ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

Identifying Prevelance Flood Attacks in Delay Tolerant Networks

C.Gnana Prakash[1], K.Shanthi[2]

Student, Dept of Computer Science, S.V. College of Engineering, Tirupathi, India[1]

Associate Professor, Dept of Computer Science and Engineering, S.V.College of Engineering, Tirupathi, India[2]

ABSTRACT: A distributed scheme to detect if a node has debased its rate limits. To address the challenge that it is difficult to count all the packets or replicas sent by a node due to lack of statement infrastructure, our detection adopts claim-carry-and check: each node itself counts the number of packets or replicas that it has sent and claims the count to other nodes; the getting nodes carry the claims when they move, and cross-check if their carried claims are inconsistent when they contact. The claim structure uses the pigeonhole standard to guarantee that an attacker will make conflicting claims which may lead to discovery. We present rigorous analysis on the probability of detection, and assess the effectiveness and efficiency of our scheme with extensive trace driven simulations. Using Lyapunov optimization, we extend this examination to design a utility Maximizing algorithm that uses explicit delay information from the head-of-line packet at each user. The consequential policy is shown to ensure deterministic worst-case delay guarantees and to yield a throughput utility that differs from the optimally fair value by an amount that is inversely proportional to the delay guarantee. Our results hold for a general class of 1-hop networks, including packet switches and multiuser wireless systems with time varying reliability.

KEYWORDS: Lyapunov optimization, claim-carry-and check, 1-hop networks, packet switches.

I. INTRODUCTION

In this paper, we employ rate limiting [15] to defend beside flood attacks in DTNs. In our approach, each node has a limit over the number of packets that it, as a source node, can send to the network in each time interval. Each node also has a boundary over the numeral of replicas that it can generate for each packet (i.e., the number of nodes that it can forward each packet to). The two limits are used to mitigate packet flood and model flood attacks, respectively. If a node violates its rate limits, it will be detected and its data traffic will be filtered. In this way, the amount of flooded traffic can be controlled.

Our main role is a procedure to notice if a node has violated its rate limits. Although it is easy to detect the violation of rate limit on the Internet and in telecommunication networks where the egress router and base station can account each user's traffic, it is challenging in DTNs due to lack of communication infrastructure and consistent connectivity. Since a node moves approximately and may send data to any contacted node, it is very difficult to count the number of packets or replicas sent out by this node. Our essential idea of detection is claim-carry-and-check. Each node itself counts the number of packets or replicas that it has sent out, and claims the count to other nodes; the receiving nodes carry the claims around when they move, swap some claims when they contact, and cross-check if these claims are inconsistent. If an attacker floods more packets or replicas than its limit, it has to use the same calculate in more than one claim according to the pigeonhole principle, and this inconsistency may lead to detection. Based on this idea, we use different cryptographic constructions to detect packet flood and replica flood attacks.

Because the contacts in DTNs are opportunistic in nature, our approach provides probabilistic detection. The more traffic an attacker floods, the more likely it will be detected. The detection probability can be flexibly adjusted by system parameters that control the amount of claims

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

exchanged in a contact. We provide a inferior and better bound of detection probability and examine the problem of parameter selection to maximize detection probability under a certain amount of exchanged claims. The effectiveness and efficiency of our scheme are evaluated with extensive trace-driven simulations. This paper fills that gap. We use a delay-based Lyapunov function and extend the examination to treat joint stability and presentation optimizat ion via the Lyapunov optimization technique from our prior work [2], [13], [14]. The extension is not obvious. Indeed, the flow control decisions in the prior work [2], [13], [14] are made immediately when a new packet arrives, which directly affects the drift of backlog-based Lyapunov functions. However, such decisions *do not* directly affect the delay value of the head-of-line packets, and hence do not directly affect the drift of delay-based Lyapunov functions. We overcome this challenge with a novel flow control policy that queues *all* arriving data, but makes packet dropping decisions just before advancing a new packet to the head-of-line. This policy is structurally different from the service optimization works [2] and [13]–[20]. This new structure leads to deterministic guarantees on the worst-case delay of any non-dropped packet and provides throughput utility that can be pushed arbitrarily close to optimal. Specifically, for any integer $D \geq 2$, we Similar $[O(1/D), O(D)]$ performance tradeoffs are shown for queue-based Lyapunov functions in the previous work [2], [13], [14] (see also [24]–[26] for improved tradeoffs), but these guarantees apply only to queue size, slightly than delay.

