# Flexible Updating of Attractors in Virtual Network Topology Control with Bayesian Attractor Model

Tatsuya Otoshi¶, Shin'ichi Arakawa†,Masayuki Murata†
Kai Wang†‡, Takeo Hosomi†‡, Toshiyuki Kanoh†‡
¶ Graduate School of Economics, Osaka University
†Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
‡Data Science Laboratories, NEC Corporation
1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa 211-8666, Japan

***Abstract*—Network virtualization is expected to handle various forms of network traffic induced by Internet of Things applications and other Internet-based services. Because traffic patterns change with time, virtual networks should be dynamically reconstructed to accommodate increasing traffic and to free unused resources. However, collecting all traffic information is difficult when applications are deployed on a wide-area network. It is therefore necessary to consider uncertainty of information due to data incompleteness or traffic dynamics. Our research group has proposed a virtual network reconstruction method based on a Bayesian attractor model that deals with uncertain information in decision-making. However, this method requires advance knowledge of the assumed environment as an attractor. When the environment changes, attractors must also be changed. In this study, we use control feedback to automatically update attractors when the environment changes. Simulation-based evaluations demonstrate that the proposed method deals with unknown situations while maintaining noise tolerance.**

***Index Terms*—Virtual Network Topology Reconstruction, Bayesian Attractor Model, Control Feedback**

## I. Introduction

Network virtualization technologies [1] are expected to handle various Internet-based services, such as Internet of Things (IoT) applications [2]. Each service generates a different type of communication, so network providers must prepare network resources to meet the requirements of each. Network virtualization allows construction of multiple virtual networks for different network services on a physical network while satisfying the communication quality requirements of each service. Additionally, virtual networks can be flexibly reconstructed to accommodate traffic changes.

Designing a virtual network topology according to actual traffic patterns is problematic in that it is difficult to grasp the entirety of network traffic information. In wide-area networks such as those used for IoT devices, collecting all traffic information at all times is infeasible. This makes it difficult to apply existing network provisioning techniques [3], which use overall traffic information to optimize networks. While there are proposed traffic estimation techniques that infer overall traffic from partially obtained information [4], estimation errors still occur. Uncertain information causes unnecessary controls, which can destabilize the network status. It is therefore necessary to determine an appropriate virtual network topology even under uncertain traffic information.

Taking as inspiration a decision-making model of the human brain, our research group has developed a virtual network topology control method [5]. A model called the *Bayesian attractor model* (BAM) [6] describes decision-making behaviors in the human brain. The BAM utilizes a probabilistic inner *decision state*, which is sequentially updated with uncertain observations. It accumulates confidence based on observed information and makes decisions upon reaching a certain confidence level. Using a BAM for virtual network topology control allows good handling of the tradeoff between response speed and control stabilization. However, BAM requires advance knowledge of attractors for possible situations, limiting its applicability to virtual network control in which new traffic situations continually arise.

Our research group previously proposed a method for adding new attractors in unknown situations [7] in the field of image recognition. This method can detect unknown categories and automatically collect data needed for training and adding attractors. Image recognition requires labeling only new inputs with new labels, while virtual network control requires mapping inputs to controls. It is thus necessary to update not only attractors but also correspondences between inputs and controls. Moreover, computation times increase with the number of attractors, so the number of attractors needs to be limited. Also, an increase in the number of similar attractors increases the difficulty of their classification, which may lead to wrong decisions. It is therefore necessary to update existing attractors as the situation changes. In this paper, we propose a method that uses BAM confidence and control feedback to update attractors in response to changing situations.

When learning with feedback, the degree of confidence in choices plays an important role [8]. For example, an error under high confidence indicates that the assumptions leading to the judgment are wrong, and thus should be significantly revised by feedback. A BAM allows the degree of confidence in a decision to be calculated. We extend that model to also provide control feedback. Changing the choice of BAM according to feedback has been investigated in the past [5], but this paper treats the decision-to-feedback loop in a unified way by incorporating feedback into the model itself. In this model, attractors are updated by considering whether a decision made with the attractors would be good or bad. Through simulation-based evaluation, we show that the addition of feedback allows a BAM to make decisions appropriate to new situations. Moreover, though response to a new situation tends to increase unnecessary control by responding to noise, the proposed method can respond to new situations while maintaining the noise regime.

