Modeling bulk transmission with a star graph

Elizabeth Varki

University of New Hampshire

email: varki@cs.unh.edu

Abstract

Bulk transmission refers to the transmission of large files - sizes in the gigabyte, terabyte, petabyte range - from a sender to a receiver, via the Internet. The time to transmit a large file depends primarily on the bandwidth capacity of the Internet links; thus, the bulk routing algorithm must find the high-capacity links. The problem is complicated because the capacity of links varies during the day depending on Internet traffic. Maximum-flow over time is the routing problem underlying the transmission of large files over the Internet. The underlying routing model is a directed graph with n nodes that represent data centers and arcs that represent end-to-end bandwidth capacity between the nodes. The inter-connectivity of the Internet ensures that there is end-to-end flow between any two nodes. Consequently, the flow network of the Internet is a complete graph. The number of nodes in the graph is O(n) and the number of arcs is $O(n^2)$. Since bandwidth capacity along arcs varies by time-of-day, the flow network models flows over time. In a discrete time model, the arc capacity is parameterized for each flow instant t=0, 1, ..., T-1. The time expanded graph of the flow network has $O(nT)$ nodes and $O(n^2T)$ arcs. This paper proves that the complete graph that models Internet flow can be mapped to an equivalent star graph. The time expanded star graph has $O(nT)$ nodes and $O(nT)$ arcs. The complexity of routing algorithms is proportional to the size of the search space. By lowering the space complexity of the graph model, this paper lowers the complexity of the corresponding flow routing algorithm. The contribution of this paper is showing that flow in a completely connected Internet can be modeled by a star graph.

1 Introduction

Maximum-flow is the problem that underlies bulk transmission via the Internet. Bulk transmission refers to the transmission of a large file - whose size may range from tens of Gigabyte to the Petabyte scale - from a sender's network to a receiver's network. The Internet parameter of significance is the bandwidth capacity available to the transmission. For example, a 100 MB file is transmitted in less than a minute over a 20 Mb/s, while a 100 GB file takes 11.36 hours over the 20 Mb/s link and less than 7 minutes over a 2 Gb/s link, a reduction of 99% of transmission time. The physical bandwidth capacity is fixed, but the available bandwidth varies during the day depending on Internet traffic. The network traffic pattern is dependent on time-of-day [5]; bandwidth usage increases during the day reaching peak usage around 8:00 PM, bandwidth usage decreases after 8:00 PM and reaches minimum around 3:00 AM. There is maximum bandwidth availability during the early morning hours of 1:00 AM to about 9:00 AM. This is similar to roadways, where roads are congested during rush-hour traffic.

The dependence of bandwidth availability on time-of-day may result in low end-to-end bandwidth when the sender and receiver are located in different time zones. Flow rate is limited by the smallest pipe, so a sender with ample bandwidth at 3:00 AM (sender's local time) may be forced to transmit at low rates due to insufficient bandwidth availability at the receiver (where the local time is 8:00 PM). This problem can be addressed by transmitting to intermediate data centers that have high bandwidth capacity at the same time as the sender and the receiver. Instead of transmitting directly (end-to-end) from the sender to the receiver, it is faster to transmit by a store-and-forward protocol [7].

NetStitcher [6] is a store-and-forward bulk transmission protocol where the underlying flow network is modeled as a time expanded graph and the routing is generated by Ford and Fulkerson's maximum-flow algorithm [4]. The routing is determined before the start of transmission; the file is divided into varying sized *segments* as determined by the routing algorithm. Each file segment travels along its predetermined path which may include transit data centers were a segment is stored until bandwidth is available. En route, at a transit data center, a file segment may be divided into further file segments that are routed separately. Once all the file segments arrive at the receiver, the file is reassembled.

Bulk routing protocols like NetStitcher are categorized as maximum flow algorithms. The modeling tool for the algorithm is the flow network which is a directed graph where the arcs have capacity that permits flow between end nodes. Thus, bulk routing algorithms require construction of a flow network of the Internet. Since the capacity of links is dependent on Internet traffic which varies over time, the flow network models flows over time. The discrete time flow network is a time expanded graph which contains a copy of the set of nodes and the set of arcs at every flow time step. Let n represent the number of data centers - sender, receiver, and transits - participating in the bulk transmission, and let T represent the number of discrete time steps in the flow duration. Since any two networks (data centers) on the Internet are connected, there are links between every pair of nodes in the graph. Thus, the time expanded graph for Internet flow has O(nT) nodes and $O(n^2T)$ arcs [6].

This paper reduces the complexity of bulk routing, not by developing a new algorithm, but by reducing the number of nodes and arcs in the underlying graph model. The paper reduces the state space by proving that the complete flow graph of the Internet can be mapped to an equivalent star graph. The number of nodes in the star graph is $n + 1$ representing the n data centers and the internal node. The star graph has O(nT) nodes and O(nT) arcs as opposed to the complete time expanded flow network with O(nT) nodes and $O(n^2T)$ arcs.

Prior papers on bulk transmission have focused on the networking protocol. For example, several papers [1, 7, 8, 9, 10] have studied the suitability of end-to-end, store-and-forward, priority, and BitTorrent protocols for bulk transmission. Other papers [2, 6] have tailored maximum flow algorithms to compute Internet flow. However, prior papers have not evaluated the flow model underlying bulk routing algorithms. The NetStitcher paper [6] briefly outlines the complete directed flow network underlying their routing algorithm. This is the **first** paper to present a detailed analysis of the Internet's flow model. The paper then proves that the complete directed flow network of the Internet can be mapped to a star graph, thereby lowering the search space complexity of the overlaying routing (maximum flow) algorithm.

Table 1: The notation and terminology is similar to one followed by Cormen et al. in the first edition of their textbook [3] where flow networks with anti-parallel arcs (a,b), (b,a) are presented.

