Scaled VIP Algorithms for Joint Dynamic Forwarding and Caching in Named Data Networks

Fan Lai, Feng Qiu, Wenjie Bian, Ying Cui*
Shanghai Jiao Tong University
Shanghai, China

Edmund Yeh†
Northeastern University
Boston, MA, USA

ABSTRACT

Emerging Information-Centric Networking (ICN) architectures seek to optimally utilize both bandwidth and storage for efficient content distribution over the network. The Virtual Interest Packet (VIP) framework has been proposed to enable joint design of forwarding and caching within the Named Data Networking (NDN) architecture. The virtual plane of the VIP framework captures the measured demand for content objects, but does not reflect interest collapse and suppression in the NDN network. We aim to further improve the performance of the existing VIP algorithms by using a modified virtual plane where VIP counts are appropriately scaled to reflect interest suppression effects. We characterize the stability region of the modified virtual plane with VIP scaling, develop a new distributed forwarding and caching algorithm operating on the scaled VIPs, and demonstrate the throughput optimality of the scaled VIP algorithm in the virtual plane. Numerical experiments demonstrate significantly enhanced performance relative to the existing VIP algorithm, as well as a number of other baseline algorithms.

CCS Concepts

•**Networks** → **Network protocol design;** *Network architectures;*

Keywords

Named data networking, content centric networking, information centric networking, forwarding, caching, interest collapse

1. INTRODUCTION

It is increasingly clear that traditional connection-based networking architectures are ill suited for the prevailing user demands for network content [1]. Emerging Information-Centric Networking (ICN) architectures aim to remedy this fundamental mismatch so as to dramatically improve the efficiency of content dissemination over the Internet. In particular, Named Data Networking (NDN) [1], or Content-Centric Networking (CCN) [2], is a proposed network architecture for the Internet that replaces the traditional client-server communication model with one based on the identity of data or content.

Content delivery in NDN is accomplished using *Interest Packets* and *Data Packets*, along with specific data structures in nodes such as the *Forwarding Information Base (FIB)*, the *Pending Interest Table (PIT)*, and the *Content Store* (cache). Communication is initiated by a data consumer or requester sending a request for the data using an *Interest Packet*. Interest Packets are forwarded along routes determined by the FIB at each node. Repeated requests for the same object can be suppressed at each node according to its PIT. The *Data Packet* is subsequently transmitted back along the path taken by the corresponding Interest Packet, as recorded by the PIT at each node. A node may optionally cache the data objects contained in the received Data Packets in its local *Content Store*. Consequently, a request for a data object can be fulfilled not only by the content source but also by any node with a copy of that object in its cache. Please see [1, 3] for details.

NDN seeks to optimally utilize both bandwidth and storage for efficient content distribution, which highlights the need for joint design of traffic engineering and caching strategies, in order to optimize network performance. To address this fundamental problem, [3] proposes the *VIP framework* for the design of high performing NDN networks. Within this VIP framework, joint dynamic forwarding and caching algorithms operating on virtual interest packets (VIPs) are developed to maximize network stability in the virtual plane, using Lyapunov drift techniques [3]. The joint dynamic algorithms in [3] extend the existing dynamic backpressure algorithm in [4], in the sense that they incorporate caching into the dynamic design. Then, using the resulting flow rates and queue lengths of the VIPs in the virtual plane, [3] develops corresponding joint dynamic algorithms in the actual plane, which have been shown to achieve superior performance in terms of user delay and cache hit rates, relative to several baseline policies.

The virtual plane of the VIP framework in [3] focuses on capturing the measured demand for content objects in

*Y. Cui gratefully acknowledges support from National Science Foundation of China grant 61401272.
†E. Yeh gratefully acknowledges support from National Science Foundation grant CNS-1423250 and a Cisco Systems research grant.

ICN'16, September 26-28, 2016, Kyoto, Japan

DOI: http://dx.doi.org/10.1145/2984356.2984377

the network, but does not reflect interest collapse and suppression in the NDN network. In this paper, we aim to further improve the performance of the existing VIP algorithms in [3] by reflecting more accurately the actual interest packet traffic in the NDN network under interest suppression. There are several potential challenges in pursuing this. First, it is not clear how one should improve the delay performance of the existing VIP algorithms by simultaneously capturing the network demand and the interest suppression effect, in a tractable manner. Second, it is not clear how to maintain the desired throughput optimality of the existing VIP algorithms when the Lyapunov-drift-based control structure is modified to reflect interest suppression in the actual network.

In the following, we shall address the above questions and challenges. We first propose a modified virtual plane where VIP counts are appropriately *scaled*. In a simple manner, the scaled VIP counts capture to some extent the measured demand, but also reflect to some extent interest suppression effects. We characterize the stability region of the modified virtual plane with VIP scaling, which is a superset of the stability region in [3] without VIP scaling. We then develop a new distributed forwarding and caching algorithm operating on the scaled VIPs. Generalizing Lyapunov drift techniques, we demonstrate the throughput optimality of the scaled VIP algorithm in the virtual plane. The scaled VIP algorithm generalizes the VIP algorithm in [3]. Numerical experiments demonstrate the superior performance of the resulting stable VIP algorithm for handling Interest Packets and Data Packets within the actual plane, in terms of low network delay, relative to a number of baseline alternatives.