The deterministic delay guarantees we obtain in this present paper are quite strong and show the advantages of our new flow control structure. However, a disadvantage is that admit/drop decisions are delayed until a packet is at the head-of-line, rather than being determined immediately upon arrival. Moreover, due to correlation issues unique to this delay-based scenario, analysis is simplified if we assume the scheduler knows the vector of arrival rates to each link (although we also generalize to cases when these rates are unknown). Furthermore, while our deterministic delay guarantees hold for general arrival sample paths, our utility analysis assumes all entrance processes are independent of each other (possibly with different rates for each process) and independent and identically distributed (i.i.d.) over time-slots. Nevertheless, it is important to analyze these delay-based policies because they improve our understanding of complex delay, and because the deterministic guarantees they offer are useful for many practical systems.

We further show via simulation that our algorithms maintain good performance when the i.i.d. arrivals are replaced by ergodic but temporally correlated “bursty” arrivals with the same rates. However, the worst-case delay required to achieve the same utility performance is amplified in this case. This is not surprising if we compare to known results for backlog-based Lyapunov algorithms. Backlog-based algorithms were first developed under i.i.d. assumptions, but later shown to work—with increased delay—for non-i.i.d. cases (see [28] and references therein). Thus, while we limit our analytical proofs to the i.i.d. setting, we expect the algorithm to approach optimal utility in more general cases, as supported by our simulations. While our algorithm can be used to enforce any desired delay guarantee, it is important to emphasize that it does *not* maximize throughput utility subject to this guarantee. Such a problem can be addressed with Markov decision theory, which brings with it the *curse of dimensionality* (see structural results and approximations in [29] and weighted stochastic shortest-path approaches in [30]). proaches in [30]). In this paper, we claim only that the achieved utility is within $O(1/D)$ of the largest probable utility of any stabilizing algorithm. However, because (for large) 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. That is because a basic stability constraint is less stringent than a worst-case delay constraint, and so the optimal utility under a stability constraint is greater than or equal to the optimal utility under a worst-case delay constraint. Furthermore, our approach offers the low-complexity advantages associated with Lyapunov drift and Lyapunov optimization. Specifically, the policy makes real-time broadcast decisions based only on the current system state and does not require *apriori* 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

II. OVERVIEW

2.1 Problem Definition

Defense against Packet Flood Attacks

We consider a scenario where each node has a rate limit L on the number of unique packets that it as a source can generate and send into the network within each time interval T. The time intervals start from time 0, T, 2T, etc. The packets generated within the rate limit are deemed genuine, but the packets generated beyond the limit are deemed flooded by this node. To defend against packet flood attacks, our goal is to detect if a node as a source has generated and sent more unique packets into the network than its rate limit L per time interval. The span of time interval should be set appropriately. If the interval is too long, rate limiting may not be very efficient against packet flood attacks. If the interval is too short, the number of associates that each node has during one interval may be too nondeterministic and thus it is difficult to set an appropriate rate limit. Generally speaking, the space should be short under the condition that most nodes can have a significant number of contacts with other nodes within one interval, but the appropriate length depends on the contact patterns between nodes in the specific deployment scenario.