Notation	**Meaning**
s	sender data center
r	receiver data center
u, v	transit data centers
V	set of data center nodes (including s, r, and transit nodes)
x_u	exchange node linked to data center u
X	set of eXchange nodes
a, b	a node in $V \cup X$ (*i.e.,* $a = u \in V$ or $a = x_u \in X$)
n	cardinality of sets V and X; ($\|V\|$ = $\|X\|$ = n)
E	set of arcs (a,b)
τ	UTC time instant
Γ	total number of UTC time instants in a day; $\tau = 0, 1, ..., \Gamma - 1$
ul(u, τ)	available uplink capacity of u $\in$ V at UTC instant τ
dl(u, τ)	available downlink capacity of u $\in$ V at UTC instant τ
t	flow time instant
T	total flow instants; t=0,1,...,T-1
θ	UTC time $\tau = \theta$ when flow instant $t = 0$
$c(a, b, t)$	capacity of edge (a,b) at flow instant t
$f(a, b, t)$	flow along edge (a,b) at flow instant t
$f(a, b, T')$	the sum total flow along edge (a,b) from t=0 until t=T'-1, $T' \leq T$.
$f(X, b, t)$	the sum total flow from nodes in set X to node b at time t
$\mid f \mid$	flow value of flow f
$\mid f_{max} \mid$	maximum flow value
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\}$
$p(t)$	simple path from u to v $\langle u, x_u, x_{w1}, x_{w2}, ..., x_{w'}, x_v, v\rangle$(t)
p*(t)	direct simple path from u to v $\langle u, x_u, x_v, v\rangle$(t)
p	path from s to r
G_f	residual network G due to flow f
$c_f(a, b, t)$	residual capacity along (a,b,t) due to flow f

2 End-to-End flow

The logistical problem underlying bulk transmission is captured by the problem statement:

What is the largest file that can be transmitted from a sender to a receiver along the Internet during a given time duration?

This is a maximum flow problem, and in order to find a solution, the first task is to construct the flow network of the Internet. The flow network is a directed graph where the arcs represent the end-to-end flow (bandwidth) capacity between the nodes. There is end-to-end flow between any two nodes of the Internet, so this graph is complete. In this section, we evaluate the Internet parameters of significance for construction of the graph model.

Static model: The input parameters to the problem are the sender and receiver's Internet bandwidth capacities. Specifically, the sender and receiver are data centers at end networks. The sender has uplink to the Internet and the receiver has downlink from the Internet. The sender's uplink consists of two links, namely, the sender's LAN uplink which connects to its backbone (BB) Internet uplink. Thus, the sender's uplink capacity is the minimum of its LAN and BB link capacities. Similarly, the receiver's downlink consists of the two links, namely, the receiver's Internet backbone downlink which connects to its LAN downlink. Let ul represent uplink capacity, and let dl represent downlink capacity; let s represent the sender and r represent the receiver. The sender's uplink capacity and the receiver's downlink capacities are given by

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

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

If an end network has less backbone transit bandwidth than its BB uplink/downlink bandwidth, then set the BB uplink/downlink bandwidth of this node to this lower transit value.

The flow network representing the above system consists of 2 nodes representing the sender, s, and the receiver, r. There is a single arc from s to r represented by (s, r). The capacity of the arc, $c(s, r)$, is given by the minimum of the sender's uplink, the receiver's downlink, and the backbone transit links between the nodes. The architecture of the Internet has the following redundancy feature: there are several paths between two nodes of the Internet. This architectural feature ensures that a failure or congestion in one region does not make the Internet unavailable. The business and pricing policy of the Internet has the following feature: end network customers pay for peak backbone bandwidth usage, so an end network has access to the BB uplink and BB downlink bandwidth it has paid for. These architectural and business features of the Internet ensure that the transit backbone links are not a bottleneck. Incorporating the architectural and business characteristics of the Internet in the model, gives:

c(s,r) = minimum$\{$ul(s), dl(r)$\}$.

Note that if the architectural characteristics do not hold, then the capacity function c(s,r) represents the upper bound on the end-to-end flow capacity between the sender's network and the receiver's network. That is, the end-to-end flow between s and r cannot exceed this limit. In reality, end networks such as the sender and receiver, pay backbone providers for transit bandwidth on the Internet. Therefore, the end-to-end bandwidth bottleneck between the sender and receiver would equal minimum $\{$LANul(s), BBul(s), LANdl(r), BBdl(r)$\}$ = c(s,r).

For the given input parameters, the flow network is given by G=($\{$s, r$\}$, $\{$(s,r)$\}$) in which arc (s,r) has capacity function c(s,r) as defined above. A flow in G is a function $f : (s, r) \longrightarrow \mathbb{N}_0$ that satisfies the capacity constraint $f(s, r) \leq c(s, r)$. ($\mathbb{N}_0$ refers to $\mathbb{N} \cup \{0\}$.) The largest file that can be transferred from s to r is given by the maximum flow value $\mid f_{max} \mid = c(s, r)$.

Dynamic model: The above model incorporates the physical characteristics (bandwidth), but the model does not incorporate the Internet's workload. The impact of the Internet's workload is directly reflected by bandwidth usage. The Internet traffic at end networks has a predictable diurnal wave distribution [5] 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. The bandwidth available for bulk transmission is the "free" remaining bandwidth. This available bandwidth depends on the time of day. Let τ represent UTC time. The *available* uplink capacity ul and the *available* downlink capacity dl of end networks are a function of τ.

