Modeling the ICN on optimal energy consumption

Jie Duan, Xiong Wang, Sheng Wang, Shizhong Xu
Key Lab of Optical Fiber Sensing and Communications (Ministry of Education), School of Communication and Information Engineering (SCIE),
University of Electronic Science and Technology of China
No.2006, Xiyuan Ave, West Hi-Tech Zone, Chengdu, Sichuan, 611731
P.R. China
duanjie@uestc.edu.cn

Abstract: - Information Centric Network (ICN) architecture as a deemed potential solution of network has attracted more and more attentions. Both routing and caching are the fundamental components of ICN. This work studies on the energy efficiency in access network of ICN, which jointly optimizes the routing scheme and caching capacity from the perspective of ISP. The energy consumption model of content distribution for the ICN is formulated with the bandwidth limit. Then the optimization problem is divided into two sub-problems: 1) RSWC, determine the routing paths for objects without considering the cache deployed and 2) RSC, take the cache into consideration to get the final resolution. The corresponding meta-heuristic algorithms are proposed. Simulation results show that our algorithm can get a better result for energy consumption with time complexity $O(|Traf|^2)$. Meanwhile, the cache capacity and Zipf parameter are important factors affecting the total cost.

Key-Words: ICN (Information-Centric Network); energy consumption; routing scheme; cache placement

1 Introduction

Nowadays, applications and services become more and more focusing on the content itself, and does not matter where the source of information, which is conflict with the end-to-end communication models. Many clean-slate designs have been proposed to solve such mismatch. ICN as a most potential solution has attracted more and more attentions, in which information become the first civil replacing the status of end node.

Energy consumption is a hot topic to be discussed recently. With the increasing scale, Internet becomes one of the leading consumers in power for content access and delivery. About 10% worldwide energy estimated is expensed by Internet, and is constantly increasing [7]. It also a critical issue for programming ICN to consider the energy efficiency.

Many researchers devote to studying the ICN architecture. The three common components are a) named content objects b) routing and c) in-network caching [2, 3]. The delivery of the ICN [1] can be classified into 1) unstructured routing like IP routing, in which the control traffic overhead is huge, 2) structured routing, tree or distributed hash table (DHT) is used, 3) reverse path of the subscription propagation is another popular choice for ICN routing. However, the routing schemes mentioned are only concerns the strategy itself without considering the optimization. Multi-commodity transportation is used for model the routing of ICN to minimize the cost [4], in which the publisher and subscriber are modeled as the producer and consumer of commodity, while content objects is modeled as commodities. But the cache has not considered in [4]. Moreover, none of them have concerned the power consumption. Our work studies on the routing scheme for an ISP to minimize the energy consumption referring little on the design of cache location and cache capacity, when the request rate is known in a period.

Cache as another important characteristic of ICN has been studied by several works. Most of existing papers is about the cache policy, that is, how to design the cache [8-11]. Our work mainly focuses on the cache location and cache capacity combining the routing scheme from the perspective of power consumption, while Least Recently Used (LRU) cache scheme is used in our work.

So far little effort has been done on in-network caching for ICN from the viewpoint of energy efficiency. Nakjung et al. [7] and Jun Li et al. [6] have formulated optimization problems with the ICN energy consumption. A genetic algorithm is used to deploy cache in [7], while an online aging popularity based caching scheme (APC) is proposed in [6]. However, both of them optimize the cache policy in the case that routing scheme is decided in advance. Meanwhile, none of them have considered the bandwidth limitation, fix costs of cache router and link.

Our contribution in this paper is three folded. First, we formulate the energy consumption model for ICN content delivery with the bandwidth limitation, and find that it is a joint optimization of the routing scheme and cache placement. Second, we divide the problem into two sub-problems, and propose a meta-heuristic algorithm to deploy the content routing and cache capacity to effectively save energy. Third, our algorithm is simulated on variety of network topologies, and simulation results show that our scheme can get a better solution within a limited time.

2 Energy consumption model

Energy efficiency is a critical issue should be considered for deploying ICN, especially for the increasing scale of network. The energy consumption is mainly contributed by content caching and data transmission. In this section, we give the summarized notations firstly. And then energy consumption model for ICN is formulated.

2.1 Notations

Let N be the set of routers. From the perspective of ISP network, the content origins can be abstracted as one remote content server S outside the access network, which connects ISP network through many routers. Consider an ISP network topology is represented by directed graph $G = (Node, E)$. $Node$ is the set of total nodes, that is, $Node = N \cup S$. E is the set of edges, which are the links between neighboring routers. *arc(i,j)* is the link from node i to node j. $arc(i,j) \in E$. Request generators are responsible for collecting the requests from their users, and spreading them through connected routers. The router forwards the request to the network, and delivers the content object back to the request generators. If a request generator connects to the ISP network through router i, then the request generated by such generator can be regarded as generated by router i. Router i can be called as generating router. As shown in figure 1, the ISP network attaches content server through router e, f and g.

$N_{(i)}^{+}$: The set of outward neighbors of router i. If *arc(i,j)* exists, router j is the outward neighbor of router i ($j \in N_{(i)}^{+}$).

$N_{(i)}^{-}$: The set of inward neighbors of router i. If *arc(j,i)* exists, router j is the inward neighbor of router i ($j \in N_{(i)}^{-}$).