Defense against Replica Flood Attacks

As motivated in Section 2, the defense against replica flood considers single-copy and multi-copy routing protocols. These protocols need that, for each packet that a node buffers no matter if this packet has been generated by the node or forwarded to it, there is a limit l on the number of times that the node can forward this packet to other nodes. The values of l may be different for different buffered packets. Our goal is to detect if a node has violated the routing protocol and forwarded a packet more times than its limit l for the packet. A node's limit l for a buffered packet is firm by the routing protocol. In multi copy routing, l ¼ L is a parameter of routing) if the node is the source of the packet, and l ¼ 1 if the node is an intermediate hop (i.e., it received the packet from another node). In single-copy routing, l ¼ 1 no matter if the node is the source or an intermediate hop. Note that the two limits L and l do not depend on each other. We discuss how to protect against replica flood attacks for quota-based routing.

III. CLAIM-CARRY-AND-CHECK

Packet Flood Detection

To detect the attackers that abuse their rate limit L, we must count the number of unique packets that each node as a basis has generated and sent to the network in the current interval. However, since the node may send its packets to any node it contacts at any time and place, no other node can monitor all of its sending behavior. To address this challenge, our idea is to let the node itself count the number of unique packets that it, as a source, has sent out, and claim the up-to-date packet count (together with a little auxiliary information such as its ID and a timestamp) in each packet sent out. The node's rate limit certificate is also attached to the packet, such that other nodes receiving the packet can learn its official rate limit L.

If an attacker is flooding more packets than its rate limit, it has to deceitfully claim a count smaller than the real value in the flooded packet, since the real value is larger than its rate limit and thus a clear indicator of attack. The claimed count must have been used before by the attacker in another claim, which is guaranteed by the pigeonhole principle, and these two claims are inconsistent. The nodes which have conventional packets from the attacker carry the claims included in those packets when they move around. When two of them contact, they check if there is any variation between their collected claims. The attacker is detected when an inconsistency is found.

IV. THE INTELRATE CONTROLLER DESIGN

fuzzy logic traffic controller for controlling traffic in the network system defined Called the IntelRate, it is a TISO (Two-Input SingleOutput) controller. The TBO (Target Buffer Occupancy) q>0 is the queue size level we aim to achieve upon congestion. The queue deviation $e(t) = q_0 - q(t)$ is one of the two inputs of the controller. In order to remove the steady state error, we choose the integration of e(t) as the other input of the controller, $e(t)$ i.e. g

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

$g(e(t)) = \int e(t)\,dt$. The aggregate output is $y(t) = \sum u_i(t - \tau_i)$. Under heavy traffic situations, the IntelRate controller would compute an allowed sending rate $u_i(t)$ for flow i according to the current IQSize so that q(t) can be stabilized around q. In our design, IQSize q(t) is the only parameter each router needs to calculate in order to complete the closed-loop control. FLC is a non-linear map of inputs into outputs, which consists of four steps, i.e., rule base building, fuzzification, inference and defuzzification. The concepts of fuzzy set and logic of FLC were introduced in 1965 by Zadeh, and it was basically extended from two-valued logic to the continuous interval by adding the intermediate values between absolute TRUE and FALSE. Interested readers are referred to some normal tutorials/texts like [36], [45] for the details of the fuzzy logic theory. In the sequel, we formulate our new controller by following those four steps along with designing the fuzzy linguistic descriptions and the membership functions. The parameter design issues and the traffic control process are also discussed at the end of the section.

