Special Issue on Internet of Vehicle

Scheduling algorithms for time-constrained big-file transfers in the Internet of Vehicles

Chuan Lin¹, Yuanguo Bi^{1*}, Hai Zhao¹, Zeshen Wang², Jinfa Wang¹

1. School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China

2. School of Business Administration, Northeastern University, Shenyang 110819, China

* Corresponding author, Email: biyanguo@ise.neu.edu.cn

Abstract: In order to transfer large files and provide high-quality services in the IoV (Internet of Vehicles), intelligent routing and scheduling are indispensable for fast transfers and efficient network utilization, particularly when multi-path routing is allowed in the wired-transfer. Thus, a network administrator must select a set of feasible paths over which the transfer can be conducted. We consider a TBTS (Time-constrained Big-file Transfer Scheduling) problem in this paper. We prove that TBTS problem is NP-hard and that the TBTS problem can be solved by addressing a corresponding maximum flow over time problem with multi-path routing technique. We then propose both a heuristic algorithm (TBTS-H) and an exact algorithm (TBTS-A) to solve the TBTS problem. Although both of the proposed approaches can solve the TBTS problem, the heuristic runs more efficiently by trading accuracy for delay, while the exact algorithm can achieve high accuracy for delay, at the cost of increased running-time. The corresponding simulation results illustrate this trade-off. Additionally, we conduct some comparisons between our proposed approaches and a traditional single-path routing scheme.

Keywords: Internet of Vehicles, wired-transfer, time-constrained, big-file, scheduling, multi-path routing

Citation: C. Lin, Y. G. Bi, H. Zhao, et al. Scheduling algorithms for time-constrained big-file transfers in the Internet of Vehicles [J]. Journal of communications and information networks, 2017, 2(2): 126-135.

1 Introduction

People are spending more time in vehicles than ever before. There is also a rapidly increasing number of users in vehicles accessing the Internet through IP-enabled smart devices. They are eager to enjoy high quality services through the IoV^[1,2], such as high-definition video streaming, data uploading and downloading, and driving data recording. Extremely large data files, frequently called “big-file”, are generated by applications in IoV. Reliable and flexible file transfer is critical for supporting these applications. Additionally, some applications may require files to be delivered within a short time

constraint. For example, timely delivery of high-definition video streams to users is critical for achieving high QoS (Quality of Service)^[3,4]. As shown in Fig. 1, file transfer in IoV includes vehicle-to-infrastructure transfer, vehicle-to-vehicle transfer, i.e., wireless-transfer and wired-transfer over the Internet. Wired-transfer in IoV is more important than wireless-transfer because bandwidth is more easily exhausted as users continue to frequently request data from remote servers over the Internet. In order to maintain high QoS during wired-transfer, a network administrator is required to compute feasible potential paths based on network topology, allocate link bandwidth for each application, and sched-

ule transfers efficiently in order to satisfy time constraints. However, real-time file transfer is impossible in traditional IP networks because centralized administration is not available for existing IP networks.

Figure 1 An example scenario of IoV

In recent years, many types of HPNs (High-Performance Networks) have emerged, which make efficient big-file transfer possible and allow intelligent TE (Traffic Engineering) services, such as SURFnet¹⁾, USN (UltraScience Net)^[5], the OSCARS (On-demand Secure Circuits and Advance Reservation System) of ESnet²⁾, and the ION of Internet2³⁾. Additionally, emergent SDN (Software-Defined Networking) technologies can coordinate the facilities in HPNs through a control plane, thus enabling intelligent scheduling and high QoS data transfer.

In order to achieve efficient big-file transfer through wired-transfer in IoV, the concept of using multi-path routing between a source node and destination node has attracted significant attention in recent years. Compared to single-path routing, multi-path routing is more efficient in terms of network utilization, congestion reduction and robustness control. However, multi-path routing introduces extra overhead to both the control plane and

data plane in a network. The complexity of multi-path routing depends on the type and the number of QoS constraints, and many multi-path routing problems have been proven to be NP-complete. Furthermore, multi-path routing with a single constraint is an NP-hard problem even when only two-path routing is considered^[6,7]. In order to achieve higher network utilization, big-file transfer based on multi-path routing has attracted significant attention, particularly for resource-limited networks^[8,9]. Multi-path file transfer problems encompass both the single-file multi-path transfer problem^[10-12] and the multi-file multi-path transfer problem^[13-16]. However, most of previous works do not consider the size of files and how to search for a path set in a given network.

