Network Information Flow in Network of Queues

Phillipa Gill[†], Zongpeng Li[†], Anirban Mahanti[‡], Jingxiang Luo[†], Carey Williamson[†]

***Abstract*— Two classic categories of models exist for computer networks: network information flow and network of queues. The network information flow model appropriately captures the multi-hop flow routing nature in general network topologies, as well as encodable and replicable properties of information flows. However, it assumes nodes in the network to be infinitely powerful and therefore does not accurately model queueing delay and loss at nodes. The network of queue model instead focuses on finite capacitied nodes and studies buffering and loss behaviors from a stochastic perspective. However, existing models on network of queues are mostly based on unrealistically simple topologies, and lacks the multi-hop flow routing dimension. In this work, we seek to combine advantages from both models. We start with the network information flow model and replace each infinitely powerful node with a finitely capacitied queue system instead. We show that the optimal routing problems for unicast, multiple unicasts and multicast can all be formulated as convex optimization problems. As a necessary step in validating the model for multicast routing, we show that network coding does not change the memoryless nature of traffic. We examine the correctness of the models through simulations and show that they behave differently than traditional link-cost based network flow models.**

I. INTRODUCTION

The routing of information flows with performance objectives has been an important direction in networking research. Depending on the application, performance objectives may include high throughput(e.g. [11]), or quality of service objectives such as loss and delay (e.g. [13], [17]). In this paper, we develop a new model for routing information flows within a computer network. Our motivation originates in the observation that existing models, including network information flow and network of queues, each has its advantages and disadvantages, as explained below. We wish to combine attractive features from both models to better represent a computer network.

The classic network flow problem [2] studies the transmission of commodity flows across a transportation network. The goal is to maximize the end-to-end flow rate or to minimize the transmission cost incurred while sustaining a certain target throughput. The routing is performed in a graph representing the network topology and is multi-hop by nature. Constraints in optimizing the routing include flow conservation at relay nodes and capacity limits at links [2], [3]. In computer network research, the network flow model has been adopted for modelling unicast routing [8], [12], where data streams flow from one sender to one receiver. The structure of multicast routing, where data flows from one sender to multiple receivers, is not a simple network flow anymore, since information may be replicated and flow conservation does not hold [6], [12]. Nonetheless, a recent break through in information theory [1], [9] suggests that, by further considering the encodable property of information flows, multicast routing can be modelled as a union of *conceptual* network flows [11], [14].

The network flow based routing model in computer networks successfully characterizes the general graph topology of a network, the multi-hop flow routing nature in data transmission, as well as the unique encodable and replicable properties of information flows (in the case of multicast). However, network flow based routing essentially ignores the limited capabilities of nodes (end-hosts, routers and switches) in the network, and assume that they are infinitely powerful. While such assumptions may be acceptable in commodity networks, where flows simply pass through joints of pipes, it is less desirable in communication networks. Information flows passing through a node in a computer network need to be buffered, switched [10] and processed [1]; such node processing, along with the associated queueing delay and packet loss, are critical features to consider in modelling a computer network. In this paper, we leverage classic modelling techniques in queueing theory to complement the network flow model, so that both multihop-routing in a graph and finite node processing power are appropriately accounted for.

Queuing theory provides a framework for modelling limited capacities at network nodes using standard queuing models such as the $M/M/1$ or $M/M/1/k$ systems. These queuing models provide methods for evaluating the average behaviors of a node based on the amount of workload it is subjected to. Of particular interest to this study is the average delay at a node; we plan to study effects of packet loss and retransmissions in future work. Queuing models of nodes can be extended to networks of many nodes (modelled as queues) using networks of queues models such as Gordon-Newell networks or Jackson networks [4]. Although these networks of queues exist to generalize single queuing models into networks of many queues, it is seldom that networks of queues are modelled using general network topologies [7], [16]. In this paper, we apply networks of queues to general network topologies in order to perform optimal multihop routing of information flows. As we will show in Sec. VII, this is different than traditional link-cost based network flow models.

