# Users at End Networks Collaborating for Maximum Flow

Elizabeth Varki
University of New Hampshire
varki@cs.unh.edu

## ABSTRACT

Transmitting large files over the Internet is not routine since transmission protocols are not designed to transmit hundreds of gigabytes at a time. A solution to this problem is to divide the large file into segments up to a few GBs in size; each segment is then transmitted separately to the receiver. The transmission of all the file's segments may take several days depending on the file's size and the bandwidth availability. This paper develops a scheduler that generates the optimal segment sizes and transmission start times. The transmission of a large data set requires proportionally large bandwidth. The available bandwidth at end networks varies according to Internet traffic - bandwidth availability is high during early morning hours when users are sleeping - so segment transmissions should be scheduled for early morning hours when Internet usage is low. When the sender and receiver are in different time zones, their hours of high bandwidth availability are not synchronized. It is still possible to take advantage of high bandwidth availability by collaborating with users in other networks. These transit users receive segments and later transmit them to other transit users or to the receiver. This paper develops a scheduler that generates optimal values for segment sizes, transit network locations, transit bandwidth capacities, and transmission start times. When there are few transit networks, segment sizes are larger and bandwidth requirements at each transit network is higher. Therefore, consider a crowd of transit users: the large file is divided into micro segments, a few megabyte in size, and each crowd user is responsible for the transmission of a single micro segment to another crowd user or to the receiver. The advantage of crowd transmission is that a large file can be transmitted quickly by reserving large bandwidth only at the sender and receiver.

## Keywords

flow networks, maximum flow network, performance models, graph models, modeling and analysis, maximum flow algorithms, parameter extraction, Internet flow model

## 1. INTRODUCTION

Consider the scenario: "a user in UK (the sender) wants to transmit a large data set to another user in Japan (the receiver). The data set consists of a collection of files with a total capacity in the TBs range. The sender and receiver are at end networks connected via the Internet. The users have permission to access available bandwidth at the end networks. The sender may divide each file into *segments* and transmit along multiple channels using transmission protocols such as ftp, HTTP, or gridFTP. It is not necessary to transmit file segments concurrently. The receiver reassembles files once the last segment arrives in Japan." This paper evaluates the scenario and answers questions such as: "What is the transmission duration?," "Does the transmission start time affect the duration of transmission?," and "What is the maximum number of bytes transmitted from a sender to a receiver in a given time duration?."

In order to answer these questions, one must know how much bandwidth is available for the transmission. The availability of bandwidth is dependent on the architecture, network policies, and traffic patterns at the sender and receiver's end networks and at the backbone long-haul networks. This paper evaluates the problem from the perspective of end users of the Internet. A user transmitting a file has no control over how the file's packets are routed. The user (with administrator permission) may use all available bandwidth at the end network by executing several ftp, HTTP, gridFTP commands simultaneously. Thus, a user has control over parameters such as a) the number of ftp transmissions, b) the start times of these transmissions, and c) the sizes of the file segments. This paper develops a scheduler that determines the optimum selection of these user controlled input parameters.

We refer to the scheduler as **Flowes**: **FLOW** of **E**lectronic **S**egments. Flowes determines the following input parameters: number of segments, sizes of segments, and transmission start times. Flowes does not route packets; it relies on the routing policies of the segment transmission protocol, which could be ftp, HTTP, or gridFTP, to name a few.

The Flowes transmissions are timed so that the combination of paths from sender to receiver has sufficient end-to-end flow capacity. Thus, the algorithm underlying Flowes is maximum flow [5]. The model for a maximum flow algorithm is the flow network which is a directed graph with flow capacity along arcs. This paper develops the flow network, which models flow paths from a sender's network to the receiver's network.

Next, the paper evaluates the Flowes scenario with several intermediate users: instead of transmitting all segments directly to the receiver, the sender transmits some segments to "transit" users at

end networks other than the sender and receiver's end networks. A transit user later transmits the segment to another transit user or to the receiver. Once all the segments arrive at the receiver, the file is reassembled. Flowes schedules segment transmissions not only from the sender's network but also to-and-from transit end networks. The paper constructs the flow network of transit end networks collaborating to maximize the flow between a sender's network and the receiver's network. With reference to the UK to Japan problem, the flow model answers questions such as: "Given a set of transit end networks, what is the maximum number of bytes that can be transmitted from UK to Japan?" and "Where should transit users be located so that a file is transmitted in the shortest time?," and "What are the bandwidth distributions at the transit users' networks that allow for optimal flow from a sender to a receiver?."

Finally, the paper considers the scenario of a "crowd" of intermediate users participating in the flow of a large data set from a sender to a receiver. In crowd Flowes, files are divided into *micro segments* where each micro segment is a few MBs in size. In transit Flowes, a segment could be several GBs in size, and require considerable bandwidth for transmission to-and-from transit end networks. With crowd Flowes, large bandwidth is required only at the sender and receiver networks.

## 2. RELATED RESEARCH

Large transmissions need large bandwidth - this principle is common to prior work and our research. Prior research has focused on transmission protocols, which fall under two categories, namely, parallel and store-and-forward (delay tolerant). Parallel protocols such as gridFTP and BitTorrent get large bandwidth by opening multiple TCP streams. Store and forward bulk transmission protocols are delay tolerant and transmit packets when large bandwidth is available [7, 11]. The delay tolerant protocols [10] rely on the sleep-wake Internet traffic pattern [9]. Packets from bulk transmission are forwarded during early morning hours when traffic is low and high bandwidth is available. Bulk protocols that transmit packets during low traffic times [10, 12] are less disruptive to Internet's users than greedy parallel protocols. Large transmissions during early morning hours are also cost-effective due to the availability of unused "purchased" bandwidth.

Flowes schedules concurrent transmissions when bandwidth utilization is low. The fundamental difference between our research and prior research is: we have evaluated user controlled parameters, while prior papers have evaluated network (administrator) controlled parameters. Consequently, Flowes generates user-level segment transmission schedules, such as when to transmit and where to transmit, while bulk protocols generate packet routing paths. The maximum flow algorithm underlies both store-and-forward bulk transmission protocols such as NetStitcher [10] and our Flowes scheduler. The inputs and outputs of a maximum flow algorithm are determined by its flow graph. The flow graphs for bulk protocols and our Flowes scheduler are different.

The bulk transmission protocols model Internet routing, and their flow model incorporates backbone network parameters outside the control of end users. The Internet has a mesh structure with several pathways between any two nodes. Consequently, the flow network for bulk protocols is a complete graph [4, 10]; when there are $n$ nodes, the number of edges is O($n^2$).

The bulk transmission flow model is complex because its input parameter values are not available to end users and its state space is large. The Flowes graph, on the other hand, has publicly available input parameters and a smaller graph state space. The input parameters are the uplink and downlink bandwidth at end networks, whose values are in the public domain. The flow network is a star graph with leaves representing end networks; when there are $n$ nodes, the number of edges is O($n$).

The contribution of this paper is the construction of a tool that allows end users to transmit large files quickly and cheaply using available network resources (such as unused bandwidth), while minimizing the negative impact of high bandwidth usage. In the following sections, we develop the flow network, which is the basis of the Flowes scheduler. The flow network is a graph model, so we map the relevant network parameters to graph parameters.

## 3. NETWORK PARAMETERS

The goal is to compute maximum flow from a sender's end network to a receiver's end network. The flow network is constructed by mapping system parameters to graph parameters.

**End network parameters:** The sender and receiver are end network ASs (Autonomous Systems). Let the sender be represented by $s$ and the receiver by $r$. The flow rate from $s$ to $r$ is determined by the bandwidth capacities of links between $s$ and $r$. The sender has uplink to the Internet and the receiver has downlink from the Internet. The sender's uplink capacity, ul, is the minimum of the sender's LAN uplink and its backbone (BB) Internet uplink.

$$\mathsf{ul}(s) = \text{minimum}\{LAN\mathsf{ul}(s), BB\mathsf{ul}(s)\}$$

Similarly, the receiver's downlink capacity, dl, is determined by the receiver's Internet backbone downlink which connects to its LAN downlink.

$$\mathsf{dl}(r) = \text{minimum}\{LAN\mathsf{dl}(r), BB\mathsf{dl}(r)\}.$$

The values of ul(s) and dl(r) are usually public.

