Computer Networks 125 (2017) 160–171

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Distributed algorithms for content placement in hierarchical cache networks

Slađana Jošilo, Valentino Pacifici*, György Dán

Department of Communication Networks and ACCESS Linnaeus Center, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden

A R T I C L E I N F O

Article history:
Received 2 September 2016
Revised 17 February 2017
Accepted 26 May 2017
Available online 27 May 2017

Keywords:
Content caching
Distributed placement
Hierarchical cache network
Mobile backhaul
Approximation ratio
Heuristics

A B S T R A C T

The growing popularity of mobile multimedia content and the increase of wireless access bitrates are straining backhaul capacity in mobile networks. A cost-effective solution to reduce the strain, enabled by emerging all-IP 4G and 5G mobile backhaul architectures, could be in-network caching of popular content during times of peak demand. Motivated by the potential benefits of caching in mobile backhaul networks, in this paper we formulate the problem of content placement in a hierarchical cache network as a binary integer programming problem. We provide a polynomial time solution when the link costs are induced by a potential and we propose a 2-approximation algorithm for the general case. The 2-approximation requires full information about the network topology and the link costs, as well as about the content demands at the different caches, we thus propose two distributed algorithms that are based on limited information on the content demands. We show that the distributed algorithms terminate in a finite number of steps, and we provide analytical results on their approximation ratios. We use simulations to evaluate the proposed algorithms in terms of the achieved approximation ratio and computational complexity on hierarchical cache network topologies as a model of mobile backhaul networks.

1. Introduction

The penetration of high speed mobile access technologies, such as HSDPA and LTE, together with the proliferation of powerful handheld devices has stimulated a rapid increase of user demand for mobile multimedia content in recent years. The traffic growth is predicted to continue in coming years, with an estimated 10-fold increase in mobile data traffic in 5 years and an increasing peak-to-average traffic ratio, and puts significant strain on mobile backhaul capacity.

Recent measurement studies of mobile data traffic indicate that caching could be an effective means of decreasing the mobile backhaul bandwidth requirements: caching could reduce the bandwidth demand by up to 95% during peak hours and could at the same time reduce content delivery time by a factor of three as shown in [1]. Such a high cache efficiency is likely due to the concentration of content popularity to relatively few content items during peak hours, a phenomenon that has been observed for, e.g., multimedia content [2]. At the same time, mobile traffic is dominated by downloads; up to 75% of daily traffic load comes from download traffic, and the demand shows significant diurnal fluctuations with low loads during early morning hours [3].

Tunelling imposed by previous 3GPP standards made backhaul in-network caching technically challenging [4], allowing only caches at the network edge, in emerging all-IP mobile backhaul architectures the caches could be co-located with every switch and could implement cooperative caching policies throughout the backhaul. Since fairly accurate content popularity predictions can be obtained for Web and video content [5,6], the most popular contents could be downloaded into the caches of the mobile backhaul in the early morning hours when the load is relatively low, thereby alleviating the traffic demand during peak hours. A similar approach could be used in content distribution networks (CDNs) and edge caching architectures in wireline networks, as done in the Netflix Open Connect program [7].

Given predicted content popularities, a fundamental problem of in-network caching is to find efficient content placement algorithms that take into consideration the characteristics of the network topology and of the content workload. The algorithms should achieve close to optimal bandwidth cost savings and should have low computational complexity. Furthermore, they should require as little information as possible, e.g., about content popularities and network topology, in order to allow fully distributed operation and scaling to large topologies with small communication overhead.

* Corresponding author.
E-mail addresses: josilo@kth.se (S. Jošilo), pacifici@kth.se (V. Pacifici), gyuri@kth.se (G. Dán).

http://dx.doi.org/10.1016/j.comnet.2017.05.029
1389-1286/

While previous works proposed centralized and distributed content placement algorithms for two-level hierarchical topologies [8], general topologies with an ultrametric [9], and topologies in a metric space [10], efficient distributed algorithms based on limited topological information have received little attention.