This work combines the concept of network flows with networks of queues to develop new optimization models for routing network flows in general topologies. Our network of

[†] Dept. of Computer Science University of Calgary, Canada. {psessini, zongpeng, jxluo, carey}@cpsc.ucalgary.ca

[‡] Department of Computer Science and Engineering, Indian Institute of Technology, Delhi, mahanti@cse.iitd.ernet.in

queues model contrasts most network information flow studies by considering nodes with finite processing capacities. These finite processing capacities can have implications for the performance of network flows (especially in terms of delay) and are important to consider when determining optimal routes for network flows. We use results from queuing theory to determine optimal routes for network flows that take into account limited processing capacity of nodes. To the best of our knowledge, this work represents the first attempt in designing a computer network model that captures both of the following features: (a) multi-hop routing of information flows in general graphs, and (b) finite node capacities and the associated queueing delays.

The main purpose of this work is to bring to the attention of the networking community the above new model for computer networks. As a proof of concept, we develop mathematical programs that compute optimal routes for network flows based on minimizing average delay for the flows. Mathematical models are developed for three types of network flows, single unicast, multiple unicast and multicast. For each model, the convexity of the mathematical program is established and numerical results are presented. Additionally, we prove that the output of a node after coding together two network flows will produce an output process that follows a Poisson distribution provided that the input flows also follow Poisson distributions.

The rest of the paper is structured as follows. We describe the model of network nodes and the network model in Section II. In Sections III and IV we present our models for optimal routing of single and multiple unicast flows in a network of queues. Optimal routing of multicast flows is considered in Section V. We present numerical results of our models in Section VI. Finally, conclusions are discussed in Section VIII.

II. NETWORK MODEL AND PRELIMINARIES

A salient feature of our work is modelling network information flows with limited node capacities. This is accomplished by considering each node as an $M/M/1$ queuing system. Queuing occurs when the incoming flow rate temporarily exceeds the node processing capacity and packets have to wait in the node buffer. Node processing here is an abstraction of real-world packet processing that includes, for example, extracting destination address from the packet header, routing table look-up, input-output port switching [10], and encoding/decoding over a finite field in the case of multicast [1].

The incoming flow to a node is assumed to follow a Poisson distribution. It is a well-known fact that if incoming traffic to an $M/M/1$ system is Poisson/memoryless, then so is the departing traffic. The special case where synchronization for network coding is necessary will be analyzed separately. No flows are prioritized at nodes and all incoming packets to a node u will enter the same queue. The output flow rate of a node u is denoted by, $f_{out}(u)$.

It is assumed that processing time per bit at each node is constant and packet sizes are exponentially distributed. Thus, the per packet service time can be modelled using exponential distribution with a mean of $\frac{1}{\mu_u}$, where μ_u denotes the processing power of node u. Using results on the average delay of an $M/M/1$ queue, we have the following function for delay on a node u:

$$\tau(u) = \frac{1}{\mu_u - f_{in}(u)}$$

Here $\tau(u)$ includes both waiting time in the buffer and processing time. Since our expression for $\tau(u)$ is derived based on steady state queuing behavior, we require that the each node be an ergodic queuing system. This means that there must exist a steady state solution for the queuing model on each node. This can be guaranteed by assuming that $f_{in}(u) < \mu_u$. We are able to generalize this single node model to an entire graph by considering it as a Gordon-Newell network [4].

III. SINGLE UNICAST

We study the new network model, with both multi-hop routing and node queueing elements, from the perspective of optimal routing algorithms. We start from the simple case of one unicast only, and generalize to multiple unicast sessions and multicast. In each case, we show that the problem of minimum-delay routing can be formulated into a convex optimization problem within the new model.

A. Mathematical Program

We now develop a mathematical program for optimal routing of a single unicast flow with respect to average weighted delay. The mathematical program takes as input a directed graph $G = (V, E)$, where each edge $\overrightarrow{uv} \in E$ has a fixed capacity denoted by $C(\overrightarrow{uv})$. The aggregated incoming flow for a node u can be described as:

$$f_{in}(u) = \sum_{v \in N(u)\uparrow} f(\overrightarrow{vu}) \qquad \forall u \in V, \tag{1}$$

where $N(u)\uparrow$ is the set of upstream neighbours of u, and $f(\overrightarrow{vu})$ is the amount of flow on edge $\overrightarrow{vu}$.