This paper models time in discrete increments; the link capacities are defined for each time instant $\tau = 0, 1, 2, ..., \Gamma - 1$, where Γ is the total number of time instants in a day. The value of Γ depends on the time unit. For example, if time unit is an hour, $\Gamma = 24$ and link capacities are defined for each hour ranging from 0:00 to 24:00; $\tau = 0$ represents UTC [0:00-1:00) and $\tau = 23$ represents UTC [23:00-0:00). If time unit is 5 minutes, $\Gamma = 288$ and link capacities are defined for each 5 minute interval starting from $\tau = 0$ representing UTC [0:00-0:05) to $\tau = 287$ representing UTC [23:55-00:00).

ul(s,τ) = minimum {LANul(s,τ), BBul(s,τ)} $\quad \tau = 0, 1, ..., \Gamma - 1;$

dl(r,τ) = minimum {LANdl(r,τ), BBdl(r,τ)} $\quad \tau = 0, 1, ..., \Gamma - 1.$

For example, suppose time unit is 1 hour: if uplink for s is 10 Gb/s from 03:00- 04:00 UTC, then ul(s,3)=4500 GB.

With the introduction of time, the flow network models flows over time instants t=0,1,2,...,T-1 when flow is permitted. The flow instant t=0 corresponds to UTC time $\tau = \theta$. The dynamic flow network is given by G=({s, r}, {(s,r)}) in which arc (s,r) has capacity function c(s,r,t) defined by:

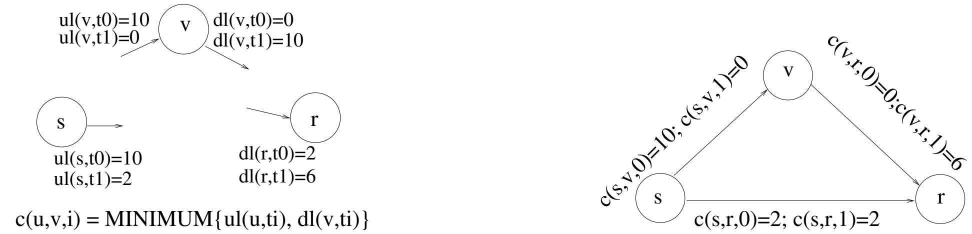
$$c(s, r, t) = \text{minimum}\{\mathsf{ul}(s, \tau), \mathsf{dl}(r, \tau)\} \qquad \tau = \theta + t \quad \text{in modulo } \Gamma \text{ arithmetic.} \tag{1}$$

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

$$\mid f_{max} \mid = \sum_{t=0}^{T-1} c(s, r, t) \tag{2}$$

3 Store-and-Forward

A characteristic of the Internet is that the bandwidth usage at its end networks has a diurnal distribution that depends on time-of-day. Internet bandwidth usage is low during early morning hours which corresponds to maximum available bandwidth. If the sender and receiver are located in different time zones, then their high free capacity times may not synchronize. For example, suppose flow is initiated



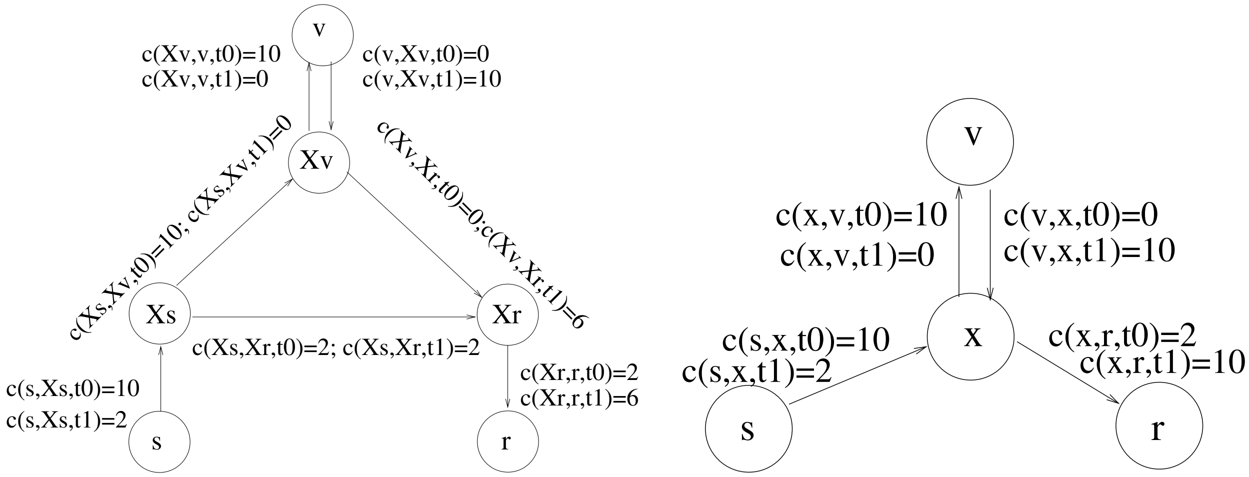
(i) Uplink/Downlink bandwidth

(ii) Complete graph

Figure 1: Flow network that does not incorporate end network bottleneck

from UTC time θ (flow instant t=0) until time $\theta + 1$ (flow instant t=1), and let ul(s,θ)=10, ul(s,θ+1)=2; and dl(r,θ)=2, dl(r,θ+1)=6. The total uplink capacity at s during flow time equals 12, and the total downlink capacity at r during flow time equals 8. Thus, the bottleneck capacity is 8. The maximum flow from s to r during flow time, computed from Equation 2, is $\mid f_{max} \mid= 4$. This value is lower than the bottleneck total capacity of 8. It is possible to increase the maximum flow by using intermediate nodes that have high capacity at the same time as the sender/receiver. Suppose there is a data center, v, with downlink capacity of 10 at t=0 and uplink capacity of 10 at $t = 1$. At time θ, flow can be transmitted from the sender to both the receiver and to the intermediate node v; at time $\theta + 1$, flow can be transmitted from s and v to r. Therefore, it is possible to transmit a flow of 8 units during flow time using a store-and-forward protocol.