The remainder of this paper is organized as follows. In Section II, we describe a BAM-based method for virtual network reconstruction and propose prediction-based decision-making. Section III describes the extended model, which uses control feedback and confidence to update attractors. In Section IV, we present simulation results for our proposed method. Finally, Section V summarizes this paper and suggests topics for future work.

## II. Virtual Network Reconstruction by BAM

When applying the BAM decision mechanism to virtual network reconstruction, the control server is expected to determine an appropriate topology even with uncertain traffic information. In this section, we describe details of the BAM-based virtual network reconstruction method. We also introduce a prediction mechanism to the conventional BAM to recognize changing traffic situations.

### A. Bayesian Attractor Model

In this study, we used a BAM as a classifier that formulates the cognitive decision-making process in the human brain [6]. In the BAM, this process is modeled as accumulating confidence through sequential observations. The BAM has several attractors, or choices, by which it estimates the confidence of which choice is most suited to the current situation. When the confidence of a certain choice exceeds a certain threshold, the BAM takes that choice.

We denote the decision state at time $t$ as $\boldsymbol{z}_t$ and the state corresponding to $S$ alternatives as $\phi_1, \cdots, \phi_S$. That is, the $i$th alternative is selected when the decision state $\boldsymbol{z}_t = \phi_i$. Usually, $\phi_i$ is a one-hot vector whose $i$th element is 1 and others are $-1$.

*1) State Update:* When a new observation value $\boldsymbol{x}_t$ is obtained, the BAM calculates the posterior probability distribution $P(\boldsymbol{z}_t|\boldsymbol{x}_t)$ by Bayesian inference. The generative model for Bayesian inference is

$$\begin{aligned} \boldsymbol{z}_t - \boldsymbol{z}_{t-\Delta t} &= \Delta t f(z_{t-\Delta t}) + \sqrt{\Delta t} w_t \quad (1) \\ \boldsymbol{x}_t &= M\boldsymbol{\sigma}(\boldsymbol{z}_t) + v_t, \quad (2) \end{aligned}$$

where $f$ is the Hopfield dynamics with $S$ fixed points (attractors) $\phi_1, \cdots, \phi_S$ representing the $S$ alternatives. In this generative model, if the decision state approaches an attractor $\phi_i$, then the selected attractor is suited to the observed value $x_t$. $M = [\boldsymbol{\mu}^{(1)}, \cdots, \boldsymbol{\mu}^{(S)}]$ is a matrix that aligns $\boldsymbol{\mu}^{(i)}$, which is representative of the observation value for the situation in which the $i$th alternative is suitable. $\boldsymbol{\sigma}$ is a sigmoid function whose image is $[0, 1]$. In this paper, functions in bold are element-wise applied to the operand. $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are noise terms respectively following Gaussian distributions $w_t \sim N(0, \frac{q^2}{\Delta t} I)$ and $v_t \sim N(0, r^2 I)$. $q$ is *dynamics uncertainty*, representing the tendency to switch among alternative decisions, and $r$ is *sensory uncertainty*, representing the magnitude of error in observation values.

*2) Decision-making:* The posterior distribution gives the confidence of choice suitability for the current situation. In the BAM, a decision is made when the confidence exceeds a certain threshold.

The BAM selects the $i$th alternative when the condition is satisfied:

$$P(\boldsymbol{z}_t = \phi_i|\boldsymbol{x}_t) \geq \lambda \quad (3)$$

Here, $\lambda$ is the threshold, and $P(\boldsymbol{z}_t = \phi_i|\boldsymbol{x}_t)$ is confidence in the $i$th alternative.

### B. Virtual Network Reconstruction with BAM(VNR-BAM)

This subsection describes the virtual network reconstruction method with BAM, which we call *VNR-BAM*. The process of virtual network reconstruction is roughly separated into three subprocesses: collecting traffic information from the virtual network, determining a virtual network topology appropriate for the current traffic pattern, and implementing that topology in the actual network. The entire process is periodically repeated in response to traffic changes. It takes an enormous amount of time to calculate the optimal solution, and it cannot keep up with the fluctuation of traffic. Therefore, we take an approach of preparing multiple candidate solutions in advance and selecting an appropriate solution according to the situation.

