Enhanced VIP Algorithms for Forwarding, Caching, and Congestion Control in Named Data Networks

Ying Cui, Fan Lai
Shanghai Jiao Tong University, Shanghai, China

Edmund Yeh, Ran Liu
Northeastern University, Boston, 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, caching, and congestion control strategies within the Named Data Networking (NDN) architecture. While the existing VIP algorithms exhibit good performance, they are primarily focused on maximizing network throughput and utility, and do not explicitly consider user delay. In this paper, we develop a new class of enhanced algorithms for joint dynamic forwarding, caching and congestion control within the VIP framework. These enhanced VIP algorithms adaptively stabilize the network and maximize network utility, while improving the delay performance by intelligently making use of VIP information beyond one hop. Generalizing Lyapunov drift techniques, we prove the throughput optimality and characterize the utility-delay tradeoff of the enhanced VIP algorithms. Numerical experiments demonstrate the superior performance of the resulting enhanced algorithms for handling Interest Packets and Data Packets within the actual plane, in terms of low network delay and high network utility.**

I. 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], [4] 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, in our previous work [3], we propose the *VIP framework* for the design of high performing NDN networks. Within this VIP framework, we develop joint dynamic forwarding, caching and congestion control algorithms operating on virtual interest packets (VIPs) to maximize network utility subject to network stability in the virtual plane, using Lyapunov drift techniques [3], [4]. Then, using the resulting flow rates and queue lengths of the VIPs in the virtual plane, we develop the corresponding joint dynamic algorithms in the actual plane, which have been shown to achieve superior performance in terms of network throughput, user utility, user delay, and cache hit rates, relative to several baseline policies.

While the VIP algorithms in [3], [4] exhibit excellent performance, they are primarily focused on maximizing network throughput and do not explicitly consider user delay. In this paper, we aim to further improve the delay performance of the existing VIP algorithms in [3], [4] by leveraging VIP information beyond one hop. 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 jointly modifying forwarding, caching and congestion control in a tractable manner. Second, it is not clear how to maintain the desired throughput optimality and utility-delay tradeoff of the existing VIP algorithms when the Lyapunov-drift-based control structure is modified for improving the delay performance.

In the following, we shall address the above questions and challenges. We first develop a new class of enhanced distributed forwarding and caching algorithms operating on VIPs to stabilize network in the virtual plane. We then extend the algorithm to include congestion control, thus achieving a favorable utility-delay tradeoff. These enhanced VIP algorithms

Y. Cui gratefully acknowledges support from the National Science Foundation of China grant 61401272. E. Yeh gratefully acknowledges support from the National Science Foundation Future Internet Architecture grant CNS-1205562 and a Cisco Systems research grant.

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

reduce the delay of the existing VIP algorithms by 1) exploiting the margin between the VIP arrival rate vector and the boundary of the VIP network stability region, and 2) making use of VIP information beyond one hop in a simple and flexible manner. Generalizing Lyapunov drift techniques, we demonstrate the throughput optimality and characterize the utility-delay tradeoff of the enhanced VIP algorithms. These enhanced VIP algorithms generalize the VIP algorithms in [3], [4] in the sense that they maintain network stability and maximize network utility while improving delay performance. In addition, these enhanced VIP algorithms (designed for NDN networks) extend the enhanced dynamic backpressure algorithms in [5] (designed for traditional source-destination networks) in the sense that they incorporate caching into the joint design of dynamic forwarding (routing) and congestion control. Numerical experiments demonstrate the superior performance of the resulting enhanced algorithms for handling Interest Packets and Data Packets within the actual plane, in terms of low network delay and high network utility.

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 [6], the authors demonstrate the gains of joint forwarding and caching in ICNs. In [7], a potential-based forwarding scheme with random caching is proposed for ICNs. The results in [7] are heuristic in the sense that it remains unknown how to choose proper potential values to ensure good performance. In [8], the authors propose throughput-optimal one-hop routing and caching to support the maximum number of requests in a single-hop Content Distribution Network (CDN) setting. In [9], assuming the path between any two nodes is predetermined, the authors consider single-path routing (equivalently cache node selection) and caching to minimize link utilization for a general multi-hop content-oriented network. In [10], cooperative caching schemes have been heuristically designed without being jointly optimized with forwarding strategies. Finally, adaptive multipath forwarding in NDN has been examined in [11], but has not been jointly optimized with caching strategies.

II. Network Model

We consider the same network model as in [3], [4], 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 [4]. 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.