In this paper, we address the wired-transfer problem in IoV and propose a TBTS problem for considering transfer over multiple end-to-end paths. We also prove that the TBTS problem is NP-hard. For addressing the TBTS problem, we relate the TBTS problem to a corresponding maximum flow over time problem, and propose two approaches to solve it: a heuristic algorithm and an exact algorithm.

The remainder of this paper is organized as follows. Our network model and definitions are presented in section 2. In section 3, we introduce the TBTS problem. Section 4 presents our heuristic approach for solving the TBTS problem. The exact algorithm is presented in section 5. The corresponding simulations are shown in section 6. Section 7 concludes the paper and discusses possible future research.

2 Network model and definitions

A network can be modeled as a weighted directed graph $G(V, E)$, where V is the set of network nodes and E is the set of network links. A link $e \in E$ has bandwidth $b(e) > 0$ and delay $d(e) \geq 0$. Let s be a source node, t be a destination node, and $p = v_1, v_2, \dots, v_l$ be a path, where $e = (v_{i-1}, v_i) \in E$

¹⁾ <http://www.surf.nl/en/about-surf/subsidiaries/surfnet>.

²⁾ <http://www.es.net/oscars>.

³⁾ <http://www.internet2.edu/ion>.

represents a link connecting nodes v_{i-1} and v_i . The bandwidth of path p is

$$b(p) = \min_{e \in p} b(e). \quad (1)$$

Normally, the delay in a computer network depends on Eq. (1) “path-delay”, which consists of propagation, queuing and processing delays accumulated along the path, and Eq. (2) the delay inherent in the available bandwidth (lower bandwidth results in longer transfer delay). The path-delay of a path p is

$$d(p) = \sum_{e \in p} d(e). \quad (2)$$

Given a set $\mathcal{P}$ of k $s-t$ paths, we wish to distribute a file of size σ units over (some of) the paths in $\mathcal{P}$. If $p_i \in \mathcal{P}$ is chosen, it will utilize a portion of its bandwidth $f(p_i) \leq b(p_i)$ to transfer a sub-file σ_i , where $\sum_{i \in [1, k]} \sigma_i = \sigma$. The total bandwidth used for the file transfer, $f(\mathcal{P})$, is represented as

$$f(\mathcal{P}) = \sum_{p_i \in \mathcal{P}} f(p_i), \quad (3)$$

and the time needed by p_i to transfer a sub-file of size σ_i across the allocated bandwidth $f(p_i)$ is

$$T(p_i, f(p_i), \sigma_i) = d(p_i) + \frac{\sigma_i}{f(p_i)}, \quad i \in [1, k]. \quad (4)$$

Thus, the total time needed to transfer the entire file of size σ by using all paths $p_i \in \mathcal{P}$ is

$$T(\mathcal{P}, f(\mathcal{P}), \sigma) = \max_{i \in [1, k]} T(p_i, f(p_i), \sigma_i). \quad (5)$$

3 Problem definition

TBTS (Time-constrained Big-file Transfer Scheduling) problem: Let σ be the size of the file that needs to be transferred from node s to node t within time τ , find a set of feasible paths $\mathcal{P}$, a feasible $s-t$ flow f , and a file distribution among the paths $p_i \in \mathcal{P}$, such that $T(\mathcal{P}, f(\mathcal{P}), \sigma) \leq \tau$.

It is difficult to derive a generalized exact solution for the multi-path routing problem because the multi-path routing problem is always NP-hard^[17]. Additionally, the fastest flow over time problem, which requires the fastest data transfer solution, has

also been proven to be NP-hard, such as the research in Ref. [12]. The TBTS problem not only needs to find a feasible path set, but must also facilitate the file transfer within a time constraint. Therefore, the TBTS problem is NP-hard.

Unlike the maximum-flow problem^[12,18], which only finds the maximum available aggregated-bandwidth between s and t , the TBTS problem considers a file of fixed size and finds a set of paths that can be used to transfer the file within a certain time. For example, consider the transfer of a file of size 13 units from node s to node t in the network shown in Fig. 2 within a time constraint of $\tau = 4$ units.

Figure 2 A network with fixed (bandwidth, delay) link weights

The maximum flow in this network is confined by link $(2, t)$ and is equal to 8. However, because there are two paths from s to t and the two sub-paths between s and node 2 have a total bandwidth of 10, there is a decision to make on how to distribute the maximum flow of 8 units over the two possible paths.

