Applied Mathematics & Information Sciences 5 (2) (2011), 61S-68S
– An International Journal

Performance monitoring methods of ForCES architecture

Zhou Jingjing
College of Electronics & Information Engineering, Zhejiang Gongshang University, China
Email Address: zhoujingjing@zjgsu.edu.cn

Received June 22, 2010; Revised January 21, 2011

The new generation open programmable architecture of network components, ForCES (Forwarding and Control Elements Separation), has efficient, open, distributed, modular and programmable features. ForCES technology and even open architecture network have developed rapidly. However, the research for ForCES architecture mainly focuses on the functions and standards at present. To ensure ForCES architecture can provide users with a more efficient computing environment, we need to monitor the real-time running situation of a ForCES architecture network to provide the basis for performance evaluation and optimization. We propose to use the SRFsTME algorithm to estimate the traffic matrix to monitor the ForCES architecture network. Simulation results show that the SRFsTME algorithm is suitable for ForCES architecture.

Keywords: traffic matrix, Kalman Filtering, ForCES architecture.

1 Introduction

The new generation open programmable architecture of network components, ForCES (Forwarding and Control Elements Separation), has efficient, open, distributed, modular and programmable features. In the context of accelerating the development of triple play, which replaces the traditional network products, becomes a trend. Currently industrial and research institutions are exploring the use of distributed architecture to improve the handling capacity of network devices. In this regard internationally new types of network architectures are emerging. New network architecture needs to model the various parts of the network components. Communication between modules uses an open, unified interface. Scalability and programmability of network are improved apparently.

ForCES is considered to be widely used in the Next-Generation Network (NGN). Our ForCES workgroup has studied related ForCES protocols for 8 years, which is one of the core strengths in the IETF ForCES working groups. Our workgroup is working at

studying architecture [1] and protocol standards [2] of the open programmable IP router and provides framework and guidance to the openness, programmability and high re-configurability for NGN. In ForCES architecture protocol messages are transmitted between CE (Control Element) and FE (Forwarding Element). The ForCES architecture can control multiple FE and CE in parallel and this can improve the scalability, flexibility and internal operability of the router.

In recent years ForCES technology and even open architecture network have developed rapidly, but achieving research goals, network efficiency, is still a difficult task. Next-generation network requires a higher network computing performance and packet forwarding performance of the network devices. However, the research for ForCES architecture mainly focuses on the functions and standards at present. To ensure ForCES architecture can provide users with more efficient computing environment, we need to monitor the real-time running situation of ForCES architecture network and provide the basis for performance evaluation and optimization.

This paper firstly introduces the concept of traffic matrix to monitor the global traffic of ForCES architecture. A traffic matrix describes the amount of traffic that flows between every possible pairs of origin and destination nodes in a network. The traffic matrix is the perspective of the whole network flow. It can clearly reflect the flows on each path of ForCES architecture. It is an important means of network monitoring and can be used for performance evaluation and optimization. However, direct measurements to obtain traffic matrices are prohibitively expensive [3] and so many estimation methods based on some available information are used to populate the traffic matrix. With the development of estimation methods and the continuous improvement of estimation accuracy, traffic matrix has been gradually applied to the various network fields (traffic engineering, network management, network security etc.). All these have a far-reaching impact on the development of network technology.

This paper is organized as follows. Section 2 briefly introduces related works about traffic matrix estimation and ForCES architecture. In Section 3 we introduce the SRFsTME algorithm. In Section 4 we test the SRFsTME algorithm performance in the ForCES architecture network. Section 5 presents our concluding remarks.

2 Related works

2.1 Traffic matrix Estimation

Many of the previous work on IP traffic matrix estimation have incorporated prior information to solve the ill-posed linear inverse problem. With recent advances in flow monitoring techniques, TM can somewhat be obtained by direct measurement, but [4] shows that the communications overhead is still too large. Hence the authors propose the

use of flow monitoring in a lightweight fashion -- that of turning flow monitors on only when (in time) and where (which nodes) needed. Based on this idea, the new generation methods, such as fanout [4], PCA [5] and Kalman Filtering [5], are proposed. Soule et al. [5] introduces the PCA method which relies on Principal Component Analysis (PCA). The final method makes use of Kalman Filtering techniques and is hence referred to as the Kalman Filtering method, which is also introduced in the paper [4] for the first time.