**Internet parameters:** The bandwidth capacity between $s$ and $r$ is determined not only by the uplink from $s$ and downlink to $r$ but also by the links between them. The sender and receiver ASs are end networks (LANs) that are connected via backbone ASs (Wide Area Networks - WANs). The Internet has a mesh structure, so there are several paths between $s$ and $r$. For maximum flow computation, the only parameter of significance is the total bandwidth given to the transmission.

Let WAN(s,r) represent the total flow capacity from $s$ to $r$ along backbone networks. The maximum data flow between $s$ and $r$ can never exceed the minimum link capacities at $s$, $r$, and WAN(s,r). Finding the value of WAN(s,r) is challenging since backbone ISPs conceal network details. End network customers pay for $95^{th}$ percentile backbone bandwidth usage, so an end network has access to the BB uplink and BB downlink bandwidth it has paid for. Moreover, the Internet has a large degree of redundancy for network availability. The architectural and business features of the Internet ensure that the backbone links are not usually a bottleneck. If WAN details are unavailable, then it is assumed that WAN(s,r) is not the bottleneck, and WAN(s,r) is abstracted out of the flow model by setting WAN(s,r) $\geq$ maximum {BBul(s),BBdl(r)}.

**Graph model - system to graph mapping:** The flow network con-

sists of only two nodes, $s$ and $r$. There is a single arc from $s$ to $r$, $(s, r)$. The capacity of the arc, c($s$,$r$), is the minimum of the sender's uplink, the receiver's downlink, and the total backbone links between the nodes.

$$c(s, r) = \text{minimum}\{\mathsf{ul}(s), WAN(s, r), \mathsf{dl}(r)\} \quad (1)$$

The flow network G=(V,E) has node set V=$\{s, r\}$ and arc set E=$\{$(s,r)$\}$) where arc (s,r) has capacity function c(s,r). A flow in G defined by function f: $(s,r) \longrightarrow \mathbb{N}_0$ that satisfies the capacity constraint f($s$,$r$) $\leq$ c($s$,$r$). ($\mathbb{N}_0 \equiv \mathbb{N} \cup \{0\}$.) The largest file transferred from $s$ to $r$ is the maximum flow value $\mid f_{max} \mid = c(s, r)$.

**Example 1:** "What is the maximum number of bytes that can be transmitted from UK to Japan during 12 hours given that the sender's end network has bandwidth capacity of 10 Gb/s, the receiver has bandwidth capacity of 20 Gb/s and the backbone links between the end networks have a total capacity of 25 Gb/s."

For the example, ul(s)=54 TB, dl(r)=108 TB, WAN(s,r)= 135 TB, so c(s,r)=minimum{54, 108, 135} = 54 TB. The maximum file size that can be transmitted from the end network in Chicago to the end network in Japan over 12 hours is 54 TB. □

# 4. INTERNET TRAFFIC

Flowes transmissions (*i.e.,* the set of segment transmissions scheduled by Flowes) do not get solo access on end networks. A user either purchases fixed bandwidth or accesses available bandwidth. When Flowes has fixed bandwidth, Equation 1 suffices for maximum flow computation. The available bandwidth refers to the backbone bandwidth purchased by end networks from WAN ISPs (Internet Service Providers); since this bandwidth is shared by users at the end network, Flowes transmission may transmit only along the remaining already paid-for bandwidth. The Internet traffic at end networks has a predictable diurnal wave distribution [9] where traffic increases gradually during the day with peak bandwidth usage between 6:00 PM and 10:00 PM; the traffic drops off sharply after midnight. Consequently, when Flowes transmits along available bandwidth, Equation 1, which assumes fixed bandwidth, is insufficient for maximum flow computation.

## 4.1 System time

The available uplink capacity ul and the available downlink capacity dl of end networks are a function of time of day. The WAN(s,r) bandwidth may also vary with time. Let $\tau$ represent UTC time. This paper models time in discrete increments, $\tau = 0, 1, 2, ..., \Gamma - 1$, where $\Gamma$ is the total number of time instants in a day. The value of $\Gamma$ depends on the time unit; $\tau = 0$ refers to the UTC time interval starting at 00:00 UTC, and $\tau = \Gamma - 1$ refers to the last UTC time interval ending at 00:00 UTC. For example, if time unit is an hour, then $\Gamma = 24$ and link capacities are defined for each hour in [00:00-23:00]; $\tau = 0$ represents UTC [00:00-01:00), $\tau = 1$ represents UTC [01:00-02:00), ..., and $\tau = 23$ represents UTC [23:00-00:00). If time unit is 5 minutes, $\Gamma = 288$, $\tau = 0$ represents UTC [00:00-00:05), $\tau = 1$ represents UTC [00:05-00:10), ..., and $\tau = 287$ represents UTC [23:55-00:00). The link capacities, ul(s,$\tau$), dl(r,$\tau$), and WAN(s,r,$\tau$), are defined for each time instant. For example, suppose time unit is 1 hour: if uplink for s is 10 Gb/s from UTC[03:00- 04:00), then ul(s,3)=4500 GB. In order not to get hung up on units such as MB, GB, Gb, we drop them when specifying bandwidth capacity per instant, so ul(s,3)=4500.

**Example 2 - mapping bandwidth distribution:** Suppose the sender's network in UK and the receiver's network in Japan permit Flowes transmission from local time [12:00 AM-12:00 PM). The available bandwidth distribution by local time is provided below:

[12:00 AM-3:00 AM): 10 units/3hours;

[3:00 AM-6:00 AM): 20 units/3hours;

[6:00 AM-9:00 AM): 18 units/3hours;

[9:00 AM-12:00 PM): 8 units/3hours.

The bandwidth distribution is specified in units/3hours for computational simplicity. Each time instant is equivalent to 3 hours, so $\Gamma = 8$. $\tau = 0$ represents UTC interval [00:00-03:00), $\tau = 1$ represents UTC interval [03:00-06:00), $\tau = 2$ represents UTC interval [06:00-09:00), ..., $\tau = 6$ represents UTC interval [18:00-21:00), $\tau = 7$ represents UTC interval [21:00-00:00).

The sender and receiver have identical distribution by local time, but they are in different time zones, so their bandwidth distribution by UTC differ.

UK is in UTC + 00:00 (zone 0), so local time equals UTC time. Therefore, the uplink, ul(s,$\tau$), from UK is given by:

ul(s,0)=10; ul(s,1)=20; ul(s,2)=18; ul(s,3)=8; ul(s,4)=0; ul(s,5)=0; ul(s,6)=0; ul(s,7)=0.

Japan is in UTC + 09:00 (zone 9), so the local time is 9 hours ahead of UTC time (*i.e.,* 3 time units ahead). Therefore the downlink, dl(r,$\tau$), to Japan is given by:

dl(r,0)=8; dl(r,1)=0; dl(r,2)=0; dl(r,3)=0; dl(r,4)=0; dl(r,5)=10; and dl(r,6)=20; dl(r,7)=18. □

The UTC time zone (offset from local time) specifies the geographic positions of $s$ and $r$; thus, with the inclusion of UTC time, the global span of the Internet is incorporated in the model. Let $\theta$ represent the UTC instant at which Flowes is initiated. In the above example, if Flowes is initiated at 09:00 UTC ($\tau = 3$) then $\theta = 3$. Suppose Flowes duration is 6 time instants; the UTC instants relating to Flowes duration are: $\tau$=$\theta$=3, $\tau$=$\theta$+1=4, $\tau$=$\theta$+2=5, $\tau$=$\theta$+3=6, $\tau$=$\theta$+4=7, $\tau$=$\theta$+5=0. Thus, Flowes runs from 09:00 UTC to 03:00 UTC the next day.

## 4.2 Graph model - mapping UTC instant $\tau$ to flow instant $t$

In graph theory, a flow network models flows over time instants t=0,1,2,...,T-1. The flow instant t=0 corresponds to the instant at which flow starts, and the flow instant t=T-1 corresponds to the instant at which flow ends. Thus, t=0 of the flow graph corresponds to UTC instant $\tau$=$\theta$ and t=T-1 of the flow graph corresponds to UTC instant $\tau$=$\theta$+T-1 in modulo $\Gamma$ arithmetic. The flow network models flow instants t=0, 1, 2, ..., T-1 which corresponds to UTC instants $\tau = \theta, \theta + 1, ..., \theta + T - 1$ modulo $\Gamma$.