If we reserve 3 units of bandwidth on the top path ($p_1 = s-1-2-t$) and the remaining 5 units on the bottom path ($p_2 = s-2-t$), the time constraint of the transfer can be satisfied with $\sigma_1 = 3$ units of the file allocated to p_1 and $\sigma_2 = 10$ units of the file allocated to p_2 . If we reserve 5 units of bandwidth on p_1 and 3 units on p_2 , then it is possible that no solutions will satisfy the delay constraint, regardless of how we distribute the file over the paths. Because both solutions are maximum-flow solutions, the TBTS problem cannot be solved by solving the maximum-flow problem.

4 TBTS heuristic

4.1 Maximum flow over time

In Ref. [10], Ford and Fulkerson studied the MFT (Maximum Flow over Time) problem. The MFT

problem is defined as follows:

Given a weighted directed graph $G(V, E)$, a source node s , a destination node t , a set $\mathcal{P}$ of k $s-t$ paths, and a time period τ , a link $e \in E$ has bandwidth $b(e) > 0$ and delay $d(e) \geq 0$. The MFT problem amounts to finding the maximum amount of flow that can be transferred from s to t in time τ .

Ford and Fulkerson proved that the MFT problem has a LP (Linear Programming) formulation and can be solved in polynomial time by solving a corresponding minimum-cost flow problem^[10] as follows:

$$\min \sum_{\forall (u,v) \in \mathcal{P}} d(u,v) \cdot f(u,v) - \tau \cdot f(\mathcal{P}) \quad (6)$$

s.t.

$$\sum_{(s,v) \in \mathcal{P}} f(s,v) - f(\mathcal{P}) = 0 \quad (7)$$

$$\sum_{(v,t) \in \mathcal{P}} f(v,t) - f(\mathcal{P}) = 0 \quad (8)$$

$$\sum_{(u,v) \in \mathcal{P}} f(u,v) - \sum_{(v,z) \in \mathcal{P}} f(v,z) = 0, \quad v \neq s, t \quad (9)$$

$$0 \leq f(u,v) \leq b(u,v), \quad \forall (u,v) \in \mathcal{P}, \quad (10)$$

where $f(u,v)$ is the allocated flow for each link $e = (u,v)$, $e \in p_i$ and $f(\mathcal{P})$ is the aggregated bandwidth over all $p_i \in \mathcal{P}$. The maximum amount of data that can be transferred from node s to node t within time τ is

$$N(\tau, \mathcal{P}) = \tau \cdot f(\mathcal{P}) - \sum_{\substack{\forall (u,v) \in p_i, \\ p_i \in \mathcal{P}}} d(u,v) \cdot f(u,v). \quad (11)$$

For an $s-t$ path set $\mathcal{P}$ to be a solution to the TBTS problem, it must hold that $\text{MFT}(\tau, \mathcal{P})$ returns $N(\tau, \mathcal{P}) \geq \sigma$. Otherwise, transfer cannot be completed within time τ . Similarly, there exists a minimum time $\bar{\tau}$ required to transfer a file of a certain size. The minimum time period is related to how the file is distributed over the set of paths. Assuming the minimum time period $\bar{\tau}$ is given, the file distribution problem asks for a positive decomposition such that $\sum \sigma_i = \sigma$ and $0 < \sigma_i \leq (\bar{\tau} - d(p_i)) \cdot f(p_i)$.

Because the MFT problem assumes that a feasible path set $\mathcal{P}$ is given, we will present a heuristic algorithm for finding such a path set in the following section.

4.2 Algorithm

We propose Algorithm 1, a heuristic algorithm called TBTS-H for solving the TBTS problem. TBTS-H uses MFT to determine if a given path set can be used to transfer a file within the specified time. It does this iteratively by starting with a path set consisting of one path and using MFT to determine if it is feasible. If not, another path is added to the set, until MFT indicates that a feasible path set has been found or that no more paths can be added. The paths to be added are computed by using a k -shortest paths algorithm^[19] and it is possible to limit the size of the path set to at most $k_{\max}$ paths.

