International Journal of Computer Trends and Technology (IJCTT) – volume 4 Issue 6–June 2013

Optimal Network Utility

Mrs.P.Radhadevi*1, Anil Ravala*2
Assistant Professor, Dept of Computer Applications, SNIST, Ghatkesar, Hyderabad, AP, India
M.C.A Student, Dept of Computer Applications, SNIST, Ghatkesar, Hyderabad, AP, India

ABSTRACT:

The problem of scheduling for maximum throughput-utility in a network with random packet arrivals and time varying channel reliability, the network controller assesses the condition of its channels and selects a set of links for transmission. The success of each transmission depends on the collection of links selected and their corresponding reliabilities. The goal is to maximize a concave, non-decreasing function of the time, average through-put on each link. Such a function represents a utility function that acts as a measure of fairness for the achieved throughput vector.

KEYWORDS: Network utility, Throughput, Reliability, Concave optimization, and optimization.

I.INTRODUCTION:

When the traffic is inside the network capacity region, the utility-optimal throughput vector is simply the vector of arrival rates, and the problem reduces to a network stability problem. In this case, it is well known that the network can be stabilized by max weight policies that schedule links every slot to maximize a weighted sum of transmission rates, where the weights are queue backlogs. This is typically shown via a Lyapunov drift argument (see [1] and references therein). This technique for stable control of a queuing network was first used for link and server scheduling, and has since become a powerful method to treat stability in different contexts, including switches and computer networks, wireless systems and ad-hoc mobile networks with rate and power allocation, and systems with probabilistic channel errors.

1.

In the case when traffic is either inside or outside of the capacity region, it is known that the max-weight policy can be combined with a flow control policy to jointly stabilize the network while maximizing throughput-utility. This is shown via a Lyapunov Optimization argument. Utility optimization for the special case of "infinitely backlogged" sources, and was perhaps first addressed for time-varying wireless downlinks without explicit queuing.

The stability works all use backlog-based trans-mission rules, as do the works in which treat joint stability and utility optimization. However, work introduces an interesting delay-based Lyapunov function for proving stability, where the delay of the head-of-line packet is used as a weight in the max-weight decision. This approach intuitively provides tighter control of the actual queuing delays. Indeed, a single head-of-line packet is scheduled based on the delay it has experienced, rather than on the amount of additional packets that arrived after it. This delay-based approach to queue stability is extended, where the Modified Largest Weighted Delay First algorithm is developed, and which uses a delay-based exponential rule. However, use delay-based rules only in the context of queue stability. To our knowledge, there are no prior works that use delay-based scheduling to address the important issue of joint stability and utility optimization. This paper fills that gap. We use a delay-based Lyapunov function, and extend the analysis to treat joint stability and performance optimization via the Lyapunov Optimization technique from our prior work.

2. RELATED WORK:

The Network Utility Maximization problem has recently been used extensively to analyze and design distributed rate allocation in networks such as the Internet. A major limitation in the state-of-the-art is that user utility functions are assumed to be strictly concave functions, modeling elastic flows. Many applications require inelastic flow models where nonconcave utility functions need to be maximized. It has been an open problem to find

the globally optimal rate allocation that solves non-concave network utility maximization, which is a difficult non-convex optimization problem. We provide a centralized algorithm for off-line analysis and establishment of a performance benchmark for non-concave utility maximization. Based on the semi algebraic approach to polynomial optimization, we employ convex sum-of-squares relaxations solved by a sequence of semi definite programs, to obtain increasingly tighter upper bounds on total achievable utility for polynomial utilities. Surprisingly, in all our experiments, a very low order and often a minimal order relaxation yields not just a bound on attainable network utility, but the globally maximized network utility. When the bound is exact, which can be proved using a sufficient test, we can also recover a globally optimal rate allocation. In addition to polynomial utilities, sigmoid utilities can be transformed into polynomials and are handled. Furthermore, using two alternative representation theorems for positive polynomials, we present price interpretations in economics terms for these relaxations, extending the classical interpretation of independent congestion pricing on each link to pricing for the simultaneous usage of multiple links.