The flow network G=(V,E) has node set V=$\{s, r\}$ and arc set E=$\{$(s,r)$\}$ in which arc (s,r) has capacity function c(s,r,t) defined by:

$$c(s, r, t) = \text{minimum}\{\mathsf{ul}(s, \tau), \text{WAN}(s, r, \tau), \mathsf{dl}(r, \tau)\} \quad (2)$$

where $0 \leq \theta < \Gamma$ is the UTC instant when flow is initiated, $0 \leq$

$t < T$ is the flow instant and T is the flow duration. The system time parameter $\tau$ is mapped to the graph time parameter t by $\tau = \theta + t$ in modulo $\Gamma$ arithmetic. A flow in G is a function $f : (s, r) \times [0, T) \longrightarrow \mathbb{N}_0$ that satisfies the capacity constraint $f(s, r, t) \leq c(s, r, t)$. The largest file that can be transferred from $s$ to $r$ is given by the maximum flow value

$$| f_{max} | = \sum_{t=0}^{T-1} c(s, r, t) \quad (3)$$

**Example 3:** Consider the setup of Example 1. Suppose Flowes is initiated from UK to Japan at $\theta = 6$ for 4 times instants; $\tau = 6, 7, 0, 1$. The values of ul(s,$\tau$) and dl(r,$\tau$) (as computed in Example 1) are: ul(s,6)=0, ul(s,7)=0, ul(s,0)=10, ul(s,1)=20; dl(r,6)=20, dl(r,7)=18, dl(r,0)=8, dl(r,1)=0. Assume that WAN(s,r) is not the bottleneck.

Mapping system parameters to graph parameters: T=4 and t=0, 1, 2, 3, so values of c(s,r,t) are

c(s,r,0)= minimum{ul(s,6), dl(r,6)} = 0,

c(s,r,1) = minimum{ul(s,7), dl(r,7)} = 0,

c(s,r,2) = minimum{ul(s,0), dl(r,0)} = 8,

c(s,r,3) = minimum{ul(s,1), dl(r,1)} = 0.

The total Flowes flow from UK to Japan, by Equation 3, is $\sum_{t=0}^{3} c(s, r, t) = 8$ when Flowes is initiated at $\theta = 6$ for 4 time instants. □

**Example 4:** Reconsider Example 1, Suppose Flowes is initiated from UK to Japan at $\theta = 1$ for 4 time instants.

c(s,r,0) = minimum{ul(s,1), dl(r,1)} = 0,

c(s,r,1) = minimum{ul(s,2), dl(r,2)} = 0,

c(s,r,2) = minimum{ul(s,3), dl(r,3)} = 0,

c(s,r,3) = minimum{ul(s,4), dl(r,4)} = 0.

The total Flowes flow from UK to Japan is $\sum_{t=0}^{3} c(s, r, t) = 0$ when Flowes is initiated at $\theta = 1$ for 4 time instants. □

A consequence of including bandwidth distribution in the Flowes model is that a new parameter, namely, start time, becomes significant to the maximum flow computation. In Example 2, maximum file size transmitted from UK to Japan is 8 when Flowes is initiated at $\theta$=6; in Example 3, maximum file size transmitted from UK to Japan is 0 when Flowes is initiated at $\theta$=1.

## 4.3 Flowes start and end time

In Example 3, Flowes is initiated at UTC instant 6, but there is no flow until UTC instant 0. The UTC instant 0 is the first instant that segment transmission occurs. Since transmission does not occur during all flow instants, we define the following terms:

**Definition** Flowes **start time** is the time instant that the first segment starts transmitting from the sender; Flowes **end time** is the time instant that the last segment arrives at the receiver; Flowes transmission **duration** is the difference between (end time + 1) and the start time.

In Example 4, when Flowes is initiated at UTC 1, there is no transmission at the next 4 instants. Therefore, as per the definition, there is no start time, end time, nor transmission duration. When Flowes is initiated at UTC 6, start time = UTC 0, end time = UTC 0, and duration = 1.

Note that start time is flow instant t=0 and UTC instant $\theta$; end time is flow instant t=T-1 and UTC instant $\tau$=$\theta$+t in modulo $\Gamma$ arithmetic. The flow network corresponding to Flowes need only model states corresponding to t=0 (start time) until t=T-1 (end time). Flow occurs between t=0 and t=T-1, so Flowes transmission duration is T. It is possible that there are time instants 0 < t < T-1 with no transmission, but Flowes transmission duration is still T since duration is determined by start time and end time.

The introduction of bandwidth distribution to the model results in new Flowes problems:

1. What is the maximum number of bytes that can be transmitted from UK to Japan given that start time is UTC instant $x$ and duration is $y$ instants?

2. What is the start time that results in the maximum bytes being transferred from UK to Japan in $y$ time instants?

3. Given a file size $x$, what time instant should Flowes be started so that the file can be transferred in minimum duration?

4. Given a file size and transmission completion time, what is the latest start time?

# 5. TRANSIT FLOWES

When the sender and receiver are in different time zones, their high capacity time instants are asynchronous. For example, maximum flow from UK to Japan is 8 even though each end network has 56 bandwidth units per day. A maximum of 56 units can be transmitted from UK to Japan if a suitable "transit" user is found. Suppose a transit user in Germany has access to 20 units/3 hours for the entire day. UK starts transmitting at UTC instant 0 when UK has uplink capacity 10 and Japan has downlink capacity 8. At UTC 0, UK transmits 8 units to Japan and the remaining 2 units to Germany. During the next 3 instants, UK transmits to Germany since Japan has 0 bandwidth. When bandwidth becomes available in Japan, Germany transmits to Japan.

It is cheaper to purchase bandwidth during low traffic sleep times. Scheduling Flowes during hours of low bandwidth utilization is also less disruptive to high priority transmissions. Therefore, it is cost-effective and unselfish to have transit networks in several time zones. A sender may transmit to a transit user; the transit user may transmit to another transit user or to the receiver. Next, we construct the flow network with a sender, receiver, and one or more transit networks.

The flow network modeling transit Flowes consists of several transit nodes, in addition to the sender and receiver nodes. Therefore, Equation 3 does not suffice for maximum flow computation; instead, maximum flow algorithms, such as Edmonds Karp algorithm [6], must search the flow network to compute maximum flow.

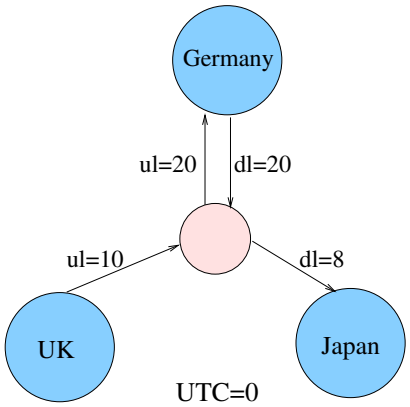


**Figure 1: Star graph**

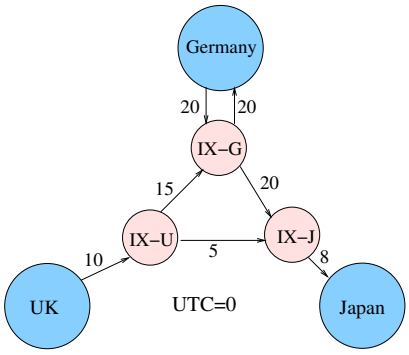


**Figure 2: Star-Complete graph**

## 5.1 Challenge to modeling transit Flowes

The input parameters to the model are the sender's uplink capacity (ul), the receiver's downlink capacity (dl), and the transit networks uplink and downlink capacities (ul and dl). Each network is represented by a node, and the arcs represent flow between the end nodes. The capacity of arc ($u$,$v$) is equal to the minimum of $u$'s uplink capacity and $v$'s downlink capacity. The sender has outgoing arcs to every node, the receiver has incoming arcs from every node, and the transit nodes have both outgoing and incoming arcs to-and-from every node. Therefore, the flow network is a complete graph. When there are $n$-2 transit networks, the number of nodes in the flow network is O($n$) and the number of arcs is O($n^2$).

The complete graph models the mesh structure of the Internet, but the graph is unable to model the end network bottleneck capacity. For example, At UTC 0, UK has capacity 10, Japan has capacity 8, and Germany has capacity 20. The arc from UK to Japan has capacity 8 (minimum of 10 and 8) and the arc from UK to Germany has capacity 10, thereby, allowing UK to transmit 18 units. However, UK has total capacity of only 10 units at UTC 0, so the complete graph is an incorrect model of Flowes.

