QoS-LI: QoS Loss Inference in Disadvantaged Networks

Vidyaraman Sankaranarayanan and Shambhu Upadhyaya
Computer Science and Engineering
University at Buffalo
Buffalo, NY USA
Email: {vs28, shambhu}@cse.buffalo.edu

Kevin Kwiat
Air Force Research Laboratory
525 Brooks Road
Rome, NY 13441
Email: kwiatk@rl.af.mil

***Abstract* — Quality of Service (QoS) of disadvantaged networks is usually considered from a purely network standpoint in existing works. Adversarial intervention in such networks is not analyzed, nor is it possible to infer if a QoS loss is benign or otherwise. In this work, we analyze the nature of a QoS loss between a remote system and a backend service infrastructure connected over a disadvantaged satellite network. We present a game theoretic framework to infer if a QoS loss is due to benign or malicious factors. An attack on the backend system (DDoS attack) or on the transmitting station (RF Jamming) is considered to be a malicious factor, while a statistical network variation due to random noise is considered to be a benign factor. We then present the implementation of the game theoretic framework to the satellite network, and verify the validity of the idea through simulations in OPNET.**

Keywords – *Disadvantaged Networks, Game Theory, K-Armed Bandit Problem, QoS, Resource Selection, Satellite Networks.*

I. INTRODUCTION

Disadvantaged Networks are defined as networked environments such as wireless and satellite networks, where the very nature of the physical medium restricts effective bandwidth. The effective Quality of Service (QoS) in such networks is measured through different techniques, ranging from QoS for IP over satellite networks [12], to multi-dimensional QoS measures [17] that take into account throughput, delay and loss rate in a unified formula. While many studies have been conducted that address the optimization of resources through resource allocation algorithms [23], routing algorithms [13, 2, 12], etc., no work has been done for inferring the nature of QoS loss over disadvantaged networks.

Consider the scenario where a complex backend system operates over a disadvantaged network, providing services to remote users. Herein, we situate the disadvantaged network as a satellite network, or a *satcom*. Figure 1 shows the schematic of a complex backend system providing service over satellite networks. Assume that a module from the backend system communicates to a corresponding module at the remote client end. The effective QoS between these modules may be reduced at some point in time. Although the mechanisms in the satcom will operate to provide the best QoS, the remote client does not have any idea of the reason behind the QoS loss. The QoS loss may be due to

- a problem on the host platform, or
- a problem with QoS issues with the network, or
- an adversarial manipulation of the network, or
- the backend operating under hostile conditions, or
- a combination of the above factors

All the above scenarios are equally important, and in a time constrained operational mode, it is crucial for the end user to not, for example, terminate the local application on the host machine assuming a localized problem, when in fact, the backend was operating under hostile conditions. This problem is compounded by the fact that a direct communication of a situation cannot always be made on account of the network being disadvantaged or the complexity of the backend system. Depending on the operation scenario, the advantage of knowing the actual reason behind a QoS loss may be beneficial in ways more than one; obviously, the knowledge that the QoS loss stems from adversarial network control is paramount.

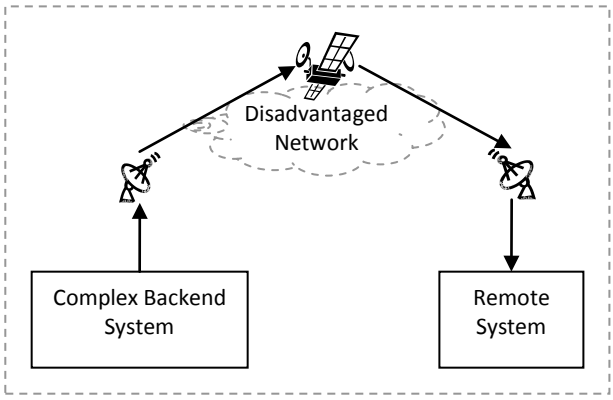

Figure 1: Complex Backend System Servicing Remote Node

As a practical example, consider DARPA's initiative into Self Regenerating Systems (SRS) [8]. Self-Regenerative Systems are complex backend systems that derive their motivation from biological systems and extend the concept of healing [27] into the field of computer science and allied fields. The main goal [8] of Self-Regenerative Systems is to provide cognitive immunity and regeneration of computer systems with granular and scalable redundancy. A number of important and crucial areas ranging from key distribution schemes [5] and