$f_{i,j}$: The fix cost of *arc(i,j)*. $arc(i,j) \in E$

f_i: The fix cost for router i to forward content object.

The fix cost of link or router refers to the energy consumption for powering on, which is the cost to maintain the equipment to work on.

$u_{i,j}$: The bandwidth limitation of *arc(i,j)*. $arc(i,j) \in E$

C_i: The capacity of memory technologies chosen for deploying on cache router i.

Given the assumed large number caches, it is reasonable to apply the independent reference model at each router since its request process results from the superposition of many independent overflow processes, each contributing a small fraction of overall demand. Moreover, the occupancy states of caches can reasonably be assumed to be statistically independent.

O is the set of content objects requested in ICN. $|O|$ is the number of objects. It has been observed that the popularity of objects follows a generalized *Zipf* law in many existing flows. The content objects are sorted by its popularity in descending order. For the n_{th} ($n \in [1, |O|]$) most popular content object, the possibility of being requested is

$$P(n) = \frac{1/n^{\alpha}}{A} \tag{1}$$

Where, $A = \sum_{i=1}^{|O|} 1/i^{\alpha}$. α is the *Zipf* parameter.

If the total request rate is q, the request rate for the n_{th} most popular content object is

$$q(n) = qP(n) \tag{2}$$

The Least Recently Used (LRU) cache replacement policy is a favorite cache scheme for its good hit ratio and low hardware complexity. The approximate value of hit ratio is proposed by Che et al. [12] with the assumption that all the content objects have the same size. The hit ratio for the n_{th} most popular object is $1 - e^{-q(n)T_c}$. T_c is a parameter determined by the request rate and cache capacity, which refers to the time an object maintained by the cache memory.

The cache capacity is the expectation of the hit ratio for all objects, that is,

$$C = \sum_{n=1}^{|O|} h(n) = \sum_{n=1}^{|O|} (1 - e^{-q(n)T_c}) \tag{3}$$

T_c can be gained by resolving the above equation.

The average hit ratio for any object with capacity C can be described as

$$hitratio = \sum_{n=1}^{|O|} P(n) \times (1 - e^{-P(n) \times q \times T_c}) \tag{4}$$

S_k: The size of object O_k ($O_k \in O$).An object can be divided into many chunks in ICN for routed by different paths. For convenience of description, we assumed the object O_k is indivisible, that is, the whole size of object S_k is delivered by the same path. It is assumed all the content objects have the same size, i.e, S_k=S_o for all $O_k \in O$.

$q_{i,j}^{k}$: The request rate of object O_k ($O_k \in O$) from router i to router j ($arc(i,j) \in E$).

$s_{i,j}^{k}$: The flow of object O_k ($O_k \in O$) on *arc(i,j)* ($arc(i,j) \in E$). Clearly, $s_{i,j}^{k} = q_{j,i}^{k} \times S_k$.

q_i^k : The request rate of object O_k ($O_k \in O$) generated at router i.

q_i: the total request rate generated on router i. $q_i = \sum_{O_k \in O} q_i^k$

P_r: Energy density of a router (J/bit).

P_l: Energy density of a link (J/bit).

P_{ca}: Power density of caching in content router (W/bit).

$y_{i,j}$: Boolean design variable. Equals 1 if *arc(i,j)* ($arc(i,j) \in E$) is used, 0 otherwise.

y_i: Boolean design variable. Equals 1 if router i is used to forward content object, 0 otherwise.

y_i^{ca} : Boolean design variable. Equals 1 if router *i* is used as cache router, 0 otherwise.

Obviously, $\exists\ j \in N_{(i)}^{+}$, $y_{i,j} \neq 0$ or $\exists\ j \in N_{(i)}^{-}$, $y_{j,i} \neq 0$, then y_i=0. Only the router used should be chosen as cache router, that is, $y_i^{ca} = y_i$.

H_d^k : The hops traversed for request generator (which connects to router *d*) to get the content O_k from content server. In other words, the number of routers on the path from request generator to content O_k 's server is $H_d^k - 1$. $H_d^k \geq 1$

$hr_i(k)$: The hit ratio for object O_k on router *i*.

hitratio$_i$: The average hit ratio of cache router *i* with the cache capacity C_i. It is reasonable assumed that all cache routers have the same hit ratio for a given content with the same cache capacity.

t: Investigated interval time

2.2 Energy consumption model

The energy consumption (*cost*) is composed by the fix cost (*fixcost*) and variable cost (*variablecost*). The fix cost is the energy consumed by powering on the equipment. The variable cost depends on the traffic flow dealt by the routers or traversed by links, which is similar as the variable cost in multi-commodity transportation problem. Because the size of request packet is too trivial to be ignored comparing to the data packet, the energy consumption model is only formulated for data packets.

The fix cost given in (5) is the fix energy consumption within a given interval *t*, which is composed by energy consumed for links, routers and caches. $f_{i,j} \times y_{i,j}$ is the fix cost on *arc(i,j)*. $f_i \times y_i$ is the energy cost for using router *i*. $P_{ca} \times C_i \times \sum_i y_i^{ca}$ is the total energy cost for cache routers.

$$fixcost = t \times (\sum_{i \in N} \sum_{j \in N_{(i)}^{+}} f_{i,j} \times y_{i,j} + \sum_{i \in N} f_i \times y_i + P_{ca} \times C_i \times \sum_{i \in N} y_i^{ca}) \tag{5}$$

$$variablecost = p_r \times \sum_{arc(i,j) \in E} \sum_{O_k \in O} s_{i,j}^k + p_l \times \sum_{i \in N} \sum_{O_k \in O} (q_i^k \times S_o \times t + \sum_{j \in N_{(i)}^{+}} s_{i,j}^k) \tag{6}$$

The variable cost consists of energy consumptions of links and routers. The former part of the equation (6) is the energy consumed on links, while the later is for cache routers. The data dealt on router *i* should contain the data requested by its connecting request generator and transformed to neighboring outward routers.