2.2 ForCES Architecture

According to ForCES framework (RFC3746) **Error! Reference source not found.**, the ForCES architecture [1] is shown in Figure 2.1.

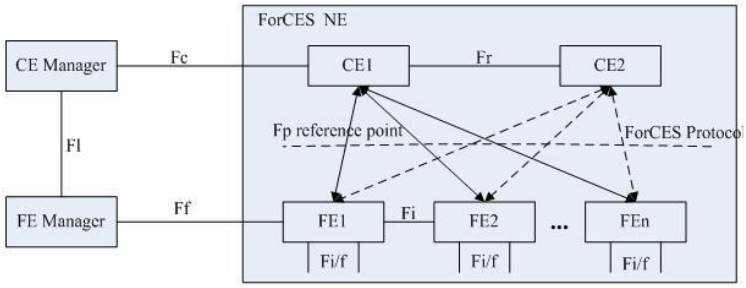

Figure 2.1: ForCES Architecture

The ForCES architecture defines the ForCES components. These components may be connected in different kinds of topologies for flexible packet processing.

Figure 2.1 shows the logical components of the ForCES architecture and their relationships. The ForCES includes two kinds of components inside the network element (NE): CE and FE. FE is the forwarding layer (or called the data layer) of the router. It processes each data packet specifically, such as classifiers, schedules, routes and so on. CE is the control layer of the router. In order to manage the router, CE sends a series of messages defined by the ForCES protocol to FE. For example, CE changes the classification rules of a data packet and changes the scheduling strategies. FE cannot modify router attributes. It is controlled by CE entirely. According to demand CE can add or delete FE, configure the properties of the various functional modules in FE and so on. The ForCES router has a high flexibility which the router of traditional architecture has not.

Figure 2.1 shows all the important reference points. The ForCES Protocol is defined on the reference point, Fp. Fi is the interface between FEs, Fr is the interface between CEs and Fi/f is the external interface of FEs.

3 SRFsTME Algorithms

When one uses the Kalman Filtering method to estimate the traffic matrix, we find that the error covariance calculation components of the Kalman Filtering are difficult to implement in realistic network systems, the numerical difficulties have occurred in

problems with singular and nearly singular estimation covariance matrices because of the ill-conditioning in the MATLAB environments. To cope with these ill-conditioning situations of Kalman Filtering, we modify or replace the algorithm with one numerically better-conditioned model. The Square Root Filtering is such a solution and it has inherently better stability and numerical accuracy than the Kalman Filtering. We design the SRFsTME algorithm [7] based on the Square root factorization [8], which involves a "square root" of the error covariance matrix and corresponds to the factorization of the Kalman Filtering/Smoothing algorithm. Our SRFsTME algorithm is implemented in a form of feedback control through the following two filtering steps and one smoothing step as follows:

(1) Filtering Prediction Step: We use the MWGS (Modified Weighted Gram-Schmidt Orthogonalization) to implement the traffic matrix prediction to replace that of the standard Kalman Filtering. It is an alternate algorithm with better overall numerical stability and accuracy. The standard forms of the prediction step and its covariance matrix can be rewritten as follows:

$$x_{t+1} = Cx_t + Gw_t \tag{3.1}$$

$$P(-) = CP(+)C^T + GQG^T \tag{3.2}$$

where G is the coefficient of the noise process. Eq. (3.3) uses triangularization of the Q matrix (if it is not yet diagonal) in the form $Q= GD_QG^T$, where D_Q is a diagonal matrix. Define

$$W = \begin{bmatrix} U_{k-1}^T(+)C_k^T \\ G_k^T \end{bmatrix} \qquad D_w = \begin{bmatrix} D_{k-1}(+) & 0 \\ 0 & D_{Q_k} \end{bmatrix} \tag{3.3}$$

