ=2.0		λ=0.5	λ=2.0		λ=0.5	λ=2.0
π	0.96	0.95	0.93	0.79	0.77	0.72	1.00	1.00	1.00
ρ	0.76	0.69	0.45	0.83	0.75	0.46	0.71	0.64	0.42
ϕ	0.85	0.80	0.61	0.81	0.76	0.56	0.83	0.78	0.59
ψ	0.76	0.70	0.50	0.74	0.70	0.54	0.72	0.65	0.44



Thank you for your attention!