The mathematical program to minimize delay for a unicast session with sender s and receiver t is presented below in (2). The average delay per node is weighted by the amount of flow that is directed to a given node (henceforth referred to as *weighted delay*). Such a weighted delay model implies a fair treatment of routing algorithms that split the total throughput into multiple paths.

SINGLE UNICAST MODEL

Minimize

$$\sum_{u \in V} f_{in}(u)\tau(u) \tag{2}$$

Subject to:

$$\begin{cases} \tau(u) = \frac{1}{\mu_u - f_{in}(u)} & \forall u \in V & 2a. \\ f_{in}(u) < \mu_u & \forall u \in V & 2b. \\ f(\overrightarrow{ts}) = xput & & 2c. \\ f_{in}(u) = f_{out}(u) & \forall u \in V & 2d. \\ f(\overrightarrow{uv}) \leq C(\overrightarrow{uv}) & \forall \overrightarrow{uv} \in E & 2e. \end{cases}$$

$\tau(u), f(\overrightarrow{uv}) \geq 0 \qquad \forall u, \forall\ \overrightarrow{uv}$

The first constraint in the Single Unicast model is on the average delay of nodes in the network, defined by the steady-state behavior of an $M/M/1$ queueing system. Constraint $2b$ requires that arrivals to a node do not exceed its processing capabilities. This constraint is required in order to ensure the system is ergodic. The third constraint places a virtual link between the receiver and the sender with flow rate equal to the target throughput of the flow, denoted as $xput$. This constraint is required for the model to satisfy the flow balance constraints given in constraint $2d$. Finally, constraint $2e$ ensures the flow along an edge $\overrightarrow{uv}$ does not exceed its capacity.

Theorem 1: The Single Unicast Model is a convex optimization problem.
Proof of Theorem 1:
We essentially need to check that both the feasibility region and the objective function in the model are convex. First, consider the convexity of the constraints in the Single Unicast Model. Clearly, constraints $2b$-$2e$ are convex since they are linear. Since constraint $2a$ can be substituted in to the objective function we consider them together. It can be verified that each term in the second derivative of the objective function is,

$$(f_{in}(u)\tau(u))'' = \left(\frac{f_{in}(u)}{\mu_u - f_{in}(u)}\right)'' = \frac{2\mu_u}{(\mu_u - f_{in}(u))^3}$$

which is positive when $f_{in}(u) < \mu_u$. Thus the Single Unicast Model is a convex optimization problem and can be solved using convex programming methods. □

We have solved this convex program using the interior-point algorithm, as implemented in CVX [5], a modelling and solution tool for convex programming in Matlab. This is demonstrated in Section VI, where we present numerical results of our models.

IV. MULTIPLE UNICAST

In this section we extend the Single Unicast Model to support s concurrent unicast sessions. To accomplish this $f_{in}(u)$ is redefined as follows:

$$f_{in}(u) = \sum_{i=1}^{s} f_{in}^i(u) \qquad \forall u \in V \tag{3}$$

where

$$f_{in}^i(u) = \sum_{v \in N(u)} f^i(\overrightarrow{vu}) \qquad \forall i \in [1..s]. \tag{4}$$

Here $f^i(\overrightarrow{vu})$ denotes the flow rate on link $\overrightarrow{vu}$ of session i. The mathematical formulation for multiple unicast strives to minimize the weighted delay for each flow where each flow is also given a weight, w^i. The Multiple Unicast Model for s concurrent unicast flows is presented in Equation 5.