hardware platforms [15] have been studied. However, with the exception of mobile networks [25, 24], a standard assumption across all areas has been the existence of an enterprise level network capable of providing the bandwidth required for the application area under consideration. In this paper, we consider the application of a Self-Regenerative system over a disadvantaged link. Given that the SRS performs as designed, we consider the unique problem of inferring the nature of the QoS loss, i.e., whether a reduced performance is due to a problem with QoS issues with the network, or an adversarial manipulation of the network/the SRS operating under hostile conditions. In this context, this paper views the notion of QoS in a holistic manner that encompasses the remote nodes and the disadvantaged network in between. As exemplified above, in operational scenarios, such notions of QoS, and an inference to the reason behind its loss, is very useful. The practical relevance of this problem lies in situations where networks have been built with a heavy investment in the underlying communications infrastructure. In such networks, it may be feasible to update the end nodes with higher processing power, but changing the communications infrastructure may not be immediately feasible (e.g., satellite networks).

The main focus of this paper is to formulate the scenario in terms of a resource allocation problem and present a conceptual framework to infer the nature of QoS loss. The game theoretic framework presented in this paper is based on the *K-Armed Bandit* [11, 18] problem. This work presents the framework presented in our prior [26] and extends it by illustrating how the theoretical framework is translated to a practical implementation for satellite networks. We also present the viability of such a translation by means of OPNET simulations.

Thus, this paper attempts to address a hitherto unconsidered problem of inferring QoS loss in disadvantaged networks. To the best of our knowledge, this problem has not been addressed before in disadvantaged networks. The contributions of this work can be summarized as follows:

- We present a conceptual framework, based on a game theoretic model to infer the nature of QoS loss in a disadvantaged network.
- A well known game theoretic structure, called the *K-Armed Bandit* problem, is adopted with suitable variations to solve the practical problem in our context.
- The QoS-LI model presented aims to infer the nature of QoS loss as exemplified by:
  - QoS loss due to benign network conditions
  - QoS loss due to adversarial network conditions/loss due to attack on the backend system
- We present the translation of the game theoretic framework to a satellite network and state the assumptions required for the framework to be practical.

The rest of the paper is organized as follows. Section 2 presents the problem formulation. Section 3 presents the model to infer the nature of QoS loss. These sections (and section 1) are present in our prior work [26]; they are presented here for completeness sake. Section 4 details the translation of the framework to a satellite network and present OPNET simulation results validating the approach. Comparison of the work in this paper to existing works is made in Section 5. Concluding remarks and directions for future work is presented in Section 6.

II. Problem Formulation

Consider two end nodes N_1 and N_2 that communicate via a disadvantaged network. The computing power of N_1 and N_2 is not a limitation; the disadvantaged network alone is a limitation in this framework. The nodes N_1 and N_2 are connected via multiple paths between them; these multiple paths may be multiple physical links, (a laptop connecting to a backend server via multiple access points) or multiple logical links on one single physical layer (a single satellite with multiple RF sources for uplinks/downlinks). For each of the k links $\ell_1, \ell_2, \ldots, \ell_k$ we associate the per-packet observed QoS measure, and denote it as $q(\ell_i)$ for all $1 \leq i \leq k$. Without loss of generality, we assume N_1 to be the sending node and N_2 to be the receiving node (at the remote end of the disadvantaged network). N_1 initially transmits packets to N_2 by multiplexing the packets between the k multiple links (physical or logical) according to an algorithm in the set $\mathbf{A}$. The following assumptions are made in the problem formulation:

- The protocol over which N_1 and N_2 communicate is not specified; it may be 802.11 as used in wireless networks [13], or proprietary protocols as used in satellite networks [6, 10, 28].
- The per-packet QoS measure function $q(\ell_i)$ for each of the k links is evaluated by the receiving node (N_2 in this case). Although this can be equated to the packet latency, this function is dependent on the nature of the disadvantaged network.