Linguistic Description and Rule Base

We define the crisp inputs e(t), g(e(t)) and output u(t) with the linguistic variables $\tilde{e}(t)$, $\tilde{g}(e(t))$ and $\tilde{u}(t)$, respectively. There aren N(N =1, 2, 3,...) LVs (Linguistic Values) assigned to each of these linguistic variables. Specifically, we let $\overline{P}_i = \left\{\overline{P}_i^j : j = 1, 2, \cdots, N\right\}$ be the input LVs with i =1for $\tilde{e}(t)$ and i =2for $\tilde{g}(e(t))$, and let $\overline{U} = \left\{\overline{U}^j : j = 1, 2, \cdots, N\right\}$ for $\tilde{u}(t)$. For example, when N =9, we can assign the following values for both the inputs e(t) and g(e(t)). $\overline{P}_i^1$ =Negative Very Large (NV)," Large (NL)," $\overline{P}_i^3$ ="Negative Medium (NM)," $\overline{P}_i^4$ ="Negative Small (NS)," $\overline{P}_i^5$ ="Zero (ZR)," $\overline{P}_i^6$ ="Positive Small (PS)," $\overline{P}_i^7$ ="Positive Medium (PM)," $\overline{P}_i^8$ ="Positive Large (PL)," and $\overline{P}_i^9$ ="Positive Very Large (PV)," $i = 1, 2$. Similarly, we can designate the output when $N = 9$ with the following linguistic values. $\overline{U}^1$ ="Zero (ZR)," $\overline{U}^2$ ="Extremely Small (ES)," $\overline{U}^3$="Very Small (VS)," $\overline{U}^4$ ="Small (SM)," $\overline{U}^5$ ="Medium (MD)," $\overline{U}^6$ ="Big (BG)," $\overline{U}^7$ ="Very Big (VB)," $\overline{U}^8$ ="Extremely Big (EB)," and $\overline{U}^9$ ="Maximum (MX)."

V. DELAY-BASED FLOW CONTROL

Let $\text{Let } \boldsymbol{\lambda} = \mathrm{E}[\boldsymbol{A}(t)]$ be the vector of arrival rates, so that $\lambda_l = \mathrm{E}[A_l(t)]$ is the arrival rate to link (in units of packets/slot). The *network capacity region* is defined as the closure of the set of all long-term throughput vectors that the scheme can support. The set is known to be the same as the closure of the set of all arrival rate vectors for which there exists a stabilizing preparation algorithm, subject to the constraint that *the flow controllers are turned off* (so that no packets are dropped and $D_l(t) = 0$ for all and all) [4], [12]. Specifically, in [12] it is shown that the set is given by the set of all time-average transmission rates that can be achieved by motionless and randomized algorithms, called only algorithms, that observe every slot and choose a (possibly random) diffusion vector $\boldsymbol{x}(t) \in \mathcal{X}$ according to a probability distribution that depends only on the observed channel state. Thus, for every vector ,with $\boldsymbol{r} = (r_1, \ldots, r_L)$, , there is an -only algorithm $\boldsymbol{x}^*(t)$, with a corresponding random service vector $\boldsymbol{\mu}^*(t) = (x_1^*(t)1_1^*(t), \ldots, x_L^*(t)1_L^*(t))$ that yields for each $l \in \{1, \ldots, L\}$

$r_l = \mathrm{E}[\mu_l^*(t)] = \mathrm{E}[x_l^*(t)\Psi_l(\boldsymbol{x}^*(t), \boldsymbol{S}(t))]$ where the expectation in is with respect to the distribution of $\boldsymbol{S}(t)$and the distribution of $\boldsymbol{x}^*(t)$ given $\boldsymbol{S}(t)$.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

A. Optimization Objective

Let $g(\boldsymbol{y})$ be a continuous and concave utility function of the dimensional vector $\boldsymbol{y} = (y_1, \ldots, y_L)$, ,where is used to signify the time-average throughput on each link (in units of packets/slot). The function can take positive or negative values And is assumed to be defined over the hyper-cube $\mathbf{0} \leq \boldsymbol{y} \leq \mathbf{1}$, where inequality is taken entrywise, and and are vectors with all entries equal to 0 and 1, respectively. Assume that is non decreasing in each entry y_l . An example is the *separable* utility function

$$g(\boldsymbol{y}) = \sum_{l=1}^{L} g_l(y_l)$$

where for each link $l, g_l(y_l)$ is a concave and non decreasing function of y_l defined over the interval $0 \leq y_l \leq 1$.We make the following additional assumption.