MULTIPLE UNICAST MODEL

Minimize

$$\sum_{i=1}^{s} d^i w^i \tag{5}$$

Subject to:

$$\begin{cases} d^i = \sum_{u \in V} f_{in}^i(u)\tau(u) & \forall i \in [1..s] & 5a. \\ \tau(u) = \frac{1}{\mu_u - f_{in}(u)} & \forall u \in V & 5b. \\ f(\overrightarrow{t_i s_i}) = xput_i & \forall i \in [1..s] & 5c. \\ f_{in}^i(u) = f_{out}^i(u) & \forall u \in V,\ \ i \in [1..s] & 5d. \\ \sum_{i=1}^{s} f^i(\overrightarrow{uv}) \leq C(\overrightarrow{uv}) & \forall\ \overrightarrow{uv} \in E & 5e. \end{cases}$$

$\tau(u), f^i(\overrightarrow{uv}) \geq 0 \qquad \forall i, \forall u, \forall\ \overrightarrow{uv}$

The first constraint of the Multiple Unicast Model defines the weighted delay for each flow i. The weight depends on f_{in}^i while the node delay depends on f_{in}. Similar to the Single Unicast Model, constraint $5b$ defines the delay at each node as a function of the incoming flow to the node, $f_{in}(u)$, and the capacity of the node, μ_u. Next, in the third constraint, virtual links are placed between the receiver and sender for each flow i. Finally, the last two constraints ensure flow balance as well as ensuring that flow along an edge $\overrightarrow{uv}$ will not exceed its capacity. The convexity of the multiple unicast model can be numerically verified, and is confirmed by CVX . Below we provide a proof on the case of uniform session weights.

Theorem 2: The Multiple Unicast Model is a convex optimization problem when $w^i = w^j\ \ \forall i, j \leq s$.
Proof of Theorem 2:
Clearly, constraints $5c$-e are convex since they are linear. Next, we turn our attention to the objective function and constraints $5a$ and $5b$. Consider $w^i = K$, $\forall i \in [1..s]$ where K is a constant.

$$h = \sum_{i=1}^{s} d^i w^i = K \sum_{i=1}^{s} d^i \tag{6}$$

By the first constraint:

$$d^i = \sum_{u \in V} f_{in}^i(u)\tau(u) \qquad \forall \quad i \in [1..s]$$

Substituting d^i into equation 6 gives the following:

$$K \sum_{i=1}^{s} \left(\sum_{u \in V} f_{in}^i(u)\tau(u) \right) = K \sum_{u \in V} \left(\sum_{i=1}^{s} f_{in}(u)^i \tau(u) \right) \tag{7}$$

Using constraint 2 it follows that $\tau(u) = 1/(\mu_u - f_{in}(u))$. Substituting $\tau(u)$ into equation 7 we get the following:

$$K \sum_{u \in V} \left(\sum_{i=1}^{s} f_{in}^i \tau(u) \right) = K \sum_{u \in V} \left(\sum_{i=1}^{s} \frac{f_{in}^i(u)}{\mu_u - f_{in}(u)} \right)$$

By equation 3 this is equivalent to,

$$K \sum_{u \in V} \left(\frac{1}{\mu_u - f_{in}(u)} \left(\sum_{i=1}^{s} f_{in}^i(u) \right) \right) = K \sum_{u \in V} \frac{f_{in}(u)}{\mu_u - f_{in}(u)}$$

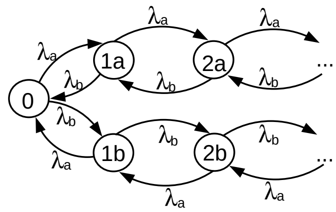

Fig. 1. Markov chain representing the amount of unmatched packets from flow a and flow b at the coding node u.

$$h = K \sum_{u \in V} f_{in}(u)\tau(u)$$

This is equivalent to the Single Unicast objective function which we have already shown to be convex. □

V. Single Multicast

A. Preliminaries

It has been shown in previous work that network coding facilitates optimal multicast algorithm design [1], [9], [12]. We consider a model for multicast that incorporates network coding while minimizing the overall multicast flow delay. Since the network is modelled as a network of queues, we first examine whether or not this model is applicable when network coding is considered, *i.e.*, whether traffic flows remain memoryless after processing due to network coding.