At each time slot, the controller collects traffic information from the network. We denote the traffic information at time $t$ as a vector $\boldsymbol{x}_t$, whose elements are traffic levels in an origin–destination flow. We distinguish observed values from true traffic amounts, denoted by $\boldsymbol{\mu}_t$, because observed values contain observation error.

Given the traffic information, the controller calculates a virtual network topology appropriate to the current traffic. We represent the virtual network topology at time $t$ as a vector $\boldsymbol{z}_t$, whose element $z_t^{ab}$ indicates the ON/OFF state of the link between nodes $a$ and $b$. Particularly, a link is established when $z_t^{ab} = 1$, and no link is established when $z_t^{ab} = -1$. The controller calculates a topology $\boldsymbol{z}_t$ satisfying a certain condition under current traffic $\boldsymbol{x}_t$ such that the maximum link utilization remains below an acceptable level when, for example, traffic is routed along shortest paths [9] or when minimizing capacity installation costs [10]. The link capacity can be represented directly as $C^{ab}$, or indirectly as the number of ON links between $a$ and $b$.

Finally, the controller reconstructs the virtual network according to $\boldsymbol{z}_t$, requesting the physical network provider to allocate physical resources to realize the new virtual network. This process, called *virtual network embedding*, has been well studied [11, 12].

This paper focuses on how to decide which topology is best suited to the current situation when the obtained traffic information includes uncertainty, such as that due to traffic dynamics or incomplete data collection.

### C. Generative Model for Virtual Network Reconstruction

To realize decision-making for a virtual network with uncertain observations, we apply the BAM to virtual network reconstruction. In this subsection, we formulate the virtual network reconstruction as a BAM.

Since the generative model (Eqs. (1)–(2)) determines the decision-making process in the BAM, each element in the model should be associated with the virtual network reconstruction problem. Namely, the observation value $\boldsymbol{x}_t$, decision state $\boldsymbol{z}_t$, Hopfield dynamics $f$, and observation equation should be determined in correspondence to the virtual network reconstruction. The following describes how these are mapped in the virtual network reconstruction scheme.

*1) Observation Value:* As mentioned in Section II-B, observation values in virtual network reconstruction are traffic information. The controller collects traffic amounts at each origin–destination flow $\boldsymbol{x}_t$, which may differ from the actual traffic amount $\boldsymbol{\mu}_t$ due to observation uncertainty.

*2) Decision State:* The virtual network controller uses observed traffic to select an appropriate virtual network topology. The BAM decision state thus corresponds to the connection state of each link $\boldsymbol{z}_t$, as defined in Section II-B.

Similarly, specific states $\phi_1, \cdots, \phi_S$ representing the BAM alternatives correspond to $S$ topology candidates. The controller selects one candidate for reconfiguring the virtual network topology according to its confidence $P(\boldsymbol{z}_t = \phi_i | \boldsymbol{x}_t)$.

*3) Hopfield Dynamics:* The Hopfield dynamics $f$ plays a role in remembering alternatives $\phi_1, \cdots, \phi_S$ and recalling one of them. Following Ref. [9], which also utilizes Hopfield dynamics for virtual network reconstruction, we define the function $f$ as

$$f(\boldsymbol{z}) = \Phi\Phi^{+}\varsigma(\boldsymbol{z}) - \boldsymbol{z}, \tag{4}$$

where $\Phi = ({}^t\phi_1, \cdots, {}^t\phi_S)$ is a matrix in which vectors $\phi_1, \cdots, \phi_S$ are arranged in columns, $\Phi^+$ is the pseudoinverse of $\Phi$, and $\varsigma$ is a sigmoid function whose image is $[-1, 1]$. Note that we distinguish two sigmoid functions $\sigma$ and $\varsigma$, whose images are $[0, 1]$ and $[-1, 1]$, respectively.