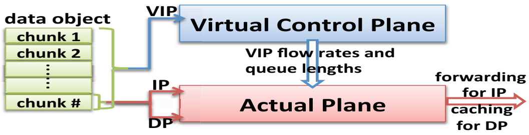


Fig. 1: VIP framework [3]. IP (DP) stands for Interest Packet (Data Packet).

III. VIP Framework

We adopt the VIP framework proposed in [3]. In the following, we briefly introduce the VIP framework 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 in the network, and represent content popularity which is empirically measured, rather than being given a priori. 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].

We now specify the 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* $V_n^k(t)$.[1] 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.[2] 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 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 as follows:

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

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

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], [4].

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:

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

where $(x)^+ \triangleq \max(x,0)$. Furthermore, $V_n^k(t) = 0$ for all $t \geq 1$ if $n = src(k)$. The detailed explanation of (3) can be found in [3], [4]. 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 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. Under these assumptions, Theorem 1 in [4] characterizes the VIP stability region in the virtual plane (or equivalently the Interest Packet stability region in the actual plane when there is no collapsing or suppression at the PITs). Note that the theoretical results in this paper also hold under these assumptions.

In the following, with the aim of improving the delay performance of the VIP algorithms in [3], [4], we focus on developing a new class of enhanced VIP algorithms within the virtual plane of the VIP framework, for the cases where $\boldsymbol{\lambda} \in \text{int}(\Lambda)$ and $\boldsymbol{\lambda} \notin \Lambda$ in Sections IV and V, respectively.

IV. Enhanced Throughput Optimal VIP Control

In this section, we consider the case where $\boldsymbol{\lambda} \in \text{int}(\Lambda)$, and develop a new class of enhanced joint dynamic forwarding and caching algorithms, within the virtual plane of the VIP framework.

A. Bias Function

The VIP algorithm, i.e., Algorithm 1 in [3], [4], focuses primarily on maximizing network throughput, and uses one-hop VIP count differences for forwarding and on per-node VIP counts for caching. This leads to a simple distributed implementation. On the other hand, by incorporating VIP count information beyond one hop in a tractable manner, one can potentially improve the delay performance of the VIP algorithm while retaining the desirable throughput optimality and distributed implementation. Toward this end, we consider a general nonnegative VIP count-dependent bias function for each node $n \in \mathcal{N}$ and object $k \in \mathcal{K}$ [5]:

$$f_n^k(\mathbf{v}) = \sum_{n'\in\mathcal{N}} \eta_{nn'}^k(\mathbf{v}) v_{n'}^k / z_{n'}^k. \tag{4}$$

Here, $\mathbf{v} \triangleq (v_n^k)_{n\in\mathcal{N},k\in\mathcal{K}}$ represents the VIP counts at a particular time slot and $\eta_{nn'}^k(\mathbf{v}) \in [0,1]$ is the weight associated with VIP count $v_{n'}^k$ at node n for object k, representing the relative importance of $v_{n'}^k$ in the bias at node n for object k. The parameter $z_{n'}^k > 0$ is designed to guarantee network stability and will be discussed below in Theorem 1 and Theorem 2. We can treat $v_{n'}^k / z_{n'}^k$ as a normalized version of $v_{n'}^k$. While the bias function in (4) is generally written as a function of the global VIP counts, one can choose the bias function to depend only on the local VIP counts within one hop [5]. For example, as in [5], we can choose the minimum next-hop VIP count bias function, i.e.,

$$f_n^k\left(\mathbf{v}_n^k\right) = \frac{1}{z} \min_{n'\in\{n':(n,n')\in\mathcal{L}^k\}} v_{n'}^k \tag{5}$$

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

[2] 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.

where $\mathbf{v}_n^k \triangleq (v_{n'}^k)_{n' \in \{n' : (n,n') \in \mathcal{L}^k\}}$.

Each specific choice of a bias function $\mathbf{f} \triangleq (f_n^k)_{n \in \mathcal{N}, k \in \mathcal{K}}$ corresponds to one enhanced VIP algorithm, and the number of VIP counts contributing to the bias function determines the implementation complexity of the corresponding enhanced VIP algorithm. The form of the bias function is carefully chosen to stabilize the network or maximize the network utility, while at the same time offering a high degree of flexibility in choosing specific enhanced VIP algorithms with manageable complexity, distributed implementation, and significantly better delay performance [5].