In network coding, nodes in a network may encode two network flows together and forward the encoded packet to the next hop in the route. Encoding requires that at least one packet from each flow to be encoded is present at the encoding node. Otherwise synchronization of the flows is required at the encoding node, before the encoded packet can leave the node. For the network of queues model to be applied, it is necessary that the output flows of all nodes still follow a Poisson distribution. We now prove that the output process of an encoding node will indeed converge to Poisson, hence making the network of queues model applicable to coded multicast.

Theorem 3: The process generated by coding together multiple flows following Poisson arrivals will also converge to a Poisson process.

Proof of Theorem 3:

For a network of queues model to be applied to multicast routing with network coding, it is necessary to examine the output process of a node after multiple network flows (following Poisson arrivals) are synchronized and coded together. For simplicity we consider the case where two flows are coded together, however our proof can be generalized to the case where more than two flows are coded together.

Consider a node u with two incoming data flows a and b with arrival rates λ_a and λ_b, respectively. Node u will encode these flows together and output the encoded flow, $e(a, b)$. The state of node u can be represented by the Markov chain shown in Figure 1. In the Markov chain, each state keeps track of how many packets from flow a or flow b are waiting to be coded at node u. At state ia, we have i unmatched packets of flow a, waiting for packets from flow b to arrive and then coded together. If the system stays at any state except state 0, the inter-packet times will follow the same distribution in the departing flow as in the non-backlogged flow.

We now consider 3 possible scenarios.

- $\lambda_a < \lambda_b$. Consider a cut between any pair of neighboring states in the Markov chain when $\lambda_a < \lambda_b$. In this case, flows (probability transitions) across the cut will not be balanced. As a consequence, the system moves toward states that are in the bottom branch and continues to progress down the bottom branch as the probability of moving further down the branch is greater than the probability of moving back toward state 0. Eventually the probability for the system to be at state 0 at steady-state is 0. As a result, the output process of node u will be a Poisson distribution with $\lambda = \lambda_a$. The flow with the lower arrival rate limits the amount of coded packets that the node may output.
- $\lambda_b < \lambda_a$. By similar arguments as in the case above, the output flow will eventually approach a Poisson flow with rate λ_b.
- $\lambda_a = \lambda_b$. For the situation we have a symmetric random walk on a line. From the classical results on random walks, we know that as sufficiently long time elapses, the probability of being at any state will tend to 0. In particular, the system is "almost surely" not at state 0. Following the argument before, the system's output process will converge to Poisson process, with rate $\lambda = \lambda_a = \lambda_b$.

Above we have shown that regardless of the relationship between the rates of the flows to be coded, the output of node u will follow a Poisson process. Thus the network of queues model may be applied to coded multicast. □

B. Mathematical Model

We now extend our previous models to support a multicast session with k receivers. The formulation of the model is presented in Equation 8.

Single Multicast

Minimize

$$\sum_{u \in V} f_{in}(u)\tau(u) \tag{8}$$

Subject to:

$$\begin{cases} \tau(u) = \frac{1}{\mu_u - f_{in}(u)} & \forall u \in V & 8a. \\ f_{in}(u) < \mu_u & \forall u \in V & 8b. \\ f(\overrightarrow{t_i s}) = xput & \forall i \in [1..k] & 8c. \\ f^i_{in}(u) = f^i_{out}(u) & \forall u \in V \;\; i \in [1..k] & 8d. \\ f^i(\overrightarrow{uv}) \leq C(\overrightarrow{uv}) & \forall \; \overrightarrow{uv} \in E \;\; i \in [1..k] & 8e. \end{cases}$$

$$\tau(u), f^i(\overrightarrow{uv}) \geq 0 \qquad \forall u, \forall i, \forall \; \overrightarrow{uv}$$

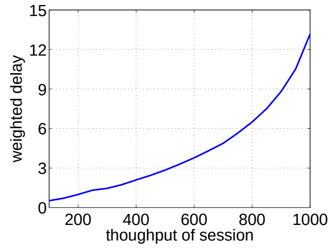

Fig. 2. Weighted delay for a single session as the throughput of the session is increased.