The standard Network Utility Maximization (NUM) problem has a static formulation, which fails to capture the temporal dynamics in modern networks. This work considers a dynamic version of the NUM problem by introducing additional constraints, referred to as delivery contracts. Each delivery contract specifies the amount of information that needs to be delivered over a certain time interval for a particular source and is motivated by applications such as video streaming or webpage loading. The existing distributed algorithms for the Network Utility Maximization problems are either only applicable for the static version of the problem or rely on dual decomposition and first order (gradient or sub gradient) methods, which are slow in convergence. In this work, we develop a distributed Newton-type algorithm for the dynamic problem, which is implemented in the primal space and involves computing the dual variables at each primal step. We propose a novel distributed iterative approach for calculating the dual variables with finite termination based on matrix splitting techniques. It can be shown that if the error level in the Newton direction (resulting from finite termination of dual iterations) is below a certain threshold, then the algorithm achieves local quadratic convergence rate to an error neighborhood of the optimal solution in the primal space. Simulation results demonstrate significant convergence rate improvement of our algorithm, relative to the existing first-order methods based on dual decomposition.

We describe Wireless Network Utility Maximization, WNUM, and compare its performance to traditional NUM along the dimensions of rate, delay and reliability under flat fading. Both coded and uncoded links are considered as are networks with interfering links. In each case, WNUM is shown to offer superior performance in simulations operating under Rayleigh fading due to its ability to adapt to changing channel conditions. A general method for finding adaptive optimal policies is presented that is sample- based and that makes no assumptions about the distribution of channel states. WNUM uses optimal policies to adapt to changing channel conditions by adjusting network resources. We present the optimal adaptive control policies for WNUM in the single link case and describe a FROEC based algorithm for the multiple interfering link case. These policies are sample-based and make no assumptions about the distribution of channel states. NUM does not model the physical layer and consequently is unable to exploit good channel conditions or respond to poor channel conditions, resulting in relatively inferior performance. Future research work includes extending this formulation to broader types of reliability mechanisms and extending the formulation to traffic with QoS requirements.

3. **PROBLEM STATEMENT:**

In the present paper, we claim only that the achieved utility is within O(1=D) of the largest possible utility achievable by any stabilizing algorithm. However, because (for large D) our utility is close to this ideal utility value, it is even closer to the maximum utility that can be achieved

subject to the worst-case delay constraint. Further, our approach offers the low complexity advantages associated with Lyapunov drift and Lyapunov Optimization. Specifically, the policy makes real-time transmission decisions based only on the current system state, and does not require a-prior knowledge of the channel state probabilities. The flow control decisions here can also be implemented in a distributed fashion at each link, as is the case with most other Lyapunov based utility optimization algorithms (this is not necessarily the case for dynamic pro-gramming or Markov decision theory approaches).

It is important to distinguish our work, which considers actual network delay, with work that approximates network delay as a convex function of a flow rate (such as in [30][27]). While it is known that average queue congestion and delay is convex if traffic is probabilistically split [31], this is not necessarily true (or relevant) for dynamically controlled networks, particularly when the control depends on the queue backlogs and delays themselves. Actual network delay problems involve not only optimization of rate based utility functions, but engineering of the Lagrange multipliers (which are related to queue backlogs) associated with those utility functions.

The network is assumed to be a 1-hop network that operates in discrete time with normalized timeslots t 2 f0; 1; 2; : : :g. There are L links, and packets arrive randomly every slot and are queued separately for transmission over each link. We let $\mathbf{A}(t) = (A_1(t); : : : ; A_L(t))$ be the process of random packet arrivals, where $A_l(t)$ is the number of packets that arrive to link l on slot t. For simplicity, we assume all packets have fixed size, and that there is at most one packet arrival to each link per slot, so that $A_l(t)$ 2 f0; 1g for all links l and slots t. The arrival vector $\mathbf{A}(t)$ is assumed to be i.i.d. over slots, and further the arrival processes $A_l(t)$ for different links in each slot are assumed to be independent. Let $\mathbf{Q}(t) = (Q_1(t); : : : ; Q_L(t))$ denote the integer number of packets currently stored in each of the L queues. All packets are marked with their integer arrival slot, which is used to determine their delay in the system. The one-step queueing equation for each link l is:

$$Q_l(t + 1) = \max[Q_l(t) \quad _l(t) \quad D_l(t); 0] + A_l(t) \quad (1)$$

where $_l(t)$ represents the amount of packets successfully served on slot t, and $D_l(t)$ represents the number of packets dropped on slot t. A packet can be dropped at any time, al-though in our specific algorithm we impose a 2-stage structure that first makes a transmission decision and then makes a dropping decision in reaction to the feedback obtained from the transmission.