B. Enhanced Forwarding and Caching Algorithm

We now present a new class of enhanced joint dynamic forwarding and caching algorithms for VIPs in the virtual plane by incorporating the general VIP count-dependent bias function in (4) into Algorithm 1 in [3], [4].

Algorithm 1 (Enhanced 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

$$\mu_{ab}^k(t) = \begin{cases} C_{ba}/D, & W_{ab}^*(t) > 0 \text{ and } k = k_{ab}^*(t) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

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

$$\max \sum_{k \in \mathcal{K}} \left(V_n^k(t) + f_n^k(\mathbf{V}(t)\right) s_n^k \quad \text{s.t.} \sum_{k \in \mathcal{K}} s_n^k \le L_n/D. \tag{7}$$

Here, $V_n^k(t) + f_n^k(\mathbf{V}(t))$ represents the enhanced caching weight of object k at node n.

Based on the forwarding and caching in (6) and (7), the VIP count is updated according to (3). For simplicity, we use the same bias function for forwarding and caching. In general, the bias functions for forwarding and caching can be different.

Remark 1 (Connection to the Algorithms in [5] and [3], [4]): Algorithm 1 (designed for NDN networks) extends the enhanced dynamic backpressure algorithms in [5] (designed for traditional source-destination networks) in the sense that it incorporates caching into the joint design of dynamic forwarding and congestion control. Algorithm 1 also generalizes Algorithm 1 in [3], [4] in the sense that it maintains network stability while improving delay performance by incorporating VIP information beyond one hop.

At each slot t and for each link (a,b), the enhanced 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 enhanced backpressure $W_{ab}^k(t)$. The enhanced max-weight caching algorithm implements the optimal solution to the max-weight knapsack problem in (7), i.e., allocate cache space at node n to the $\lfloor L_n/D \rfloor$ objects with the largest enhanced caching weights $V_n^k(t) + f_n^k(\mathbf{V}(t))$. The enhanced forwarding and caching algorithm maximally balances out the VIP counts by joint forwarding and caching, to prevent congestion building up in any part of the network, thereby reducing delay.

It is important to note that with local VIP count-bias functions, such as the minimum next-hop VIP count bias function in (5), both the enhanced backpressure-based forwarding algorithm and the enhanced max-weight caching algorithm are *distributed*. Following the complexity analysis in [5], we know that the enhanced VIP algorithm in Algorithm 1 with the minimum next-hop VIP bias function in (5) has the same order of implementation complexity as Algorithm 1 in [3], [4] (without any bias functions). In general, for ease of implementation, one should intelligently choose the VIP counts which contribute to the bias function, leading to enhanced VIP algorithms with distributed implementation and good delay performance.

C. Throughput Optimality

We now show that Algorithm 1 adaptively stabilizes all VIP queues for any $\boldsymbol{\lambda} \in \text{int}(\Lambda)$, without knowing $\boldsymbol{\lambda}$.[3]

Theorem 1 (Generalized Throughput Optimality): Given $\boldsymbol{\epsilon} \triangleq (\epsilon_n^k)_{n \in \mathcal{N}, k \in \mathcal{K}} \succeq \mathbf{0}$ such that $\boldsymbol{\lambda} + \boldsymbol{\epsilon} \in \text{int}(\Lambda)$, there exist $\boldsymbol{\delta} \triangleq (\delta_n^k)_{n \in \mathcal{N}, k \in \mathcal{K}} \succ \mathbf{0}$ such that $\boldsymbol{\lambda} + \boldsymbol{\epsilon} + \boldsymbol{\delta} \in \Lambda$, and $\mathbf{z} \triangleq (z_n^k)_{n \in \mathcal{N}, k \in \mathcal{K}} \succ \mathbf{0}$ such that $\boldsymbol{\epsilon}_z \triangleq \left(\frac{2C_{\max}L^k + Nr_{\max}}{z_n^k}\right)_{n \in \mathcal{N}, k \in \mathcal{K}} \preceq \boldsymbol{\epsilon}$. Then, the network of VIP queues under Algorithm 1 satisfies