The objective function is the overall multicast flow delay across the network. The first constraint, similar to our previous models, defines the delay at a node based on results from queuing theory. Constraint $8b$ ensured that the system will be ergodic as the processing capacity of the node is not exceeded. Virtual links are defined between the receiving nodes and the sender, with flow rate equal to the desired throughput for each receiver. The flow balance constraint is enforced on the conceptual flows of each of the k receivers in constraint $8d$. Finally, it must be the case that no single flow can exceed the capacity of a link. In this model each f^i is a conceptual unicast flow from s to t_i. Note that constraint $8e$ requires each conceptual flow rate be upper-bounded by link capacity, instead of having the summation of them bounded by link capacity. This is due to the celebrated multicast feasibility characterization with network coding[1], [9]: *a multicast rate is feasible in a directed network if and only if it is feasible as an independent unicast to each receiver.*

Theorem 4: The Single Unicast Model is a convex optimization problem.
Proof of Theorem 4:
It is clear that constraints $8b$-$8e$ are convex because they are linear. Similar to the Single Unicast model, when the first constraint is substituted into the objective function the result will also be a convex function. Therefore, the Single Multicast Model is also a convex program and may be solved using standard techniques. □

VI. NUMERICAL RESULTS

In this section, we numerically solve our models to examine their performance under varying conditions. We use the CVX [5] package for Matlab to solve our convex optimization problems. The CVX package provides a convenient way to represent and solve convex optimization problems. Sample network topologies are generated using the BRITE [15] topology generation tool. We also generate μ_u for each node u using a uniform distribution between 1000 and 1500.

First, results are derived for the Single Unicast Model, as shown in Figure 2. We consider the weighted delay for a session as its throughput is increased from 100 to 1000. In this test, we fix the number of nodes in the topology at 50. We observe that the weighted delay for the unicast session increases in an exponential fashion as the throughput of the session is increased linearly. This is consistent with the behavior of an $M/M/1$ queue as the load on the server increases.

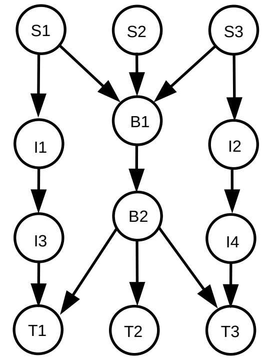

Fig. 6. Example graph to compare our model to previous models

The Multiple Unicast Model is solved in Figures 3 and 4. Impacts of multiple sessions competing for network resources are shown in Figure 3. Each session has a throughput requirement of 500 and the network has 50 nodes. The number of sessions in the network is increased from 1 to 10 and the weighted delay for the first session is plotted. Weighted delay for the first session remains relatively constant until the around 8-9 sessions. At this point, the load placed on the network by the additional sessions begins to have an adverse impact on the existing session by increasing queuing delay at nodes along the flow's route.

We now consider the inter-session fairness of the Multiple Unicast Model. We examine the situation where 2 sessions share a network of 50 nodes. The first session has a constant throughput of 100 while the throughput of the second session is varied from 100 to 1000. Figure 4 illustrates the results of this simulation. We observe that the weighted delay for session 2 increases exponentially, similar to results seen in the single unicast case, while the weighted delay for session 1 remains relatively constant. Increased throughput of the second session does not have an adverse impact on the performance of the first session with lower throughput requirements.

Performance of the Single Multicast Model is considered in Figure 5. We examine impacts of increasing the throughput for each receiver on the weighted delay of the entire multicast session. For the evaluation we use a network of 20 nodes and a multicast session with 5 receivers. The throughput to each receiver is varied from 100 to 1000. We observe an increase in weighted delay that is roughly exponential for linear increase in the throughput of the session. This is to be expected given results from queuing theory and is consistent with observations for the Single and Multiple Unicast Models.

VII. COMPARISON WITH LINK BASED MODEL

A unique characteristic of our model is that we consider queuing delays on nodes instead of links in the network. In this section we compare our node based approach to a

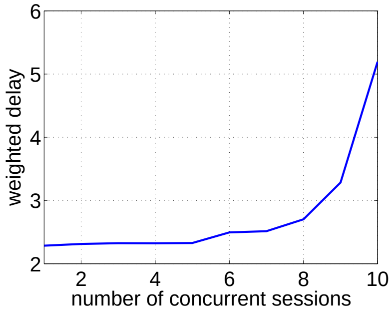

Fig. 3. Weighted delay for a single session as the number of concurrent sessions is increased.