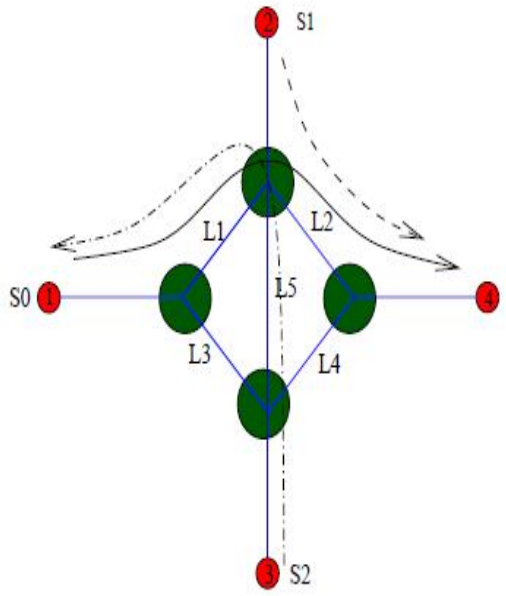

Fig. Example illustrating network resource allocation. The large circles indicate routers that route packets in the core network. We assume that links L1, L3 and L5 have capacity 2, while L2 and L4 have capacity 1. The access links of the sources are unconstrained. There are three flows in the system.

4. IMPLEMENTATION METHODS

A. Time Varying Link Reliability

For simplicity, we assume that each link can transmit at most one packet per slot, so that $_l(t)$ 2 f0; 1g for all links l and all slots t. Let $\mathbf{x}(t) = (x_1(t); : : : ; x_L(t))$ denote a transmission vector, where $x_l(t)$ 2 f0; 1g, and $x_l(t) = 1$ if link l attempts transmission on slot t. Let X denote the set of all allowable link transmission vectors, possibly being the set of all 2^L such vectors, but also

possibly incorporating some constraints (such as permutation constraints for N N packet switches). In principle, it is useful to assume a link can transmit even if it does not have a packet, in which case a null packet is transmitted. Let $\mathbf{S}(t) = (S_1(t); : : : ; S_L(t))$ denote a link condition vector for slot t, which determines the probability of successful transmission on each slot. Specifically, given particular $\mathbf{x}(t)$ and $\mathbf{S}(t)$ vectors, the probability of successful transmission on link l is given by a reliability function:

$$\Pr[\text{ link l success } j\mathbf{x}(t); \mathbf{S}(t)] = {}_l(\mathbf{x}(t); \mathbf{S}(t)) \quad (2)$$

The reliability function ${}_l(\mathbf{x}; \mathbf{S})$ for each l 2 f1; : : : ; Lg is general and is assumed only to take real values between 0 and 1 (representing probabilities), and to have the property that ${}_l(\mathbf{x}; \mathbf{S}) = 0$ whenever $x_l = 0$. We assume that the channel condition vector $\mathbf{S}(t)$ is i.i.d. over slots, taking values in a set S of arbitrary cardinality, and that $\mathbf{S}(t)$ is known to the network controller at the beginning of each slot t. In practice, $\mathbf{S}(t)$ represents the result of a channel measurement or estimation that is done every slot. The estimate might be inexact, in which case the reliability function ${}_l(\mathbf{x}(t); \mathbf{S}(t))$ represents the probability that the actual network channels on slot t are sufficient to support the attempted transmission over link l (given $\mathbf{x}(t)$ and the estimate $\mathbf{S}(t)$ for slot t).

We assume the reliability function is known. Recent online techniques for estimation of packet error rates are considered in [32]. In the context of [32], a number of other decision parameters to be chosen on each slot also affect reliability, such as modulation, power levels, sub band selection, coding type, etc. These choices can be represented as a parameter space I. In this case, the reliability function can be extended to include the parameter choice I(t) 2 I made every slot: ${}_l(\mathbf{x}(t); \mathbf{S}(t); I(t))$. This does not change our mathematical analysis (see also Remark 1 in Section III-F), although for simplicity we focus on the reliability function structure of (2).