Then the MWGS orthogonalization procedure produces a unit lower triangular $n \times n$ matrix L and a diagonal matrix D_β such that

$$\begin{aligned} &W = BL \\ &L^T D_\beta L = L^T B^T D_w BL = (BL)^T D_w BL = W^T D_w W \\ &\quad = \left[C_{k-1}U_{k-1}(+)G_{k-1} \right] \begin{bmatrix} D_{k-1}(+) & 0 \\ 0 & D_{Q_{k-1}} \end{bmatrix} \begin{bmatrix} U_{k-1}^T(+)C_{k-1}^T \\ G_{k-1}^T \end{bmatrix} \\ &\quad = C_{k-1}U_{k-1}(+)D_{k-1}(+)U_{k-1}^T(+)C_{k-1}^T + G_{k-1}D_{Q_{k-1}}G_{k-1}^T \\ &\quad = C_{k-1}P_{k-1}(+)C_{k-1}^T + Q_{k-1} \\ &\quad = P_k(-) \end{aligned} \tag{3.4}$$

Consequently the factors $U_k(-) = L^T, D_k(-) = D_\beta$ are the solutions of the prediction step for the UD Filtering. The prediction step continues with the factorization used in the

estimation step such that we can use the obtained UD matrices to obtain an estimate of $P(-)$.

(2) Filtering Estimation Step: This step updates the state and variance using a combination of the predicted state and the observation y. The Kalman Filtering form of the estimation step of the covariance matrix is

$$P(+) = P(-) - \frac{P(-)A^T AP(-)}{R + AP(-)A^T} \tag{3.5}$$

where P(+) indicates the prior covariance of the errors, and P(-) indicates the posterior covariance of the errors. They can be decomposed as $P(-) \overset{def}{=} U(-)D(-)U^T(-)$, $P(+) \overset{def}{=} U(+)D(+)U^T(+)$. So we have

$$\begin{aligned} U(+)D(+)U^T(+) &= U(-)D(-)U^T(-) - \frac{U(-)D(-)U^T(-)A^T AU(-)D(-)}{R + AU(-)D(-)U^T(-)A^T} U^T(-) \\ &= U(-)D(-)U^T(-) - \frac{U(-)D(-)vv^T D(-)U^T(-)}{R + AU(-)D(-)U^T(-)A^T} \\ &= U(-)\left[D(-) - \frac{D(-)vv^T D(-)}{R + v^T D(-)v} \right] U^T(-) \\ &= U(-)\left[BD(+)B^T \right] U^T(-) \\ &= \{U(-)B\}D(+)\{U(-)B\}^T \end{aligned} \tag{3.6}$$

where $v=U^T(-)A^T$ is an n-vector and n is the dimension of the state vector. The posterior U factors can be solved from the equation $U(+)=U(-)B$. Therefore, for the estimation step of the UD factors of the covariance matrix P, it suffices to find a numerically stable and efficient method for the appropriate B. This estimation step continues with the factorization used in the smoothing updating step such that we can use the obtained UD matrices to obtain an estimate of $P(+)$.

(3) Smoothing updating step: We use the UD covariance factorization of Rank-2 Smoother algorithm [9] to implement the smoothing updating step for realizing the EM algorithm to estimate the parameters C, Q and R of the filtering equation.

The above three steps are repeated until the system parameters of the maximization step of EM algorithm have converged. When the iterations are terminated, the output of the filtering prediction step, x_t, t=T,T-1,…,1, is our estimated traffic matrix.

4 Experiments and Results

Traffic matrices may be estimated or measured at varying levels of detail [10]: between Points-of-Presence (PoPs), routers, links or even IP prefixes. We estimated the routers level traffic matrix based on the simulation data of the NS2 to evaluate our SRFsTME algorithm in the ForCES architecture network. Figure 4.1 illustrates the 5-

nodes routers level topology of the ForCES architecture network. In this topology there are one CE node and four FE nodes. This topology can represent the traffic flow of the ForCES architecture.

