**APPENDIX A**

**PRELIMINARIES OF HHMM**

The proposed NHMM is based on the frame of the hierarchical hidden Markov model (HHMM) [8]. The HHMM is an extension of the HMM [30]. It has two advantages: (i) it can model the different stochastic levels and length scales that are present in the time series; and (ii) it can infer correlated observations over long periods in the observation sequence via different levels of the hierarchy.

The basic structure of the HHMM is shown in Fig. 14. The structure consists of three types of Markov states: internal states, production states and end states. Among these are three types of state transitions: horizontal, vertical, and forced transitions. An internal state may have sub-models in its higher layer. The sub-model is a HHMM by itself and consists of one or more internal states, or one or more production states and one end state. The production states output symbols in the same way as the states of an ordinary HMM. The end state is the only state which ends the current sub-model. The hierarchical structure can be maintained in such a way that 'the lower layer' cannot change state unless the 'higher layer' state reaches the end state. Thus, the HHMM is a structural multi-layer stochastic process, which is especially important in cases where data is hierarchical. The most prominent feature of HHMMs is that long-term dependency can be captured via the additional layers designed to model lower-level activities evolving at slower timescales. This feature makes it suitable to describe the hierarchical dynamic process of network traffic.

**APPENDIX B**

**ALGORITHM FOR THE NHMM**

**B.1 Inference of model’s parameters**

Similar to the classical HMM [30], the proposed NHMM model should be able to solve three problems: (i) given a model, find the probability of the observations; (ii) given a model and an observation sequence, find the most likely underlying state transition trajectory; and (iii) maximize either (i) or (ii) by adjusting the model’s parameters. Now, we introduce a new efficient Forward-backward algorithm to solve these problems for the proposed model.

We define the forward process by

\[
\alpha_t(i, d, m_i) = \sum_{n_i} \alpha_{t-1}^{n_i} (i, d, m_i) + \sum_{j \neq i, n_j} \tilde{\alpha}_{t}^{n_j} (i, d, m_i) \tag{2}
\]

where \(\tilde{\alpha}_{t}^{n_j}(i, d, m_i)\) and \(\tilde{\alpha}_{t}^{n_j}(i, d, m_i)\) are defined by the following equations, respectively.

\[
\tilde{\alpha}_{t}^{i, n_i}(i, d, m_i) = P(\alpha_{t-1}^{i, d, m_i}, q_{t-1}^{i, d, m_i} = (i, n_i), (q_t^{i, d, m_i} = (i, d), q_t^{n_i} = m_i, i, d + 1, n_i) | a_{m_i}^{n_i} b_{m_1}^{n_1} o_{m_2}^{n_2}) \tag{3}
\]

and

\[
\tilde{\alpha}_{t}^{j, n_j}(i, d, m_i) = P(\alpha_{t-1}^{i, d, m_i}, q_{t-1}^{i, d, m_i} = (j, n_j), (q_t^{i, d, m_i} = (i, d), q_t^{n_j} = m_i, j \neq i) | a_{m_i}^{n_i} b_{m_1}^{n_1} o_{m_2}^{n_2}) \tag{4}
\]

The initial condition of the forward variable can be calculated by

\[
\alpha_1(i, d, m_i) = \pi_{m_i}^{i, d, m_i} \prod_{t=1}^{T} p_{m_i}^{i, d, m_i} S_{m_i}^{n_i} b_{m_1}^{n_1} \tag{5}
\]

We define the backward variable by

\[
\beta_t(i, d, m_i) = P(\alpha_{T+1}^{i, d, m_i}, \tau_t^{i, d, m_i} = (i, d), q_t^{n_i} = m_i) \tag{6}
\]

where \(i \in Q^M, d = \{1, 2, ..., D\}, m_i \in Q^{S_l_i} \). The possible states that follow \((q_t^{i, d, m_i} = (i, m_i))\) must be \((q_{t+1}^{i, d, m_i} = (i, n_i), n_i \in Q^{S_l_i}, q_{t+1}^{i, d, m_i} = (j, n_j), j \neq i)\) for \(j \neq i\) and \(n_j \in Q^{S_l_j}\). Therefore, we obtain the following backward recursion formula:

\[
\beta_t(i, d, m_i) = \sum_{n_i} \beta_{t+1}^{n_i} (i, d, m_i) + \sum_{j \neq i, n_j} \tilde{\beta}_{t}^{n_j} (i, 1, m_j) \tag{7}
\]

where \(\tilde{\beta}_{t}^{n_i}(i, d, m_i)\) and \(\tilde{\beta}_{t}^{n_j}(i, 1, m_j)\) are defined by the following equations, respectively.