If all the cache routers are assumed deploying with the same cache capacity *C*, the hit ratio for object O_k is labeled as *hr(k)*. The variable cost can be described as expression (7). Because we only care about the energy cost in ISP access network, the final hop from router to request generator should be ignored, which is “-1” in the expression.

$$\begin{aligned} variablecost\_C = {} & t \times p_l \times S_o \times \sum_{i \in N} \sum_{O_k \in O} q_i^k \times (Hops_i^k - 1) \\ & + t \times p_r \times S_o \times \sum_{i \in N} \sum_{O_k \in O} q_i^k \times R\_num_i^k \end{aligned} \tag{7}$$

$$\begin{aligned} R\_num_i^k = {} & \sum_{h=1}^{H_i^k - 1} h \times hr(k) \times (1 - hr(k))^{h-1} \\ & + (H_i^k - 1) \times [1 - \sum_{h=1}^{H_i^k} hr(k) \times (1 - hr(k))^{h-1}] \end{aligned} \tag{8}$$

$$\begin{aligned} Hops_d^k = {} & \sum_{h=1}^{H_i^k - 1} h \times hr(k) \times (1 - hr(k))^{h-1} \\ & + H_i^k \times [1 - \sum_{h=1}^{H_i^k} hr(k) \times (1 - hr(k))^{h-1}] \end{aligned} \tag{9}$$

The expectation number of hops traversed by generating router *i* to get an object O_k is given by equation (8), while the expectation number of routers bypassed for generating router *i* to get object O_k is shown in (9). $hr(k) \times (1 - hr(k))^{h-1}$ is the hit ratio for object O_k at *h* hops away from request generator, which missed at the closer routers to request generator. The objects can always be hit in content server, so the final hit ratio for content O_k in content server is $1 - \sum_{h=1}^{H_i^k} hr(k) \times (1 - hr(k))^{h-1}$, which is the possibility for request O_k arriving at content server. The discrimination between (8) and (9) is the parameters before $1 - \sum_{h=1}^{H_i^k} hr(k) \times (1 - hr(k))^{h-1}$. If request arrives at content server, the number of routers bypassed is $H_i^k - 1$, but the hop number is H_i^k .

Our objective is to minimize the total energy consumptions (*cost*). Then the optimal energy consumption for ICN can be formulated as (10). Constraint 1) in (10) is the bandwidth limitation, which ensures the total flow on a link does not exceed the capacity. Constraint 2) is the flow conservation constraint. Because only the missing requests should be send upward. For an object, the ratio of inward traffic to outward traffic is the miss ratio of a cache router. Constraint 2) is a non-linear equation. The $hr_i(k)$ achieved through resolving the exponential equation (3) under the knowledge of cache capacity and arriving request rate which is the variable programmed. Constraint 3) enables the non-negativity traffic on links. Constraints 4)-6) are variable constraints. If a neighboring link of a router is used, the router is chosen, given in constraint 4). Constrait 5) ensures that only the router powered on should be deployed with cache. constrait 6) represents whether the link is chosen.

$$
\begin{aligned}
&\textbf{Minimize}\quad cost = fixcost + variablecost\\
&\textbf{Subject to: } 1)\ \sum_{O_k\in O} s^k_{i,j} \le u_{i,j}\times y_{i,j} \qquad \forall arc(i,j)\in E\\
&2)\ \sum_{j\in N^-_{(i)}} s^k_{i,j} - \Big(\sum_{j\in N^+_{(i)}} s^k_{j,i} + t\times q^k_i \times S_o\Big)\times(1-hr_i(\\
&\qquad\qquad\qquad \forall i\in N, \forall O_k \in\\
&3) s^k_{i,j}\ge 0,\ \forall arc(i,j)\in E, \forall O_k\in O\\
&4) y_i = \begin{cases} 1, & \exists j\in N^+_{(i)}\cup N^-_{(i)}, y_{i,j}=1\\ 0, & else\end{cases},\quad \forall i\in\\
&5) y^{ca}_i = y_i, \forall i\in N\\
&6) y_{i,j}\in\{0,1\}, \forall arc(i,j)\in E
\end{aligned}
\tag{10}
$$

Theorem: The problem of the optimal energy consumption for ICN is NP-hard;

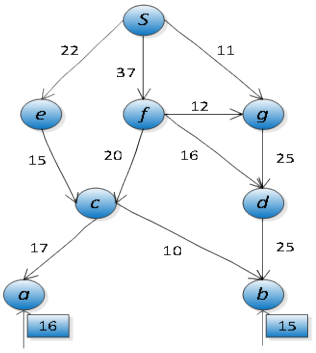
Proof: Without considering the fix costs of routers, cache scheme, and the inseparability of object, the problem can be formulated as multi-commodity transportation problem, which have been proven to be NP-hard [4]. Moreover, the constraint 2) is non-linear. So the problem shown in (9) is also an NP-hard problem.