Although there is now a rapidly growing literature in ICN, the problem of optimal joint forwarding and caching for content-oriented networks remains challenging. In [5], the authors demonstrate the gains of joint forwarding and caching in ICNs. In [6], the authors propose throughput optimal one-hop routing and caching in a single-hop Content Distribution Network (CDN). Throughput-optimal caching and routing in multi-hop networks remains an open problem. In [7], assuming the path between any two nodes is predetermined, the authors consider single-path routing and caching to minimize link utilization for a general multi-hop content-oriented network. The benefits of selective caching based on the concept of betweenness centrality, relative to ubiquitous caching, are shown in [8]. In [9], cooperative caching schemes have been heuristically designed without being jointly optimized with forwarding strategies. Finally, adaptive multipath forwarding in NDN has been examined in [10], but has not been jointly optimized with caching.

2. NETWORK MODEL

We consider the same network model as in [3], which we describe for completeness. Consider a connected multi-hop (wireline) network modeled by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ denote the sets of N nodes and L directed links, respectively. Assume that $(b, a) \in \mathcal{L}$ whenever $(a, b) \in \mathcal{L}$. Let $C_{ab} > 0$ be the transmission capacity (in bits/second) of link $(a, b) \in \mathcal{L}$. Let $L_n \geq 0$ be the cache size (in bits) at node $n \in \mathcal{N}$.

Assume that content in the network are identified as *data objects*, each consisting of multiple data chunks. Content delivery in NDN operates at the level of data chunks. That is, each Interest Packet requests a particular data chunk, and a matching Data Packet consists of the requested data chunk, the data chunk name, and a signature. A request for a data object consists of a sequence of Interest Packets which request all the data chunks of the object. We consider a set $\mathcal{K}$ of K data objects, which may be determined by the amount of control state that the network is able to maintain, and may include only the most popular data objects in the network, typically responsible for most of the network congestion.[1] For simplicity, we assume that all data objects have the same size D (in bits). The results in the paper can be extended to the more general case where object sizes differ.We consider the scenario where $L_n < KD$ for all $n \in \mathcal{N}$. Thus, no node can cache all data objects. For each data object $k \in \mathcal{K}$, assume that there is a unique node $src(k) \in \mathcal{N}$ which serves as the content source for the object. Interest Packets for chunks of a given data object can enter the network at any node, and exit the network upon being satisfied by matching Data Packets at the content source for the object, or at the nodes which decide to cache the object. For convenience, we assume that the content sources are fixed, while the caching points may vary in time.

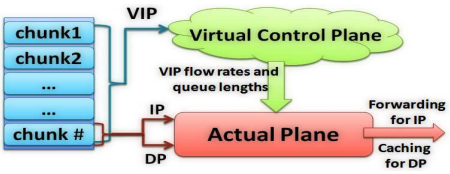

Figure 1: VIP framework [3]. IP (DP) stands for Interest Packet (Data Packet).

3. VIP FRAMEWORK

We first review the VIP framework proposed in [3] to facilitate the discussion of the algorithms developed in later sections. Please refer to [3] for the details on the motivation and utility of this framework. As illustrated in Fig. 1, the VIP framework relies on virtual interest packets (VIPs), which capture the *measured demand* for the respective data objects (i.e., represent content popularity which is empirically measured, rather than being given a priori). Note that the demand is unavailable in the interior of the actual network due to interest collapsing and suppression. The VIP framework employs a *virtual* control plane operating on VIPs *at the data object level*, and an *actual* plane handling Interest Packets and Data Packets *at the data chunk level*. The virtual plane facilitates the design of distributed control algorithms operating on VIPs, aimed at yielding desirable performance in terms of network metrics of concern, by taking advantage of local information on network demand (as represented by the VIP counts). The flow rates and queue lengths of the VIPs resulting from the control algorithm in the virtual plane are then used to specify the control algorithms in the actual plane [3].

While the VIP counts in [3] capture the measured demand for the respective data objects in the network, they do not reflect interest collapsing and suppression in the actual network. To provide potentially better guidance for designing

[1]The less popular data objects not in $\mathcal{K}$ may be distributed using simple techniques such as shortest-path forwarding with little or no caching.

efficient control algorithms in the actual plane, we propose a modified virtual plane operating on VIPs with dynamics different from those in [3]. The VIP counts in the modified virtual plane capture to some extent the measured demand, but also reflect to some extent the actual interest packet traffic under interest collapse and suppression.