\[
\tilde{\beta}_{t}^{i, n_i}(i, d, m_i) = P(\alpha_{T+1}^{i, d, m_i}, q_{T+1}^{i, d, m_i} = (i, n_i), (q_t^{i, d, m_i} = (i, d), q_t^{n_i} = m_i) \tag{8}
\]

\[
= a_{m_i}^{n_i} b_{m_1}^{n_1} o_{m_2}^{n_2} \beta_{t+1}(i, d-1, n_i)
\]
\[ \bar{\beta}^n_i(i, 1, m_i) = P[ a^T_i, (q^M_{i-1}, q^S_{i-1}) = (j, n_j) | (q^M_t, \tau^M_t) = (i, 1), q^S_t = m_i], j \neq i \]
\[
= \sum_d a^M_{d,j} P^M_{i,d} \pi^S_{d,j} b^S_{d,m_i+1,j} \bar{\beta}_{i+1}(j, d, n_j)
\]  
(9)

The initial condition of the backward variable can be calculated by

\[ \beta_T(i, c, m, d) = 1 \]
(10)

In order to simplify the following derivation, we define an auxiliary function:

\[ v_t(i, j, d, m_i, n_j) \equiv P[ a^T_t, (q^M_{t-1}, q^S_{t-1}) = (i, m_i), (q^M_t, \tau^M_t) = (j, d), q^S_t = n_j] \]
(11)

The possible MS of \( v_t(i, j, d, m_i, n_j) \) at time \( t - 1 \) must be \((q^M_{t-1}, \tau^M_{t-1}) = (i, d+1) \) or \((q^M_{t-1}, \tau^M_{t-1}) = (j, 1) \) for \( j \neq i \). Hence, we can calculate the auxiliary function by the following equations:

\[ v_t(i, j, d, m_i, n_j) = \]
\[
\begin{cases} 
P[o^T_t, (q^M_{t-1}, \tau^M_{t-1}) = (i, d+1), q^S_{t-1} = m_i, (q^M_t, \tau^M_t) = (j, d), q^S_t = n_j], \text{for } j = i \\
P[o^T_t, (q^M_{t-1}, \tau^M_{t-1}) = (i, 1), q^S_{t-1} = m_i, (q^M_t, \tau^M_t) = (j, d), q^S_t = n_j], \text{for } j \neq i
\end{cases}
\]
\[
= \alpha_{t-1}(i, d+1, m_i) a^S_{t-1,m_i} \beta_{t-1}(i, d, n_j) \\
\alpha_{t-1}(i, 1, m_i) a^M_{t-1,m_i} P^M_{t-1,d} \pi^S_{t-1,n_j} b^S_{t-1,m_i+1,n_j} \beta_{t-1}(j, d, n_j), j \neq i
\]
(12)

Now we define the joint probability functions to infer the parameters. The joint probability of observing \( o^T_t \) and a transition from \((i, m_i)\) to \((j, n_j)\) at time \( t \) can be expressed by:

\[ \xi_t(i, j, m_i, n_j) \equiv P[ o^T_t, (q^M_{t-1}, q^S_{t-1}) = (i, m_i), (q^M_t, q^S_t) = (j, n_j)] \]
\[
= \sum_d v_t(i, j, d, m_i, n_j)
\]  
(13)

To simplify the notations, we define the joint probability of observing \( o^T_t \) and a transition from MS \( i \) to \( j \) at time \( t \) by:

\[ \tilde{\xi}_t(i, j, m_i, n_j) \equiv P[ o^T_t, q^M_{t-1} = i, q^M_t = j], i \neq j \]
\[
= \sum m_i, n_j \xi_t(i, j, m_i, n_j)
\]  
(14)

The joint probability of observing \( o^T_t \) and a transition from SS \( m_i \) to \( n_i \) at time \( t \) within a MS \( i \) can be expressed by:

\[ \tilde{\xi}_t(i, m_i, n_i) \equiv P[ o^T_t, q^M_{t-1} = i, (q^S_{t-1}, q^S_t) = (m_i, n_i)] \]
\[
= \tilde{\xi}_t(i, i, m_i, n_i)
\]  
(15)

The joint probability of observing \( o^T_t \) and MS \( i \) beginning with SS \( m_i \) at time \( t \) can be expressed by:

\[ \tilde{\xi}_t(i, m_i) \equiv P[ o^T_t, q^M_{t-1} \neq i, q^M_t = i, q^S_t = m_i] \]
\[
= \sum_{j, n_j} \tilde{\xi}_t(i, j, n_j, m_i), j \neq i
\]  
(16)

The joint probability of observing \( o^T_t \) and a transition to MS state \( i \) at time \( t \) and remaining in state \( i \) for time units \( d \) can be expressed by:

\[ \eta_t(i, d) \equiv P[o^T_t, q^M_{t-1} \neq i, (q^M_t, \tau^M_t) = (i, d)] \]
\[
= \sum_{j \neq i, m_i} \sum_n v_t(i, j, d, m_j, n_i)
\]  
(17)

In order to estimate the states \((q^M_t, q^S_t)\) from the observation sequence \( o^T_t \), let us consider the joint probability of \( o^T_t \) and \((q^M_t, q^S_t) = (i, m_i)\)

\[ \gamma_t(i, m_i) \equiv P[o^T_t, (q^M_t, q^S_t) = (i, m_i)] \]
(18)