Next, we model Flowes with a star graph. Each end network is represented by a leaf node, with arcs to-and-from the hub node. The arc from a leaf to the hub represents the downlink from the end network, and the arc from a hub to a leaf node represents the uplink to the end network. The hub node represents an Internet eXchange (IX) where networks connect. Figure 1 shows the star graph of UK, Japan, Germany Flowes; At UTC 0, UK may transmit at most 10 units of flow. Therefore, a star graph models the complete flow connectivity between end networks and also models their bottleneck bandwidth capacity.

So far, we have assumed that total backbone capacity between end networks is not a bottleneck. Normally, the Internet's architecture and business policies guarantee that total backbone capacity is greater than the LAN and BB bandwidth capacities at end networks. However, Flowes, a bandwidth greedy application, stresses Internet resources and may result in backbone bottlenecks. The probability of backbone bottlenecks is higher when the communicating end networks are separated by several time zones since the IXs where long-haul networks connect (such as DE-CIX [2], AMS-IX [1], and LINX [3], to name a few), display sleep-wake traffic patterns. Reconsider the UK, Japan, Germany example with the additional constraint that backbone capacity from UK to Japan at UTC 0 is 5, WAN(UK,Japan,0)=5, and the backbone capacity from UK to Germany is 15, WAN(UK,Germany,0)=15. Thus, WAN(UK,Japan,0) has lower bandwidth capacity than the bandwidth capacity at the end networks and is the bottleneck in end-to-end maximum flow from UK to Japan at UTC 0. At UTC 0, UK would transmit 5 units to Japan and 5 units to Germany. The star graph incorrectly models UK transmitting 8 units to Japan and 2 units to Germany.

The challenge is to model both the end network bottlenecks and the total backbone flow capacity between end networks. One possibility is to combine the star graph and the complete graph as shown in Figure 2. The nodes in the complete sub-graph represent Internet Exchanges and the arcs in the complete sub-graph represent backbone flow capacities between the IXs. The capacity of arc (u-X,v-X) connecting the exchange nodes for end networks u and v is WAN(u,v). If the complete sub-graph is viewed as a single hub, then the graph has a star structure. The leaf nodes of this star-complete graph represent end networks and the arcs between a leaf nodes and their corresponding exchange node represent Internet flow capacity to-and-from the end networks.

Unfortunately, this star-complete graph is an inaccurate model of Flowes since it permits routing options between end networks as explained here: flow from end network $u$ to end network $v$ should only take the path $\langle u, u\text{-}X, v\text{-}X, v\rangle$ (where u-X represents u's exchange node). However, the star-complete graph permits routing options between exchange nodes. For example, consider Flowes collaboration amongst UK, Germany, and Japan represented by Figure 2. Suppose a transmission is scheduled at UTC 0 from UK to Japan. A model for Flowes should direct the flow along $\langle$ UK, UK-IX, Japan-IX, Japan $\rangle$. A maximum flow algorithm searching the complete sub-graph, however, may select the routing path $\langle$ UK, UK-IX, Germany-IX, Japan-IX, Japan $\rangle$. A Flowes graph should not model routing options between end nodes.

A Flowes transmission is defined from end network to end network, so a flow from end network $u$ to end network $v$ can only pass through nodes representing IXs $u$-$X$ and $v$-$X$. The challenge is to model both the total backbone capacity between every pair of end networks and the bottleneck bandwidth capacity at each end network while ensuring that flows from $u$ to $v$ only traverse nodes $u$-$X$ and $v$-$X$. In short, the challenge is to model end-to-end flow connectivity without routing. In the next section, we present our solution to this problem.

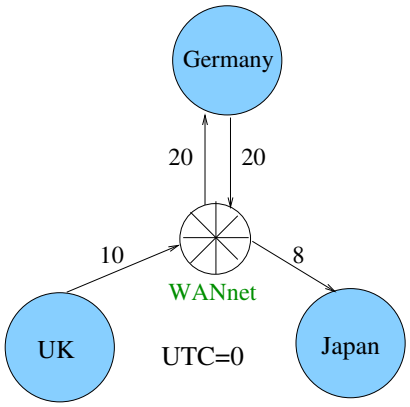


**Figure 3: Star Flowes graph**

## 5.2 Star Flowes network

This section presents the Flowes model, which is a flow network, characterizing end-to-end flow. Since the Flowes model is a flow network, it satisfies flow properties of capacity constraint, skew symmetry, and flow conservation [5]. Maximum flow algorithms such as Edmonds Karp algorithm [6] and Ford Fulkerson algorithm [8] can run over the star Flowes model.

Before presenting model definition, we explain star Flowes graph informally. As the name suggests, the Flowes graph has a star structure similar to that of Figure 1. The fundamental difference between star and star Flowes is the hub node - in Flowes, the hub node models WAN(u,v) between end nodes $u$, $v$ for all leaf nodes of the star graph. In Figure 3, the lines within the hub node represent WAN links. When flow is initiated from leaf nodes $u$ to leaf node $v$, the flow cannot exceed the minimum of ul(u), dl(v), and WAN(u,v). Thus, the WANnet hub models the backbone WAN capacity constraints between end nodes without modeling WAN routes.

**Definition** For a flow duration t=0 until t=T-1 (UTC instants $\tau = \theta$ until $\tau$ = $\theta$+T-1, $0 \leq \tau, \theta < \Gamma$), the Flowes network $G^\star$=(V$\cup x$, E,T) where

- V is the set of $n$ nodes representing end networks which includes the sender $s$, the receiver $r$, and the ($n$-2) transits;
- $x$ represents a single exchange node; and
- E is the set of arcs $\{(s,x),(x,r)\} \cup \{\ (v,x),(x,v)\ \mid\ \forall v \in$ V-$s$-$r$ $\}$.

$G^\star$ is a star network where the leaves are in node set V and the hub is node $x$. The nodes in V have infinite storage capacity, and the node $x$ has zero storage capacity. The arcs in E have bandwidth capacity:

c($v$,$x$,$t$) = ul($v$,$\tau$) $\quad \forall\ (v,x) \in$ E

c($x$,$v$,$t$) = dl($v$,$\tau$) $\quad \forall\ (x,v) \in$ E

In addition, bandwidth capacity between leaf nodes is defined by:

c($u$,$v$,$t$) = WAN($u$,$v$,$\tau$) $\quad \forall\ (u,x), (x,v) \in$ E

$\forall t = 0, 1, ..., T-1$ where $\tau$ = $\theta$ + t in modulo $\Gamma$ arithmetic. □

*The inclusion of* c($u$,$v$,$t$) *where* ($u$,$v$) $\notin$ E *deviates from the standard definition of flow networks where capacity function is only defined over arcs.* When WAN($u$,$v$,$\tau$) is not the bottleneck (or its value is unknown), set WAN($u$,$v$,$\tau$)=$\infty$.

A maximum flow algorithm computes maximum flow when the networks consists of several nodes. The majority of maximum flow algorithms ignore time by assuming instantaneous flow from sender to receiver; the corresponding flow network is *static*. Transit Flowes is modeled by a flow network where arc capacity varies with time; such a flow network is *dynamic*. Ford and Fulkerson [8] proposed the following solution for dynamic flows: convert a dynamic flow to a static flow using a *time expanded* flow network. The time expanded flow network is a static network in which there is a copy of the graph for each time instant $0 \leq t <$ T. Figure 4 shows the time expanded version of Figure 3 when duration is 4 time units. In this Figure, Flowes starts from UK at UTC 6 and ends at ends at UTC (6+4-1) modulo 8 = 1; here $\theta = 6$ and graph flow instants are $0 \leq t < 4$. (In the figure, we have modified the bandwidth distribution at the end networks from the one presented in Example 2.) There are 4 sub-graphs, each representing an instant of time. The dashed lines are called *holdover* arcs and they represent storage capacity at end nodes. The holdover arc $(u_t, u_{t+1})$, $u \in V$, models $u$ at $t$ storing segments for transmission at $t+1$ Storage capacity at end networks is not a bottleneck, so capacity of holdover arcs is set to infinity.