Motivated by the observation that mobkile backhaul networks and edge caching architectures can often well be modeled by a cache hierarchy, in this paper we formulate the problem of content placement based on predicted demands in a hierarchical cache network with asymmetric link costs. We formulate the problem as a 0-1 integer programming problem, and show that under the potential induced link cost model the problem can be solved in polynomial time. For the general case we show that a 2-approximation to the problem can be obtained using a distributed greedy algorithm when global information is available, and propose two computationally simple distributed algorithms that do not require global information. We evaluate the algorithms through extensive simulations on various network topologies. Our results show that information about object demands at descendants is not sufficient for achieving good performance, but the proposed *h*-Push Down algorithm achieves consistently good performance based on a limited amount of information about object placements.

The rest of the paper is organized as follows. Section 2 describes the system model and provides the problem formulation. Section 4 describes the 2-approximation algorithm based on global information, and Section 5 describes the distributed algorithms based on limited information. Section 6 shows performance results based on simulations. Section 7 discusses related work and Section 8 concludes the paper.

2. System model and problem formulation

We consider a typical mobile backhaul, which is the primary motivating use case of our work, and model its active topology by a symmetric acyclic directed graph $\mathcal{G}(\mathcal{N}, E)$, where the vertices $\mathcal{N}$ are routers that connect cell sites and may aggregate traffic from other routers (and thus cell cites), and for every connected pair of nodes $i, j \in \mathcal{N}$ there exist edges $(i, j) \in E$ and $(j, i) \in E$. Observe that since $\mathcal{G}$ is connected and acyclic, $\mathcal{G}$ is a tree. The assumption that the active topology is a tree is realistic for the access part of the mobile backhaul in urban environments. We denote by $\mathcal{L}$ the set of leaf nodes in $\mathcal{G}$, by $\mathcal{I}$ the set of internal nodes and by n_0 the root node, i.e., $\mathcal{N} = \mathcal{L} \cup \mathcal{I} \cup n_0$. We denote the unique simple path from node i to node j by $P_{i,j} = \Big((i, v_1), (v_1, v_2), \ldots, (v_{|P_{i,j}|-1}, j)\Big)$, and we denote by $|P_{i,j}|$ the number of edges in path $P_{i,j}$. Observe that $|P_{i,j}| = |P_{j,i}|$. We define the level $l(i)$ of node $i \in \mathcal{N}$ in the tree $\mathcal{G}$ as the number of edges from node i to the tree's root node n_0 in the unique simple path from i to n_0, i.e., $l(i) = |P_{i,n_0}|$. We denote the children of node $i \in \mathcal{N}$ by $\mathcal{C}(i) \triangleq \{j | (i, j) \in E \wedge l(j) > l(i)\}$ and the parent of node i by $\mathcal{P}(i)$, where $\mathcal{P}(i) \in \mathcal{N}$ such that $i \in \mathcal{C}(\mathcal{P}(i))$. We denote by $\mathcal{P}^l(i)$ the l^{th}-ancestor of node i, e.g., $\mathcal{P}^2(i) = \mathcal{P}(\mathcal{P}(i))$. By definition $\mathcal{P}^0(i) = i$. We refer to an edge (i, j) as the downlink direction if $j \in \mathcal{C}(i)$ and as the uplink direction if $i \in \mathcal{C}(j)$.

We say that two nodes are siblings if they have the same parent, and define the sibling set $\mathcal{S}(i) \triangleq \{j | \mathcal{P}(j) = \mathcal{P}(i) \wedge i \neq j\}$. We denote the descendants of node i by $\mathcal{D}(i) \triangleq \{j | l(j) > l(i) \wedge \mathrm{LCA}(i, j) = i\}$, where LCA$(i, j)$ denotes the lowest common ancestor of nodes i and j, furthermore we use the notation $\mathcal{G}_i(\mathcal{N}_i, E_i)$ for the subgraph induced by $\mathcal{N}_i = \{i\} \cup \mathcal{D}(i)$ rooted in i.

2.1. Objects, demand and storage

We denote the set of objects requested by mobile nodes by $\mathcal{O}$. We follow common practice and consider that every object has unit size [11,12], which is a reasonable simplification if content is divisible into unit-sized chunks. We denote the average request rate (demand) predicted for the peak hours for object $o \in \mathcal{O}$ at the cell site connected to node i by w_i^o.