Figure 1(i) shows the uplink and downlink capacities of nodes, s, r, and the transit node v. (In the figure, $\theta = t0$ and $\theta + 1 = t1$.) Figure 1(ii) shows the complete graph for store-and-forward flow corresponding to the nodes. For this simple example, it is easy to compute maximal flow from s to r. At flow instant t=0: f(s,r,0)=2; f(s,v,0)=6; At flow instant t=1: f(s,r,1)=2; f(v,r,1)=6. The value of flow $\mid f \mid$ = f(s,r,0)+f(s,r,1)+f(s,v,0)=10 units of flow during flow duration. But the flow value of 10 exceeds the bottleneck capacity 8. The reason that flow exceeds bottleneck capacity is that the complete graph does not model the bottleneck constraints of the end networks. For example, while the uplink from s at time θ equals 10 units, the complete graph models an uplink of 12 units at θ (2 units along (s,r) and 10 units along (s,v)). While the complete graph models the end-to-end capacity between the nodes, it does not model the end network bottleneck constraint.



(i) Bottleneck modeling in complete graph

(ii) Equivalent star graph

Figure 2: Graph modeling Internet flow

4 End network bottlenecks

Figure 2(i) represents the graph modeling bottlenecks corresponding to the example. The bottlenecks are modeled in the complete graph by linking each data center node, v, to an exchange node, x_v. The arcs between v and x_v represent data center uplinks and downlinks, while the arcs between the exchange nodes x_u and x_v represent the end-to-end flow between the data centers u and v. Next, we define the flow network that models the end network bottleneck. For convenience, the paper represents notation such as V-{s} by V-s.

Definition For a given flow duration starting from UTC instant $\tau = \theta$ when t=0 until t=T-1 (UTC instant τ = θ+T-1), the flow network modeling end network bottleneck in a completely interconnected network is given by G=($V \cup X$, E) where

- V is the set of n nodes representing data centers which includes the sender s, the receiver r, and the (n-2) transit nodes;
- X is the set of n nodes representing exchanges such that $\forall v \in V$, $\exists$ $x_v \in X$;
- E is the set of arcs $\{(s, x_s), (x_r, r)\} \cup \{(x_s, x_v) \mid \forall x_v \in$ X-$x_s\} \cup \{(x_v, x_r) \mid \forall x_v \in$ X-$x_r\} \cup$

$\{(x_v, v)(v, x_v) \mid \forall v \in$ V-s-r$\} \cup \{(x_u, x_v) \mid \forall x_u, x_v \in$ X-x_s-$x_r\}$.

The nodes in V have infinite storage capacity, and the nodes in X have zero storage capacity. The arcs in E have bandwidth capacity given by:

$$\begin{aligned} c(v, x_v, t) &= \mathsf{ul}(v, \tau) \qquad\qquad \forall v \in V - r \\ c(x_v, v, t) &= \mathsf{dl}(v, \tau) \qquad\qquad \forall v \in V - s \\ c(x_u, x_v, t) &= \text{minimum}\{\mathsf{ul}(u, \tau), \mathsf{dl}(v, \tau)\} \\ &= \text{minimum}\{c(u, x_u, t), c(x_v, v, t)\} \qquad \forall x_u \in X - x_r,\ \forall x_v \in X - x_s \qquad (3) \end{aligned}$$

$\forall t = 0, 1, ..., T-1$ where τ = θ + t in modulo Γ arithmetic.

The exchange nodes model the border between the end networks and the Internet. Equation 3 states that the exchange arcs in G model **end-to-end constraint** between the end nodes.

A notational convention we use is the following: u, v refer to nodes in set V, x_u, x_v refer to nodes in set X, a, b refer to nodes in either V or X.

A flow in G is a function f: E × [0,T) $\longrightarrow \mathbb{N}_0$ that satisfies the following properties [3]:

Capacity constraint: For all nodes $a, b \in V \cup X,\ t \in$[0,T), $f(a, b, t) \leq c(a, b, t)$.

Skew symmetry: For all (a,b) ∈ E, t ∈[0,T), f(a,b,t) = -f(b,a,t).

Flow conservation:

For all $u \in$ V-s-r, $\displaystyle\sum_{\substack{t=0 \\ f(x_u,u,t)>0}}^{T-1} f(x_u, u, t) = \sum_{\substack{t=0 \\ f(u,x_u,t)>0}}^{T-1} f(u, x_u, t)$

For all $x_u \in X$ and $t \in$ [0,T),

if $f(x_u, u, t) > 0$, then $\displaystyle\sum_{\substack{x_v \\ f(x_v,x_u,t)>0}} f(x_v, x_u, t) = \sum_{\substack{x_v \\ f(x_u,x_v,t)>0}} f(x_u, x_v, t) + f(x_u, u, t)$

if $f(u, x_u, t) > 0$, then $\displaystyle\sum_{\substack{x_v \\ f(x_v,x_u,t)>0}} f(x_v, x_u, t) + f(u, x_u, t) = \sum_{\substack{x_v \\ f(x_u,x_v,t)>0}} f(x_u, x_v, t)$

The flow conservation property states that the positive flow in must equal the positive flow out. The flow conservation constraints for nodes in V reflect the availability of storage at the end nodes. The positive inflow to u at t can be greater than the positive outflow from u at t, but the total positive inflow into u over the flow duration must equal the total positive outflow from u over the flow duration [0,T). Nodes in X have zero storage, so flow conservation for $x_u \in X$ states that the total positive flow into x_u at t must equal the total positive flow out of x_u at t.

For convenience and readability, we use implicit summation notation when functions are summed over time, and when arguments of a function are a set of nodes. For example, the flow conservation equations are written as:

Flow conservation (for positive flows) with implicit summation notation:

For all $u \in$ V-s-r, and positive flows, $\quad f(x_u, u, T) = f(u, x_u, T)$

For all $x_u \in X$, $t \in$ [0,T), and positive flows

$$\begin{aligned} \text{if } f(x_u,u,t) > 0 \text{ then, } f(X,x_u,t) &= f(x_u,X,t) + f(x_u,u,t); \\ \text{if } f(u,x_u,t) > 0 \text{ then, } f(x_u,X,t) &= f(X,x_u,t) + f(u,x_u,t). \end{aligned} \tag{4}$$

When all flows, positive, negative, or zero, are considered, the flow conservation equations are:

For all $u \in$ V-s-r, $\quad f(x_u, u, T) = 0$

For all $x_u \in X$, $t \in$ [0,T), $\quad f(X, x_u, t) + f(u, x_u, t) = 0$

The value $\mid f \mid$ of a flow f is defined as [3]:

$\mid f \mid = \sum_{t=0}^{T-1} f(s, x_s, t) = \sum_{t=0}^{T-1} f(x_r, r, t)$

Using implicit summation notation, the above equation is rewritten as:

$\mid f \mid = f(s, x_s, T) = f(x_r, r, T)$

The next result proves that the maximum flow value of G is at most equal to the end network bottleneck value.