**Definition** The dynamic Flowes network $G^\star$=(V $\cup$ $x$,E,T) transforms to the static Flowes network G=($V_T \cup X_T$, $E_T \cup H_T$) where

- $V_T$ is the set of end nodes $u_t$, $\forall u \in V$ and t=0,1,...,T-1;
- $X_T$ is the set of hub nodes $x_t$, t=0,1,...,T-1;
- $E_T$ is the set of arcs $(u_t, x_t)$, $(x_t, u_t)$, $\forall\ (u,x), (x,u) \in E$ and t=0,1,...,T-1;
- $H_T$ is the set of *holdover* arcs $(u_t, u_{t+1})$, $\forall\ u \in V$ and t=0,1,...,T-2.

The arcs have capacity:

c($u_t$,$x_t$) = c($u$,$x$,$t$), c($x_t$,$u_t$) = c($x$,$u$,$t$), and c($u_t$,$u_{t+1}$) = $\infty$.

In addition, bandwidth capacity between end networks at time $t$ is defined by c($u_t$,$v_t$) = c($u$,$v$,$t$) □

The dynamic flow from $s$ to $r$ is equivalent to a corresponding static flow from $s_0$ to $r_{T-1}$ [8]. Therefore, finding maximum flow in the dynamic network can be solved by finding maximum flow in the corresponding static time expanded network. For the rest of the paper, we refer to the time expanded version of the Flowes network.

**Definition** A flow in time expanded G is a function

f: $V_T \times V_T \longrightarrow \mathbb{N}_0$ satisfying the property: $\forall$ nodes $u_t, v_t \in V_T, t \in$[0,T),

f($u_t$,$v_t$) $\leq$ minimum {c($u_t$,$x_t$), c($u_t$,$v_t$), c($x_t$,$v_t$)} and

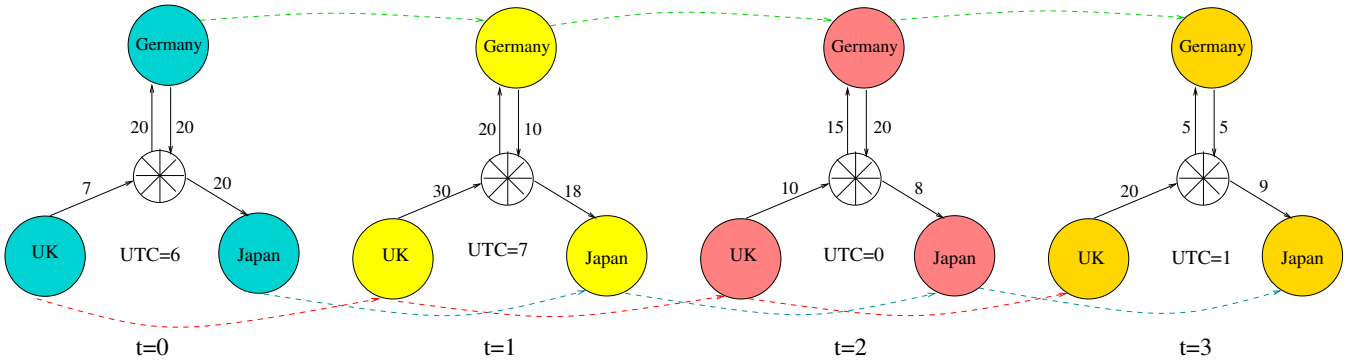


**Figure 4: Time expanded star Flowes graph**

$f(u_t,u_{t+1}) \leq c(u_t,u_{t+1})$.

The value $| f |$ of a flow f in G is

$$| f | = \sum_{v \in V_0} f(s_0,v_0) + f(s_0,s_1) = \sum_{v \in V_{T-1}} f(v_{T-1},r_{T-1}) + f(r_{T-2},r_{T-1})$$

□

The flow function, f, is not defined over hub nodes $x_t$ because Flowes only schedules transmissions between end networks. In the Appendix, we prove that f satisfies all flow properties, and hence, f is a valid flow.

When the Flowes graph is input to a maximum flow algorithm, the output is the maximum flow from sender to receiver along with the corresponding Flowes schedule. The Flowes schedule lists the times at which segments are to be transmitted to-and-from various nodes starting from the sender and ending at the receiver. Since maximum flow algorithms compute flow as per the standard definitions, the flow algorithm must be modified according to the Flowes definition. Edmonds Karp maximum flow algorithm searches for maximum flow paths using a breadth first search of all the nodes. We modified the breadth first search routine as follows: when a hub node is reached, the search continues to the end node; the flow is computed from end-to-end as the minimum of the capacities of three arcs, namely, end-to-hub, hub-to-end, and end-to-end WAN. This is the only modification to the Edmonds Karp algorithm.

## 5.3 Motivating example

The objective is maximal flow from Chicago (Ch) to Japan (Jp) when the sender in Chicago and the receiver in Japan are in end networks with identical bandwidth distribution by local time. Since Japan is 15 hours ahead of Chicago, the sender and receiver have different distributions by UTC time. Suppose transit users are located in time zones that are 3 hours apart. Starting from Chicago, the Flowes users, in UTC zone order, are in: Chicago (Ch), Argentina (Ag), United Kingdom (UK), Jordan (Jd), Bhutan (Bh), Japan (Jp), New Zealand (NZ), and Alaska (Ak). (Argentina is 3 hours ahead of Chicago, Alaska is 3 hours behind Chicago.) We analyze Flowes with two bandwidth distributions. (It is assumed that a network's upload and download distributions are identical.)

**Bandwidth distribution 1:** The bandwidth distribution, by local time, at Chicago and Japan is as follows: 12:00 AM-3:00 AM: 10 units; 3:00 AM-6:00 AM: 20 units; 6:00 AM-9:00 AM: 18 units; 9:00 AM-12:00 PM: 8 units; 12:00 PM-3:00 PM: 3 units; 3:00 PM-

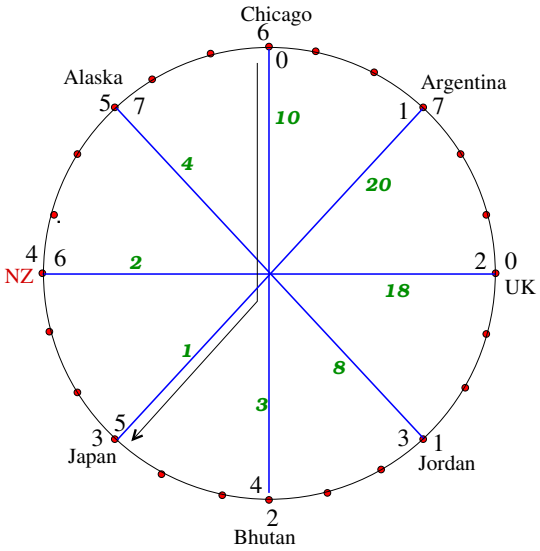


**Figure 5: WANnet showing UTC and local time at each node. The numbers outside the hub represent UTC instant while the numbers just inside the hub represent local time instant. The numbers along the links represent WAN bandwidth (distribution 1) at the corresponding instant.**