Points $\phi_1, \cdots, \phi_S$ are fixed points of the dynamics as determined by the above function, and a certain state $\boldsymbol{z}_t$ approaches the closest fixed point $\phi_i$ by the dynamics. In our formulation, a fixed point $\phi_i$ has multiple elements with values of 1 corresponding to ON links in the topology, while in Ref. [6] each fixed point has only one element whose value is 1. This difference prevents utilization of the observation equation (2) in our formulation, so we define a new observation equation to handle multiple 1 values in $\phi_i$, detailed in the next subsection.

*4) Observation Equation:* The BAM observation equation maintains the relation between observation values and decision states. More precisely, it decides which decision seems to be correct when obtaining observation value $\boldsymbol{x}_t$. The observation equation in virtual network reconstruction should thus be defined as a mapping between observed traffic and the virtual network topology most appropriate to the observation.

Unfortunately, the original observation equation defined as Eq. (2) cannot be directly applied to our formulation, because the original BAM assumes a special type of Hopfield dynamics in which each fixed point $\phi_i$ has only one element with value 1. In Ref. [6], the fixed point $\phi_i$ is a unit vector whose $i$th element is 1, so the product $M\boldsymbol{\sigma}(\boldsymbol{z})$ nearly equals $\boldsymbol{\mu}^{(i)}$ (the $i$th column of $M$) when $\boldsymbol{z}_t = \phi_i$. The decision state $\phi$ thus corresponds with the observation value $\boldsymbol{\mu}^{(i)}$. However, this logic is invalid if $\phi_i$ contains multiple elements with value 1.

We therefore extended the observation equation to handle multiple 1 values in the fixed points. In the original BAM, the $i$th element of $\boldsymbol{\sigma}(\boldsymbol{z}_t)$ behaves as an indicator function that indicates whether $\boldsymbol{z}_t$ equals $\phi_i$. ¿From this perspective, we introduce the indicator function

$$\delta_{\phi_i}(\boldsymbol{z}) = \prod_{j;\phi_i[j]=1} \sigma(z_j) \prod_{j;\phi_i[j]=0} (1 - \sigma(z_j)), \tag{5}$$

where $\phi_i[j]$ is the $j$th element of $\phi_i$. $\delta_{\phi_i}(\boldsymbol{z})$ is close to 1 if $\boldsymbol{z} = \phi_i$ and close to 0 otherwise. Using this indicator function, we formulate an observation function similar to Eq. (2):

$$\boldsymbol{x}_t = M\boldsymbol{\delta}(\boldsymbol{z}_t) + \boldsymbol{v}_t, \tag{6}$$

where $\boldsymbol{\delta}(\boldsymbol{z}) = (\delta_{\phi_1}(\boldsymbol{z}), \cdots, \delta_{\phi_S}(\boldsymbol{z}))$ is a vector of indicator functions for all alternatives.

*5) Decision-making:* The posterior distribution gives confidence in the alternative being suited to the current situation. In the VNR-BAM, this decision is made when the confidence exceeds a certain threshold.

The VNR-BAM selects the $i$th alternative when the condition

$$P(\boldsymbol{z}_t = \phi_i | \boldsymbol{x}_t) \geq \lambda \tag{7}$$

is satisfied, where $\lambda$ is the threshold, and $P(\boldsymbol{z}_t = \phi_i | \boldsymbol{x}_t)$ is the confidence in the $i$th alternative.

## III. Attractor Update with Control Feedback

This section describes how we incorporate feedback of the control results into the VNR-BAM, so that the VNR-BAM can be adapted to the situation. We propose a new model that can automatically update attractors. Considering a simple VNR-BAM extension, we first show a model for Bayesian estimation with classification results as feedback. In this model, feedback is discrete information about whether the classification is correct, but the control results are generally continuous. We therefore propose a model for modifying the attractor based on the control results as evaluated by continuous values. In the proposed model, a performance indicator that takes a continuous value, such as maximum link utilization, is used as feedback and to modify attractors by comparing decision confidence and feedback.

*1) Bayesian Estimation with Feedback:* A Bayesian filter uses correct data as training data to modify the classifier's parameters. The magnitude of confidence in classifier judgments plays an important role in modifying parameters [8]. For example, a mistake with high confidence implies an error in the classifier's parameters, so the parameters need to be significantly modified.