We now specify the modified dynamics of the VIPs within the virtual plane. Consider time slots of length 1 (without loss of generality) indexed by $t = 1, 2, \ldots$. Specifically, time slot t refers to the time interval $[t, t+1)$. Within the virtual plane, each node $n \in \mathcal{N}$ maintains a separate VIP queue for each data object $k \in \mathcal{K}$. Note that no data is contained in these VIPs. Thus, the VIP queue size for each node n and data object k at the beginning of slot t is represented by a *counter* $V_n^k(t)$.[2] An exogenous request for data object k is considered to have arrived at node n if the Interest Packet requesting the starting chunk of data object k has arrived at node n. Let $A_n^k(t)$ be the number of exogenous data object request arrivals at node n for object k during slot t.[3] For every arriving request for data object k at node n, a corresponding VIP for object k is generated at n. The long-term exogenous VIP arrival rate at node n for object k is $\lambda_n^k \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} A_n^k(\tau)$. Let $\mu_{ab}^k(t) \geq 0$ be the allocated transmission rate of VIPs for data object k over link (a, b) during time slot t. Note that VIP transmissions between nodes do not involve actual packets being sent, and a single message between node a and node b can summarize all the VIP transmissions during each slot. Data Packets for the requested data object must travel on the reverse path taken by the Interest Packets. Thus, in determining the transmission of the VIPs, we consider the link capacities on the reverse path below:

$$\sum_{k\in\mathcal{K}} \mu_{ab}^k(t) \leq C_{ba}/D, \text{ for all } (a,b) \in \mathcal{L} \tag{1}$$

$$\mu_{ab}^k(t) = 0, \text{ for all } (a,b) \notin \mathcal{L}^k \tag{2}$$

where C_{ba} is the capacity of "reverse" link (b, a) and $\mathcal{L}^k$ is the set of L^k links which are allowed to transmit the VIPs of object k. Let $C_{\max} \triangleq \max_{(a,b)\in\mathcal{L}} C_{ab}/D$.

In the virtual plane, we may assume that at each slot t, each node $n \in \mathcal{N}$ can gain access to any data object $k \in \mathcal{K}$ for which there is interest at n, and potentially cache the object locally. Let $s_n^k(t) \in \{0, 1\}$ represent the caching state for object k at node n during slot t, where $s_n^k(t) = 1$ if object k is cached at node n during slot t, and $s_n^k(t) = 0$ otherwise. Note that even if $s_n^k(t) = 1$, the content store at node n can satisfy only a limited number of VIPs during one time slot. This is because there is a maximum rate r_n (in objects per slot) at which node n can produce copies of cached object k [3].

In contrast to [3], in order to reflect network demand as well as interest suppression in the actual network in a simple manner, we consider *scaled VIP counts* in the virtual plane. Specifically, for object k at node n, we scale down the total exogenous and endogenous VIP arrivals $A_n^k(t) + \sum_{a\in\mathcal{N}} \mu_{an}^k(t)$ at slot t by a constant $\theta_n^k \geq 1$. Note that θ_n^k is a design parameter for the virtual plane, and can be chosen to reflect the average interest suppression effect within a slot for object k at node n.

Initially, all VIP counters are set to 0, i.e., $V_n^k(1) = 0$. The time evolution of the VIP count at node n for object k is as follows:

$$V_n^k(t+1) \leq \left(\left(V_n^k(t) - \sum_{b\in\mathcal{N}} \mu_{nb}^k(t)\right)^+ + \frac{A_n^k(t) + \sum_{a\in\mathcal{N}} \mu_{an}^k(t)}{\theta_n^k} - r_n s_n^k(t)\right)^+ \tag{3}$$

where $(x)^+ \triangleq \max(x, 0)$. Note that when $\theta_n^k = 1$ for all $k \in \mathcal{K}$ and $n \in \mathcal{N}$, the dynamics of VIPs in (3) reduces to the VIP dynamics in [3]. Thus, VIP counts in [3] capture the demand for the respective data objects without interest suppression. By setting $\theta_n^k \geq 1$ for all $k \in \mathcal{K}$ and $n \in \mathcal{N}$, VIP counts here can reflect the demand for the respective data objects with interest scaling.

From (3), it can be seen that the VIPs for data object k at node n at the beginning of slot t are transmitted during slot t at the rate $\sum_{b\in\mathcal{N}} \mu_{nb}^k(t)$. The exogenous and endogenous VIP arrivals $A_n^k(t) + \sum_{a\in\mathcal{N}} \mu_{an}^k(t)$ during slot t are scaled down by θ_n^k. The remaining VIPs $(V_n^k(t) - \sum_{b\in\mathcal{N}} \mu_{nb}^k(t))^+$, as well as the scaled-down exogenous and endogenous VIP arrivals $(A_n^k(t) + \sum_{a\in\mathcal{N}} \mu_{an}^k(t))/\theta_n^k$ during slot t, are reduced by r_n at the end of slot t if object k is cached at node n in slot t ($s_n^k(t) = 1$). The VIPs still remaining are then transmitted during the next slot $t + 1$. Note that (3) is an inequality because the actual number of VIPs for object k arriving to node n during slot t may be less than $\sum_{a\in\mathcal{N}} \mu_{an}^k(t)$ if the neighboring nodes have little or no VIPs of object k to transmit [4]. Furthermore, $V_n^k(t) = 0$ for all $t \geq 1$ if $n = src(k)$. In other words, the VIPs for an object exit the network once they reach the source node of the object. Physically, the VIP count can be interpreted as a *potential*. For any data object, there is a downward "gradient" from entry points of the data object requests to the content source and caching nodes.