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} \sum_{n \in \mathcal{N}, k \in \mathcal{K}} \mathbb{E}[V_n^k(\tau)] \le \frac{NB}{\beta_z} \tag{8}$$

where $B \triangleq \frac{1}{2N} \sum_{n \in \mathcal{N}} \left((\mu_{n,\max}^{out})^2 + (A_{n,\max} + \mu_{n,\max}^{in} + Kr_n)^2 + 2\mu_{n,\max}^{out} Kr_n\right)$, $\beta_z = \sup_{\{(\boldsymbol{\epsilon},\boldsymbol{\delta}):\boldsymbol{\epsilon} \succ \boldsymbol{\epsilon}_z, \boldsymbol{\delta} \succ \mathbf{0}, \boldsymbol{\lambda}+\boldsymbol{\epsilon}+\boldsymbol{\delta} \in \in \Lambda\}} \min_{n \in \mathcal{N}, k \in \mathcal{K}} \left\{\epsilon_n^k + \delta_n^k - \frac{2C_{\max}L^k + Nr_{\max}}{z_n^k}\right\}$, with $\mu_{n,\max}^{in} \triangleq \sum_{a \in \mathcal{N}} C_{an}/D$, $\mu_{n,\max}^{out} \triangleq \sum_{b \in \mathcal{N}} C_{nb}/D$, $A_{n,\max} \triangleq \sum_{k \in \mathcal{K}} A_{n,\max}^k$, and $r_{\max} \triangleq \max_{n \in \mathcal{N}} r_n$.

Similar to Theorem 1 in [5], Theorem 1 should be interpreted as follows. When it is given that $\boldsymbol{\lambda}$ is bounded away from the boundary of Λ by at least $\boldsymbol{\epsilon} \succ \mathbf{0}$, i.e., $\boldsymbol{\lambda} + \boldsymbol{\epsilon} \in \text{int}(\Lambda)$, one can choose a *finite* $\mathbf{z} \succ \mathbf{0}$ such that $\boldsymbol{\epsilon}_\mathbf{z} \prec \boldsymbol{\epsilon}$. In this case, Algorithm 1 can improve the delay performance of Algorithm 1 in [3], [4] (which will be demonstrated numerically in Section VI) while maintaining a generalized notion of throughput optimality, by

[3] We omit proofs due to the page limitation. Please refer to [12] for details.

exploiting the margin $\boldsymbol{\epsilon} \succ \mathbf{0}$ to incorporate VIP counts beyond one-hop [5]. When it is only known that $\boldsymbol{\lambda} \in \text{int}(\Lambda)$ and no extra margin is given ($\boldsymbol{\epsilon} = \mathbf{0}$), then by Theorem 1, z_n^k must be chosen to be infinity for all $n \in \mathcal{N}$ and $k \in \mathcal{K}$ (i.e., $\mathbf{f} = \mathbf{0}$). In this case, Algorithm 1 reduces to Algorithm 1 in [3], [4], and Theorem 1 reduces to Theorem 2 in [4]. Theorem 1 can be seen as the generalization of the throughput optimal results in Theorem 1 of [5] and Theorem 2 of [4].

V. Enhanced VIP Congestion Control

The VIP forwarding and caching algorithm first described in [3] is extended to incorporate congestion control in [4]. Here, we develop a new class of enhanced algorithms which generalize the enhanced forwarding and caching algorithm (Algorithm 1) described above to incorporate congestion control.

A. Transport Layer and Network Layer VIP Dynamics

Even with throughput optimal forwarding and caching, excessively large request rates ($\boldsymbol{\lambda} \notin \Lambda$) can overwhelm a NDN network with limited resources. When $\boldsymbol{\lambda} \notin \Lambda$, newly arriving Interest Packets (equivalently VIPs) first enter transport-layer storage reservoirs before being admitted to network-layer queues. Let $Q_{n,\max}^k$ and $Q_n^k(t)$ denote the transport layer VIP buffer size and VIP count for object k at node n at the beginning of slot t, respectively. Let $\alpha_n^k(t) \geq 0$ denote the amount of VIPs admitted to the network layer VIP queue of object k at node n from the transport layer VIP queue at slot t. Assume $\alpha_n^k(t) \leq \alpha_{n,\max}^k$, where $\alpha_{n,\max}^k$ is a positive constant which limits the burstiness of the admitted VIPs to the network layer. We have the following time evolutions of the transport and network layer VIP counts [4], [13]:

$$Q_n^k(t+1) = \min\left\{\left(Q_n^k(t) - \alpha_n^k(t)\right)^+ + A_n^k(t), Q_{n,\max}^k\right\} \tag{9}$$

$$V_n^k(t+1) \leq \left(\left(V_n^k(t) - \sum_{b\in\mathcal{N}} \mu_{nb}^k(t)\right)^+ + \alpha_n^k(t) + \sum_{a\in\mathcal{N}} \mu_{an}^k(t) - r_n s_n^k(t)\right)^+. \tag{10}$$