Algorithm 1 *TBTS-H*($G, s, t, k_{\max}$)

```

1: Let  $k \leftarrow 1$ .
2:  $\mathcal{P} \leftarrow$  Compute the  $k$ th-shortest path  $p$ .
3: if  $p \neq \emptyset$  and  $d(p) < \tau$  then
4:   Compute the LP of  $\text{MFT}(\tau, \mathcal{P})$ .
5:   if  $N(\tau, \mathcal{P}) < \sigma$  and  $k < k_{\max}$  then
6:      $k \leftarrow k + 1$ .
7:     Goto Line 2.
8:   end if
9:    $\bar{\tau} = \left\lceil \frac{\sum_{\forall (u,v) \in \mathcal{P}} d(u,v) \cdot f(u,v) + \sigma}{f(\mathcal{P})} \right\rceil$ 
10:  for each  $p_i \in \mathcal{P}$  do
11:     $f(p_i) \leftarrow \min_{(u,v) \in p_i} f(u,v)$ .
12:    for each link  $(u,v) \in p_i$  do
13:       $f(u,v) \leftarrow f(u,v) - f(p_i)$ .
14:    end for
15:  end for
16:  Compute the file distribution such that:
     $\sum \sigma_i = \sigma$ , and  $0 < \sigma_i \leq (\bar{\tau} - d(p_i)) \cdot f(p_i)$ .
17: else
18:   Return FALSE
19: end if
```

In the TBTS-H algorithm, line 2 computes the k th shortest path and adds it to the path set $\mathcal{P}$. The length measure used is $d(p_i)$, because $T(p_i, f(p_i), \sigma_i) \leq \tau$ only if $d(p_i) < \tau$ (according to Eq. (4)). Line 3 determines if a feasible path exists that can be added to the path set. The new path set is subsequently verified by the MFT linear program. Lines 4-7 determine if another new path is needed for solving the TBTS problem. Line 9 computes the minimum time period $\bar{\tau}$ needed to transfer a file of size σ over the path set by using Eq. (11).

Figure 3 An example showing the steps of TBTS-H. (a) G with (bandwidth, delay) pair; (b) $\mathcal{P}=\{p_1\}$; (c) $\mathcal{P}=\{p_1, p_2\}$

Lines 10-15 compute the maximum available bandwidth for each $p_i \in \mathcal{P}$ and line 16 determines how to split and transfer the file over the paths. According to Eq. (4), it takes $(d(p_i) + \sigma_i/f(p_i))$ units of time to transfer data of size σ_i along path p_i . If the time period is limited to $\bar{\tau}$, the maximum amount of data that can be transferred during $\bar{\tau}$ is equal to $(\bar{\tau} - d(p_i)) \cdot f(p_i)$. Thus, each p_i is allocated a sub-file of at most $(\bar{\tau} - d(p_i)) \cdot f(p_i)$ units. The file distribution problem boils down to solving a set of linear equations and can be solved through linear programming.

The MFT problem can be solved in pseudo-polynomial time bounded by $S(\varepsilon)$ where ε is an instance of the TBTS problem with input size $S(\varepsilon)$. Assuming k paths are required to satisfy the time-constrained transfer, the time complexity of computing a feasible path set (lines 2-7) is $O(k|V|(|E| + |V| \log |V|))$. Additionally, lines 10-15 require $O(n^2)$ time for allocating bandwidth to each path. Therefore, the time complexity of the TBTS-H algorithm is bounded by the number of shortest paths k .

4.3 Illustrative example for TBTS-H

In this section, we use a simple network to illustrate the steps performed by the TBTS-H algorithm. Consider the network with fixed (bandwidth, delay) link weights in Fig. 3(a). We wish to transfer a file of size 21 units from node s to node t with delay constraint of $\tau = 6$. TBTS-H starts by computing the shortest path ($k = 1$) such that $\mathcal{P}$ contains $p_1 = s, 1, 2, t$, as shown in Fig. 3(b). Because $d(p_1) = 2$, $\mathcal{P}$ is verified by solving a corresponding MFT($\tau, \mathcal{P}$) problem. The flows are shown next to the link weights in Fig. 3(b). The maximum amount of data that can be transferred in time τ is $N(\tau, \mathcal{P}) = 20$ units.

Therefore, TBTS-H continues by searching for another path to add to the existing path set, namely $p_2 = s, 1, 2, t$ with $d(p_2) = 3$. The iteration stops because $N(\tau, \mathcal{P}) = 29$. The flows are shown next to the link weights in Fig. 3(c). With this path set, the file transfer can be completed in at least $\bar{\tau} = \lceil (3 + 3 + 5 + 8 + 21)/8 \rceil = 5$ units of time. $p_1 = s, 2, t$ is allocated 5 units of bandwidth, and $p_2 = s, 1, 2, t$ is allocated 3 units of bandwidth. Now, we need to choose σ_1 and σ_2 such that $\sigma_1 + \sigma_2 = 21$, $0 < \sigma_1 \leq 15$ and $0 < \sigma_2 \leq 6$. One possible solution is to choose $\sigma_1 = 15$ and $\sigma_2 = 6$.