3 Heuristic algorithm design

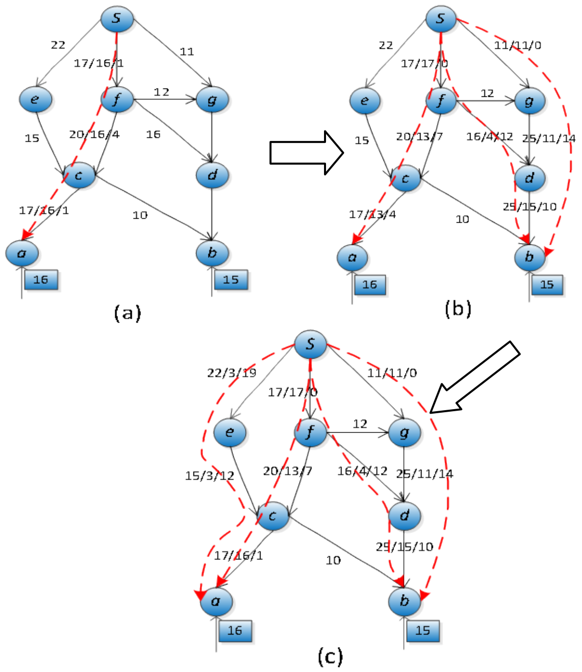
As the optimal energy consumption of ICN routing scheme is NP-hard, we propose a heuristic algorithm for efficiently solving the problem of an access network under condition that the request traffic for a near future is priori-known.

It is too complicated to compute the bandwidth requirement when considering the cache function for each possible routing path for a given traffic. The problem described in (10) can be done in the following two sub-problems: 1) RSWC: determine the Routing Scheme Without considering the Cache deployed; 2) RSC (Routing Scheme with Cache): take the cache into consideration to get the final resolution. If the bandwidth constraint can be satisfied by a solution without cache, the solution can be feasible with considering cache, because the cache may decrease the traffic upwards. The sub-problem 1) can be thought as getting a feasible solution, while sub-problem 2) is to used to get a near optimal solution with cache.

Figure 1. Problem illustration (**A**) Simple example of an ISP network. (**B**) RSWC. (**C**) RSC



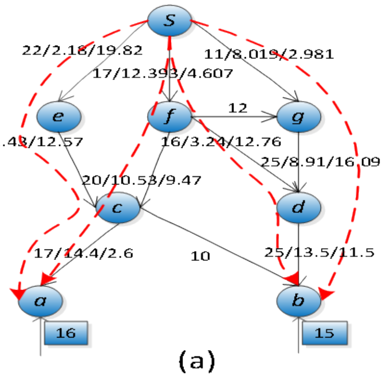
(A) simple example of an ISP network

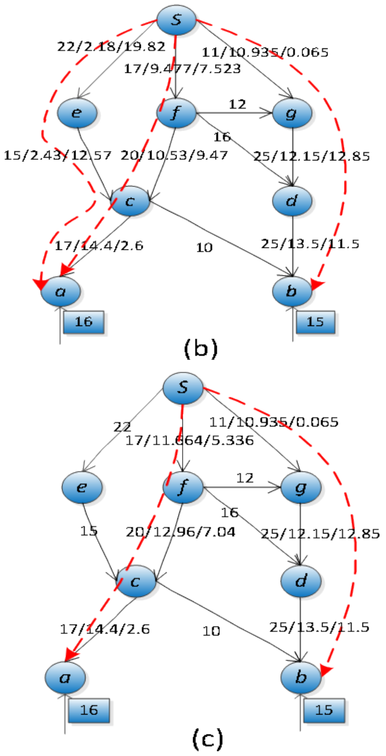


(B) RSWC

3.1. Algorithm for RSWC

The sub-problem 1) is used to get a feasible solution of the whole problem, which can be modeled as an integer linear programming (ILP) problem.





(b)

(c)

(C) RSC

The fix cost (11) is only composed by powering on the chosen routers and links, without considering the cost of deploying cache. Similarly, the variable cost (12) is comprised by the energy consumption of flows traversed on links and routers. The RSWC is formulated as (13). The object is also to minimize the total energy consumption. The constraint 1), 3), 4) and 5) in (13) is the same as the constraint in (10). However, the constraint 2) in (13) is different from the flow conservation in (10). If *i* is the generating router, the difference between outward flow and inward flow of router *i* is the traffic generated. Otherwise, the inward traffic equals the outward traffic of a router.