B. Enhanced Congestion Control Algorithm

The goal of congestion control is to support a portion of the VIPs which maximize the sum utility when $\boldsymbol{\lambda} \notin \Lambda$. Let $g_n^k(\cdot)$ be the utility function associated with the VIPs admitted into the network layer for object k at node n. Assume $g_n^k(\cdot)$ is non-decreasing, concave, continuously differentiable and non-negative. Define a $\boldsymbol{\theta}$-optimal admitted VIP rate [4], [13]:

$$\overline{\boldsymbol{\alpha}}^*(\boldsymbol{\theta}) \triangleq \arg\max_{\overline{\boldsymbol{\alpha}}} \quad \sum_{n\in\mathcal{N},k\in\mathcal{K}} g_n^k\left(\overline{\alpha}_n^k\right) \tag{11}$$

$$s.t. \quad \overline{\boldsymbol{\alpha}} + \boldsymbol{\theta} \in \Lambda, \ \mathbf{0} \preceq \overline{\boldsymbol{\alpha}} \preceq \boldsymbol{\lambda} \tag{12}$$

where $\overline{\boldsymbol{\alpha}}^*(\boldsymbol{\theta}) \triangleq (\overline{\alpha}_n^{k*}(\boldsymbol{\theta}))_{n\in\mathcal{N},k\in\mathcal{K}}$, $\overline{\boldsymbol{\alpha}} \triangleq (\overline{\alpha}_n^k)_{n\in\mathcal{N},k\in\mathcal{K}}$ and $\mathbf{0} \preceq \boldsymbol{\theta} \triangleq (\theta_n^k)_{n\in\mathcal{N},k\in\mathcal{K}} \in \Lambda$. Due to the non-decreasing property of the utility functions, the maximum sum utility over all $\boldsymbol{\theta}$ is achieved at $\overline{\boldsymbol{\alpha}}^*(\mathbf{0})$ when $\boldsymbol{\theta} = \mathbf{0}$.

We now develop a new class of enhanced joint congestion control, forwarding and caching algorithms that yield a throughput vector which can be arbitrarily close to the optimal solution $\overline{\boldsymbol{\alpha}}^*(\mathbf{0})$. We introduce auxiliary variables $\gamma_n^k(t)$ and the virtual VIP count $Y_n^k(t)$ for all $n \in \mathcal{N}$ and $k \in \mathcal{K}$. Set $Y_n^k(0) = 0$ for all $n \in \mathcal{N}$ and $k \in \mathcal{K}$.

Algorithm 2: (Enhanced Congestion Control, Forwarding and Caching) At the beginning of each slot t, observe the VIP counts $\mathbf{V}(t)$ and $\mathbf{Y}(t) \triangleq (Y_n^k(t))_{n\in\mathcal{N},k\in\mathcal{K}}$, and perform the following congestion control, forwarding and caching in the virtual plane as follows.[4]

Congestion Control: For each node n and object k, choose the admitted VIP count at slot t, which also serves as the output rate of the corresponding virtual queue:

$$\alpha_n^k(t) = \begin{cases} \min\left\{Q_n^k(t), \alpha_{n,\max}^k\right\}, & Y_n^k(t) > V_n^k(t) \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Then, choose the auxiliary variable, which serves as the input rate to the corresponding virtual queue:

$$\gamma_n^k(t) = \arg\max_{\gamma} \quad W g_n^k(\gamma) - Y_n^k(t)\gamma \tag{14}$$
$$s.t. \quad 0 \leq \gamma \leq \alpha_{n,\max}^k$$

where $W > 0$ is a control parameter which affects the utility-delay tradeoff of the algorithm. Based on the chosen $\alpha_n^k(t)$ and $\gamma_n^k(t)$, the transport layer VIP count is updated according to (9) and the virtual VIP count is updated according to:

$$Y_n^k(t+1) = \left(Y_n^k(t) - \alpha_n^k(t)\right)^+ + \gamma_n^k(t). \tag{15}$$

Forwarding and Caching: Same as Algorithm 1. The network layer VIP count is updated according to (10).