Assumption 1*:* For each $l \in \{1, \ldots, L\}$ the th right partial derivative of $g(\boldsymbol{y})$, over all $\boldsymbol{y} \in [0,1]^L$ such that $y_l < 1$, is bounded above by a finite constant ν_l ,where $\nu_l \geq 0$.[2] Assumption 1 implies that for any vectors and such that

$\boldsymbol{y} \in [0,1]^L, \boldsymbol{w} \in [0,1]^L$, and $\boldsymbol{y} + \boldsymbol{w} \in [0,1]^L$, and we have

$$g(\boldsymbol{y} + \boldsymbol{w}) \leq g(\boldsymbol{y}) + \sum_{l=1}^{L} \nu_l w_l.$$

The above is also an approximation of proportional fairness when $\nu_l = \nu$ for all for some large value . For each link define as

$$Y_l(t) \triangleq \lambda_l - D_l(t).$$

B. Problem Transformation With Virtual Queues

It is not difficult to show that the stochastic network optimization problem (9)–(11) can be transformed using a vector $\boldsymbol{\gamma}(t) = (\gamma_1(t), \ldots, \gamma_L(t))$ of *auxiliary variables* that are chosen every slot according to the constraints. The transformed problem is

$$\begin{aligned}
&\text{Maximize:} && \hat{g}(\overline{\boldsymbol{\gamma}}) \\
&\text{Subject to:} && \overline{Q}_l < \infty \quad \text{for all } l \in \{1, \ldots, L\} \\
&&& \overline{y}_l \geq \overline{\gamma}_l \quad \text{for all } l \in \{1, \ldots, L\} \\
&&& -1 \leq \overline{\gamma}_l \leq 1 \quad \text{for all } l \in \{1, \ldots, L\} \\
&&& \overline{\boldsymbol{Q}} \text{ and } \overline{\boldsymbol{y}} \text{ are achievable on the network}
\end{aligned}$$

where $\overline{Q}_l$ is defined

$$\overline{Q}_l \triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[Q_l(\tau)\right].$$

We say that a nonnegative discrete-time stochastic process $Q_l(t)$ is strongly stable if $\overline{Q}_l < \infty$. This transformation can be intuitively understood as follows. The constraint (11) automatically holds for any achievable control policy, as the throughput cannot be larger than the raw arrival rate, and hence is satisfied whenever (18) holds. The constraint (10) is ensured by the stability constraint (15) in the transformed problem. Finally, one can always choose the auxiliary vector $\boldsymbol{\gamma}(t) = \boldsymbol{y}(t)$ to ensure that (16) and (17) are satisfied (note that for all because arrival rates cannot be larger than 1). The fact that $0 \leq \overline{y}_l \leq 1$ is non decreasing in each entry and that $\hat{g}(\boldsymbol{y}) = g(\boldsymbol{y})$ whenever $0 \leq \boldsymbol{y} \leq 1$ ensures that it suffices to consider all constraints (16) holding with equality, so that any control algorithm that solves (14)–(18) also solves (9)–(11). The auxiliary variables $\boldsymbol{\gamma}(t)$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

are important for solving problems of maximizing a concave function of a time average and are critical for network utility maximization with randomly arriving traffic [2], [28]. To ensure that the constraints (16) are satisfied, we use a *virtual queue* for each link , with update equation as follows:

$$Z_l(t+1) = \max\left[Z_l(t) - \lambda_l + D_l(t) + \gamma_l(t), 0\right].$$

C. Delay-Based Lyapunov Function

We now impose the following structure on our control policy. Every slot, a packet transmission decision is made *first*.Ifa transmission over link l is successful (so that), then the packet is removed from the queue, and no packet is dropped from link l (so that). Else, if link either did not attempt transmission or if its transmission was unsuccessful, we can decide whether or not to drop the packet, but no other packet can be dropped from link l Thus, every slot ,we have $0 \le \mu_l(t) + D_l(t) \le 1$. We show later that this structure does not hinder our maximum utility objective. Furthermore, it is useful to consider the possibility of transmitting or dipping a *null packet* when the queue is empty, so that $\mu_l(t)$ and $D_l(t)$ in principle can be chosen independently of queue backlog. Let $H_l(t)$ represent the waiting time of the head-of-line packet in link on slot (being at least one if there is a packet), and define $H_l(t) = 0$ if there are no packets in link at this time. A new packet that arrives to an unfilled queue on slot is not placed to the head-of-line until the next slot and is designated to have a waiting time of 1 at slot $t+1$.Define as an indicator variable that is 1 if $Q_l(t) > 0$, and is zero if the queue is empty. Let $\beta_l(t) = 1 - \alpha_l(t)$. . Similar to [21], we observe that satisfies the following:

$$H_l(t+1) = \alpha_l(t)\max\left[H_l(t) + 1 - (\mu_l(t) + D_l(t))T_l(t), 0\right]$$