$$fixcost^{*}=t\times(\sum_{i\in N}\sum_{j\in N^{+}_{(i)}} f_{i,j}\times y_{i,j}+\sum_{i\in N} f_{i}\times y_{i}) \qquad (11)$$

$$fixcost^{*}=t\times(\sum_{i\in N}\sum_{j\in N^{+}_{(i)}} f_{i,j}\times y_{i,j}+\sum_{i\in N} f_{i}\times y_{i}) \qquad (12)$$

$$\textbf{Minimize} \quad cost^{*}=fixcost^{*}+variablecost^{*}$$

$$\textbf{Subject to: } 1)\ \sum_{O_k\in O} s^{k}_{i,j}\le u_{i,j}\times y_{i,j} \qquad \forall arc(i,j)\in E$$

$$2)\ \sum_{j\in N^{-}_{(i)}} s^{k}_{i,j}-\sum_{j\in N^{+}_{(i)}} s^{k}_{j,i}=\begin{cases}-t\times q^{k}_{i}\times S_{o},\ i \text{ is generating router}\\ t\times q^{k}_{i}\times S_{o}, \quad i=S\\ 0, \qquad else\end{cases},$$

$$\forall i\in Node, \forall O_k\in O \qquad (13)$$

$$3) s^{k}_{i,j}\ge 0, \forall(i,j)\in N, \forall O_k\in O$$

$$4) y_{i}=\begin{cases}1, & \exists j\in N^{+}_{(i)}\cup N^{-}_{(i)}, y_{i,j}=1\\ 0, & else\end{cases}, \quad \forall i\in N$$

$$5) y_{i,j}\in\{0,1\}, \forall(i,j)\in N$$

Without considering the fix cost of router and the inseparability of object, the sub-problem can be seen as a capacitated multi-commodity min-cost flow problem (CMMCF). A meta-heuristic algorithm is proposed to resolve the sub-problem RSWC in (13) described in figure 2. The requested traffic generated by router *r* within an interval *t* is always $S_o\times q_r$. *Traf* is the set of data rates generated by requested routers. *traf*$_i$ is the i_{th} largest request rate in *Traf*. *PATH*$_i$ is the set of optional paths for *traf*$_i$. The available bandwidth of a path is defined as the minimal available bandwidth of links on path. If multiple paths share one link, then the sum of available bandwidth for multiple paths could not exceed the available bandwidth of the common link. If a link *arc(m,n)* with available bandwidth $B_{m,n}$ is shared by *k* paths in *PATH*$_i$, the available link bandwidth of *arc(m,n)* on each path can be simply computed as $B_{m,n}/k$. $relBW_j$ is the released bandwidth of path *j* for *traff*$_i$, $j\in PATH_i$. *released_traffic* is the set of traffic released to be rearranged. $re\_traf^{i}_{k}$ is the sub-traffic of *traf*$_k$ affected by *traf*$_i$, $traf_i\in Traf$, $traf_k\in Traf$, i>k. un_traf_i is the sub-traffic of *traf*$_i$ unarranged, $traf_i\in Traf$.

The requested traffics are arranged in a descending order by the request rate on the topology. For the traffic *traf*$_i$, if the available bandwidth of optional paths can accommodate the *traf*$_i$, successively choose the path with smallest $cost^{*}$ to locate $traf_i$, then renew the *path_arranged*, and compute the available bandwidth for links, until the traffic *traf*$_i$ is arranged completely, step 4-6. Otherwise, if the available bandwidth of optional paths can't accommodate the *traf*$_i$, determine whether the *traf*$_i$ itself can't be located in the network, if so, the sub-traffic unarranged should be $traf_i-\sum_{j\in PATH_i} BW_j$, and the traffic can be arranged equals $\sum_{j\in PATH_i} BW_j$, step 8-11. Then release the bandwidth occupied by other sub-traffics of larger traffics to locate *traf*$_i$ step 12-18. The released paths should affect minimal number of traffics. The traffic that could not be located is *un_traf*$_i$ + *traf*$_i$- $\sum_{j\in PATH_i} relBW_j$. Finally, rearrange the located traffic, step 17-18. The result of unlocated

Figure 2. Pseudo code of RPWC algorithm

```
Algorithm for RPWC
1. Traf←requested_traffic
2. sort the requested traffics of Traf in a descending order
3.  for $traf_i\in Traf$ , do
4.    if $\sum_{j\in PATH_i} avaBW_j\ge traf_i$ , then
5.      choose the smallest cost paths to locate $traf_i$
6.      renew path_arranged and available bandwidths of links
7.    else
8.      if $\sum_{j\in PATH_i} BW_j < traf_i$ then
9.        un_traf_i = $traf_i-\sum_{j\in PATH_i} BW_j$
10.       traf_i= $\sum_{j\in PATH_i} BW_j$
```

```
11.     end if
12.         release the bandwidth of $path_j$ $relBW_j$ with minimal number of affected traffics $re\_traf_k^i$ ( $k>i$) to locate $traf_i$.
13.     $un\_traf_i = un\_traf_i + traf_i - \sum_{j\in PATH_i} relBW_j$
14.     renew path_arranged and available bandwidths of links
15.     $released\_traffic \leftarrow re\_traf_k^i$
16.     sort the released_traffic by $k$ in an ascending order
17.     for $traf_t \in released\_traffic$, do
18.         go to step 12
19. Return un_located_traffic, path_arranged
```

traffic and path arranged is returned. The time complexity of the algorithm is $O(|Traf|^2)$.