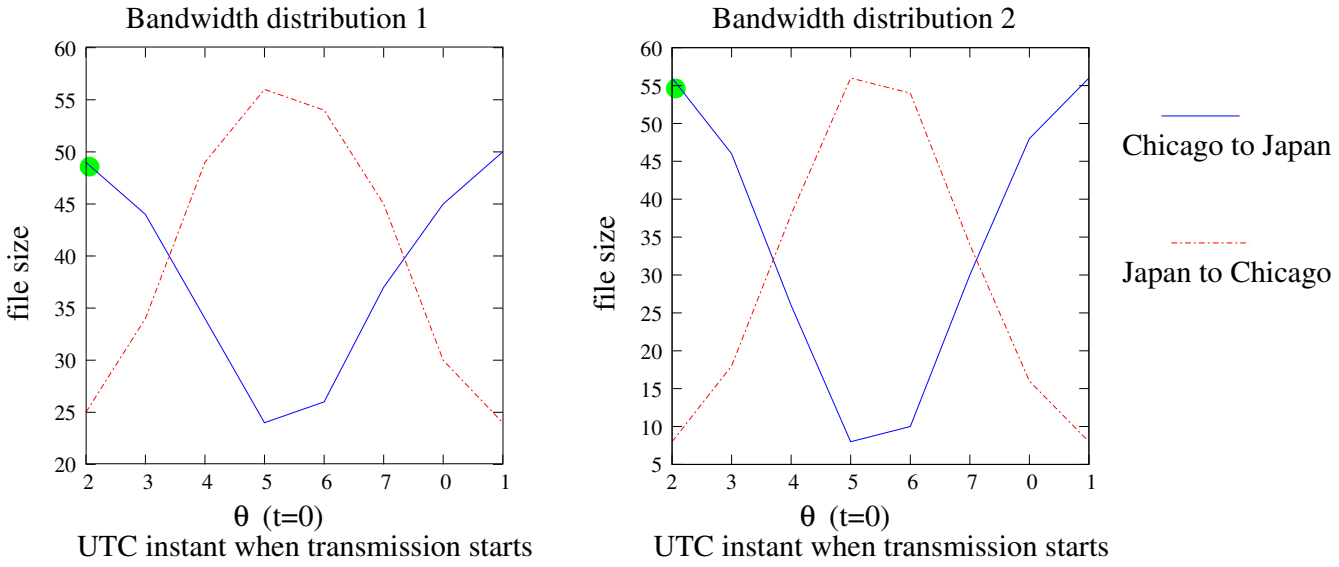


**Figure 6: Impact of Flowes start time on size of transmitted file over duration of at most 24 hours ($0 \leq t < 8$). The marked point on the Chicago to Japan plots of graphs 1 and 2 correspond to total flow shown in the last row, first column of Tables 1 and 2, respectively.**

**Table 1: Bandwidth distribution 1: Flowes scheduler output for Chicago to Japan transmission with start time UTC 06:00 (local time in Chicago 12:00 AM) and Flowes duration of 24 hours. Each row presents a segment flow path from Chicago to Japan via one or more transit nodes. The first column of the last row presents the total units transmitted from t=0 until t=7; other columns in the last row present the units transmitted to Japan at the corresponding time instant.**

| Segment size | UTC 06:00 ($\tau$=2,t=0) | UTC 09:00 ($\tau$=3,t=1) | UTC 12:00 ($\tau$=4,t=2) | UTC 15:00 ($\tau$=5,t=3) | UTC 18:00 ($\tau$=6,t=4) | UTC 21:00 ($\tau$=7,t=5) | UTC 00:00 ($\tau$=0,t=6) | UTC 03:00 ($\tau$=1,t=7) |
|---|---|---|---|---|---|---|---|---|
| 1 | Ch-to-Jp | | | | | | | |
| 2 | | Ch-to-Jp | | | | | | |
| 4 | | | Ch-to-Jp | | | | | |
| 8 | | | | Ch-to-Jp | | | | |
| 2 | | Ch-to-Ak | | Ak-to-Jp | | | | |
| 3 | | | | | Ch-to-Jp | | | |
| 3 | | Ch-to-Ak | | | Ak-to-Jp | | | |
| 5 | | | Ch-to-NZ | | NZ-to-Jp | | | |
| 2 | | | Ch-to-Ak | | Ak-to-Jp | | | |
| 1 | | | | | | Ch-to-Jp | | |
| 5 | Ch-to-Jd | | | | | Jd-to-Jp | | |
| 2 | | | | | | | Ch-to-Jp | |
| 4 | Ch-to-UK | | | | | | UK-to-Jp | |
| 1 | | Ch-to-UK | | | | | UK-to-Jp | |
| 3 | | | Ch-to-Ak | Ak-to-NZ | | NZ-to-Jp | | |
| 3 | | | | | | | | Ch-to-Jp |
| 49* | 1 | 2 | 4 | 10 | 13 | 9 | 7 | 3 |

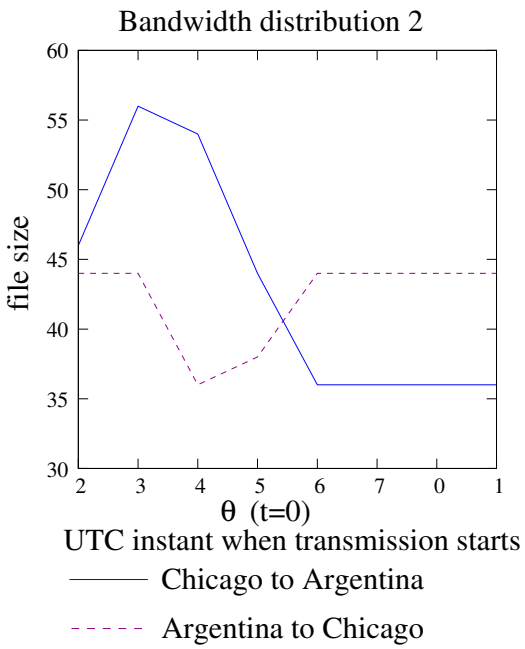


**Figure 7: Impact of Flowes start time on size of transmitted file over duration <= 24 hours ($0 \leq t < 8$).**

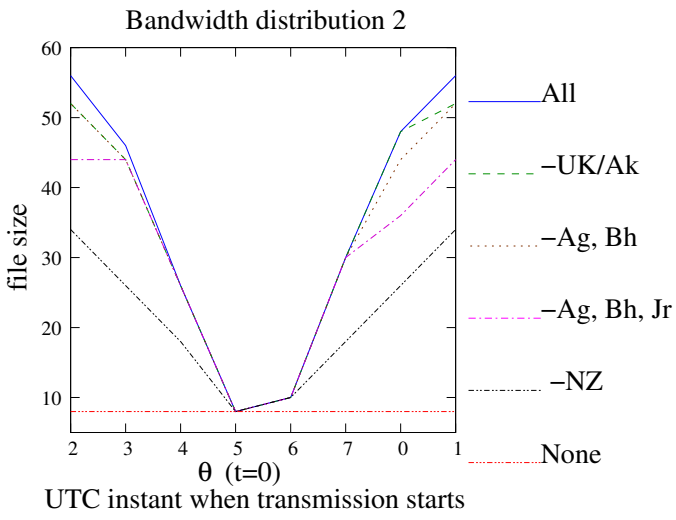


**Figure 8: Impact of transit nodes on size of transmitted file. All: with all transits; -UK/Ak: minus either UK or Alaska; -Ag,Bh: minus both Argentina and Bhutan; -Ag,Bh,Jr: minus Ag, Bh, and Jordan; -NZ: minus NZ; None: no transit nodes, direct sender to receiver.**

6:00 PM: 1 unit; 6:00 PM-9:00 PM: 2 units; 9:00 PM-12:00 AM: 4 units. (We have moved away from bandwidth units such as b/s and simply give bandwidth as flow values.) The transit networks have the following end network distribution by local time: 12:00 AM-3:00 AM: 5 units; 3:00 AM-6:00 AM: 5 units; 6:00 AM-9:00 AM: 5 units; 9:00 AM-12:00 PM: 5 units; 0 units for the remaining time. Thus, transit networks only permit Flowes from midnight until noon and have far less bandwidth than either the sender or the receiver. Assume that WAN distribution is not the bottleneck at any of the end nodes. □

**Bandwidth distribution 2:** Suppose all networks - sender, receiver, and transits - have identical bandwidth distribution by local time. All networks are only allowed to transmit from midnight until noon. The bandwidth distribution, by local time, is as follows: 12:00 AM-3:00 AM: 10 units; 3:00 AM-6:00 AM: 20 units; 6:00 AM-9:00 AM: 18 units; 9:00 AM-12:00 PM: 8 units; 12:00 PM-3:00 PM: 0; 3:00 PM-6:00 PM: 0; 6:00 PM-9:00 PM: 0; 9:00 PM-12:00 AM: 0. □

Both bandwidth distributions model the sleep-wake diurnal cycle with more bandwidth being available during the early morning hours. The fundamental difference between the two distributions is the bandwidth at the transit networks - in distribution 1, the transit networks have small bandwidth for Flowes. Figure 5 shows the WAN-net hub of the flow graph for the example's infrastructure.

The objective is to transmit maximum flow from Chicago to Japan in 24 hours. Mapping to the graph model: since time unit is 3 hours, there are 8 star Flowes sub-graphs in the time expanded graph (Figure 4), where each star sub-graph represents flow time t, $0 \leq t < 8$. For each change in input parameter value, a new graph is constructed and the maximum flow algorithm is run.