5 Exact TBTS algorithm

We have proposed a polynomial-time heuristic algorithm for solving the TBTS problem in the previous section. In this section, we propose an exact pseudo-polynomial time algorithm to solve the TBTS problem by using an auxiliary graph technique.

5.1 Auxiliary graph

An auxiliary graph, also known as a time-expanded graph, has already been used in the context of routing. For instance, the authors in Ref. [20] solved a constrained maximum-flow problem by transforming the input graph into an auxiliary graph, computing a solution for that auxiliary graph, and then transforming it back into a solution for the original input graph.

In the following, we will demonstrate that the auxiliary graph technique can also be used for solving the TBTS problem. We assume that the link delay values are represented as integral positive weights. We create the auxiliary graph $G^A(V^A, E^A, \tau)$ in the following manner:

1. V^A contains τ nodes: $u^0, u^1, \dots, u^{\tau-1}$ for each $u \in V, u \neq s$, and only contains s^0 for $s \in V$. For each $(u, v) \in E, u \neq s, E^A$ contains $(\tau - d(u, v))$ links in the form $(u^i, v^{i+d(u,v)})$, $i = 0, 1, \dots, \tau - d(u, v) - 1$. And E^A only contains $(s^0, v^{d(s,v)})$ for each $(s, v) \in E$. E^A also contains links in the form: (t^i, t^{i+1}) , $i = 0, 1, \dots, \tau - 2$.

2. Add node t^τ to N^A and link $(t^{\tau-1}, t^\tau)$ to E^A . Remove any nodes $u \neq s, t$ from V^A whose out-degree or in-degree is 0 and remove their attached links.

Figure 4 $G^A(V^A, E^A, \tau)$ corresponding to the instance shown in Fig. 2

The auxiliary graph $G^A(V^A, E^A, \tau)$ corresponding to the instance shown in Fig. 2 is shown in Fig. 4 and we can observe the following:

1. Any $s - t$ path in G^A is also an $s - t$ path in G whose delay is smaller than τ . For example, $p = s^0, 2^1, t^2$ in G^A corresponds to path $p^* = s, 2, t$ in G , which has a delay of 2 units.

2. The aggregated-flow over links $(u^i, v^{i+d(u,v)})$ may not exceed $b(u, v)$, which is the capacity of link (u, v) . For instance, the maximum aggregated-flow on $(2^1, t^2)$ and $(2^2, t^3)$ may not exceed the capacity of $(2, t) \in E$, which is 8.

Using these two observations, the maximum flow over time problem based on auxiliary graph can be expressed as follows:

$$\min \sum_{(u,v) \in E^A} d(u, v) \cdot f(u, v) - f(G^A) \cdot \tau \quad (12)$$

s.t

$$\sum_{(s^i, v^{i+d(s,v)}) \in E^A} f(s^i, v^{i+d(s,v)}) = f(G^A) \quad (13)$$

$$\sum_{(v^i, t^{i+d(v,t)}) \in E^A} f(v^i, t^{i+d(v,t)}) = f(G^A) \quad (14)$$

$$\sum_{(x,y) \in E^A} f(x, y) = \sum_{(y,z) \in E^A} f(y, z), \quad y \neq s, t \quad (15)$$

$$\sum_{i=0} f(u^{[i]}, v^{[i+d(u,v)]}) \leq b(u, v), \quad \forall (u, v) \in E^A \quad (16)$$

$$f(u, v) \geq 0, \quad \forall (u, v) \in E^A, \quad (17)$$

such that the maximum amount of data that can be transferred within time τ in G^A from node s^0 to node t^τ is equal to

$$M(\tau, G^A) = \tau \cdot f(G^A) - \sum_{(u,v) \in E^A} d(u, v) \cdot f(u, v), \quad (18)$$

where $f(G^A)$ is the corresponding aggregated-flow. We refer to this LP formulation as the MAF (Maximum Auxiliary Flow) formulation.

A solution to the TBTS problem only exists when the corresponding MAF formulation returns a solution for which $M(\tau, G^A) \geq \sigma$.