For a sequence of T packets, the effective QoS as seen by the receiving node N_2 is defined as:

$$Q_T = f(q(\ell_i)) \quad \forall \; i = 1, \ldots, k \qquad (1)$$

Where the function f is dynamically defined at the receiving node N_2 and computed at runtime. The effective QoS for a sequence of T packets is also called the *return at time horizon* T for the multiplexing algorithm $\mathbf{A}$. Three different conditions under which N_1 and N_2 may operate are defined.

- ***Normal operating conditions*** are said to exist if (a) the backend SRS operates normally and (b) the network condition is stable (subject only to natural variations).
- ***Abnormal operating conditions*** are said to exist if there is a stochastic variation in the QoS measure $q(\ell_i)$ of the k links.
- ***Adversarial network conditions*** are said to exist if an adversary controls the k network links ($\ell_1, \ell_2, \ldots, \ell_k$) and deterministically manipulates $q(\ell_i)$.

We now define the problem as follows: Find a set of multiplexing algorithms $\mathbf{A}$ such that Q_T has a well defined range for the three operating conditions (*normal*, *abnormal* and *adversarial*). Hence for a sequence of T packets, depending on the range within which Q_T lies in, an inference may be made about the operating conditions of the network by applying the appropriate multiplexing algorithm in $\mathbf{A}$.

III. QoS Loss Inference (QoS-LI) Model

Towards solving the problem, we formulate a model by leveraging on a game theoretic problem called the *K-Armed Bandit Problem*, and explain its relation to the problem under consideration.

A. K-Armed Bandit Problem

In the *k*-armed bandit problem [11, 18], a gambler in a casino must play between *k* different slot machines and maximize his reward. Each slot machine provides a monetary reward (payoff) based on its characteristics. The reward of each slot machine may be stochastically distributed or deterministically manipulated by an adversary. This classical problem provides a simple, yet illustrative tradeoff between the notion of *exploration* and *exploitation*. On the one hand, the player may try out each arm of the slot machine to discover the one with the best payoff (*exploration*). On the other hand, the player may choose to repeatedly play a particular slot machine believed to give the highest payoff (*exploitation*) at the cost of missing out on a higher payoff slot machine.

The practical relevance of this problem is related to networking in the context of choosing a network path among *k* different paths to find the link with the maximum operational QoS. We use suitable variations of the game theoretic problem, so as to optimize the flow of data and ensure a minimal level of QoS (or alternatively, compute the minimal level of QoS that can be assured). The optimal path decision algorithm has an upper bound on the time taken to find such a path. Together with the network specifics, this measure can directly be translated to an expected QoS level for the receiving node. In our setting, since the network specifics and the algorithms are initially known to both the back-end Self-Regenerative System and to the end application, an inference can be made on the nature of the QoS loss, if any. The game is related to the states of the network (or the *k* slot machines, in the games parlance) in the following manner:

- Each of the *k* slot machines is assumed to be non-identical.
- The payoffs from each of the machines are assumed to be independent stochastic processes, i.e. their distribution may be assumed to be independent (and known or unknown depending on the scenario).
- In our scenario, it may be that the network is under adversarial control, in addition to being disadvantaged. In this case, no stochastic assumptions are made on the distributions of the payoffs of the slot machines [22].
- In certain situations, there might be an external agent that is able to offer 'expert' advice on the states of the network paths, based on past history [22]. The network analysis can also incorporate these inputs to provide an inference on the expected time delay.

The end result of the game analysis is a set of strategies that dictate the packet multiplexing algorithm between the *k* different paths. The performance hit that the network takes during the process is measured by a metric termed "regret" [9] which is the difference between the best possible payoff and current payoff. We do not consider this notion in our framework, as our objective is only to infer the nature of the QoS loss.