The VIP queue at node n is *stable* if

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} 1_{[V_n^k(\tau)>\xi]} d\tau \to 0 \text{ as } \xi \to \infty,$$

where $1_{\{\cdot\}}$ is the indicator function. The *VIP network stability region* Λ is the closure of the set of all VIP arrival rates $\boldsymbol{\lambda} \triangleq (\lambda_n^k)_{n\in\mathcal{N},k\in\mathcal{K}}$ for which there exists some feasible (i.e., satisfying (1)-(2) and the cache size limits $(L_n)_{n\in\mathcal{N}}$) joint forwarding and caching policy which can guarantee that all VIP queues are stable [3]. Assume (i) the VIP arrival processes $\{A_n^k(t) : t = 1, 2, \ldots\}$ are mutually independent with respect to n and k; (ii) for all n and k, $\{A_n^k(t) : t = 1, 2, \ldots\}$ are i.i.d. with respect to t; and (iii) for all n and k, $A_n^k(t) \leq A_{n,\max}^k$ for all t. The theoretical results in this paper hold under these assumptions.[4] We now characterize the VIP stability region in the modified virtual plane (with VIP scaling).

THEOREM 1 (SCALED VIP STABILITY REGION). *The VIP stability region of the network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ with link capacity constraints* (1)-(2)*, and with VIP queue evolution* (3)*,*

[2]We assume that VIPs can be quantified as a real number. This is reasonable when the VIP counts are large.

[3]We think of a node as an aggregation point combining many network users, and hence it is likely to submit many requests for a data object over time.

[4]We omit proofs due to the page limitation. Please refer to [11] for details.

is the set Λ *consisting of all* $(\lambda_n^k)_{k\in\mathcal{K},n\in\mathcal{N}}$ *such that there exist flow variables* $(f_{ab}^k)_{k\in\mathcal{K},(a,b)\in\mathcal{L}}$ *and storage variables* $(\beta_{n,i,l})_{n\in\mathcal{N};i=1,\cdots,\binom{K}{l};\ l=0,\cdots,i_n\triangleq\lfloor L_n/D\rfloor}$ *satisfying*

$$f_{ab}^k \geq 0, f_{nn}^k = 0, f_{src(k)n}^k = 0,\ \forall a,b,n\in\mathcal{N}, k\in\mathcal{K} \quad (4)$$

$$f_{ab}^k = 0,\ \forall a,b\in\mathcal{N},\ k\in\mathcal{K},\ (a,b)\notin\mathcal{L}^k \quad (5)$$

$$0\leq\beta_{n,i,l}\leq 1,\ i=1,\cdots,\binom{K}{l},\ l=0,\cdots,i_n,\ n\in\mathcal{N} \quad (6)$$

$$\frac{\lambda_n^k}{\theta_n^k}\leq\sum_{b\in\mathcal{N}}f_{nb}^k-\frac{\sum_{a\in\mathcal{N}}f_{an}^k}{\theta_n^k}+r_n\sum_{l=0}^{i_n}\sum_{i=1}^{\binom{K}{l}}\beta_{n,i,l}\mathbf{1}[k\in\mathcal{B}_{n,i,l}],$$
$$\forall n\in\mathcal{N},\ k\in\mathcal{K}, n\neq src(k) \quad (7)$$

$$\sum_{k\in\mathcal{K}}f_{ab}^k\leq C_{ba}/D,\ \forall(a,b)\in\mathcal{L} \quad (8)$$

$$\sum_{l=0}^{i_n}\sum_{i=1}^{\binom{K}{l}}\beta_{n,i,l}=1,\ \forall n\in\mathcal{N} \quad (9)$$

Here, $\mathcal{B}_{n,i,l}$ *denotes the caching set consisting of the* i*-th combination of* l *data objects out of* K *data objects at node* n*, where* $i=1,\cdots,\binom{K}{l}$, $l=0,\cdots,i_n\triangleq\lfloor L_n/D\rfloor$.