5.2 Algorithm

Algorithm 2 TBTS-A(G, s, t)

```

1: Construct the auxiliary graph  $G^A(V^A, E^A, \tau)$ .
2: Solve the MAF formulation.
3: if  $M(\tau, G^A) \geq \sigma$  then
4:   Compute the shortest  $s - t$  path  $p^*$ .
5:    $x \leftarrow d(p^*) + 1$ ;  $y \leftarrow \tau$ .
6:   while  $y \neq x + 1$  do
7:      $\bar{\tau} \leftarrow \lfloor (x + y) / 2 \rfloor$ .
8:     Construct the auxiliary graph  $G^A(V^A, E^A, \bar{\tau})$ .
9:     Solve the MAF formulation.
10:    if  $M(\bar{\tau}, G^A) \geq \sigma$  then
11:       $y \leftarrow \bar{\tau}$ .
12:    else
13:       $x \leftarrow \bar{\tau}$ .
14:    end if
15:  end while
16:   $\bar{\tau} \leftarrow y$ .
17:  Output the  $s - t$  path set  $\mathcal{P}$  obtained from the MAF
    solution with the corresponding flow allocation  $f(p_i)$ 
    for each  $p_i \in \mathcal{P}$ .
18:  Compute the file distribution such that:
     $\sum \sigma_i = \sigma$ , and  $0 < \sigma_i \leq (\bar{\tau} - d(p_i)) \cdot f(p_i)$ .
19: else
20:   Return FALSE.
21: end if

```

We propose Algorithm 2, called TBTS-A, for solving the TBTS problem. After constructing the aux-

Figure 5 An example showing the steps of TBTS-A. (a) $G^A(V^A, E^A, 6)$; (b) $G^A(V^A, E^A, 4)$; (c) $G^A(V^A, E^A, 5)$

iliary graph G^A (line 1) and solving MAF (line 2), line 3 determines if the transfer can be completed within the specified time. If so, lines 4-16 compute the minimum time period by using a binary search with scope $[d(p^*) + 1, \tau]$, where p^* is the shortest $s-t$ path in G . Lines 17 and 18 compute the $s-t$ path set and the corresponding flow decomposition based on the minimum time period $\bar{\tau}$. In TBTS-A, the number of the MAF problems that must be solved is bounded by $\lceil \log(\tau - d(p^*) - 1) \rceil + 1$ and the time complexity of an MAF problem is bounded by the size of the instance. Therefore, TBTS-A is a polynomial time algorithm.

5.3 Illustrative example for TBTS-A

In this section, we use the TBTS-A algorithm to solve the TBTS instance shown in Fig. 3. After executing line 1 of TBTS-A, we obtain graph G^A , shown in Fig. 5(a). Because $M(6, G^A) = 29 > 21$, the problem can be solved. TBTS-A then uses its binary search within scope $[3, 6]$ ($p^* = s, 2, t$ with $d(p^*) = 2$ is the shortest path in G). After constructing $G^A(V^A, E^A, 4)$ (Fig. 5(b)), $G^A(V^A, E^A, 5)$ (Fig. 5(c)), and solving the MAF, we find that the minimum time to complete the transfer is $\bar{\tau} = 5$. The aggregated-flow in G^A runs over two paths:

$p_1 = s^0, 2^1, t^2$ and $p_2 = s^0, 1^1, 2^2, t^3$. The flow values are shown next to each link in Fig. 5(c). Again (refer to Sec. 4.3), choosing $\sigma_1 = 15$ and $\sigma_2 = 6$ provides a solution for distributing the file.

6 Simulations

In order to evaluate the performance of our algorithms, we conduct simulations on ER (Erdős-Rényi)^[21] topologies ranging in size from 10 to 100 nodes. For each generated network, the probability of link existence is $(\lg |V|)/|V|$, the link delay $d(e)$ is uniformly distributed between 5 and 15 units, and the link bandwidth $b(e)$ is uniformly distributed between 10 and 20 units. We randomly select $2 \cdot |V|$ $s-t$ pairs for each topology and transfer files ranging in size from 10 to 100 units, within time constraint 40 units. Our algorithms are implemented in Python on an Intel(R) Core i7-4710MQ 2.5 GHz machine of 16 GM memory. In particular, we use Networkx⁴⁾ library to generate ER topologies and Pulp⁵⁾ to solve the LP problems

We first compare the performance of TBTS-H with that of the SP (Shortest Path) routing algorithm in terms of total-delay and the probability of finding a path or path set fulfilling the delay-constrained routing (i.e., success-rate). We set $k_{\max} = 10$ for TBTS-H and compare the TBTS-H

⁴⁾ <https://networkx.github.io/>.

⁵⁾ <http://www.coin-or.org/PuLP/index.html>.

Figure 6 Success-rate and delay. (a) Success-rate comparison; (b) delay comparison

and TBTS-A algorithms in terms of total-delay, the number of $s - t$ paths in $\mathcal{P}$, aggregated-bandwidth, and running-time.