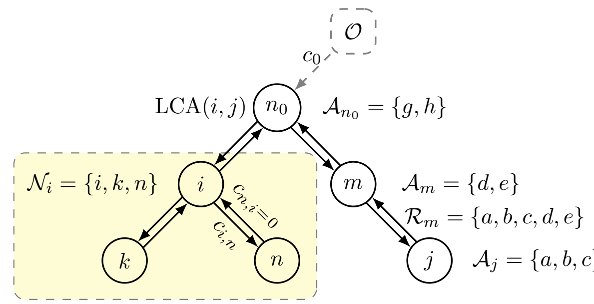

Fig. 1. Example hierarchical cache network with nodes in three levels, showing commonly used notation.

Every node $i \in \mathcal{N}$ is equipped with a cache, and we denote the size of the cache at node i by K_i. We denote the set of objects stored in the cache at node i by $\mathcal{A}_i \subset \mathcal{O}$, $|\mathcal{A}_i| \le K_i$. We use the shorthand notation $\mathcal{A}_V \triangleq (\mathcal{A}_j)_{j \in V}$, where $V \subseteq \mathcal{N}$, and $\mathcal{A}_{-i} \triangleq (\mathcal{A}_j)_{j \in \mathcal{N} \setminus \{i\}}$. We denote by $\mathfrak{A}_i$ the set of object placements that satisfy the storage capacity constraint at node i, i.e. $\mathfrak{A}_i = \{\mathcal{A}_i \in 2^{\mathcal{O}} : |\mathcal{A}_i| \le K_i\}$, where $2^{\mathcal{O}}$ is the powerset of $\mathcal{O}$. Finally, we denote the set of objects stored at node i and at its descendants by $\mathcal{R}_i(\mathcal{A}) = \mathcal{A}_i \bigcup_{j \in \mathcal{D}(i)} \mathcal{R}_j(\mathcal{A})$. Fig. 1 shows an example topology with a maximum level of 2, illustrating some of the commonly used notation.

2.2. Cost model

We denote the unit cost of using edge (i, j) by $c_{i,j}$. Since during peak hours most of the traffic in a mobile backhaul is flowing downlink (serving users' requests for content) [1,3], we consider that uplink edges have zero unit cost, i.e., $c_{i,\mathcal{P}(i)} = 0$. Without loss of generality, the cost of downlink edges is $c_{\mathcal{P}(i),i} > 0$. We consider that edge costs are additive, i.e., if a request for object o arrives at node i and is served from node j then the unit cost is $d_{i,j} = \sum_{(v,w) \in P_{j,i}} c_{v,w}$. We call $d_{i,j}$ the *distance* from node j to node i. Note that the terms $c_{v,w}$ are zero if they correspond to an uplink, i.e., if $w = \mathcal{P}(v)$. Furthermore, observe that in general $d_{j,i} \neq d_{i,j}$, thus distance is not symmetric (hence it is a hemimetric).

A request for object o generated by a mobile user connected to the cell site at node $i \in \mathcal{N}$ is served locally if $o \in \mathcal{A}_i$. Otherwise, if node i has a descendant $j \in \mathcal{D}(i)$ for which $o \in \mathcal{A}_j$, the node forwards the request to the closest such descendant. Otherwise, node i forwards the request to its parent $\mathcal{P}(i)$, which follows the same algorithm for serving the request. If an object o is not stored in any node (i.e., $o \notin \mathcal{R}_{n_0}$) then it needs to be retrieved through the Backbone via the root node n_0 at a unit cost of c_0.

Given a placement $\mathcal{A} = (\mathcal{A}_j)_{j \in \mathcal{N}}$ we can define the unit cost to serve a request for object o at node i as

$$d_i(o, \mathcal{A}) = \begin{cases} \min\limits_{\{j \in \mathcal{N} | o \in \mathcal{A}_j\}} d_{i,j} & \text{if } o \in \mathcal{R}_{n_0} \\ d_{i,n_0} + c_0 & \text{if } o \notin \mathcal{R}_{n_0}, \end{cases}$$

which together with the demand w_i^o determines the cost incurred by node i as

$$C_i(\mathcal{A}) = \sum_{o \in \mathcal{O}} C_i^o(\mathcal{A}) = \sum_{o \in \mathcal{O}} w_i^o d_i(o, \mathcal{A}). \tag{1}$$

Finally, we define the total cost $C(\mathcal{A}) = \sum_{i \in \mathcal{N}} C_i(\mathcal{A})$.

2.3. Problem formulation

Motivated by minimizing the congestion in the mobile backhaul during peak hours, our objective is to find a placement that mini-