To interpret Theorem 1, note that the flow variable f_{ab}^k represents the long-term VIP flow rate for data object k over link (a,b). The storage variable $\beta_{n,i,l}$ represents the long-term fraction of time that the set $\mathcal{B}_{n,i,l}$ (the i-th combination of l data objects out of K data objects) is cached at node n. Inequality (7) states that the scaled-down exogenous VIP arrival rate for data object k at node n is upper bounded by the total long-term outgoing VIP flow rate minus the total scaled-down endogenous long-term incoming VIP flow rate, plus the long-term VIP flow rate which is absorbed by all possible caching sets containing data object k at node n, weighted by the fraction of time each caching set is used. It is easy to see that the stability region in Theorem 1 (with VIP scaling) is a superset of the stability region in Theorem 2 of [3] (without VIP scaling). It is also clear that the stability region in Theorem 1 becomes larger when θ_n^k increases for all $k\in\mathcal{K}$ and $n\in\mathcal{N}$. Note that when $\theta_n^k=1$ for all $k\in\mathcal{K}$ and $n\in\mathcal{N}$, the VIP stability region in Theorem 1 reduces to the VIP stability region in Theorem 2 of [3].

4. THROUGHPUT OPTIMAL VIP CONTROL

First, we present a new joint dynamic forwarding and caching algorithm for VIPs in the virtual plane when $\boldsymbol{\lambda}\in\text{int}(\Lambda)$.

ALGORITHM 1 (SCALED FORWARDING AND CACHING). *At the beginning of each slot* t*, observe the VIP counts* $\mathbf{V}(t)\triangleq(V_n^k(t))_{n\in\mathcal{N},k\in\mathcal{K}}$ *and perform forwarding and caching in the virtual plane as follows.*

Forwarding*: For each data object* $k\in\mathcal{K}$ *and each link* $(a,b)\in\mathcal{L}^k$*, choose*

$$\mu_{ab}^k(t)=\begin{cases}C_{ba}/D, & W_{ab}^*(t)>0 \text{ and } k=k_{ab}^*(t)\\ 0, & \text{otherwise}\end{cases} \quad (10)$$

$W_{ab}^k(t)\triangleq V_a^k(t)-\frac{V_b^k(t)}{\theta_b^k}$, $W_{ab}^*(t)\triangleq\left(W_{ab}^{k_{ab}^*(t)}(t)\right)^+$, *and* $k_{ab}^*(t)\triangleq\arg\max_{k\in\{k:(a,b)\in\mathcal{L}^k\}}W_{ab}^k(t)$. *Here,* $W_{ab}^k(t)$ *is the backpressure weight of object* k *on link* (a,b) *at slot* t*, and* $k_{ab}^*(t)$ *is the data object which maximizes the backpressure weight on link* (a,b) *at time* t.

Caching*: At each node* $n\in\mathcal{N}$*, choose* $(s_n^k(t))_{n\in\mathcal{N},k\in\mathcal{K}}$ *to*

$$\max\sum_{k\in\mathcal{K}}V_n^k(t)s_n^k\quad s.t.\ \sum_{k\in\mathcal{K}}s_n^k\leq L_n/D. \quad (11)$$

Here, $V_n^k(t)$ *serves as the caching weight of object* k *at node* n.

Based on the forwarding and caching in (10) *and* (11)*, the VIP count is updated according to* (3).

Note that when $\theta_n^k=1$ for all $k\in\mathcal{K}$ and $n\in\mathcal{N}$, Algorithm 1 reduces to Algorithm 1 in [3]. The computation complexity of Algorithm 1 has the same order as that of Algorithm 1 in [3]. Later, in Section 5, we shall see that Algorithm 1 yields superior delay performance to Algorithm 1 in [3]. The forwarding part in Algorithm 1 is different from that in Algorithm 1 in [3]. At each slot t and for each link (a,b), the scaled backpressure-based forwarding algorithm allocates the entire normalized "reverse" link capacity C_{ba}/D to transmit the VIPs for the data object $k_{ab}^*(t)$ which maximizes the backpressure $W_{ab}^k(t)$. The scaled backpressure-based forwarding algorithm captures to some extent the interest suppression at receiving node b, and also spreads out the VIP levels from the high potential at the entry nodes to the low potential of the content source nodes and caching nodes, thus lowering the overall potential in the network. The caching part in Algorithm 1 is the same as that in Algorithm 1 in [3]. The max-weight caching algorithm implements the optimal solution to the max-weight knapsack problem in (11), i.e., allocate cache space at node n to the $\lfloor L_n/D\rfloor$ objects with the largest caching weights $V_n^k(t)$. The scaled forwarding and caching algorithm maximally balances out the scaled VIP counts by joint forwarding and caching, in order to prevent congestion building up in any part of the network, thereby reducing delay. Similar to Algorithm 1 in [3], Algorithm 1 can be implemented in a distributed manner.

We now show that Algorithm 1 adaptively stabilizes all VIP queues for any $\boldsymbol{\lambda}\in\text{int}(\Lambda)$, without knowing $\boldsymbol{\lambda}$. Thus, Algorithm 1 is *throughput optimal*, in the sense of adaptively maximizing the VIP throughput, and therefore the user demand rate satisfied by the network.