C. Utility-Delay Tradeoff

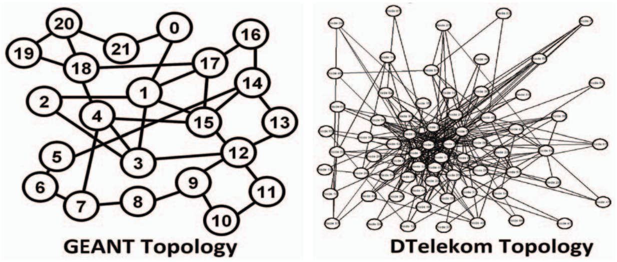
We now show that by tuning control parameter $W > 0$, Algorithm 2 adaptively achieves a utility-delay tradeoff for VIP queues, for any $\boldsymbol{\lambda} \notin \Lambda$, without knowing $\boldsymbol{\lambda}$. In addition, Algorithm 2 yields a throughput vector which can be arbitrarily close to $\overline{\boldsymbol{\alpha}}^*(\mathbf{0})$ by letting $W \to 0$.

Theorem 2 (Utility-Delay Tradeoff): For an arbitrary VIP arrival rate $\boldsymbol{\lambda}$ and for any control parameter $W > 0$, given $\boldsymbol{\epsilon} \in \text{int}(\Lambda)$, there exist $\boldsymbol{\delta} \succ \mathbf{0}$ such that $\boldsymbol{\epsilon} + \boldsymbol{\delta} \in \Lambda$, and $\mathbf{z} \succ \mathbf{0}$ such that $\boldsymbol{\epsilon}_z \preceq \boldsymbol{\epsilon}$. Then, the network of VIP queues under Algorithm 2 satisfies

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} \sum_{n\in\mathcal{N},k\in\mathcal{K}} \mathbb{E}[V_n^k(\tau)] \leq \frac{2N\hat{B} + WG_{\max}}{2\hat{\beta}_z} \tag{16}$$

$$\liminf_{t\to\infty} \sum_{n\in\mathcal{N},k\in\mathcal{K}} g_n^k\left(\overline{\alpha}_n^k(t)\right) \geq \sum_{n\in\mathcal{N},k\in\mathcal{K}} g_n^k\left(\overline{\alpha}_n^{k*}(\boldsymbol{\epsilon}_z)\right) - \frac{2N\hat{B}}{W} \tag{17}$$

[4] Note that the congestion control part of Algorithm 2 is the same as that in [13, page 90] and Algorithm 3 in [4]. The difference lies in the forwarding and caching part. We describe the congestion control part here for the purpose of completeness.



(a) GEANT (22 nodes).

(b) DTelekom (68 nodes).

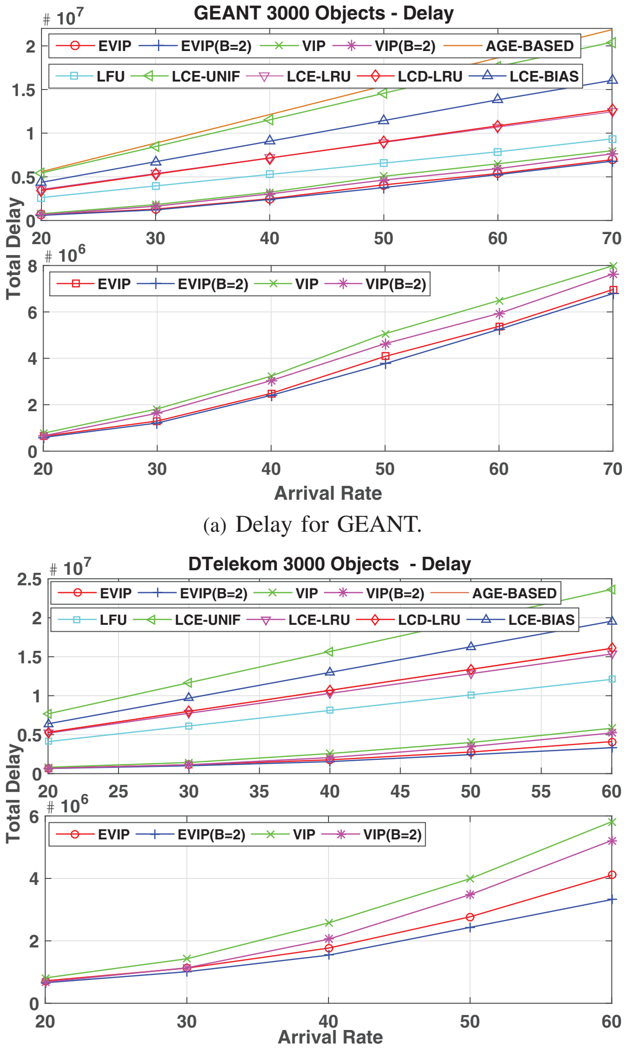
Fig. 2: Network topologies.