D. Minimizing the Drift-Plus-Penalty

Define $\Delta(\boldsymbol{\Theta}(t))$ as the one-step conditional Lyapunov drift

$$\Delta(\boldsymbol{\Theta}(t)) \triangleq \mathbb{E}\left[L(\boldsymbol{\Theta}(t+1)) - L(\boldsymbol{\Theta}(t)) \,|\boldsymbol{\Theta}(t)\right]$$

Using the Lyapunov optimization framework in [2] and [28], our strategy is to make transmission and packet dropping decisions to minimize a bound on the following "drift-plus-penalty" expression every slot:

$$\Delta(\boldsymbol{\Theta}(t)) - V\mathbb{E}\left[\hat{g}(\boldsymbol{\gamma}(t)) \,|\boldsymbol{\Theta}(t)\right]$$

where is a nonnegative control parameter that is chosen as desired and will affect an explicit utility–delay tradeoff. Here, the "penalty" for slot t is considered to be 1 times the "reward" $\hat{g}(\boldsymbol{\gamma}(t))$. We later show that our resulting algorithm has a certain independence property, defined as follows.

Definition 1*:* We say that a control algorithm implemented over time has the independence property if for any slot t ,every Link l such that $H_l(t) > 0$ has a value of $T_l(t)$ that is independent of $\boldsymbol{\Theta}(t), \mu_l(t),$ and $D_l(t)$.

$$\Delta(\boldsymbol{\Theta}(t)) \le B - \sum_l Z_l(t)\mathbb{E}\left[\lambda_l - D_l(t) - \gamma_l(t)|\boldsymbol{\Theta}(t)\right]$$
$$- \sum_l \lambda_l H_l(t)\mathbb{E}\left[(\mu_l(t) + D_l(t))T_l(t) - 1|\boldsymbol{\Theta}(t)\right]$$

where is a finite constant that does not depend on .

Proof: The proof is given in Appendix B, where the constant is also specified.

Lemma 2: Every slot , for any value of , and under any control policy that satisfies the independence property, we have

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

$$\begin{aligned}&\Delta(\boldsymbol{\Theta}(t)) - V\mathbb{E}\left[\hat{g}(\boldsymbol{\gamma}(t))|\boldsymbol{\Theta}(t)\right]\\&\leq B - V\mathbb{E}\left[\hat{g}(\boldsymbol{\gamma}(t))|\boldsymbol{\Theta}(t)\right]\\&\quad-\sum_l Z_l(t)\mathbb{E}\left[\lambda_l - D_l(t) - \gamma_l(t)|\boldsymbol{\Theta}(t)\right]\\&\quad-\sum_l H_l(t)\mathbb{E}\left[\mu_l(t) + D_l(t) - \lambda_l|\boldsymbol{\Theta}(t)\right]\end{aligned}$$

where is the same constant from Lemma 1.

E. Delay-Based Flow Control and Scheduling Algorithm

Every slot , observe , $\boldsymbol{Z}(t), \boldsymbol{H}(t),$,and $\boldsymbol{S}(t),$, and perform the following operations, described as four control phases:
1.Auxiliary Variable Selection: Choose
$\boldsymbol{\gamma}(t) = (\gamma_1(t), \ldots, \gamma_L(t))$ as the solution to the following:

$$\begin{aligned}&\text{Maximize}: \quad V\hat{g}(\boldsymbol{\gamma}(t)) - \sum_l Z_l(t)\gamma_l(t)\\&\text{Subject to}: \quad -1 \leq \gamma_l(t) \leq 1 \qquad \text{for all } l \in \{1, \ldots, L\}.\end{aligned}$$

Subject to for all In the case of the separable utility function (4), this amounts to solving single-variable concave optimizations over an interval and has a closed-form solution when $\hat{g}_l(\gamma_l)$ has a derivative with a closed-form inverse.