Fig. 6(a) presents a comparison of median success-rate between TBTS-H and SP. For each algorithm, the success-rate decreases as network size and file size increases. Compared to the success-rate for SP (ranging from 16.67% to 85%), TBTS-H achieves a higher success-rate (ranging from 31.2% to 91%) because TBTS-H attempts to take full advantage of each $s - t$ path in order to satisfy the delay constraint. Additionally, the median delay achieved by each algorithm increases as network size and file size increases. As shown in Fig. 6(b), SP can achieve smaller delays than TBTS-H when file size ranges from 10 to 60 units. However, TBTS-H can achieve smaller delays than SP when large files are transferred, implying that TBTS-H is more efficient at transferring larger files across a network.

Compared to TBTS-H, TBTS-A uses a larger number of $s - t$ paths (Fig. 7(a)) with a larger aggregated-bandwidth (Fig. 7(b)) to achieve a smaller delay than TBTS-H (Fig. 7(c)). Although both TBTS-H and TBTS-A can obtain a feasible path set for solving the TBTS problem, the complexity of TBTS-A is significantly higher than TBTS-H, particularly when the network is large, as shown in Fig. 8. With TBTS-H, we can control running-time by setting $k_{\max}$. With TBTS-A, we do not have such a control, although we could scale the link weights to obtain a polynomial-time approximation algorithm, similar to the solution in Ref. [22].

Figure 7 Path-number, aggregated-bandwidth, and delay comparison between TBTS-H and TBTS-A. (a) Path-number comparison; (b) aggregated-bandwidth comparison; (c) delay comparison

Figure 8 Running-time comparison between TBTS-H and TBTS-A.

7 Conclusion

We have addressed the wired-transfer problem in IoV and studied the problem of routing and scheduling big-file transfer within a time constraint. For the TBTS problem, we proposed a heuristic algorithm based on the maximum flow over time problem and an exact algorithm based on an auxiliary graph technique. Our simulation results indicate that, both of the proposed algorithms perform well at solving the TBTS problem. The proposed heuristic algorithm in particular can address the TBTS problem within a short time frame, while the exact algorithm requires a longer running-time to address the TBTS problem in an optimal way.

Possible future work comprises designing traffic engineering strategies for files with different priorities in IoV.

References

- [1] Y. G. Bi, H. B. Zhou, W. C. Xu, et al. An efficient PMPv6-based Handoff Scheme for urban vehicular networks [J]. *IEEE transactions on intelligent transportation systems*, 2016, 17(12): 3613-3628.
- [2] Y. G. Bi, L. X. Cai, X. S. Shen, et al. Medium access control for QoS provisioning in vehicle-to-infrastructure communication networks [J]. *Mobile networks and applications*, 2013, 18(2): 174-185.
- [3] S. Kandula, I. Menache, R. Schwartz, et al. Calendaring for wide area networks [J]. *ACM SIGCOMM computer communication review*, 2015, 44(4): 515-526.
- [4] H. Zhang, K. Chen, W. Bai, et al. Guaranteeing deadlines for inter-data center transfers [J]. *IEEE/ACM transactions on networking*, 2016, 77(3): 415-512.
- [5] N. S. V. Rao, W. R. Wing, S. M. Carter, et al. Ultrascent net: network testbed for large-scale science applications [J]. *IEEE communications magazine*, 2005, 43(11): 12-17.
- [6] W. F. Liang. Robust routing in wide-area WDM networks [C]//*IEEE 15th International Proceedings on Parallel and Distributed Processing Symposium*, San Francisco, USA, 2001: 8-21.
- [7] B. H. Shen, B. Hao, A. Sen. On multipath routing using widest pair of disjoint paths [C]//*IEEE Workshop on High Performance Switching and Routing*, Phoenix, USA, 2004: 134-140.
- [8] D. Jurca, P. Frossard. Media flow rate allocation in multipath networks [J]. *IEEE transactions on multimedia*, 2007, 9(6): 1227-1240.
- [9] M. Macit, V. C. Gungor, G. Tuna. Comparison of QoS-aware single-path vs. multi-path routing protocols for image transmission in wireless multimedia sensor networks [J]. *Ad hoc networks*, 2014, 19(8): 132-141.
- [10] Jr L. R. Ford, D. R. Fulkerson. *Flows in Networks* [M]. Princeton: Princeton university press, 2015.
- [11] C. Y. Chiu, T. H. Tsai, Y. C. Liou, et al. Near-duplicate subsequence matching between the continuous stream and large video dataset [J]. *IEEE transactions on multimedia*, 2014, 16(7): 1952-1962.
- [12] G. L. Xue. Optimal multichannel data transmission in computer networks [J]. *Computer communications*, 2003, 26(7): 759-765.
- [13] M. Groß, M. Skutella. Maximum multicommodity flows over time without intermediate storage [C]//*Springer Berlin Heidelberg European Symposium on Algorithms*, Ljubljana, Slovenia, 2012: 539-550.
- [14] A. Hall, S. Hippler, M. Skutella. Multicommodity flows over time: Efficient algorithms and complexity [J]. *Theoretical computer science*, 2007, 379(3): 387-404.
- [15] B. Klinz, G. J. Woeginger. Minimum-cost dynamic flows: the series-parallel case [J]. *Networks*, 2004, 43(3): 153-162.
- [16] S. Sharma, D. Katramatos, D. T. Yu. End-to-end network QoS via scheduling of flexible resource reservation requests [C]//*ACM International Conference on High Performance Computing, Networking, Storage and Analysis*, Seattle, USA, 2011: 68.