THEOREM 2 (THROUGHPUT OPTIMALITY). *If there exists* $\boldsymbol{\epsilon}=(\epsilon_n^k)_{n\in\mathcal{N},k\in\mathcal{K}}\succ\mathbf{0}$ *such that* $\boldsymbol{\lambda}+\boldsymbol{\epsilon}\in\Lambda$*, then the network of VIP queues under Algorithm 1 satisfies*

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=1}^{t}\sum_{n\in\mathcal{N},k\in\mathcal{K}}\mathbb{E}[V_n^k(\tau)]\leq\frac{NB}{\epsilon} \quad (12)$$

where $B\triangleq\frac{1}{2N}\sum_{n\in\mathcal{N}}\left((\mu_{n,\max}^{out})^2+\frac{A_{n,\max}+\mu_{n,\max}^{in}}{\theta_{n,min}}+r_{n,\max})^2+2\mu_{n,\max}^{out}r_{n,\max}\right)$, $\epsilon\triangleq\min_{n\in\mathcal{N},k\in\mathcal{K}}\epsilon_n^k$, $\mu_{n,\max}^{in}\triangleq\sum_{a\in\mathcal{N}}C_{an}/D$, $\mu_{n,\max}^{out}\triangleq\sum_{b\in\mathcal{N}}C_{nb}/D$, $\theta_{n,\min}\triangleq\min_{k\in\mathcal{K}}\theta_n^k$, $A_{n,\max}\triangleq\sum_{k\in\mathcal{K}}A_{n,\max}^k$, *and* $r_{n,\max}=Kr_n$.

The upper bound (12) on the average total number of VIPs is smaller than the upper bound in Theorem 1 of [3]. This is because during transmission within the modified virtual plane, the number of VIPs for an object is scaled down. In addition, the upper bound in (12) decreases when θ_n^k increases for all $k\in\mathcal{K}$ and $n\in\mathcal{N}$. Note that when $\theta_n^k=1$ for all $k\in\mathcal{K}$ and $n\in\mathcal{N}$, Theorem 2 reduces to Theorem 1 in [3].

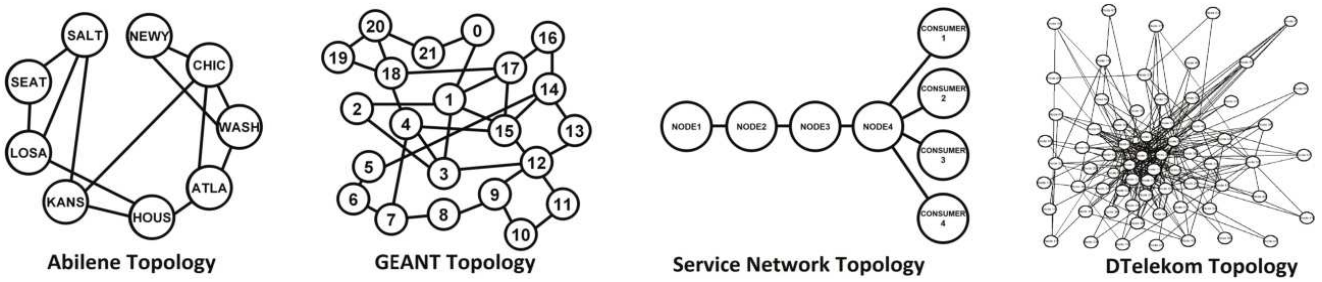

Figure 2: Network topologies [3].

5. EXPERIMENTAL EVALUATION

Using the scaled VIP algorithm in Algorithm 1, we can develop a corresponding stable caching VIP algorithm for handling Interest Packets and Data Packets in the actual plane using the mapping in [3] (please see therein for details). We now compare the delay performance of the new VIP algorithm for the actual plane resulting from Algorithm 1, denoted by NVIP, with the performance of the VIP algorithm for the actual plane resulting from Algorithm 1 in [3], denoted by VIP, and with the performance of six other baseline algorithms. These baseline algorithms use popular caching algorithms (LFU, LCE-UNIF, LCE-LRU, LCD-LRU, and LCE-BIAS) in conjunction with shortest path forwarding and a potential-based forwarding algorithm. The detailed descriptions of these baseline algorithms can be found in [3].

We consider two ways of choosing θ_n^k. In the first way, for simplicity, we choose $\theta_n^k = \theta$ for all $k \in \mathcal{K}$ and $n \in \mathcal{N}$. In the second way, we choose θ_n^k to be a moving average of $A_n^k(t) + \sum_{a\in\mathcal{N}} \mu_{an}^k(t)$ for all $k \in \mathcal{K}$ and $n \in \mathcal{N}$. In particular, we consider an exponential moving average (EMA), and update $\theta_n^k(t)$ according to $\theta_n^k(t) = (1-\beta)\theta_n^k(t-1) + \beta\left(A_n^k(t) + \sum_{a\in\mathcal{N}} \mu_{an}^k(t)\right)$ at each time t, where β is set to be 0.125 in the simulation.