Result 1 $\mid f_{max} \mid \leq$ *minimum*$\{\sum_{\tau=\theta}^{\theta+T-1}ul(s,\tau)$, $\sum_{\tau=\theta}^{\theta+T-1}dl(r,\tau)\}$

Proof: Every flow f in G is at most equal to the minimum of the total sender's uplink and the total receiver's downlink over flow duration. By definition,

$\mid f \mid = f(s, x_s, T) \leq c(s, x_s, T) = \sum_{\tau=\theta}^{\theta+T-1} \mathsf{ul}(s, \tau)$

and

$\mid f \mid = f(x_r, r, T) \leq c(x_r, r, T) = \sum_{\tau=\theta}^{\theta+T-1} \mathsf{dl}(r, \tau)$

The result follows from the above inequalities and the fact that maximum flow is also a flow. □

5 Direct-flow property

The capacity constraint, skew symmetry, and flow conservation are flow properties satisfied by general flow networks. The end-to-end constraint (defined by Equation 3) modeled by exchange arcs is specific to G, the network modeling Internet flow. Here, we prove a flow property, direct-flow property, that is unique to G, the network modeling Internet flow.

The time expanded graph for G, G(T), has one set of nodes and arcs for each flow time instant - G(t=0), G(t=1), ..., G(t=T-1). A node in G(t) is referred to by v or x_v if time t is clear from context; if not, a node is referred to by v(t) or $x_v(t)$. Similarly, an arc is referred to as (a,b,t) if time is not clear from the context. Each sub-graph G(t) is connected to G(t+1) by holdover arcs, (v(t),v(t+1)), between end nodes v(t) ∈ G(t) and v(t+1) ∈ G(t+1), $\forall v \in V$. The holdover arc capacity is infinite to match the storage capacity of end nodes.

Let $p(t)$ represent a simple path from node u to node v in G(t). The path $p(t)$ is a sequence $\langle u, x_u, x_{w1}, x_{w2}, ..., x_{w'}, x_v, v\rangle(t)$. The **direct** path, p*(t) from u to v in G(t) is the sequence $\langle u, x_u, x_v, v\rangle(t)$. The path p_t models end-to-end flow from u to v via an indirect route that covers several exchange nodes, while the path p*(t) models end-to-end flow directly from u to v via the exchange nodes x_u and x_v. The capacity of a path is the minimum of the capacities of all edges that lie along the path. We prove that the capacity of $p(t)$, $c(p, t)$, is at most equal to the capacity of p*(t), c(p*,t).

Lemma 1 $c(p, t) \leq c(p*, t)$

Proof:

$$\begin{aligned} c(p,t) &= \text{minimum}\{c(u,x_u,t), c(x_u,x_{w1},t), c(x_{w1},x_{w2},t), ..., c(x_{w'},x_v), c(x_v,v,t)\} \\ &\leq \text{minimum}\{c(u,x_u,t), c(x_v,v,t)\} \\ &= \text{minimum}\{c(u,x_u,t), c(x_u,x_v,t), c(x_v,v,t)\} \quad \text{from Equation 3} \\ &= c(p*,t) \qquad \square \end{aligned}$$

An *augmenting* path in the time expanded G is a simple path from s at time t' to r at time t" where $t' \leq t''$ and the capacity along each arc is greater than 0. Let p represent an augmenting path from $s(t')$ to $r(t'')$ where $t' \leq t''$ and $t', t'' \in$ [0,T). The augmenting path, p, is a sequence $\langle\langle u, x_u, x_{w1}, x_{w2}, ..., x_{v1}, v1\rangle(\mathbf{t'}), \langle v1, x_{v1}, ..., x_{v2}, v2\rangle(\mathbf{t1}), \langle v2, ..., \rangle(\mathbf{t2}), ..., \langle ..., x_r, r\rangle(\mathbf{t''})\rangle$ where $t' < t1 < t2 < ... < t''$. Thus, an augmenting path p is a sequence of sub-paths $\langle p(t'), (v1(t'), v1(t1)), p(t1), (v2(t1), v2(t2)), p(t2), ..., p(t'')\rangle$ with holdover arcs linking sub-paths. If t' = t", then p consists of one sub-path p(t'). For notational convenience, the path notation allows holdover arc representation to span more than one flow time unit. For example, an arc (v(t),v(t+3)) represents 3 linked arcs, namely, (v(t),v(t+1)), (v(t+1),v(t+2)), (v(t+2),v(t+3)). Mapping to the bulk transmission application, a holdover arc spanning more than one time unit represents en-route storage of the file segment in a transit node v. Let $p*$ represent the augmenting path from $s(t')$ to $r(t'')$ where each subpath $p(t)$ is replaced by the direct subpath p*(t). The next result follows from Lemma 1.

Result 2 $c(p) \leq c(p*)$

With reference to the bulk transmission application, the augmenting path represents a transmission route for a file segment, where the route starts from s at flow instant t' and ends at r at t"; enroute at time instants t ($t' < t < t''$), the segment may be stored at transit data centers.

Let f_p represent a flow along augmenting path p, $f_p : E \times [0, T) \longrightarrow \mathbb{N}_0$ defined by

$$f_p(a, b, t) = \begin{cases} c(p) & \text{if (a,b,t) on p} \\ -c(p) & \text{if (b,a,t) on p} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Thus, $\mid f_p \mid = c(p)$.