where $\hat{B} \triangleq \frac{1}{2N} \sum_{n\in\mathcal{N}} \Big((\mu_{n,\max}^{out})^2 + (\alpha_{n,\max} + \mu_{n,\max}^{in} + K r_n)^2 + 2\left(\alpha_{n,\max}\right)^2 + 2\mu_{n,\max}^{out} K r_n \Big)$ and $\hat{\beta}_z = \sup_{\{(\boldsymbol{\epsilon},\boldsymbol{\delta}):\boldsymbol{\epsilon}\succeq\boldsymbol{\epsilon}_{\mathbf{z}},\boldsymbol{\delta}\succ\mathbf{0},\ \boldsymbol{\epsilon}+\boldsymbol{\delta}\in\Lambda\}} \min_{n\in\mathcal{N},k\in\mathcal{K}} \Big\{ \epsilon_n^k + \delta_n^k - \frac{2C_{\max}L^k + N r_{\max}}{z_n^k} \Big\}$ with $\alpha_{n,\max} \triangleq \sum_{k\in\mathcal{K}} \alpha_{n,\max}^k$, $G_{\max} \triangleq \sum_{n\in\mathcal{N},k\in\mathcal{K}} g_n^k\left(\alpha_{n,\max}^k\right)$, $\overline{\alpha}_n^k(t) \triangleq \frac{1}{t}\sum_{\tau=1}^{t}\mathbb{E}[\alpha_n^k(\tau)]$.

Similar to Theorem 2 in [5], Theorem 2 should be interpreted as follows. When $\mathbf{0} \prec \boldsymbol{\epsilon} \in \text{int}(\Lambda)$, one can choose a *finite* $\mathbf{z} \succ \mathbf{0}$ such that $\boldsymbol{\epsilon}_{\mathbf{z}} \preceq \boldsymbol{\epsilon}$. In this case, Algorithm 2 can improve the utility-delay tradeoff of Algorithm 3 in [4] (which will be demonstrated numerically in Section VI), by exploiting the margin $\boldsymbol{\epsilon}$ to incorporate VIP counts beyond one-hop [5]. When $\boldsymbol{\epsilon} = \mathbf{0}$, i.e., no margin is given, z_n^k is chosen to be infinity for all $n \in \mathcal{N}$ and $k \in \mathcal{K}$ (i.e., $\mathbf{f} = \mathbf{0}$). In this case, Algorithm 2 reduces to Algorithm 3 in [4], and Theorem 2 reduces to Theorem 3 in [4]. Theorem 2 can be seen as the generalization of the utility-delay tradeoff results in Theorem 2 of [5] and Theorem 3 of [4].

VI. Experimental Evaluation

Based on the enhanced VIP algorithms in Algorithm 1 and Algorithm 2 operating on VIPs in the virtual plane, we can develop the corresponding algorithms for handling Interest Packets and Data Packets in the actual plane using a mapping similar to that in [3], [4]. We omit the details due to the page limitation. In this section, we first compare the delay performance of the enhanced VIP algorithm for the actual plane resulting from Algorithm 1 (using the minimum next-hop bias function in (5) with $z = 1$), denoted by EVIP, with the VIP algorithm for the actual plane resulting from Algorithm 1 in [3], [4], denoted by VIP, as well as with six other baseline algorithms. In particular, 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 [4]. We then compare the utility-delay tradeoff of the enhanced VIP algorithm for the actual plane resulting from Algorithm 2 (using the minimum next-hop bias function in (5) with $z = 1$), also denoted by EVIP, with the VIP algorithm for the actual plane resulting from Algorithm 3 in [4], also denoted by VIP, with some abuse of notation. To our knowledge, there are no other congestion control algorithms for NDN networks which can easily control the tradeoff between utility and delay. In evaluating delay and utility-delay tradeoff, as in [3], [4], we also consider the constant shortest path bias versions of EVIP and VIP, with B being the per-link cost.