Experimental scenarios are carried on four network topologies: the Service Topology, the Abilene Topology, the GEANT Topology and the DTelekom Topology, as shown in Fig. 4. In the Service Topology, NODE 1 is the content source for all objects, requests can be generated only by the CONSUMER nodes, and the cache size at each node is 5 GB. The cache size at each node is 5 GB in the Abilene Topology and is 2 GB in the GEANT Topology and the DTelekom Topology. In the Abilene Topology, the GEANT Topology and the DTelekom Topology, object requests can be generated by any node, and the content source for each data object is independently and uniformly distributed among all nodes. At each node requesting data, object requests arrive according to a Poisson process with an overall rate λ (in requests/node/slot). Each arriving request requests data object k (independently) with probability p_k, where $\{p_k\}$ follows a (normalized) Zipf distribution with parameter 0.75. We choose $K = 3000$, $C_{ab} = 500$ Mb/slot, and $D = 5$ MB. The Interest Packet size is 125B, and the Data Packet size is 50 KB. Each simulation generates requests for 10^4 time slots. Each curve is obtained by averaging over 10 simulation runs. The delay for an Interest Packet request is the difference (in time slots) between the fulfillment time (i.e., time of arrival of the requested Data Packet) and the creation time of the Interest Packet request. We use the total delay for all the Interest Packets generated over 10^4 time slots as the delay measure.

Fig. 3–Fig. 6 illustrate the delay performance for the four topologies. From these figures, we can observe that the VIP algorithm in [3] and the proposed new VIP algorithms with different scaling constants achieve much better delay performance than the six baseline schemes. In addition, the proposed new VIP algorithms with $\theta = 1.5$ and EMA $\theta_n^k(t)$ significantly improve the delay performance of the VIP algorithm (e.g., 58% at $\lambda = 30$ in Service, 40% at $\lambda = 60$ in Abilene, 58% at $\lambda = 40$ in GEANT and 57% at $\lambda = 60$ in DTelekom with EMA $\theta_n^k(t)$), indicating the advantage of the proposed VIP scaling in reflecting the interest suppression in the actual network. Note that a large θ may overemphasize interest suppression and underestimate measured demand, leading to performance degradation.

6. CONCLUSION

In this paper, we aimed to further improve the performance of the existing VIP algorithms by using a modified virtual plane where VIP counts are appropriately scaled to reflect interest suppression effects. We characterized the scaled VIP stability region in the virtual plane, developed a new distributed forwarding and caching algorithm, and proved the throughput optimality of the scaled proposed algorithm in the virtual plane.

Numerical experiments demonstrate significantly enhanced performance relative to the existing VIP algorithm, as well as a number of other baseline algorithms.

7. REFERENCES

[1] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, kc claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh. Named data networking (ndn) project. Oct. 2010.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, pages 1–12, New York, NY, USA, 2009. ACM.

[3] E. Yeh, T. Ho, Y. Cui, M. Burd, R. Liu, and D. Leong. Vip: A framework for joint dynamic forwarding and caching in named data networks. In *Proceedings of the 1st International Conference on Information-centric Networking*, ICN '14, pages 117–126, New York, NY, USA, 2014. ACM.

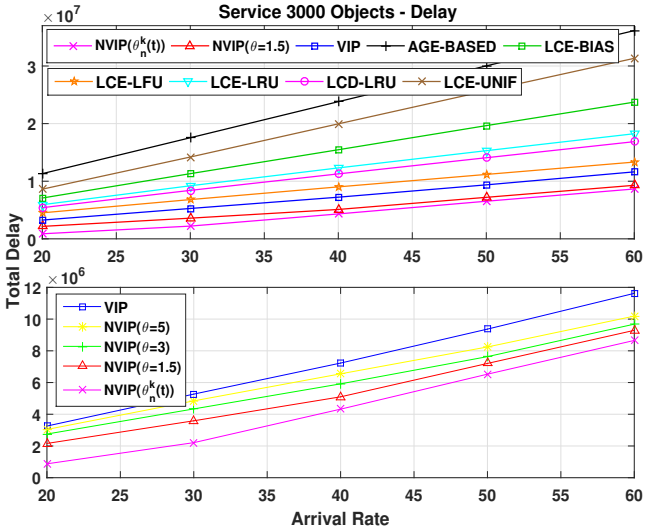

Figure 3: Average delay for Service Topology.