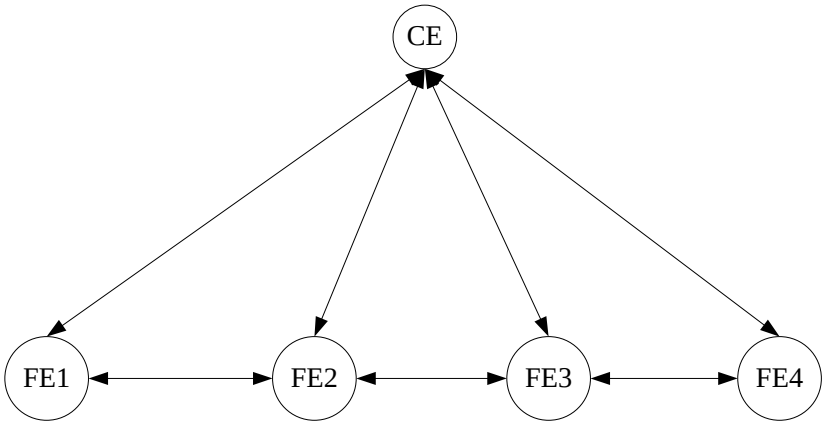

Figure 4.1: 5-Nodes Topology of ForCES Architecture

Based on this topology we simulate the SNMP data and route information to validate the accuracy and stability of the SRFsTME algorithm in the ForCES architecture. To evaluate the estimation results of the algorithms we need to measure some OD TM directly through simulation.

In this 5-nodes abstract topology every link is bidirectional so that there are 14 links and 20 ODs altogether. In this part we discuss the performance of the estimation methods based on the estimation results obtained from the computation of 20 ODs. To outline the estimation results clearly, we only illustrate them at 10 time slots at most.

Figure 4.2 shows the estimation results about 20 ODs at 5th and 9th time slots. It presents the estimated TM by SRFs traffic matrix estimation (SRFsTME) algorithm and also presents the direct measurement OD traffic. From Figure 4.2(a) we notice that the estimated TM fits well with the original OD traffic at most points while that of the KF method is not accurate at the 3rd, 5^{th} and 15th points. From Figure 4.2(b) we can see that the estimated TM is more accurate and stable compared with the original OD traffic matrix.

The estimation results about 2 ODs at 10 time slots are shown in Figure 4.3. From these results we can see that the estimation results of estimated TM is stable at most time slots. So, from the perspectives of the accuracy and stability of the estimation methods, we can say that the SRFsTME algorithm is suitable for ForCES architecture.