Let f_{p*} represent a flow along p* in G, $f_{p*} : E \times [0, T) \longrightarrow \mathbb{N}_0$ defined by

$$f_{p*}(a, b, t) = \begin{cases} c(p) & \text{if (a,b,t) on p*} \\ -c(p) & \text{if (b,a,t) on p*} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

From Result 2, f_{p*} is a valid flow along path p* with $\mid f_{p*} \mid = \mid f_p \mid$. By construction,

$$\begin{aligned} f_{p*}(u, x_u, t) &= f_p(u, x_u, t) \quad \forall u \in V \\ f_{p*}(x_u, u, t) &= f_p(x_u, u, t) \quad \forall u \in V \end{aligned} \tag{7}$$

5.1 Residual networks G_f and G_{f*}

Let $G_f = (V \cup X, E_f)$ represent the residual network of G induced by f_p, where

$E_f = \{(a, b) \mid a, b \in V \cup X, c(a, b, t) - f_p(a, b, t) > 0, \forall t \in [0, T)\}$

Similarly, define $G_{f*} = (V \cup X, E_{f*})$, the residual network of G induced by f_{p*}, where

$E_{f*} = \{(a, b) \mid a, b \in V \cup X, c(a, b, t) - f_{p*}(a, b, t) > 0, \forall t \in [0, T)\}$.

The graph G and its residual networks G_f, G_f* have the same node set. In order to distinguish between the three graphs, we refer to parameters in G_f and G_{f*} by the subscript f and $f*$, respectively. For example $c_f(a, b, t)$ represents the capacity of arc (a,b,t) of residual network G_f. From Equation 7, it follows that:

$$c_{f*}(u, x_u, t) = c_f(u, x_u, t) \qquad \forall u \in V, t \in [0, T)$$
$$c_{f*}(x_u, u, t) = c_f(x_u, u, t) \qquad \forall u \in V t \in [0, T) \tag{8}$$

Equation 8 shows the relationship between the residual networks G_f and G_{f*}. The capacity of the arcs linking u to x_u, $\forall u \in V$ of graph G_f, is equal to the capacity of the corresponding arcs in G_{f*}. Next, we prove that the residual graph G_{f*} satisfies the end-to-end constraint, namely, the capacity of an exchange arc (x_u, x_v, t) is equal to the minimum of the capacities of the uplink arc from u and the downlink arc to v (Equation 3).

Result 3 *The exchange arcs in the residual network G_{f*} satisfy end-to-end constraint:*

$c_{f*}(x_u, x_v, t) =$ *minimum*$\{c_{f*}(u, x_u, t), c_{f*}(x_v, v, t)\} \quad \forall x_u, x_v \in X, \;\; \forall t = 0, 1, ..., T - 1.$

Proof: If $\langle u, x_u, x_v, v\rangle$(t) is not a sub-path in $p*$, then the result holds since arcs in G satisfy Equation 3.

Let $\langle u, x_u, x_v, v\rangle$(t) be a sub-path in $p*$. Then, by definition of residual network capacities:

$$\begin{aligned} c_{f*}(x_u, x_v, t) &= c(x_u, x_v, t) - f_{p*}(x_u, x_v, t) \\ &= \text{minimum}\{c(u, x_u, t), c(x_v, v, t)\} - f_{p*}(x_u, x_v, t) \quad \text{by Equation 3} \\ &= c(u, x_u, t) - f_{p*}(x_u, x_v, t) \quad \text{W.L.O.G., let } c(u, x_u, t) = \text{min}\{c(u, x_u, t), c(x_v, v, t)\} \\ &= c_{f*}(u, x_u, t) \\ &= \text{minimum}\{c_{f*}(u, x_u, t), c_{f*}(x_v, v, t)\} \qquad \Box \end{aligned}$$

We use the above result to prove the following theorem about Internet flow network G.

Theorem 1 *For any flow f in G, there exists an equivalent flow f* in G with* $\mid f \mid = \mid f* \mid$, *where f* is the sum of flows along direct augmenting paths: f** = $f_{p1*} + f_{p2*} + ... + f_{p*}$

Proof: First, consider flow along an augmenting path, p, in G. Equation 6 defines a flow f_{p*} along direct path p* that is equivalent to flow f_p defined by Equation 5.

Next, consider the residual networks, G_f due to flow f_p in G and G_{f*} due to flow f_{p*} in G.

Every augmenting path in G_f and G_{f*} consists of a sequence of sub-paths of the form $\langle u, x_u, ..., x_v, v\rangle$(t). Therefore the capacity of a sub-path in both residual graphs is at most equal to the minimum of capacity of uplink/downlink arcs (u, x_u, t) and (x_v, v, t).

Given any path p in G_f, consider the corresponding path p* in G_{f*}.

$$
\begin{aligned}
c_f(p,t) &= \text{minimum}\{c_f(u,x_u,t), c_f(x_u,x_{w1},t), c_f(x_{w1},x_{w2},t), ..., c_f(x_{w'},x_v), c_f(x_v,v,t)\} \\
&\leq \text{minimum}\{c_f(u,x_u,t), c_f(x_v,v,t)\} \\
&= \text{minimum}\{c_{f*}(u,x_u,t), c_{f*}(x_v,v,t)\} \quad \text{from Equation 8} \\
&= \text{minimum}\{c_{f*}(u,x_u,t), c_{f*}(x_u,x_v,t), c_{f*}(x_v,v,t)\} \quad \text{from Result 3} \\
&= c_{f*}(p*,t)
\end{aligned}
$$

Thus, $c_f(p) \leq c_{f*}(p*)$. Hence, in each iteration of Ford-Fulkerson's method, a flow f_p in G_f can be mapped to an equivalent flow f_{p*} in G_{f*}.