B. Model Formulation

The QoS-LI model applies the *K-Armed Bandit* game to the problem under consideration. A high level description is first given where the application of the game is broadly described. Then the QoS-LI Model is formally presented with the algorithmic descriptors.

1) Description

The QoS-LI model operates in two distinct stages. During the first stage, called the ***setup stage***, the end nodes N_1 and N_2 send and receive packets via the *k* multiple links. The sending node N_1 *chooses* an algorithm $a_{best} \in \mathbf{A}$ where a_{best} provides the best possible QoS payoff. During this phase, the optimal operational QoS (Q^*) obtainable is determined by the receiving node N_2. This QoS measure (Q*) is used as the relevance measure, against which decisions are made to detect changes in the network or the backend system.

The second stage, called the ***detection stage***, is activated only if the optimal operational QoS drops below Q^*. After a predetermined time interval (or after packets received), the node N_1 switches to an algorithm $a_{stochastic} \in \mathbf{A}$, where the links payoffs are assumed to be stochastically distributed. The effective *QoS* (Q) is measured; if it converges within the specified time duration of the algorithm to the known limit of $a_{stochastic}$, the network is assumed to be operating under ***abnormal*** conditions. If the QoS convergence is not observed, N_1 switches to the algorithm $a_{adversarial}$, which assumes that the QoS of the *k* links is adversarial-controlled. If the measured operational QoS converges to the know limit for $a_{adversarial}$, the network is said to be operating under ***adversarial conditions***.

Intuitively, the setup stage achieves two purposes, viz. it determines the link ℓ_i among the *k* links has the maximum QoS throughput *and* the operational QoS (Q^*) achievable under normal operating conditions. The nodes now communicate through ℓ_i. In the detection stage, the nodes switch to other stochastic algorithms and multiplex among them to route traffic among the *k* links. If the operational QoS achieved converges to the known value (for the particular algorithm) within the specified time limit, then the assumption of the stochastic algorithms, viz. the link payoffs (QoS) vary stochastically, is correct, and hence the network is under abnormal conditions. If the convergence is not observed, the nodes switch to the adversarial algorithm and check for convergence, i.e. assuming that the links are adversary-controlled.

2) Algorithmic Description

The essence of the QoS-LI model is the set of packet multiplexing algorithms $\mathbf{A}$, that produce different QoS payoffs with different convergence times. We first characterize each algorithm a_i in the set $\mathbf{A} = \{a_1, a_2, \ldots, a_n\}$. For a sequence of T packets, each algorithm a_i is characterized by the QoS payoff Q*, the rate of convergence O(f(T)) and the probability distribution $p_i(t)$ which dictates the probability of choosing link ℓ_i, for the packets t = 1,…,T. Furthermore, let $a_1 = a_{best}$, i.e. that algorithm that affords the best optimal operational QoS Q*, assuming the link payoffs (QoS measures) are stochastically distributed. Let $a_n = a_{adversarial}$, i.e. the algorithm that operates assuming the link QoS is adversarial-controlled. Let all other algorithms a_2 to a_{n-1} be stochastic algorithms, i.e. algorithms that operate assuming the link payoffs (QoS measures) are stochastically distributed, but whose achievable operational QoS is less than a_{best}.

Setup Stage Algorithm

The setup stage follows a ε-greedy strategy ($0 \le \varepsilon \le 1$) towards achieving maximum possible QoS. The strategy and its variants are described in [4]. The work [14] details empirical evaluation that supports the hypothesis that this strategy is the best in terms of achieving greatest payoff. For the k links, the sending node N_1 first performs what is known as an exploration phase, where for the T packets to be sent, each link is randomly (uniformly) tried for εT packets. For the remaining (1-ε)T packets, the link with the highest estimated mean QoS $q(\ell_i)$ is chosen. In its simplest form, this strategy is always suboptimal, since the constant ε value prevents the optimal link from being chosen. A natural strategy, investigated by [19], is to decrease the value of ε progressively, so as to get close to the optimal link asymptotically. With this background, the algorithm a_{best} is described in Figure 2, with the results from [14, 19]. At the end of this stage, the nodes N_1 and N_2 communicate on the link ℓ_m, with the per-packet QoS of $q(\ell_m)$ and operational QoS of at least Q*. Over the operation of the system, $q(\ell_m)$ is constantly monitored and Q* is calculated over a running window, typically over T packets.