Tables 1 and 2 list all the maximum flow paths from Chicago to Japan which are output by Edmonds Karp algorithm when transmission starts at midnight, Chicago time. The transmission schedules show the transit networks along a Flowes path. Transit networks in Argentina and Bhutan do not appear on the schedule in Table 1. Summing column 1 of Tables 1 and 2 gives maximum flow of 49 and 56 units, respectively. The maximum flow of 49 units takes 24 hours while the maximum flow of 56 units takes 21 hours.

Figure 6 plots the file sizes transmitted from sender to receiver as starting time instant is varied. Flowes duration (for each start instant) is at most 24 hours; each time instant represents 3 hours. These graphs show the impact of start time on the transmitted file size. The first graph assumes distribution 1 and the second graph assumes distribution 2. The solid line references Chicago to Japan transmission and the dashed line references Japan to Chicago transmission. The sender and receiver have more bandwidth with distribution 1, but the transit networks are the bottleneck. Consequently, more data are transmitted with distribution 2. Figure 7 plots file sizes as start time is varied when transmissions are scheduled from Chicago to Argentina and vice versa.

Figure 8 shows the impact of removal of one or more transit networks from the Flowes collaboration. When there are no transit networks, the start time has no effect on maximum flow since 8 units are transmitted over 24 hours with each start instant. The removal of the transit user in New Zealand has a major negative impact on maximum flow. The removal of one or more of the other transit net-

**Table 2: Flowes scheduler output: Chicago to Japan for bandwidth distribution 2 with Flowes start time at UTC 06:00 and Flowes duration of 21 hours. Each row represents a segment flow path from sender to receiver via one or more transit nodes.**

| Segment size | UTC 06:00 ($\tau$=2,t=0) | UTC 09:00 ($\tau$=3,t=1) | UTC 12:00 ($\tau$=4,t=2) | UTC 15:00 ($\tau$=5,t=3) | UTC 18:00 ($\tau$=6,t=4) | UTC 21:00 ($\tau$=7,t=5) | UTC 00:00 ($\tau$=0,t=6) |
|---|---|---|---|---|---|---|---|
| 8 | | | | Ch-to-Jp | | | |
| 2 | | Ch-to-Ak | | Ak-to-Jp | | | |
| 8 | | Ch-to-Ak | | | Ak-to-Jp | | |
| 10 | | | Ch-to-NZ | | NZ-to-Jp | | |
| 8 | Ch-to-Jd | | | | | Jd-to-Jp | |
| 2 | | | Ch-to-Ak | Ak-to-NZ | NZ-to-Jp | | |
| 2 | Ch-to-UK | | | | | | UK-to-Jp |
| 6 | | Ch-to-UK | | | | | UK-to-Jp |
| 6 | | | Ch-to-Ak | Ak-to-NZ | | NZ-to-Jp | |
| 2 | | Ch-to-Ag | Ag-to-Ak | Ak-to-NZ | | NZ-to-Jp | |
| 2 | | Ch-to-Ag | Ag-to-Ak | Ak-to-NZ | NZ-to-Bh | Bh-to-Jp | |
| 56* | 0 | 0 | 0 | 10 | 20 | 18 | 8 |

works does not have significant impact on the maximum flow. The graph shows that the location of a transit node and its bandwidth distribution is relevant to its participation in Flowes transmission.

The addition of transit networks has added several interesting Flowes problems:

1. evaluate problems listed in earlier sections (Sections 1 and 4 with transit networks included in the transmission;
2. for a given sender and receiver, evaluate optimal values for transit locations and bandwidth;
3. for a given set of transits, find the minimum bandwidth and storage capacity to ensure maximum flow from sender to server;
4. evaluate how the selection of the time unit impacts maximal flow;

   If time unit is 1 minute then $\Gamma$, the total number of minutes in a day, is 1440. For Flowes duration of 24 hours, T=1440, for duration of 48 hours, T=2880, as opposed to T=8 and T=16 when time unit is 3 hours. The time unit impacts the state space of the maximum flow algorithm. From a systems perspective, we are interested in evaluating how a change in time unit effects location and bandwidth requirements of transit networks, the optimal start time, and the maximal flow value.
5. evaluate Flowes schedule using PERT/CPM methods.

   The Flowes schedule has slack in some segment paths, where a slack refers to the time difference between segment arrival and segment transmission from transit nodes. At a slack, it is possible for a segment to arrive later than its scheduled arrival time without impacting the maximal flow duration. The output of Flowes is a schedule, which could be input to a PERT/CPM algorithm to estimate latest arrival times of segments at transit hops.

Sometimes, the only invariants in a Flowes scheduler are the sender and receiver's network parameters, and the goal is to transmit the file quickly and cheaply. The locations and bandwidth capacities of the intermediate nodes, the backbone bandwidth capacities, and the optimal start time are to be determined by the maximum flow

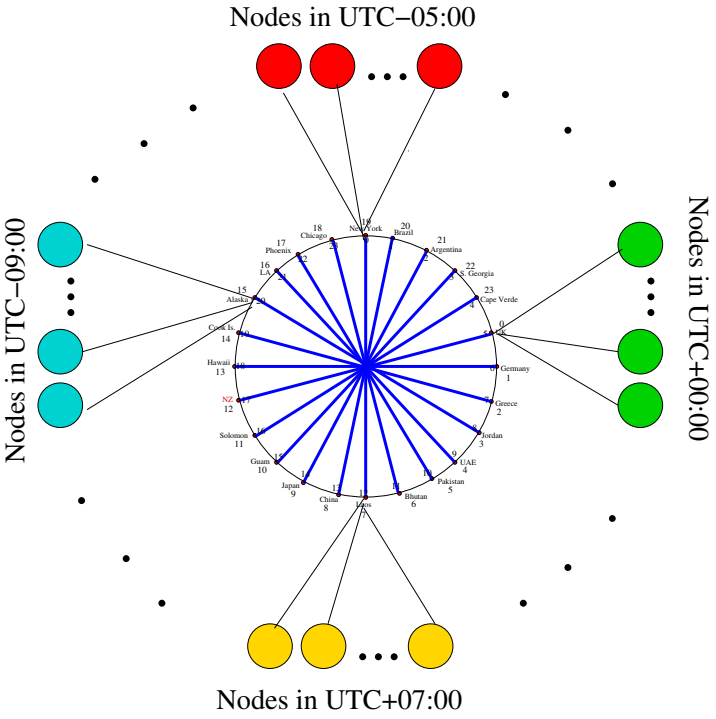


**Figure 9: Crowd Flowes model when backbone bandwidth in time zones is the bottleneck.**

algorithm. With each change in the value of an input parameter, a new flow model must be constructed and solved. Finding the optimal solution requires several executions of the maximum flow algorithm with various combinations of the input parameters; each execution requires the construction of a new flow model.

## 6. CROWD FLOWES

Flowes transmission is bandwidth intensive not only at the sender and receiver networks but also at the transit networks. We now consider another scenario where heavy bandwidth usage is restricted to just the sender and the receiver. The sender divides the file into micro segments with sizes in MB range that are typical of file sizes in a single ftp or HTTP transmission. The micro segments are trans-

mitted to a crowd of transit users' end networks. A crowd end network transmits its micro segment, on schedule, to another crowd end network or to the receiver. When all the micro segments arrive at the receiver, the file is reassembled. The transmission of a micro segment uses little bandwidth. The advantage of transmitting micro segments to a crowd of end networks is that bandwidth need not be purchased at transit end networks.

Crowd Flowes is modeled by the star Flowes graph. The fundamental difference between the transit model and the crowd model is scale, both in terms of $n$, the number of nodes, and $T$, the duration of flow. The complexity of the scheduler's maximum flow algorithm depends on the graph search space which is a function of $n$ and T. Transit Flowes has at most 100 nodes while crowd Flowes may have several thousands of nodes. The duration of flow, T, is determined not only by the duration of flow but also by the time unit. A time unit of 3 hours is specified in the motivating example of last section, so T=8 for Flowes duration of 24 hours. We set T=8 for convenience only: it is large enough to showcase the impact of start times on transit Flowes, yet small enough for graph display and readability. In reality, a time unit of 5 minutes is sufficient to capture the variance in bandwidth distribution over time [11]; if variance is higher, then time unit should be set to 1 minute. If Flowes duration is 24 hours, then T=288 when time unit is 5 minutes, and T=1440 when time unit is 1 minute. In crowd Flowes, however, the time unit is determined by the transmission time of a micro segment. Micro segments are transmitted in a few seconds, and a time unit must be small enough to model their transmission. Thus, for crowd Flowes, a time unit is at most a minute, but it is often much smaller. If time unit is 30 seconds, then T=2880 for a duration of 24 hours.