Algorithm 1: FORD-FULKERSON-METHOD for Internet Flow along direct augmenting paths

```
copy time expanded G to $G_f$, $G_{f*}$
initialize flows f, f* to 0
while there exists an augmenting path p in residual network $G_f$ do
    augment flow f by | $f_p$ | along p in $G_f$
    augment flow f* by | $f_p$ | along p* in $G_{f*}$
    update the residual capacities of edges in $G_f$ and $G_f*$ to reflect flow
return f*
```

The output of the algorithm is a flow that is the sum of flows along direct augmenting paths. □

Since maximum flow is also a flow, Theorem 1 states that maximum flow can be mapped to a maximum flow that is a sum of augmenting flows along direct paths.

Since every flow in G can be mapped to a direct flow, let f be a direct flow. The flow conservation equations (Section 4, Equation 9) states that the sum of positive in-flows to node x_u equals the sum of positive out-flows from x_u. Since f is a direct flow, all positive in-flows to node x_u from exchange nodes $x_v, \forall x_v \in X$, will be directed as an out-flow from x_u to u. This leads to the following corollary:

Corollary 1 *For all $x_u \in X$ and $t \in [0, T)$,*

$$f(x_u, u, t) = \sum_{\substack{x_v \\ f(x_v, x_u, t) > 0}} f(x_v, x_u, t) \quad when\ f(x_u, u, t) > 0$$

From skew symmetry, there is positive flow either along (u, x_u) or (x_u, u), but not both. (If there is physical flow in both direction, cancellation allows this to be represented by positive flow in at most one direction.) Thus, if $f(u, x_u, t) > 0$ then $f(x_u, u, t) < 0$ and from Corollary 1, it follows that all in-flow to x_u from exchange nodes x_v, $\forall x_v \in X$ must be less than or equal to 0. This leads to the following:

Corollary 2 *For all $x_u \in X$ and t ∈ [0,T),*

$$f(u, x_u, t) = \sum_{\substack{x_v \\ f(x_v, x_u, t) > 0}} f(x_u, x_v, t) \quad when\ f(u, x_u, t) > 0$$

Corollaries 1 and 2 are stronger than flow conservation equations for flow networks, and holds for Internet flow graph G with direct flow f. We refer to the property as direct flow property. Using implicit summation notation, the corollaries are written as:

Direct flow property: For all positive flows, and $x_u \in X$, $t \in [0, T)$, the direct flow f satisfies:

$$\begin{aligned} f(x_u, u, t) &= f(X, x_u, t); \\ f(u, x_u, t) &= f(x_u, X, t) \end{aligned}$$

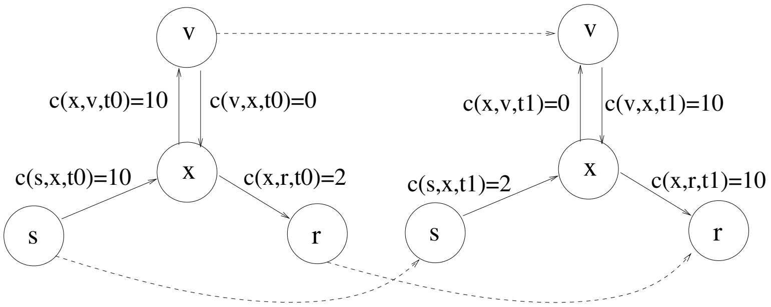


Figure 3: Time expanded star graph modeling Internet flow

6 Star model

We show that the bottleneck graph G=(V∪X,E) is equivalent to a star graph, $G^{\star}$=S_n=(V∪{x},$E^{\star}$). The star graph $G^{\star}$ is generated from G by replacing the complete sub-graph consisting of node set X and edges (x_u, x_v), $\forall x_u, x_v \in X$, by a single node x. That is, a single node x replaces sub-graph (X, $\{(x_s, x_v)|\forall x_v \in$ X-$x_s\} \cup \{(x_v, x_r)|\forall x_v \in$ X-$x_r\} \cup \{(x_u, x_v) \mid \forall x_u, x_v \in$ X-x_s-$x_r\}$) of G. Each edge (v,x_v) of G is replaced by edge (v,x) in $G^{\star}$. This leads to a star graph where the n data center nodes are leaves and the exchange node x is the internal hub. Figure 2(ii) presents the star model corresponding to Figure 2(i). The nodes s, r, v are leaves while x is an interior node. The capacity of arc (v,x) is equal to the uplink capacity of v, and the capacity of (x,v) is equal to the downlink capacity if v. Figure 3 shows the time expanded star graph; there is a copy of the node set and the arc set for each flow instant.

Definition For a given flow duration starting from UTC instant $\tau = \theta$ when t=0 until t=T-1 (UTC instant τ = θ+T-1), the Internet flow network is given by $G^{\star}$=(V∪x, $E^{\star}$) where

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

The nodes in V have infinite storage capacity, and the node x has zero storage capacity. The arcs in $E^\star$ have bandwidth capacity given by:

$c(v,x,t) = \mathsf{ul}(v,t) \quad \forall (v,x) \in E^\star$

$c(x,v,t) = \mathsf{dl}(v,t) \quad \forall (x,v) \in E^\star$

$\forall t = 0, 1, ..., T-1$ where τ = θ + t in modulo Γ arithmetic.

A flow in $G^\star$ is a function f: $E^\star$ $\times$ [0,T) $\longrightarrow$ $\mathbb{N}_0$ that satisfies the following flow properties:

Capacity constraint:

For all nodes $v \in V,\ t$ ∈[0,T), $\ f(v,x,t) \leq c(v,x,t)$, and $f(x,v,t) \leq c(x,v,t)$.

Skew symmetry: For all (u,x) ∈ $E^\star$, t ∈[0,T), f(u,x,t) = -f(x,u,t).