Algorithm 1: $a_{best}(\varepsilon_o)$ = The link ℓ_m with maximum Q* = $f(q(\ell_m))$, where $0 \le \varepsilon_o \le 1$

1. Let $\varepsilon_t = \min\{1, \varepsilon_o/t\}$, where t = 1,…,T
2. For ε_tT packets, choose the links ℓ_1 through ℓ_k randomly.
3. Calculate the mean $q(\ell_m)\ \forall\ 1 \le i \le k$.
   Let $q(\ell_m) = \max(q(\ell_i)\ \forall 1 \le i \le k$.
4. Assign probabilities $p(\ell_m) = (1-\varepsilon_t)$
   $p(\ell_i) = \varepsilon_t \qquad \forall\ 1 \le i \le k : i \ne m$
5. Choose and transmit on the links according to this distribution
6. Calculate the mean $q(\ell_i)\ \forall\ 1 \le i \le k$.
   Let $q(\ell_m) = \max(q(\ell_i)\ \forall\ 1 \le i \le k$.
7. After T packets, calculate Q* = $f(q(\ell_i))\ \forall\ 1 \le i \le k$
8. Return link ℓ_m with maximum Q*

Figure 2: Algorithm a_{best}: Achieving optimal operational QoS

Detection Stage Algorithm

The detection stage is activated if the running operation QoS Q falls below Q* for more than a predetermined number of windows, where each running window typically runs over T packets. The sending node N_1 now switches between the stochastic algorithms in $a_{stochastic} \in \mathbf{A}$, i.e., those algorithms that assume that the fluctuation of the link payoffs is stochastic. With this assumption, the algorithm **Exp4** presented in [22] is run, where each of the stochastic algorithms (also called strategies in the game theoretic parlance) is run. This selection algorithm is presented in Figure 3.

Algorithm 2: $a_{stochastic}(\gamma)$ = abnormal or unknown, where $\gamma \in (0,1]$

Initialize: $w_i(1) = 1$, for $i = 1\ldots N$
For t=1,2,…,T

1. Get the 'advice' vectors of each strategy $\xi^1(t),\ldots, \xi^N(t)$,
2. Set $W_t = \sum_{i=1}^{N} w_i(t)$ and
3. Set $p_j(t) = (1-\gamma)\sum_{i=1}^{N} \frac{w_i(t)\xi_j^i(t)}{W_t} + \frac{\gamma}{K}$ for $j = 1,2,\ldots,k$
4. Choose link ℓ_m according to the probability distribution $p_i(t)$
5. For $j = 1,2,\ldots,k$ if $j = \arg(\ell_i)$, $q`(\ell_j) = q(\ell_j)/p_j(t)$, else $q`(\ell_j) = 0$
6. For $i = i,\ldots,N$, update $w_j(t+1) = w_j(t)\ e^{\frac{\gamma \xi^i(t) q(\ell j)}{K}}$
7. After T packets, calculate Q* = $f(q(\ell_j))\ \forall\ 1 \le j \le k$
8. If max(Q*) converges in $O((\log N)^{1/2} \cdot T^{-1/2})$, return abnormal, else return unknown.

Figure 3: **Selection Algorithm**

This is the algorithm **Exp4**, presented in [22], modified for our scenario. The different stochastic algorithms that can be used are the LEASTTAKEN [14] and variants like GREEDYMIX, the SOFTMAX strategy [7] and variants. They are not presented here as they are available in standard literature. Assuming that we have a total of N such strategies, the recommendation of any one strategy (say i, of the N total strategies) is represented as an infinite sequence of probabilities $\xi^i(1), \xi^i(2), \ldots \in [0,1]^k$, where each $\xi^i(t)$ is a probability vector for the k links at time step t. Thus the j^{th} component $\xi_j^i(t)$ of $\xi^i(t)$,represents the probability of choosing link j (of the k links) at time t, by strategy i.