Regarding a BAM as a kind of classifier, its parameters are representative of the attractor. BAM classifies observed data into attractors with close representative values. Namely, the matrix of representative values $M$ is a parameter, the BAM takes $x_t$ as input and selects the $d$th attractor, and the label $l$ of the correct attractor is given. Then, the posterior distribution of parameters can be updated as

$$\begin{aligned} P(M|x_t, d=l) &\propto P(d=k|l=k, M_t, x_t)P(M) \\ P(M|x_t, d\neq l) &\propto (1-P(d=k|l=k, M_t, x_t))P(M), \end{aligned} \tag{8}$$

where $P(d=k|l=k, M_t, x_t)$ is the selection confidence and $P(M)$ is the prior distribution of $M$.

However, such feedback-based updates present several issues regarding application to controlling a virtual network. The first is that it is difficult to provide correct answers as feedback. Classifications by attractors are introduced for distinction of the observation space, and there are no strictly correct answers. Rather, the results of control are measured using a conventional performance index, such as maximum link utilization. A second issue is that feedback is limited to discrete values, while most performance indicators, such as maximum link utilization, are continuous values. In the next section, therefore, we propose a model for modifying the representative BAM value for control feedback.

*2) VNR-BAM with Control Feedback:* We propose a model for modifying the representative BAM value using control performance as feedback. As described above, it is important to reflect confidence levels when using feedback to modify parameters. In particular, it is necessary to vary the extent of model modification in response to gaps in confidence and feedback. The problem is thus how to compare control performance with confidence.

Here, we normalize the range of control performance and confidence values, thereby allowing numerical comparisons. Note that index values falling within a $[0\dots1]$ range, such as maximum link utilization, can be directly used. In contrast, a BAM confidence value is a probability density with value range $[0\dots\infty)$. Therefore, by using the softmax function

$$P'(\boldsymbol{z_t}=\phi_i) = \frac{P(\boldsymbol{z_t})}{\sum_k P(\boldsymbol{z_t}=\phi_k)}, \tag{9}$$

we can scale the confidence of all attractors so that their total is 1.

Using the scaled confidence and control performance described above, we add the following equation to the original generative model (Eqs. 1–2) to create a new generative model:

$$\alpha'_t = P'(\boldsymbol{z_t}=\phi_d) + \nu_t \tag{10}$$

$$\boldsymbol{\mu}_t^{(i)} = \boldsymbol{\mu}_t^{(i)} + \boldsymbol{\eta}_t^{(i)}, \tag{11}$$

where $\boldsymbol{\mu}_t^{(i)}$ is the representative value of the $i$th attractor and $\nu_t, \boldsymbol{\eta}_t^{(i)}$ is a noise term following the Gaussian distribution $\nu_t \sim N(0, u^2), \boldsymbol{\eta}_t^{(i)} \sim N(0, v^2 I)$.

Eq. 10 requires confidence and control performance to be equivalent, so the representative value of the attractor is modified to close any gaps between confidence and control performance. For example, if the control performance is high despite a low confidence level, the representative values of the corresponding attractors are brought closer to the current observed values to increase selection confidence. On the other hand, if the control performance is low despite a high confidence level, the representative value is updated to lower the current attractor's confidence level so that other choices can be taken. Eq. 11 shows how the representative value is updated with a noise term as its driving force.

*3) State and Representative Values Estimation in the Proposed Model:* The new model uses newly added feedback information as input to estimate representative values. In a conventional BAM, the state $\boldsymbol{z}_t$ is estimated using the observed value $\boldsymbol{x}_t$ as input. This can be performed by using Bayesian filters such as the unscented Kalman filter (UKF) [13] or a particle filter. The same estimation can be performed in the new model by setting the input as $(\boldsymbol{x}_t, \alpha'_t)$, and the state to be estimated as $(\boldsymbol{z}_t, \boldsymbol{\mu}_t^{(i)})$. In this paper, we use the UKF to estimate state and representative values.

## IV. Evaluation

Through simulations, we show that the proposed new model allows selection of suitable solutions for new situations while maintaining the BAM noise tolerance. Simulations compared cases with and without feedback to demonstrate that feedback helps cope with new situations. We also compare the proposed method with a heuristic method that randomly searches for solutions in new situations, and show that the proposed method can select a more suitable topology without significantly increasing the number of topology changes.