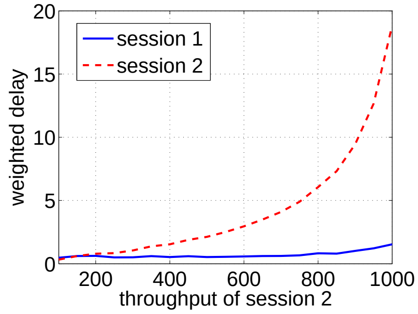

Fig. 4. Weighted delay for two sessions as the throughput of session 2 is increased.

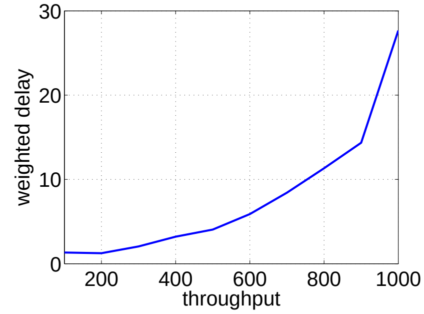

Fig. 5. Weighted delay for the multicast session as the throughput for the receivers is increased.

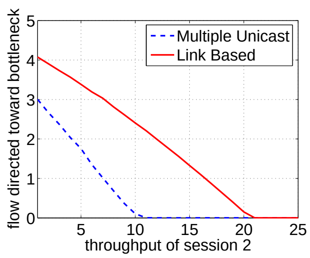

Fig. 7. Amount of data from flow 1 routed toward the bottleneck as the throughput of session 2 is increased.

traditional link-delay based model, and show that they are different.

We consider a formulation of the link based model for multiple sessions where each session is given an equal weight. The problem can be expressed by the convex program presented below which supports s concurrent sessions:

LINK BASED MODEL

Minimize

$$\sum_{\forall \overrightarrow{uv} \in E} f(\overrightarrow{uv}) t(\overrightarrow{uv}) \qquad (9)$$

Subject to:

$$\begin{cases} t(\overrightarrow{uv}) = \frac{1}{C(\overrightarrow{uv}) - f(\overrightarrow{uv})} & \forall\ \overrightarrow{uv} \in E & 9a. \\ f(\overrightarrow{uv}) = \sum_{i=1}^{s} f^i(\overrightarrow{uv}) & \forall\ \overrightarrow{uv} \in E & 9b. \\ f(\overrightarrow{uv}) \leq C(\overrightarrow{uv}) & \forall\ \overrightarrow{uv} \in E & 9c. \\ f(\overrightarrow{t_i s_i}) = xput_i & \forall i \in [1..s] & 9d. \\ f^i_{in}(u) = f^i_{out}(u) & \forall u \in V, \quad i \in [1..s] & 9e. \end{cases}$$

$$\tau(u), f^i(\overrightarrow{uv}) \qquad \forall i, \forall u, \forall\ \overrightarrow{uv}$$

The convex program above aims to minimize the total weighted delay of the links. Each link is modelled as an $M/M/1$ queue with a mean service time given by $C(\overrightarrow{uv})$. It is subject to similar constraints as our previous models. Delay is modelled using the $M/M/1$ queue formula for average delay. Flow along links is constrained by the capacity of the link as well as by flow balance constraints. Also similar to our previous models, a virtual link is placed between the receiver and sender of each session with a flow rate equal to the target throughput of the session.

We now consider the behavior of our Multiple Unicast node based model with the behavior of the Link Based Model. The graph we choose for the comparison is given in Figure 6. This graph is considered with three sessions with senders $S1$, $S2$ and $S3$ and receivers $T1$, $T2$ and $T3$, respectively. We select this graph because of the bottleneck link that may be shared by all three sessions. The bottleneck consists of 1 link but 2 nodes. Intuitively, this bottleneck will have a higher cost for the node based model than it will for the link based model.