Flow conservation for positive flows:

For all $v \in$ V-s-r, $\quad f(x,v,T) = f(v,x,T)$

For all $t \in$ [0,T), $\ f(V,x,t) = f(x,V,t)$

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

The value $\mid f \mid$ of a flow f in $G^\star$ is defined as

$\mid f \mid = f(s,x,T) = f(x,r,T)$

We prove that graph G=(V∪X,E) is equivalent to $G^\star$=(V∪x,$E^\star$). The proof outline is as follows: for every flow in G, we construct a flow in $G^\star$ with the same value as the flow in G. Similarly, for every flow in $G^\star$, we construct a flow in G with the same value as the flow in $G^\star$. Since maximum flow is also a flow, both G and $G^\star$ are equivalent.

Theorem 2 *The graph G=(V∪X,E) is equivalent to $G^\star$=(V∪x,$E^\star$)*

Proof: Let f be a direct flow in G. (By Theorem 1, every flow in G can be mapped to a direct flow.) Construct $f^\star$ in $G^\star$ as follows:

$f^\star(x,v,t) = f(x_v,v,t) \qquad \forall v \in V, x_v \in X$

$f^\star(v,x,t) = f(v,x_v,t) \qquad \forall v \in V, x_v \in X$

We prove that $f^\star$ satisfies all the required properties of a flow.

capacity constraint: $f^\star$(v,x,t) = f(v,x_v,t) $\leq$ c(v,x_v,t) = ul(v,θ+t) = c(v,x,t). By similar argument, $f^\star$(x,v,t) $\leq$ c(x,v,t).

skew symmetry: $f^\star(v,x,t) = f(v,x_v,t) = -f(x_v,v,t) = -f^\star(x,v,t)$

flow conservation for v: For positive flows,

$f^\star$(x,v,T) = f(x_v,v,T) = f(v,x_v,T) = $f^\star$(v,x,T)

flow conservation for x: For positive flows,

$$\begin{aligned}
f^\star(V,x,t) &= \sum_{v \in V} f(v,x_v,t) \\
&= \sum_{x_v \in X} \sum_{x_u \in X} f(x_v,x_u,t) \quad \text{by direct flow property} \\
&= \sum_{x_v \in X} f(x_v,X,t) \\
&= \sum_{x_v \in X} f(X,x_v,t) \quad \text{by rearranging the terms} \\
&= \sum_{x_v \in X} f(x_v,v,t) \quad \text{by direct flow property} \\
&= \sum_{v \in V} f^\star(x,v,t) \\
&= f^\star(x,V,t)
\end{aligned}$$

Thus, $f^\star$ is valid flow in $G^\star$. The proof showing that values of the flow for f and $f^\star$ are equal is trivial. To prove that for every flow in $G^\star$, there is an equivalent flow in G. Theorem 1 shows that a direct flow can be constructed for every flow in G. For $f^\star$ in $G^\star$, construct a direct flow in G with the same value as $f^\star$ using Algorithm 1 where G_f is replaced by $G^\star_f$. □.

7 Conclusions

The Internet is a completely connected network and there are several paths between any two nodes. The Internet flow network is the underlying model for routing algorithms of flow applications such as bulk transmission. The complexity of routing algorithms is dependent on the search space of the underlying model. The contribution of this paper is showing that the Internet flow network can be modeled by a star graph. The inputs to the model are the uplink and downlink capacities of the end networks. The star graph has a lower state complexity than the complete graph, thereby lowering the complexity of the overlaying routing algorithm.

The simpler star model makes it easier to look at other maximum flow problems of interest related to bulk transmission. Some of these problems are:

1. find the maximum flow when the transmission start time and end time are given;
2. find the transmission start time that results in the maximum flow for a given time duration;
3. find optimal (time zone) locations for intermediate data centers that result in maximal flow;
4. for a given set of intermediate data centers, find the minimum bandwidth and storage capacity to ensure maximum flow from sender to server;
5. for a given file size, find the start time that results in the quickest transmission; and
6. for a given file size and transmission completion time, find the latest start time.

Sometimes, the only invariants in bulk transmission are the sender and receiver, 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 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. The star graph reduces the complexity of this process by an order of magnitude.

References

[1] Eli Brosh, Salman Abdul Baset, Dan Rubenstein, and Henning Schulzrinne. The delay-friendliness of tcp. In *2008 ACM SIGMETRICS*, pages 49–60. ACM, 2008.

[2] Parminder Chhabra, Vijay Erramilli, Nikolaos Laoutaris, Ravi Sundaram, and Pablo Rodriguez. Algorithms for constrained bulk-transfer of delay-tolerant data. In *ICC'10*, pages 1–5, 2010.

[3] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, 1 edition, June 1990.

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

[5] Anukool Lakhina, Konstantina Papagiannaki, Mark Crovella, Christophe Diot, Eric D. Kolaczyk, and Nina Taft. Structural analysis of network traffic flows. In *SIGMETRICS'04*, pages 61–72, 2004.

[6] Nikolaos Laoutaris, Michael Sirivianos, Xiaoyuan Yang, and Pablo Rodriguez. Inter-datacenter bulk transfers with netstitcher. In *Proceedings of the ACM SIGCOMM 2011*.

[7] Nikolaos Laoutaris, Georgios Smaragdakis, Pablo Rodriguez, and Ravi Sundaram. Delay tolerant bulk data transfers on the internet. In *SIGMETRICS*, pages 229–238, 2009.

[8] Jiayin Qi, Hongli Zhang, Zhenzhou Ji, and Liu Yun. Analyzing bittorrent traffic across large network. In *Cyberworlds*, pages 759 –764, 2008.

[9] Cong Shi, Mostafa H. Ammar, and Ellen W. Zegura. idtt: Delay tolerant data transfer for p2p file sharing systems. In *GLOBECOM'11*, pages 1–5, 2011.

[10] Arun Venkataramani, Ravi Kokku, and Michael Dahlin. Tcp nice: A mechanism for background transfers. In *OSDI'02*, pages –1–1, 2002.