Figure 1A) presents a simple undirected example of ISP access network, where the numbers in rectangles next to the arrows represent the amount of data rate received from request generators, and the numbers next to the edges represent the link bandwidth. The algorithm of RPWC is shown in figure 2(B). The number $n_1/n_2/n_3$ next to the link in chosen path refers to the bandwidth constraint, bandwidth occupied and available bandwidth, $n_3=n_1-n_2$. We first program the path of traffic generated by router *a*, 16, which is larger than the traffic of router *b*, 15. The fix costs on routers and links are assumed equality, that is, $f_{i,j}=f_{\mathrm{m,n}}\ \forall arc(i,j)\neq arc(m,n)$, $f_i=f_j\ \forall(i\neq j)$. The minimal cost paths for $traf_a$ is *S-f-c-a*, which is shown in (a) of figure 1 (B). When manage $traf_b$, because the total available bandwidth 11 is smaller than $traf_b$ 15, we first locate the traffic on available paths. The minima cost paths *S-g-d-b* and *S-f-d-b* are chosen with the sub-traffic is 10 and 1, respectively. To locate the remaining part of $traf_b$ (15-11=4), a part of bandwidth occupied by $traf_a$ should be released. So the sub-traffic on path *S-f-c-a* is 12. The traffic on path *S-f-d-b* is tuned to be 5, as shown in (b) of figure 1 (B). So far, the $traf_b$ is arranged completely with relocating sub-traffic of $traf_a$ 4. The path *S-e-c-a* is used to route the remaining sub-traffic. The final solution is given in (c) of figure 1 (B). $traf_a$ is transformed through paths *S-f-c-a* and *S-e-c-a*, while $traf_b$ is routed through path *S-g-d-b* and *S-f-d-b*.

The algorithm of RPWC is used to meet the bandwidth constraint as much as possible. Because of the arrangement in order, the choice of minimal cost paths, and the rearrangement stage, the feasible solution can be better.

3.2. Algorithm for RSC

The main idea of RPWC algorithm is arranging the data traffic step by step, while the RPC algorithm is adjusting the link bandwidth occupied successively.

The cache deployed can decrease the data rate, i.e, bandwidth requirement. According to expression (4), because $q\times T_C$ can be seen as a variable parameter only affected by cache capacity and *Zipf* parameter. Given the cache capacity C_i and *Zipf* parameter (α) of objects distribution, the average hit ratio ($hitratio_i$) on router *i* can be gained. Accordingly, if the total data rate for an uplink *arc(j,i)* of router *i* is programmed without cache, $\sum_{O_k\in O} q_{i,j}^k$, the bandwidth occupied for *arc(j,i)* under considering cache is

$$BW_{j,i}=S_o\times\sum_{O_k\in O} q_{i,j}^k\times(1-hitratio_i) \qquad (14)$$

The RPC algorithm is given in figure 3. After considering the cache on each router, the bandwidth occupied should be recomputed for each link based on (14), which may release some of the bandwidth for locating the flow in the un_located_traffic, step 1. The flow unallocated is arranged based on the RSWC algorithm, step 2-7, until no traffic is remained or the unallocated traffic should not be arranged. If any traffic in un_located_traffic can be allocated, the bandwidth occupied is recomputed again for the newly arranged traffic. The cost is computed, and the path_arranged is renewed. Then merge the paths of traffic in ascending order, step 8-11. The sub-traffic of data rate generated by router *m* on a path can be replaced by other paths from *m* to *S*, only when the bandwidth is

Figure 3. Pseudo code of RSC algorithm

```
Algorithm for RPC
1:  re-compute the bandwidth occupied, compute the cost
2:  if un_located_traffic is not null
3:      locate the un_located_traffic based on RSWC algorithm
4:      if un_located_traffic changed, then
5:          re-compute the bandwidth occupied.
6:          compute the cost, renew the path_arranged
7:          go to 2
8:  sort the data rate in ascending order → Traf
9:  for $traf_i \in Traf$
10:     if the sub-traffic on a path can be replaced by other paths
11:         renew the path_arranged, compute cost
12: return un_located_traffic, path_arranged, cost
```

occupied by the sub-traffic can be relocated on the other paths, and the cost is reduced. If *arc(i,j)* is replaced by paths, the path_arranged is renewed, and the cost is computed. Finally, the un_located_traffic, path_arranged, cost are returned.

An example of RSC algorithm is shown in figure 1 (C). The average hit ratio of each router is assumed to be 10%. The energy consumed by cache routers deploying memory technology is the same. The bandwidth re-computed is given in (a), which is driven from the results of RSWC, (c) in figure 1(B). Then the sub-traffic of $traf_b$ on the path *S-f-d-b* can be relocated on the path *S-g-d-b*, as shown in (b) of figure 1(C). The sub-traffic of $traf_a$ on the path *S-e-c-a* can be relocated on the path *S-f-c-a*. The final solution is given in (c) of figure 1(C).

4. Performance Evaluation

In this section, we simulate the performance of our routing scheme described in section 3 vary the cache capacity and *Zipf* parameter.

4.1. simulation settings

The simulation is carried out in a variety of network topologies varying in size and bandwidth limit. Because of the space limitation, we only present the simulation results over AS1755 Ebone (Europe) [13]. It includes 18 routers, among which there are 2 gateways with max bandwidth connecting to server *S*. The available bandwidth of a link between gateway and server are large enough to transmit the data. The number on the link refers to the bandwidth comparing to the content size. The total number of object items is 5000. The content size is assumed to be 100*M*. The total size for content objects is about 500*G*. The total request rate generated on whole network is assumed to be 2500 content/s. The requests can be generated by any routers except for gateways. The request generator are randomly chosen, and the request rate is smaller than the bandwidth outwards of the request generator.