Both transit Flowes and crowd Flowes are modeled by the star Flowes graph. A Flowes scheduler performs two tasks: first, it constructs the flow graph, and second, it execute the maximum flow algorithm overlaying the graph. To find an optimal schedule, the Flowes scheduler varies input parameters such as start time, transit end network locations, bandwidth distributions, and duration of flow. Therefore, the Flowes scheduler repeats the tasks of graph construction and algorithm execution until it finds the optimal solution. The scale of a crowd Flowes graph makes it prohibitively expensive to repeat the above tasks for more than a few iterations. Therefore, we consider alternative models for crowd Flowes:

1. A star Flowes network with each crowd network represented by a node. This is similar to the transit Flowes graph shown in Figure 3. The time expanded network has O(n*T) nodes where both n and T are in the thousands for crowd Flowes.

2. A star Flowes network where all the crowd nodes in each time zone connect to a single exchange (IX) node as shown in Figure 9. Thus, all crowd nodes within a time zone map to the same WAN capacity. The transmission of a micro segment requires little bandwidth, but transmission of a crowd of micro segments requires large bandwidth. The sleep hours of high bandwidth availability are not significant to crowd end networks since the transmission of a micro segment is no different from regular file transmissions. A micro segment may be transmitted at any time. A crowd of micro segments transmitting over backbone links, however, has an impact on backbone bandwidth usage. In order to prevent disproportionate use of backbone bandwidth by crowd users, Flowes should ensure that transmissions to-and-from crowd users are scheduled during low bandwidth sleep times. By linking all crowd networks in a zone to a single WAN link, the total crowd capacity would never exceed the backbone capacity.

3. A star Flowes network consisting of a sender node, a receiver node, and time zone nodes. Each UTC time zone is represented by a node in the Flowes graph. All crowd networks in a time zone are encapsulated in the time zone node for that region. The value of T remains high since the time unit is determined by the end-to-end time to transmit a micro segment from one end network to another. The value of $n$, the number of nodes, however, reduces from thousands to less than 50 nodes. Thus, the search space in the time expanded Flowes graph is much smaller than that of the two alternative models. This approximate model could be used in the early stages of an evaluation when a large number of candidate input parameters must be evaluated. Subsequent to the elimination of non-suitable configurations, the first two models may be used to generate suitable schedules.

We plan to evaluate the alternative models with the goal of finding the model with the smallest state space that captures the behavior of crowd Flowes. We are also trying to develop a flow network that models clock time and the relationship between UTC time, local time, and graph flow time. The performances of transit Flowes and crowd Flowes are dependent on clock time. The bandwidth distribution has a diurnal cycle, but this periodicity is not captured in the graph model. If a graph could explicitly model clock time then it may be possible to develop a Flowes model with a smaller state space.

# 7. CONCLUSIONS

This paper develops a quick and cheap solution to the problem of how Internet users transmit large files from a sender's end network to a receiver's end network using existing infrastructure and transmission protocols. The solution is based on the premise that large transmissions need large bandwidth, and large bandwidth is available during early morning hours when Internet traffic is down. We are not the first to propose this approach, but we are the first to evaluate this approach from the end users' perspective. Earlier papers have addressed this problem from the perspective of administrators who control packet routing along backbone networks. Our solution only evaluates user controlled input parameters such as transmission start times. We have developed a scheduler called Flowes that generates optimal transmission start times, which is a deceptively simple problem.

The complexity arises when the sender and receiver are in different time zones and the hours of high bandwidth are unsynchronized. In this case, transmission is sped up by dividing the file into segments and transmitting them to the receiver via one or more intermediate users at networks in various time zones. Thus, the path of a segment from sender to receiver may include hops at one or more end networks. The number of intermediate networks involved in the transmission from sender to receiver results in: transit Flowes with tens of intermediate networks; crowd Flowes with thousands of intermediate networks. The Flowes scheduler determines optimal values for segment sizes, transit network locations, transit network bandwidth, and transmission start times. The Flowes scheduler is based on the maximum flow algorithm.

A contribution of this paper is the development of the flow network that models Flowes. The Flowes graph models backbone links be-

tween end networks while ensuring that end-to-end flow is direct. Earlier papers on bulk transmission have modeled end-to-end connectivity with routing, and their flow network is a complete graph. Flowes, on the other hand, is modeled by a star graph, which has a lower order of magnitude edge complexity than the complete graph.

We have identified and evaluated user controlled parameters that impact the maximum flow between two end networks. Flowes is based on the Edmonds Karp algorithm, which is a popular maximum flow algorithm based on graph search techniques. Flowes, however, is structured around clock time - both UTC and local time. An algorithm that incorporates clock time may lower search complexity of the Flowes star graph. This is especially beneficial for crowd Flowes, which has thousands of nodes. In the future, we plan to develop a maximum flow algorithm for Flowes.

# 8. REFERENCES


[1] Ams ix amsterdam internet exchange; statistics, https://www.ams-ix.net, 2015.

[2] De-cix where networks meet; statistics, http://www.de-cix.net/about/statistics/, 2015.

[3] Linx london internet exchange, https://www.ams-ix.net, 2015.

[4] P. Chhabra, V. Erramilli, N. Laoutaris, R. Sundaram, and P. Rodriguez. Algorithms for constrained bulk-transfer of delay-tolerant data. In *ICC'10*, pages 1–5, 2010.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, 1 edition, June 1990.

[6] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, Apr. 1972.

[7] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. *SIGCOMM Comput. Commun. Rev.*, 34(4):145–158, Aug. 2004.

[8] L. F. Jr. and D. Fulkerson. Constructing maximal dynamic flows from static flows. *Operations Research*, 6:419–433, 1958.

[9] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *SIGMETRICS'04*, pages 61–72, 2004.

[10] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez. Inter-datacenter bulk transfers with netstitcher. In *Proceedings of the ACM SIGCOMM 2011*.

[11] N. Laoutaris, G. Smaragdakis, P. Rodriguez, and R. Sundaram. Delay tolerant bulk data transfers on the internet. In *SIGMETRICS*, pages 229–238, 2009.

[12] C. Shi, M. H. Ammar, and E. W. Zegura. Idtt: Delay tolerant data transfer for p2p file sharing systems. In *GLOBECOM'11*, pages 1–5, 2011.


# APPENDIX

**Result:** *The function f in time expanded G is a valid flow.*

**Proof:** A valid flow function satisfies the properties of capacity constraint, skew symmetry, and flow conservation [5].

**Capacity constraint:**

For all nodes $u_t, v_t \in V_T, t \in$[0,T),

f($u_t$,$v_t$) $\leq$ minimum {c($u_t$,$x_t$), c($u_t$,$v_t$), c($x_t$,$v_t$) }

f($u_t$,$u_{t+1}$) $\leq$ c($u_t$,$u_{t+1}$)

*The definition of flow function* f *from leaf node $u_t$ to leaf node $v_t$ is stricter than the standard where the only constraint is* f($u_t$,$v_t$) $\leq$ c($u_t$,$v_t$).

**Skew symmetry:** For all $u_t$,$v_t \in V_T$, $t \in$[0,T),

f($u_t$,$v_t$) = -f($v_t$,$u_t$),

f($u_t$,$u_{t+1}$) = -f($u_{t+1}$,$u_t$).

**Flow conservation:** All positive flow into $u_t \in V_T$-$s_0$-$r_{T-1}$ equals all positive flows out of $u_t$.

$$\sum_{\substack{v_t \in V_t \\ f(v_t,u_t)>0}} f(v_t,u_t)+f(u_{t-1},u_t) = \sum_{\substack{v_t \in V_t \\ f(u_t,v_t)>0}} f(u_t,v_t)+f(u_t,u_{t+1})$$

When all flows, positive, zero, negative are considered, the flow conservation equations for $u_t$ sum to zero.

$$\sum_{v_t \in V_t} f(u_t,v_t) + f(u_{t-1},u_t) + f(u_t,u_{t+1}) = 0$$

The function f satsfies all the flow properties, so f is a valid flow. □