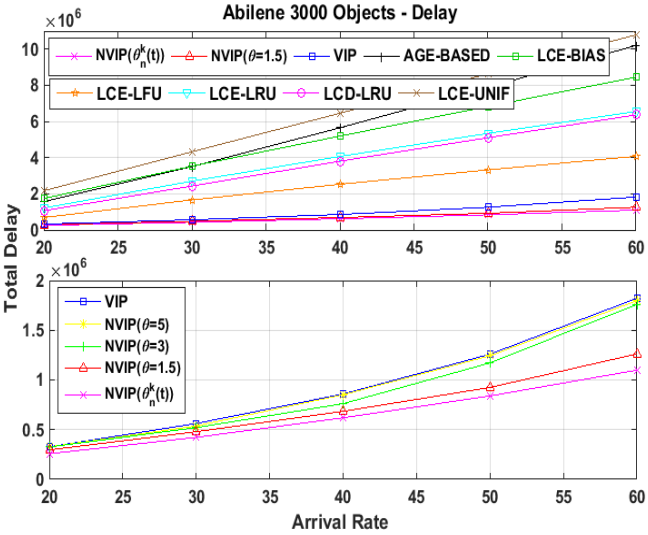

Figure 4: Average delay for Abilene Topology.

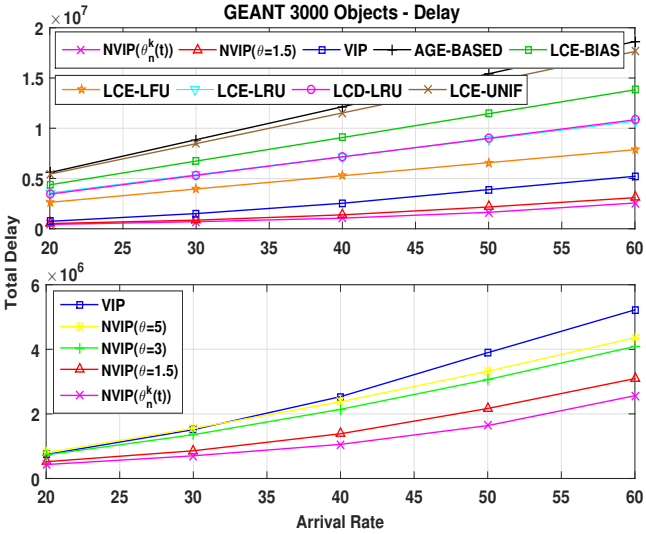

Figure 5: Average delay for GEANT Topology.

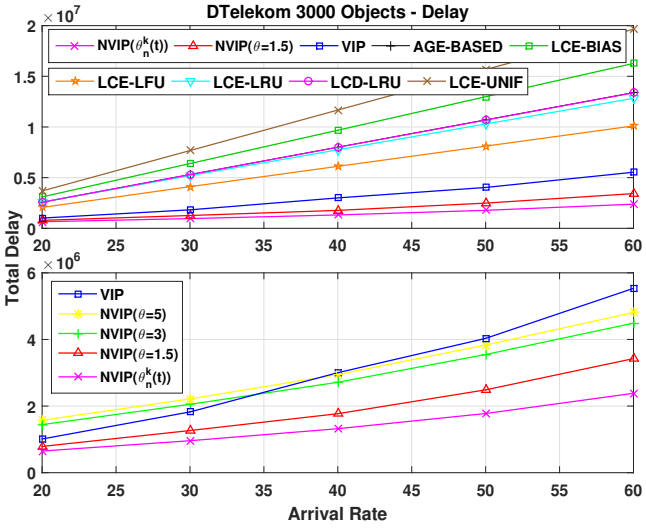

Figure 6: Average delay for DTelekom Topology.

[4] L. Georgiadis, M. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1(1):1-144, 2006.

[5] G. Rossini and D. Rossi. Coupling caching and forwarding: Benefits, analysis, and implementation. In ICN '14, pages 127–136, 2014.

[6] M. M. Amble, P. Parag, S. Shakkottai, and L. Ying. Content-aware caching and traffic management in content distribution networks. In *Proceedings of IEEE INFOCOM 2011*, pages 2858–2866, Shanghai, China, Apr. 2011.

[7] H. Xie, G. Shi, and P. Wang. Tecc: Towards collaborative in-network caching guided by traffic engineering. In *Proceedings of IEEE INFOCOM 2012:Mini-Conference*, pages 2546–2550, Orlando, Florida, USA, Mar. 2012.

[8] W. Chai, D. He, L. Psaras, and G. Pavlou. Cache "less for more" in information-centric networks. In *Proceedings of the 11th International IFIP TC 6 Conference on Networking - Volume Part I*, IFIP'12, pages 27–40, Berlin, Heidelberg, 2012. Springer-Verlag.

[9] Z. Ming, M. Xu, and D. Wang. Age-based cooperative caching in information-centric networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 268–273, March 2012.

[10] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang. Adaptive forwarding in named data networking. *SIGCOMM Comput. Commun. Rev.*, 42(3):62–67, June 2012.

[11] F. Lai, F. Qiu, W. Bian, Y. Cui, E. Yeh, "Scaled VIP Algorithms for Joint Dynamic Forwarding and Caching in Named Data Networks," *Report*, 2016. Available: http://arxiv.org/abs/1608.04198.