Considering the following identity:

\[ P[o^T_t, (q^M_t, q^S_t) = (i, m_i)] = P[o^T_t, (q^M_t, q^S_t) = (i, m_i)] - P[o^T_t, (q^M_t, q^S_t) \neq (i, m_i)] \]
\[
= P[o^T_t, (q^M_{t+1}, q^S_{t+1}) = (i, m_i)] - P[o^T_t, (q^M_{t+1}, q^S_{t+1}) \neq (i, m_i)]
\]  
(19)

we have the following equation:

\[ P[o^T_t, (q^M_t, q^S_t) = (i, m_i)] = P[o^T_t, (q^M_{t+1}, q^S_{t+1}) = (i, m_i)] - P[o^T_t, (q^M_t, q^S_t) \neq (i, m_i)] \]
\[
= \gamma_t(i, m_i) + \gamma_{t+1}(i, m_i)
\]  
(20)

Thus, we obtain the following backward recursion formula for \( \gamma_t(i, m_i) \):

\[ \gamma_t(i, m_i) = \sum_{j \neq i, n_j} \xi_{t+1}(i, j, m_i, n_j) - \xi_{t+1}(i, n_i, m_i) \]
\[
= \sum_{j \neq i, n_j} \xi_t(i, j, m_i, n_j) - \xi_t(i, i, m_i, n_i)
\]  
(21)

with the initial condition \( \gamma_T(i, m_i) = \sum_j \alpha_T(i, d, m_i) \).

Now we can solve the above three problems based on these equations. First, we calculate the probability of a trace given a model by

\[ L = \sum_{t=1}^T P[o^T_t | \lambda] = \sum_{t=1}^T \sum_{m_i} \alpha_T(i, d, m_i) \]  
(22)

Then, the maximum a posteriori (MAP) estimate of hidden states \((q^M_t, q^S_t)\) is

\[ (\hat{q}^M_t, \hat{q}^S_t) = \arg \max_{q^M_t, q^S_t} P[o^T_t, (q^M_t, q^S_t) = (i, m_i)] \]
\[
= \arg \max_{q^M_t, q^S_t} \gamma_t(i, m_i) \]  
(23)

The re-estimation formulae can be derived from the above equations.

\[ \hat{P}_t^{M} = \sum_{m_i} \gamma_t(i, m_i) \sum_{i, m_i} \gamma_t(i, m_i) \]  
(24)

\[ \tilde{\hat{P}}_t^{ij} = \sum_{t=1}^T \tilde{\xi}_t(i, j) / \sum_{t=1}^T \sum_{j} \tilde{\xi}_t(i, j) \]  
(25)
\[
\hat{p}_{id}^M = \sum_{t=1}^{T} \eta_t(i, d) / \sum_{t=1}^{T} \sum_d \eta_t(i, d) \tag{26}
\]

\[
\hat{\pi}_{m_i}^S = \sum_{t=1}^{T} \bar{\xi}_t(i, m_i) / \sum_{t=1}^{T} \sum_{m_i} \bar{\xi}_t(i, m_i) \tag{27}
\]

\[
\hat{a}_{m_i, n_i}^S = \sum_{t=1}^{T} \bar{\xi}_t(i, m_i, n_i) / \sum_{t=1}^{T} \sum_{n_i} \bar{\xi}_t(i, m_i, n_i) \tag{28}
\]

\[
\hat{b}_{m_i, k}^S = \sum_{t=1}^{T} \gamma_t(i, m_i) \delta(o_t - k) / \sum_{t=1}^{T} \gamma_t(i, m_i) \tag{29}
\]

**B.2 Complexity Analysis of the proposed algorithm**

From Eq.(2) we can see that updating the forward variables \(\alpha_t(i, d, m_i)\) at every \(t\) requires \(O(D(|Q^M||Q^S|)^2)\) computation steps. Similarly, the backward process (6) at each \(t\) also needs \(O(D(|Q^M||Q^S|)^2)\) steps. Hence, the total number of computation steps for evaluating the forward and backward variables is \(O(TD(|Q^M||Q^S|)^2)\) where \(T\) is the total number of observations. The recursive processes for both the forward and backward variables have to be calculated separately. However, the estimation algorithm can be combined with the backward process, i.e., the auxiliary functions \((\xi, \eta, \gamma)\), and the update for each parameter of the NHMM can be calculated in the backward process. Thus, the backward variables and the auxiliary functions \((\xi, \eta, \gamma)\) do not need to be stored. What we need to store is \(\alpha_{1:T}(i, d, m_i)\) because they are used to calculate the auxiliary functions \((\xi, \eta, \gamma)\). Thus, the storage requirement is \(O(TD|Q^M||Q^S|)\). Since the summation operations of the auxiliary functions \((\xi, \eta, \gamma)\) at every \(t\) can be obtained during the computation of the forward-backward variables, the number of computation steps required for evaluating the auxiliary functions \((\xi, \eta, \gamma)\) is linearly proportional to the number of parameters. Hence, the computational complexity of the re-estimation algorithm is \(O(|\lambda|T)\), where \(|\lambda| = |Q^M| + |Q^M|^2 + D|Q^M| + |Q^S||Q^M| + |Q^S|^2|Q^M| + K|Q^S||Q^M|\), where \(K\) stands for the number of distinct observations.