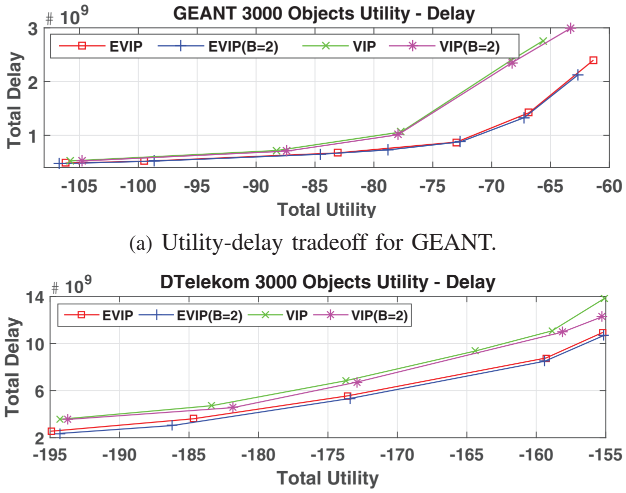


(a) Delay for GEANT.

(b) Delay for DTelekom.

Fig. 3: Average delay.

Experimental scenarios are carried on two network topologies: the GEANT Topology and the DTelekom Topology, as shown in Fig. 2. In the two topologies, object requests can be generated by any node, and the content source for each data object is independently and uniformly distributed among



(a) Utility-delay tradeoff for GEANT.

(b) Utility-delay tradeoff for DTelekom.

Fig. 4: Utility-delay tradeoff.

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, $L_n = 2$ GB, $D = 5$ 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. We consider α-utility function with $\alpha = 2$, i.e., $g_n^k(x) = -\frac{1}{x}$, and use the sum utility over all nodes and all objects as the utility measure.

Fig. 3 illustrates the delay performance. We can observe that the VIP algorithms in [3], [4] and the proposed enhanced VIP algorithms achieve much better delay performance than the six baseline schemes, especially in DTelekom. In addition, the proposed enhanced VIP algorithms significantly improve the delay performance of the VIP algorithms (e.g., about 28% at $\lambda = 30$ in GEANT and 30% at $\lambda = 40$ in DTelekom). Fig. 4 illiterates the utility-delay tradeoff. We can observe that the proposed enhanced VIP algorithms achieve significantly better utility-delay tradeoff than the VIP algorithms in [4]. In summary, the proposed enhanced VIP algorithms improve the delay performance of the VIP algorithms in [3], [4] by intelligently exploiting the VIP counts beyond one hop for forwarding, caching and congestion control.

VII. Conclusion

In this paper, we develop a new class of enhanced distributed forwarding, caching and congestion control algorithms within the VIP framework, which adaptively stabilize the network and maximize network utility, while improving delay performance. We prove the throughput optimality and characterize the utility-delay tradeoff of the enhanced VIP algorithms in the virtual plane. Numerical experiments demonstrate the superior performance of the resulting algorithms for handling Interest Packets and Data Packets in the actual plane, in terms of low network delay and high network utility, relative to a number of baseline alternatives.

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*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1658939.1658941

[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*, ser. ICN '14. New York, NY, USA: ACM, 2014, pp. 117–126. [Online]. Available: http://doi.acm.org/10.1145/2660129.2660151

[4] E. M. Yeh, T. Ho, Y. Cui, R. Liu, M. Burd, and D. Leong, "Forwarding, caching and congestion control in named data networks," *CoRR*, vol. abs/1310.5569, 2013. [Online]. Available: http://arxiv.org/abs/1310.5569

[5] Y. Cui, E. M. Yeh, and R. Liu, "Enhancing the delay performance of dynamic backpressure algorithms," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–14, 2015.

[6] G. Rossini and D. Rossi, "Coupling caching and forwarding: Benefits, analysis, and implementation," in *Proceedings of the 1st International Conference on Information-centric Networking*, ser. ICN '14, 2014, pp. 127–136.

[7] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga, "Catt: Potential based routing with content caching for icn," in *Proceedings of SIGCOMM 2012 ICN*, Helsinki, Finland, Aug. 2012, pp. 49–54.

[8] 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*, Shanghai, China, Apr. 2011, pp. 2858–2866.

[9] 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*, Orlando, Florida, USA, Mar. 2012, pp. 2546–2550.

[10] 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*, March 2012, pp. 268–273.

[11] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 3, pp. 62–67, Jun. 2012. [Online]. Available: http://doi.acm.org/10.1145/2317307.2317319

[12] Y. Cui, F. Lai, E. M. Yeh, and R. Liu, "Enhanced forwarding, caching and congestion control in named data networks," *Report*, 2016. [Online]. Available: https://arxiv.org/abs/1607.03270

[13] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.