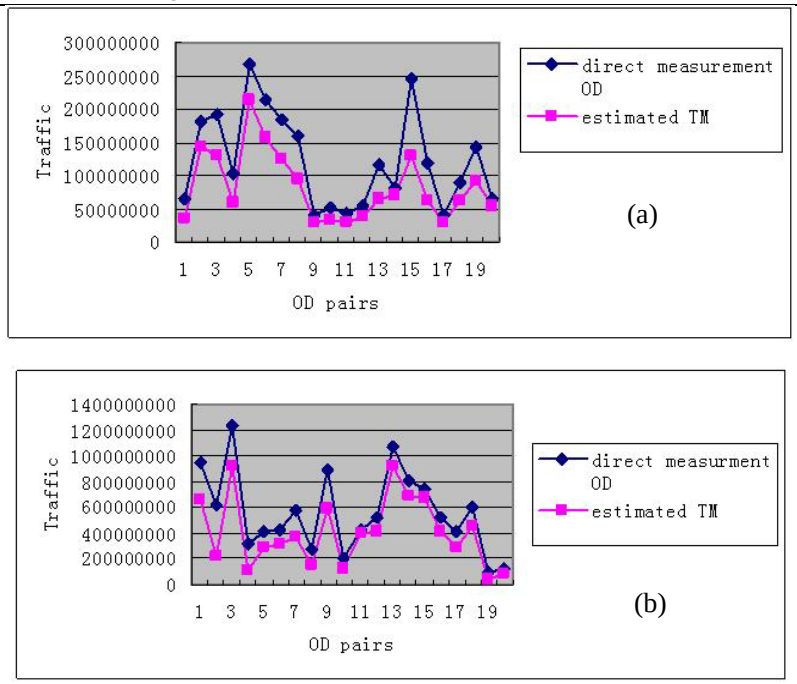

Figure 4.2: Estimation results about 20 ODs at 5th and 9th time slots

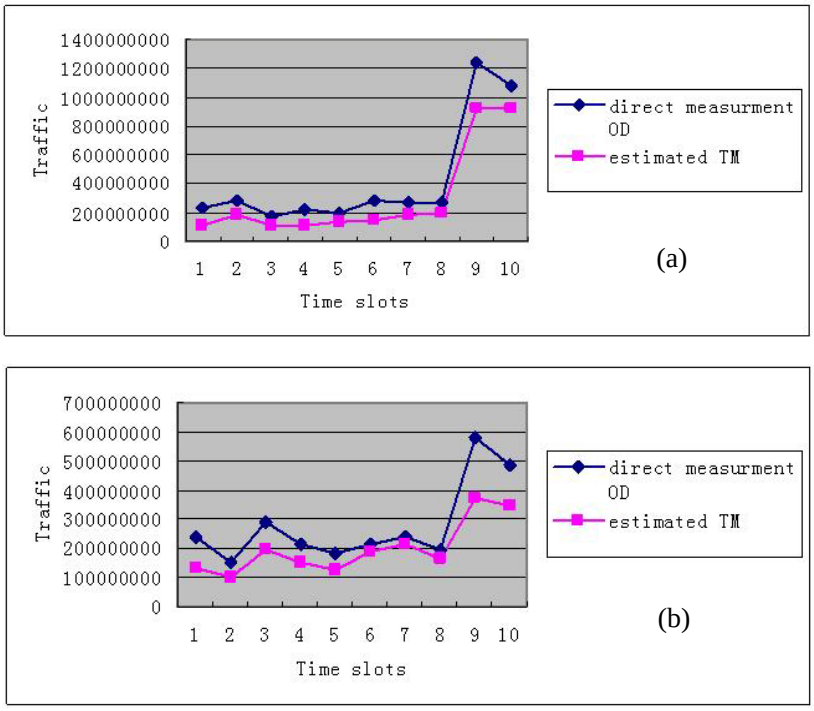

Figure 4.3: Estimation results about 3rd and 7th ODs at all 10 time slots

5 Conclusions

In this paper we firstly introduced the ForCES architecture. ForCES is an efficient, open, distributed, modular and programmable network architecture. To ensure ForCES architecture can provide users with more efficient computing environment, we proposed to use the SRFsTME algorithm to estimate traffic matrix to monitor the ForCES architecture network. Simulation results show that the SRFsTME algorithm is suitable for ForCES architecture. We will validate the utility of traffic matrix estimation algorithm in the ForCES architecture further in our future work.

Acknowledgements: This work was supported by National Natural Science Foundation of China (No. 60903214, 60970126), Zhejiang Sci & Tech Project (No. 2009C31066, 2009C11050), A Project Supported by Scientific Research Fund of Zhejiang Provincial Education Department (Y200908196, Z20097549), Zhejiang Provincial NSF China (Y1090452, Y1100871).

References

[1] W. M. Wang, L. G. Dong and B. Zhuge, *Analysis and implementation of an open programmable router based on forwarding and control element separation*, Journal of Computer Science and Technology. 23(5), 2008.

[2] A. Doria et al., *ForCES protocol specification*, http://www.ietf.org/id/draft-ietf-forces-protocol-22.txt, March 2009.

[3] K. Papagiannaki, N.T. and A. Lakhina, *A Distributed Approach to Measure Traffic Matrices*, in IMC, 2004, ACM Taormina, Italy.

[4] A. Soule, A.L., N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M.Crovella and C. Diot, *Traffic matrices: Balancing measurements, inference and modeling*, in *SIGMETRICS '05*. June 2005: Banff, Canada.

[5] A. Soule, K.S., A. Nucci and N. Taft, *Traffic Matrix tracking using Kalman Filtering*, in *Taft-LSNI2005*. 2005.

[6] L. Yang, R. Dantu, T. Anderson and R. Gopal. *Forwarding and Control Element Separation (ForCES) Framework*, http://www.ietf.org/rfc/rfc3746.txt. 2004.

[7] J. Zhou, J. Yang, Y. Yang et al., *Traffic Matrix Estimation Using Square Root Kalman Filtering/Smoothing Method*, APNOMS2008, Beijing, Springer Berlin, 5297, 519-522.

[8] S.G. Mohinder and P. A. Angus, *Kalman Filtering: Theory and Practice using MATLAB*, second edition, Wiley-Interscience (2001).

[9] G. J. Bierman, *A new computationally efficient fixed-interval, discrete-time smoothers*, Automatica, 19(1983), 503-511.

[10] M. Alberto, F. Chuck, T. Nina, B. Supratik and Christophe, D., *A Taxonomy of IP Traffic Matrices*, SPIE ITCOM: Scalability and Traffic Control in IP Networks II, Boston (August 2002).

Zhou Jingjing received the MS degree in Software Engineering from Shangdong University in 2004 and the PhD degree in Computer Science from the University of Science & Technology, Beijing, in 2009. She currently works in the Zhejiang Gongshang University.