### A. Simulation Setting

The proposed model is expected to adapt to new environments while avoiding unnecessary changes in topology. To demonstrate this, we performed simulations with repeated environmental changes as follows:

1) Generate a traffic pattern for a given number of attractors.
2) Calculate a quasi-optimal topology for the generated traffic patterns.
3) Initialize the attractor set with the generated traffic pattern as the representative value and the topology as the choice.
4) Generate an average traffic value.
5) Set the actual traffic pattern as the average of the traffic values plus noise.
6) Based on observations and feedback, update the state and representative values and select a topology over $T_{int}$ time slots.
7) Generate traffic averages for the new environment in Step 4, and repeat Steps 5 and 6.

During actual operation, the attractor is first designed to match some of the expected traffic patterns. We then generate multiple traffic patterns in Step 2 and prepare an appropriate topology for the solution. Here, we randomly generate topologies and set the topology with minimum link utilization under each traffic pattern as the quasi-optimal topology. Actual traffic patterns are generated in subsequent steps, where the initial attractors are not suited to the patterns without updates.

We performed these simulations under the following configuration. We used four virtual nodes, generating traffic flows of 0.1 units between 10 random pairs of virtual nodes. Noise was generated from Gaussian distribution with mean 0 and standard deviation 0.01. Each link has a capacity of 0.4 units, and flows are linked by shortest path. We set $T_{int} = 20$, $u = 0$, and $v = 1$. We use the best $s, q$ for the BAM parameters, changing the parameters from 0 to 1 in increments of 0.2. We assume the substrate network provides resources to the virtual network strictly according to a service level agreement, so performance is determined by the virtual network topology.

*1) Heuristic for Topology Selection:* The proposed model uses feedback to change the representative value of BAM, thereby changing the selection. A simpler way to use feedback is to directly change the selection based on good or bad performance. However, since this method does not consider traffic, it does not have the same noise tolerance as BAM and thus would frequently change topologies. To clarify these differences, we compare the proposed model with a heuristic method that uses feedback to directly modify selections.

When congestion occurs under this heuristic (i.e., when $\alpha_t$ exceeds 1), the current topology is switched to another. It is possible to maintain a history of selected topologies to avoid reselecting one, but doing so is unrealistic, because the timing of environmental changes must be accurately monitored to reset the history. We therefore do not maintain a history; rather, we randomly select the next topology from among all other possibilities. To allow fair comparison, those selections have the same topology as the BAM.

*2) Performance Metrics:* There is a tradeoff between adaptation speed and control change frequency. If the sensitivity is increased to quickly adapt to environmental changes, it will also respond to noise, causing network instability. If we retain the current settings, however, there will be delays before responding to environmental changes. By accumulating the BAM confidence level, stable selection is possible even in the presence of noise. The proposed model can respond to new situations while maintaining this noise tolerance by updating the representative value. We use two indicators to confirm this: the *congestion period* and the *number of topology changes*.

The congestion period indicates the duration of congestion, defined here as the time when the maximum link utilization exceeds 1. The number of topology changes is defined as the number of times each method makes a topology change. Each environmental change has time slots for a period of $T_{int}$, and we use the average index value for each environmental change in subsequent evaluations. Therefore, the upper limit on the congestion period and the number of topology changes is $T_{int}$.

## B. Results

*1) Congestion Period:* We first discuss the congestion period as a performance indicator to confirm that the feedback facilitates appropriate topology selections. The horizontal axis in Fig. 1 shows the number of attractors, and the vertical axis shows the congestion period. The congestion period is the average value for fifty cases with varied initial attractors. In the figure legend, “BAM_nof” is the conventional VNR-BAM with no feedback, “BAM_wf” is the proposed model, “heuristic” is the heuristic method, and “opt” indicates setting the topology from attractors by minimizing the maximum link utilization. This figure shows that BAM_wf always has a shorter congestion period than does BAM_nof, because the conventional BAM can only deal with traffic patterns assumed in the initial setting, but the proposed model can deal with new traffic patterns by updating representative values of attractors through control feedback.