2) Transmission Scheduling: Observe $\boldsymbol{Z}(t), \boldsymbol{H}(t), \boldsymbol{S}(t)$ and choose a transmission vector $\boldsymbol{x}(t)$ to solve the following:

$$\begin{aligned}&\text{Maximize}: \quad \sum_l x_l(t)\min\left[H_l(t), Z_l(t)\right]\Psi_l(\boldsymbol{x}(t), \boldsymbol{S}(t))\\&\text{Subject to}: \quad \boldsymbol{x}(t) \in \mathcal{X}.\end{aligned}$$

3) Packet Dropping: For each link l that has a head-of-line packet that was not successfully transmitted in the scheduling phase (either because its transmission was not attempted or its transmission failed), drop the packet if $Z_l(t) \leq H_l(t)$.. Else, keep it in the head-of-line.

4) Queue Updates: Update the virtual queues $Z_l(t)$ according to (19), using the values of $\gamma_l(t)$ and $D_l(t)$ as determined from the above auxiliary variable and packet dropping phases. Also update the actual queues and the head-of-line values according to (1) and (21) by simply removing any packet that was either successfully transmitted or dropped.

VI. CONCLUSIONS

we employed rate limiting to ease flood attacks in DTNs, and a scheme which exploits claim-carry-and-check to probabilistically notice the violation of rate limit in DTN environments. Our scheme uses efficient constructions to keep the computation, communication and storage cost low. Also, we analyzed the lower bound and upper bound of detection probability We have established a delay-based policy for joint stability and utility optimization. The policy provides deterministic worst-case delay bounds, with total throughput value 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. We believe these results add significantly to our understanding of network delay and delay-efficient control laws.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