Table 1. Simulation Parameter

parameter	symbol	value
Content number	O	5000
Cache capacity percentage	β	$10^{-3}-10^{-2}$
Request rate (content/s)	q	2500
Zipf parameter	α	0.6-2
Energy density of a router (*J/bit* ）	P_r	2×10^{-8}
Energy density of a link(*J/bit* ）	P_l	0.15×10^{-8}
Power density of caching (*W/bit)*	P_{ca}	2.5×10^{-9}
Fix cost of a router(*W*)	f_i	709
Fix cost of a link(*mW*)	$f_{i,j}$	130
Content size *(M)*	S_o	100
Interval time (min)	t	10-60

The device’s parameters are referred to [5]. Energy density of a router is $P_r = 2\times 10^{-8}$*J/bit*. Energy density of a link is $P_l = 0.15\times 10^{-8}$*J/bit.* Power density of caching in content router is $P_{ca} = 2.5\times10^{-9}$*W/bit*. Caching capacity is usually much smaller than the overall number of content objects. β is the percentage of the caching capacity to the total size of requesting contents, which is varying between 10^{-3} and 10^{-2} . Because the power consumption of fiber TOSA is about 130 *mW*, we assume $f_{i,j}$ is 130 *mW*. If a router SR8808-X deploys two control board SR05SRP1H1, two service board SPC-XP4LB, two switching board SFC-08B and a SR8808-X fan, the maximum energy consumption is about 709W, that is, f_i=709*W*. Many flows have proven that the Zipf parameter is between 0.6 and 2. The simulation parameters are summarized in table 1. The investigated interval time is supposed to vary between 10 minutes to 1 hour.

4.2. Simulation Results

The simulations are repeated several times. And the proposed algorithm is finished just in few seconds. Figure 4 is the average hit ratio for a router with different cache capacities when varying the Zipf parameter. Obviously, the average hit ratio has an increasing trend with a gradual increase n Zipf parameter, if the cache capacity is determined. Moreover, the increase in cache capacity percentage leads to an increase in average hit ratio. When the Zipf parameter is 2 and cache capacity percentage is 1%, the average hit ratio is nearly 100%. However, the average hit ratio is close to 0, when the Zipf parameter is 0.6 and the cache capacity percentage is smaller than 1%.

Figure 4. Average hit ratio

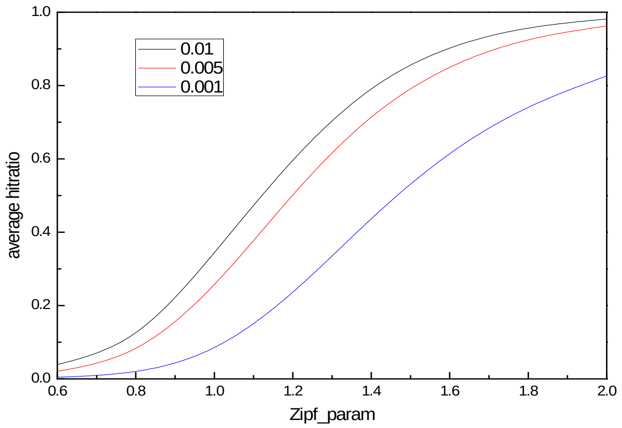
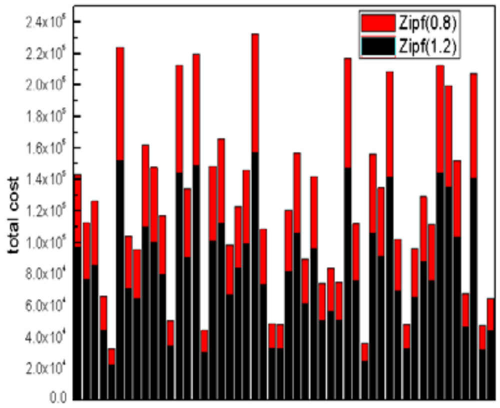


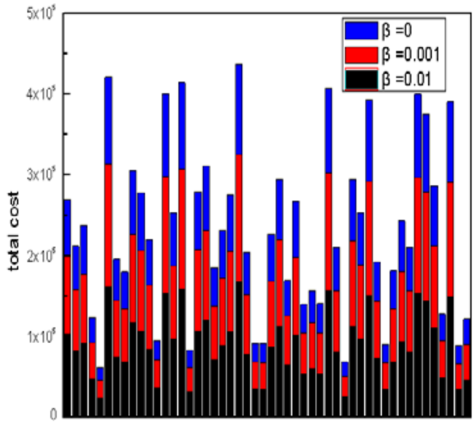
Figure 5 is the total energy consumed per second for several simulation times. Figure 5 A) is total cost for different Zipf parameters when the cache capacity percent is 0.001. The total cost of Zipf 0.8 is always more than that of Zipf 1.2. Because the average hit ratio of Zipf 1.2 is larger than Zipf 0.8, and the upward interest is smaller. The energy consumed for different cache capacities is shown in figure 5 B). Clearly, the hit ratio is increasing with the increase of cache capacity. So the total cost decreases with the increase of cache capacity percentage.

The meta-heuristic algorithm presented before is to ensure the bandwidth constraint firstly. The remained rate is composed by the following reasons: 1) the request rate excesses the outward bandwidth of generation router; 2) request rate remained caused by the arrangement algorithm. The former part is unresolved, while the other one is our concerns, which is shown in figure 6. No matter of the cache capacity percentage, the rate remained is always smaller than 0.04%. And the rate remained decreases with the increasing of cache capacity.