In addition, when the number of attractors increases, the proposed model is superior to that of the heuristic model, which simply uses feedback. In heuristic models, solutions are found by trial and error without traffic pattern recognition, and this search takes time. In contrast, the proposed model recognizes traffic patterns, potentially reducing search times by selecting a topology that can accommodate traffic patterns similar to the current pattern.

When the number of attractors increases, the number of potential traffic patterns that can be accommodated also increases, thereby lowering the congestion period for opt. BAM_wf similarly modifies the feedback to improve selections, so its performance is similar to that of opt. The heuristic method also lowered the congestion period as the number of attractors increased, but the increased search time resulted in a large gap from opt.

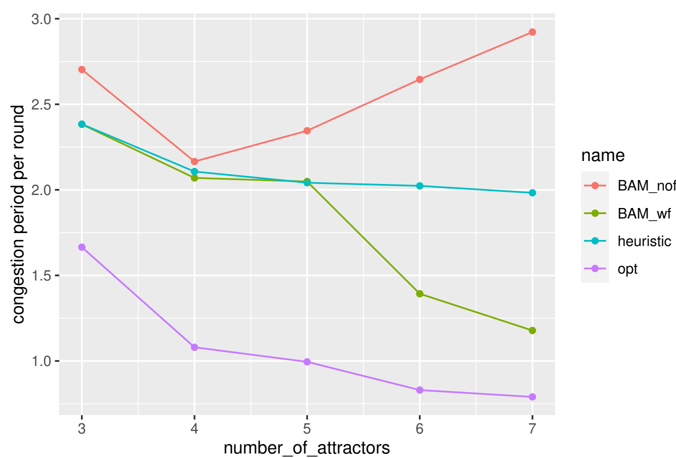


Fig. 1. Congestion periods with different numbers of attractors.

*2) Number of Topology Change:* The above shows that use of feedback can reduce congestion, even when a new environment arises. In general, however, increased sensitivity to environmental change can make control unstable due to noise. Therefore, Fig. 2 shows the average numbers of topology changes. The figure shows that opt, which sets the optimal topology each time, requires large numbers of topology changes. This is because it responds not only to environmental changes, but also to minor variations such as noise. In contrast, despite BAM_wf being able to reduce congestion to the level closest to opt, the number of topology changes is similar to or slightly less than that in the heuristic model. This confirms that the proposed model can respond to new environmental changes while maintaining the BAM noise tolerance. There are few topology changes in the conventional

BAM, because there are no corresponding attractors to avoid congestion under environment changes.

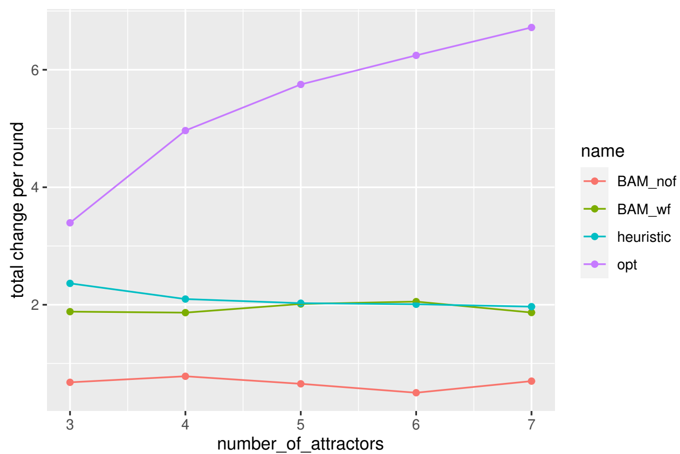


Fig. 2. Numbers of topology changes with different numbers of attractors.

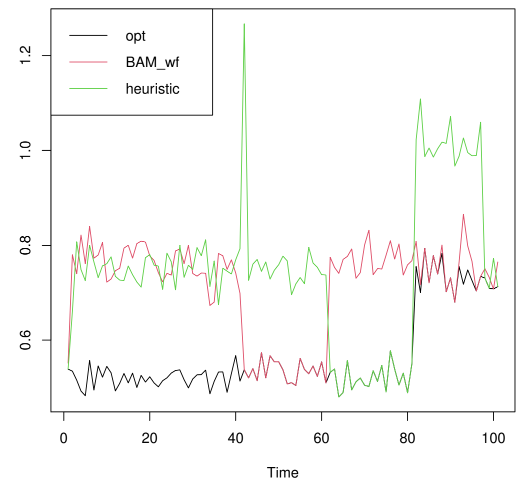


Fig. 3. Timeseries of maximum link utilization with 20 attractors.