We assume that ACK/NACK information is given at the end of the slot to inform each link if its transmission was successful or not. Packets that are not successful do not leave the queue (unless they are dropped in a packet drop decision). With this model of link success, the transmission variable ${}_l(t)$ in (1) is given by:

$${}_l(t) = x_l(t)1_l(t)$$

where $1_l(t)$ is an indicator variable that is 1 if the transmission over link l is successful, and 0 otherwise. That is:

$$1_l(t) = \begin{cases} 1 & \text{with probability } {}_l(\mathbf{x}(t); \mathbf{S}(t)) \\ 0 & \text{with probability } 1 - {}_l(\mathbf{x}(t); \mathbf{S}(t)) \end{cases}$$

The successes/failures over each link on slot t are assumed to be independent of past history given the current $\mathbf{x}(t)$ and $\mathbf{S}(t)$ values. The successes/failures might be correlated over each link. This is not captured in the ${}_l(\mathbf{x}; \mathbf{S})$ functions alone, and can only be fully described by a joint success distribution function for all 2^L possible success/failure outcomes for a given $\mathbf{x}$ and $\mathbf{S}$. However, it turns out that the network capacity region, and hence the associated maximum utility point, is independent of such inter-link success correlations [11]. Hence, it suffices to use only the marginal distribution functions ${}_l(\mathbf{x}; \mathbf{S})$ for each l 2 f1; : : : ; Lg.

B. Examples of Packet Switches and Wireless Networks

The above model applies to a wide class of 1-hop networks. For example, it applies to the N N packet switch models of [4][6] by defining $\mathbf{S}(t)$ to be a null vector (so that there is no notion of channel variation), and by defining the set X of all allowable link vectors to be the set of all vectors that satisfy the permutation constraints associated with the N N crossbar. For wireless networks with interference but without time varying channels, the set X can be defined as all link activations that do not interfere with each other (i.e., that do not produce collisions), as in [2]. The reliability function ${}_l(\ )$ can be used to extend the model to treat cases where interfering links result in probabilistic reception (rather than collision).

Further, the opportunistic scheduling systems of [3] with time-varying ON/OFF channels can be modeled with $\mathbf{S}(t)$ being the vector of ON/OFF channel states on each slot, and with the function ${}_l(\mathbf{x}; \mathbf{S})$ taking the value 1 whenever $x_l = 1$ and $S_l =$ ON, and 0 otherwise. Finally, the model supports probabilistic reception in the case when the link reliability can vary from slot to slot.

A simple example is when $S_l(t)$ represents the current probability that a link l transmission would be successful, so that:

$$x_l(t) = 1\ 1^{(\mathbf{x}(t);\ \mathbf{S}(t)) = S_l(t) \text{ if } x_l(t) = 1}_{0 \text{ if } x_l(t) = 0}$$

This example has the success probability over link l a pure function of $x_l(t)$ and $S_l(t)$, and hence

implicitly assumes that the set X limits all simultaneous link transmissions to orthogonal channels. More complex inter-channel interference models can be described by more complex $_l(\mathbf{x}; \mathbf{S})$ functions.

5. CONCLUSION

We have established a delay-based policy for joint stability and utility optimization. The policy provides deterministic worst-case delay bounds, with total throughput-utility that is inversely proportional to the delay guarantee. The Lyapunov Optimization approach for this delay-based problem is significantly different from that of backlog-based policies. Further, delay-based scheduling must overcome difficult issues involving the correlation between inter-arrival times and virtual queue states. Several new techniques were introduced to solve the problem, including the structure of dropping packets at the head-of-line (rather than immediately upon arrival), introducing the concept of concavely extending a utility function, and using a delayed arrival process in the virtual queues to maintain required independence. We believe these results add significantly to our understanding of network delay and delay-efficient control laws.

6. REFERENCES

[1] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. Foundations and Trends in Networking, vol. 1, no. 1, pp. 1-149, 2006.

[2] L. Tassiulas and A. Ephremides. Stability properties of constrained queuing systems and scheduling policies for maximum throughput in multichip radio networks. IEEE Transactions on Automatic Control, vol. 37, no. 12, pp. 1936-1949, Dec. 1992.

[3] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. IEEE Transactions on Information Theory, vol. 39, pp. 466-478, March 1993.

[4] N. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. Proc. IEEE INFOCOM, 1996.

[5] P. R. Kumar and S. P. Meyn. Stability of queuing networks and scheduling policies. IEEE Trans. on Automatic Control, vol.40,.n.2, pp.251-260, Feb. 1995.