Figure 5. Total energy consumed (**A**) Total cost for different Zipf parameter (**B**) Total cost for different cache capacity

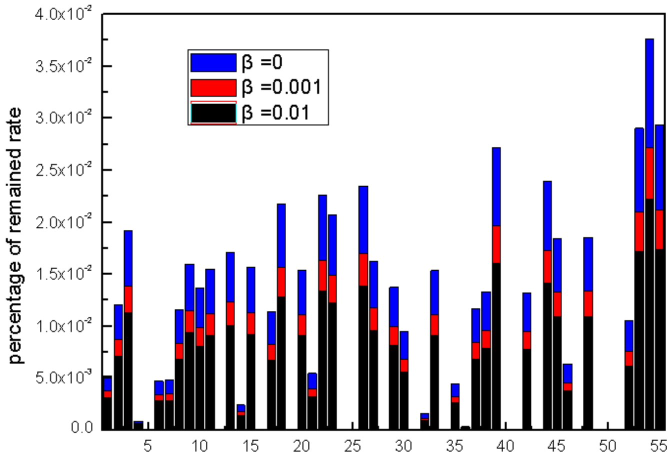


(A) Total cost for different Zipf parameter



(B) Total cost for different cache capacity

Figure 6. Rate remained



5 Conclusion

Energy consumption is a hot topic to be discussed for ICN with the increasing scale of Internet. Our model the routing scheme of ICN with the cache scheme firstly with the knowledge of request rate within an interval time. And then the problem is divided into two sub- problems. A meta-heuristic algorithm is presented to solve the problem. The simulation result shows the Zipf parameter and cache capacity are the important elements to affect the total cost. Our algorithm can arrange the requested traffic to obtain a better energy saving while meets the bandwidth constraint with little rate remained.

Acknowledgments

This work is partially supported by 973 Program (2013CB329103), NSFC Fund (61271165, 61301153), Program for Changjiang Scholars and Innovative Research Team (PCSIRT) in University and the 111 Project B14039.

References:

[1] Jaeyoung Choi; Jinyoung Han; Eunsang Cho; et al. A Survey on Content-Oriented Networking for Efficient Content Delivery, *IEEE communication magazine*, March 2011, 49, pp. 121-127.

[2] Ali Ghodsi; Teemu Koponen; Barath Raghavan; et al. Information-centric Networking: Seeing the forest for the trees, Proceedings of ACM Workshop on Hot Topics in Networks (HotNets-X), Publisher: ACM, Cambridge, MA, November 2011.

[3] Bengt Ahlgren; Christian Dannewitz; Claudio Imbrenda; et al. A Survey of Information-Centric Networking, *IEEE Communications Magazine*, 2012, 50(7), pp 26-36.

[4] Ambily Pankajakshan; Shrisha Rao. Modeling A Publish/Subscribe System As A Multi-Commodity Transportation Problem, Proceedings of 6th Annual IEEE International Systems Conference (IEEE SysCon 2012). Publisher: IEEE, Vancouver, Canada, March 2012.

[5] Jun Li; Bin Liu; Hao Wu. Energy-Efficient In-Network Caching for Content-Centric Networking, *IEEE COMMUNICATIONS LETTERS*, April 2013, 17(4), pp.797-800.

[6] Nakjung Choi; Kyle Guan; Daniel C. Kilper; and Gary Atkinson. In-Network Caching Effect on Optimal Energy Consumption in Content-Centric Networking. Proceedings of IEEE ICC'12 Next Generation Network Symposium. Publisher: IEEE, Jakarta, Indonesia, June 2012.

[7] L. Chiaraviglio; M. Mellia; F. and Neri. Minimizing ISP network energy cost: formulation and solutions, *IEEE/ACM Transaction on Networking*, April 2012**,** 20(2), pp. 463 – 476.

[8] Giovanna Carofiglio; Vinicius Gehlen; and Diego Perino. Experimental Evaluation of Memory Management in Content-Centric Networking. Proceedings of 2011 IEEE International Conference on Communications (ICC). Publisher: IEEE, Kyoto, Japan, 2011.

[9] Haiyong Xie; Gangyu Shi, Pengwei Wang. TECC: Towards Collaborative In- network Caching Guided by Traffic Engineering. Proceedings of the 31st Annual IEEE International Conference on Computer Communications: Mini-Conference. Publisher: IEEE, Orlando, Florida USA, 2012.

[10] Ioannis Psaras, Wei Koong Chai, George Pavlou. Probabilistic In-Network Caching for Information-Centric Networks. Proceedings of the second edition of the ICN workshop on Information-centric networking. Publisher: IEEE, New York, NY, USA, 2012.

[11] Jason Min Wang, Brahim Bensaou. Progressive Caching in CCN. Proceedings of 2012 IEEE Global Communications Conference (GLOBECOM). Publisher: IEEE, Anaheim, CA, 2012.

[12] Hao Che, Ye Tung, and Zhijun Wang. Hierarchical web caching systems: modeling, design and experimental results. *IEEE Journal on Selected Areas in Communications* (*IEEE JSAC)*, 2002**,** 20(7), pp 1305–1314.

[13] Neil Spring Ratul Mahajan David Wetherall. Measuring ISP topologies with rocketfuel. Proceedings of the 2002 SIGCOMM conference on Applications, technologies, architectures, and protocols for computer communications. Publisher: ACM, New York, USA, 2002.