The impact of the bottleneck link on the two different models is examined by numerically solving the model on the sample graph. These results consider the graph when all nodes and links having equal capacity of 60. Sessions 1 and 3 have a constant throughput requirement of 10. However, the throughput for session 2 is varied from 1 to 25. The flow directed toward the bottleneck by session 1 for each value of session 2 throughput is considered for each value of session 2 throughput. Our example graph is symmetric, so the amount of flow directed toward the bottleneck by session 3 is equivalent to the amount directed toward the bottleneck by session 1.

Figure 7 plots the amount of flow directed toward the bottleneck link by session 1 for both of the models and the various values of session 2 throughput. As we expect the amount of flow directed toward the bottleneck by the Link Based Model is greater than the amount directed toward the bottleneck in the Multiple Unicast Model. The Multiple Unicast Model stops sending data to the bottleneck when the throughput of session 2 is approximately 10. Session 2 throughput when the Link Based Model stops directing flow toward the bottleneck is twice as large. This scenario illustrates the importance of considering network delays both on nodes and links. If only one of these delays is considered it is possible that the full impact of a bottleneck link on performance may not be realized.

VIII. CONCLUSIONS AND FUTURE DIRECTIONS

This work is mainly intended to bring to the attention of the networking community a new network model, which leads to different optimal routing algorithm design and new understandings of multi-hop routing of stochastic information flows. In short, the new model is derived by combining advantages from both the network information flow model and network of queues model. It takes into account both the multi-hop routing nature and the stochastic processing nature of information flows across a general network topology. As a proof of concept, we show that minimum delay routing in the new model can be casted into convex programs, and solve them using the interior-point algorithm. A side result we also presented is that the flow synchronization due to network coding does not disturb the memoryless property of stochastic information flows.

REFERENCES

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network Information Flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, New Jersey, 1993.

[3] D. P. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.

[4] G. Bolch, S. Greiner, H. de Meer, and K. Trivedi. *Queuing Networks and Markov Chains*. Wiley, Hoboken, New Jersey, 2006.

[5] CVX: Matlab Software for Disciplined Convex Programming. http://www.stanford.edu/ boyd/cvx/.

[6] S. Deering. Host Extensions for IP Multicasting. In *Internet Requests for Comments (RFC1112)*, 1989.

[7] M. Gerla and L. Kleinrock. On the topological design of distributed computer networks. *IEEE Transactions on Communications*, 25(1), 1977.

[8] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact Of Interference On Multi-hop Wireless Network Performance. In *Proceedings of the 9th Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2003.

[9] R. Koetter and M. Médard. An Algebraic Approach to Network Coding. *IEEE/ACM Transactions on Networking*, 11(5):782–795, October 2003.

[10] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach*. Addison Wesley, Boston, MA, 2007.

[11] Z. Li, B. Li, D. Jiang, and L. C. Lau. On Achieving Optimal Throughput with Network Coding. In *Proceedings of IEEE INFOCOM*, 2005.

[12] Z. Li, B. Li, and L. C. Lau. On Achieving Optimal Multicast Throughput in Undirected Networks. *IEEE Transactions on Information Theory*, 52(6):2467–2485, June 2006.

[13] D. Lorenz and A. Orda. Optimal partition of qos requirements on unicast paths and multicast trees. *IEEE/ACM Transactions on Networking*, 10(1), 2005.

[14] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee. Achieving Minimum-cost Multicast: a Decentralized Approach Based on Network Coding. In *Proceedings of IEEE INFOCOM*, 2005.

[15] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: An approach to universal topology generation. In *MASCOTS'01*, Cincinnati, USA, 2001.

[16] M. Saad, A. Leon-Garcia, and W. Yu. Rate Allocation under Network End-to-End Quality-of-Service Requirements. In *IEEE Globecom*, San Francisco, USA, 2006.

[17] S. Siachalou and L. Georgiadis. Algorithms for precomputing constrained widest paths and multicast trees. *IEEE/ACM Transactions on Networking*, 13(5), 2005.