*3) Result with More Attractors:* If an appropriate selection can be made, a topology can be set that avoids congestion by increasing the number of attractors. However, increasing attractors increases the difficulty of making correct selections. The following shows the results for the case of many (20) attractors.

Fig. 3 shows the maximum link utilization rate at each time for each method. This figure confirms that the link utilization rate exceeds 1 under the heuristic method several times, indicating that congestion has occurred. Especially after time 80, the congestion remains unresolved for a long time. This is because the heuristic method changes selections by trial and error, so it takes time to make an appropriate selection from among the many attractors. In contrast, the BAM_wf method avoids congestion by recognizing traffic and selecting the most suitable topology.

## V. Conclusion

We proposed an extended BAM model for virtual network reconstruction under dynamic environments. The proposed model uses control feedback to automatically update the attractors, updating their representative values by Bayesian estimation based on the confidence gap between the BAM selection and control performance feedback. Simulation-based evaluations showed that even when the environment changes, the proposed method can select a topology that quickly resolves any resulting congestion. We showed that in comparison with heuristics that use feedback for direct selections, congestion periods are shortened without increasing the number of topology changes.

Future research will investigate nonparametric methods such as the Dirichlet process mixture model that can automatically set a reasonable number of attractors.

## References


[1] H. Cao, H. Hu, Z. Qu, and L. Yang, "Heuristic solutions of virtual network embedding: A survey," *China Communications*, vol. 15, no. 3, pp. 186–219, 2018.

[2] S. Li, L. Da Xu, and S. Zhao, "5g internet of things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.

[3] P. Zhang, S. Wu, M. Wang, H. Yao, and Y. Liu, "Topology based reliable virtual network embedding from a qoe perspective," *China Communications*, vol. 15, no. 10, pp. 38–50, 2018.

[4] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and Internet traffic matrices," in *Proceedings of ACM SIGCOMM 2009*, Aug. 2009, pp. 267–278.

[5] T. Ohba, S. Arakawa, and M. Murata, "A Bayesian-based approach for virtual network reconfiguration in elastic optical path networks," in *Proceedings of Optical Fiber Communication Conference*, Mar. 2017.

[6] S. Bitzer, J. Bruineberg, and S. J. Kiebel, "A Bayesian attractor model for perceptual decision making," *PLOS Computational Biology*, vol. 11, no. 8, p. e1004442, Aug. 2015.

[7] T. Otoshi, S. Arakawa, M. Murata, K. Wang, T. Hosomi, and T. Kanoh, "Acquiring new categories by self data gathering with bayesian attractor model," in *Proceedings of 1st IEEE International Conference on Human-Machine Systems*, September(to be presented) 2020.

[8] A. Pouget, J. Drugowitsch, and A. Kepecs, "Confidence and certainty: distinct probabilistic quantities for different goals," *Nature neuroscience*, vol. 19, no. 3, p. 366, 2016.

[9] T. Ohba, S. Arakawa, Y. Koizumi, and M. Murata, "Scalable design method of attractors in noise-induced virtual network topology control," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 9, pp. 851–863, Sep. 2015.

[10] T. Bauschert, C. Busing, F. D'Andreagiovanni, A. M. C. A. Koster, M. Kutschka, and U. Steglich, "Network planning under demand uncertainty with robust optimization," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 178–185, 2014.

[11] C. Fuerst, S. Schmid, and A. Feldmann, "Virtual network embedding with collocation: Benefits and limitations of pre-clustering," in *IEEE 2nd International Conference on Cloud Networking*, Nov. 2013, pp. 9–98.

[12] S. Qing, Q. Qi, J. Wang, T. Xu, and J. Liao, "Topology-aware virtual network embedding through Bayesian network analysis," in *IEEE GLOBECOM*, Dec. 2012, pp. 2621–2627.

[13] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.