- [17] J. C. Chen, S. H. G. Chan, V. O. K. Li. Multipath routing for video delivery over bandwidth-limited networks [J]. *IEEE journal on selected areas in communications*, 2004, 22(10): 1920-1932.
- [18] L. Fleischer, M. Skutella. Quickest flows over time [J]. *SIAM journal on computing*, 2007, 36(6): 1600-1630.
- [19] J. Y. Yen. Finding the k shortest loopless paths in a network [J]. *Management science*, 1971, 17(11): 712-716.
- [20] W. Y. Zhang, J. Tang, C. G. Wang, et al. Reliable adaptive multipath provisioning with bandwidth and differential delay constraints [C]//*IEEE INFOCOM 2010*, San Diego, USA, 2010: 1-9.
- [21] P. Erdős, A. R. W. On random graphs I [J]. *Publ math Debrecen*, 1959, 6: 290-297.
- [22] Y. Li, D. L. Du, N. H. Xiu, et al. Improved approximation algorithms for the facility location problems with linear/submodular penalties [J]. *Algorithmica*, 2015, 73(2): 460-482.

About the authors

Chuan Lin Ph.D. candidate at the School of Computer Science and Engineering, Northeastern University, Shenyang, China. He received a B.S. degree in computer science and technology from Liaoning University, Shenyang, China in 2011 and an M.S. degree in computer science and technology from

Northeastern University, Shenyang, China in 2013. His research interests include network performance analysis, software defined networking, and network modeling. (Email: chuanlin1988@gmail.com)

Yuanguo Bi [corresponding author] associate professor at the School of Computer Science and Engineering, Northeastern University, Shenyang, China. He was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada between 2007 and 2009. His current research focuses on medium access control, QoS routing, multihop broadcasting, and transmission control in vehicular networks. (Email: biyuanguo@ise.neu.edu.cn)

Hai Zhao Professor at the School of Computer Science and Engineering, Northeastern University, Shenyang, China. He received a B.Sc. degree in electrical engineering from the DaLian Maritime University, Dalian, China, in 1982, the M.Sc. and Ph.D. degrees from Northeastern University, Shenyang, China, in

1987 and 1995, respectively, both in computer science. He serves as the director of the Liaoning Provincial Key Laboratory of Embedded Technology. His current research focuses on embedded Internet technology, wireless sensor networks, pervasive computing, operating systems, data and information fusion, computer simulation, and virtual reality. He has led programs such as the National Natural Science Foundation of China, the National High Technology Research and Development Program of China, and the Nation Class Lighted Torch Plan. He has published more than 300 academic papers, four books, and international standards. He has successfully applied for 10 patents and received six awards for science and technology from the Liaoning Province and Ministry of China. He received recognition from the State Council due to his special contributions to the development of education. (Email: zhaohai@ise.neu.edu.cn)

Zeshen Wang Master's degree candidate at the School of Business Administration, Northeastern University, Shenyang, China. She received a B.S. degree from the School of Business Administration, Northeastern University, Shenyang, China in 2015. Her research interests include traffic engineering, network performance analysis, software defined networking and network modeling. (Email: jiwei@mail.neu.edu.cn)

Jinfa Wang Ph.D. candidate at the School of Computer Science and Engineering, Northeastern University, Shenyang, China. He received a B.S. degree in information and electrical engineering from LuDong University, Yantai, China in 2011 and an M.S. degree in information science and engineering from Northeastern University, Shenyang, China in 2014. His research interests are complex networks, Internet measurement, Internet topology analysis, and data science. (Email: jinfa.wong@gmail.com)