REFERENCES

[1] M. J. Neely, "Delay-based network utility maximization ," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[2] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*,vol.1, no. 1, pp. 1–149, 2006.
[3] L. Tassiulas and A. Ephremides, "Stability proper ties of constrained queueing systems and scheduling policies for maxi mum throughput in multihop radio networks," *IEEE Trans. Autom. Con trol*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.
[4] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEE E Trans. Inf. Theory*, vol. 39, no. 2, pp. 466–478, Mar. 1993.
[5] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued sw itch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.
[6] P. R. Kumar and S. P. Meyn, "Stability of queueing networks and scheduling policies," *IEEE Trans. Autom. Co ntrol*,vol.40,no.2,pp. 251–260, Feb. 1995.
[7] E. Leonardi, M. Mellia, F. Neri, and M. A. Mars an, "Bounds on average delays and queue size averages and va riances in input-queued cell-based switches," in *Proc. IEEE INFOCO,M*, 2001, vol. 2, pp. 1095–1103.
[8] N. Kahale and P. E. Wright, "Dynamic glob al packet routing in wireless networks," in *Proc. IEEE INFOCOM*, 1997, vol. 3, pp. 1414–1421.
[9]M.Andrews,K.Kumaran,K.Ramanan,A.Stolyar, P. Whiting, and R. Vijaykumar, "Providing quality of service over a shared wireless link,"*IEEE Commun. Mag.*, vol. 39, no. 2, pp. 150–154, Feb. 2001.
[10] M. J. Neely, E. Modiano, and C. E. Rohrs, " Dynamic power allocation and routing for time varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.
[11] M. Kobayashi, G. Caire, and D. Gesbert, " Impact of multiple transmit antennas in a queued SDMA/TDMA downl
ink," in *Proc. 6th IEEE SPAWC*, Jun. 2005, pp. 540–544.
[12] M. J. Neely and R. Urgaonkar, "Optimal backpressure routing in wireless
networks with multi-receiver diversity," *Ad Hoc Netw.*, vol. 7, no.5, pp. 862–881, July 2009.
[13] M. J. Neely, E. Modiano, and C. Li, "Fa
irness and optimal stochastic control for heterogeneous networks," in *Proc. IEEE INFOCOM*,Mar. 2005, vol. 3, pp. 1723–1734.
[14] M. J. Neely., "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2003, LIDS.
[15] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm,"*Queueing Syst.*,vol.50,no.4,pp.
401–457, 2005.
[16] X. Lin and N. B. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proc. 43rd IEEE Conf. Decision Control*,Paradise Island, Bahamas, Dec. 2004, vol. 2, pp. 1484–1489.
[17] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," in *Proc. IEEE INFOCOM*,Mar.2005, vol. 3, pp. 1794–1803.
[18] J. W. Lee, R. R. Mazumdar, and N. B
. Shroff, "Opportunistic power scheduling for dynamic multi server wireless systems," *IEEE Trans.Wireless Commun.*, vol. 5, no. 6, pp. 1506–1515, Jun. 2006.
[19] R. Agrawal and V. Subramanian, "Optimality of certain channel aware scheduling policies," in *Proc. 40th Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, Oct. 2002, pp. 1532–1541.
[20] H. Kushner and P. Whiting, "Asymptotic properties of proportionalfair
sharing algorithms," in *Proc. 40th Annu. Allerton Conf. Commun.,Control, Comput.*, 2002, pp. 1051–1059.
[21] A. Mekkittikul and N. McKeown, "A starvation free algorithm for achieving 100%throughput in an input-queued switch," in *Proc. ICCN*,1996, pp. 226–231.
[22] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijaykumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," *Probab.Eng.Inf.Sci.*, vol. 18, no. 2,pp. 191–217, April 2004.
[23] S. Shakkottai and A. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *Anal. Methods Appl. Probab., Amer. Math. Soc. Transl.*, vol. 207, pp. 185–202, 2002.
[24] R. Berry and R. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.
[25] M. J. Neely, "Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1489–1501, Aug. 2006.
[26] L. Huang and M. J. Neely, "Delay reduction via Lagrange multipliers in stochastic network optimization," *IEEE Trans. Autom. Control*, vol. 56, no. 4, pp. 842–857, Apr. 2011.
[27] D. P. Bertsekas and R. Gallager,*DataNetworks*. Upper Saddle River, NJ: Prentice-Hall, 1992.
[28] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Francisco, CA: Morgan & Claypool, 2010.
[29] B. Sadiq, S. Baek, and G. de Veciana, "Delay-optimal opportunistic scheduling and approximations: The log rule," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1692–1700.
[30] M. J. Neely, "Stochastic optimization for Markov modulated networks with application to delay constrained wireless scheduling," in *Proc. IEEE CDC*, Shanghai, China, Dec. 2009, pp. 4826–4833.
[31] J. K. MacKie-Mason and H. R. Varian, "Pricing congestible network resources," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1141–1149, Sep. 1995.
[32] M. J. Neely and E. Modiano, "Convexity in queues with general inputs," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 706–714, Feb. 2005.
[33] P. Giaccone, E. Leonardi, and D. Shah, "Throughput region of finite buffered networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 2, pp. 251–263, Feb. 2007.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

[34] M. J. Neely, "Energy optimal control for time varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.
[35] L. B. Le, E. Modiano, and N. B. Shroff, "Optimal control of wireless networks with finite buffers," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.
[36] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 841–854, Jun.2011.
[37] M. J. Neely, "Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks," in *Proc. IEEE INFOCOM*, 2011, pp. 1728–1736.
[38] R. Daniels and R. W. Heath, Jr., "An online learning framework for link adaptation in wireless networks," in *Proc. Inf. Theory Appl. Workshop*, San Deigo, CA, Feb. 2009, pp. 138–140.
[39] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, no. 1, pp. 33–37, Jan.–Feb. 1997.