Algorithm 3: $a_{adversarial}(\gamma)$ = adversarial or unknown, where $\gamma \in (0,1]$

Initialize: $w_i(1) = 1$, for $i = 1\ldots k$
For t=1,2,…,T

1. Set $p_i(t) = (1-\gamma)\frac{w_i}{\sum_{j=1}^{K} w_j} + \frac{\gamma}{K}$ for $i = 1,2,\ldots,k$
2. Choose link ℓ_i according to the probability distribution $p_i(t)$
3. For $j = 1,2,\ldots,k$, if $j = \arg(\ell_i)$, $q`(\ell_j) = q(\ell_j)/p_j(t)$, else $q`(\ell_j) = 0$
4. Update $w_j(t+1) = w_j(t)\ e^{\frac{\gamma q`(\ell_j)}{k}}$
5. After T packets, calculate Q* = $f(q(\ell_i))\ \forall\ 1 \le i \le k$
6. If max(Q*) converges in $O(T^{-1/2})$, return abnormal, else return unknown.

Figure 4: Adversarial Algorithm

If the selection algorithm converges within time $O((\log N)^{½} \cdot T^{-½})$ as proved in [22], then the network is said to be under abnormal conditions. If the convergence is not observed, the detection stage switches to the adversarial algorithm, where the convergence is expected in time $O(T^{-½})$ [22]. The adversarial algorithm is given in Figure 4. This is the algorithm **Exp3**, presented in [22], modified for our scenario.

IV. PRACTCAL IMPLEMENTATION

The practical relevance of the QoS-LI models lies in situations where the current communications infrastructure cannot be overhauled due to either a huge investment in it or inherently due to the physical medium characteristics. The *QoS-LI* module assumes the existence of multiple links between the remote system and the SRS backend. This may physically be true in the case of mobile networks (802.11x) where the last link may operate on different channels, with a suitable receiver at the remote system. However, in a satellite network, the uplink and the downlink are different insofar as their transmission characteristics are concerned; they are both time and frequency multiplexed, with each incoming packet stream transmission being subject to a reservation on the uplink and scheduling on the downlink. The uplink is a multiple access channel, while the downlink is a statistically multiplexed broadcast channel. We derive the motivation for the practical translation of the *QoS-LI* module to satellite networks from the work by Pandya et. al, [21], where the authors proposed a dynamic resource assignment algorithm for effective link layer utilization. This work uses a similar approach in terms of link layer resource allocations for obtaining multiple links on the satcom.

In this work, we consider a packet switched military satellite network (similar to [21]), with a provision for dynamic resource allocation on the uplink and downlinks. Each terminal in a satcom is capable of operating in multiple transmission modes (burst rates). The uplink and downlink are thus characterized by the modulation format (QPSK, BPSK, etc), the coding rate and the data burst rates. This triple is referred to as the *transmission mode* [21]. *QoS-LI* uses multiple transmission modes as the logical equivalent of multiple links over the same physical channel. The testing approach in this work is an absolutist one: we verify that if a consistent transmission mode were used, providing a predetermined (statistically variant) QoS level to the remote system, then a variation in the QoS due to a Jammer (at the SRS end) will be detected by a suitable switch of the transmission mode. With this verification, any resource assignment algorithm (predictable or dynamically assigned) may be used for normal communications: when the QoS drops below a predetermined level, the transmission modes may be switched and an inference on the existence of a Jammer could be made based on the observed QoS.

A. Simulation Setup

Figure 5 depicts the simulation setup overview. We simulate the *QoS-LI* module with a single aggregation point, i.e., we assume all streams from the remote backend are aggregated at one point (the ingress). It is also possible for satellite networks to have multiple uplink points (when the number of discrete terminals is high), as shown in Figure 6. In the context of this work, figure 6 depicts the situation when multiple remote terminals send back data to the single backend (the reverse situation of figure 5).

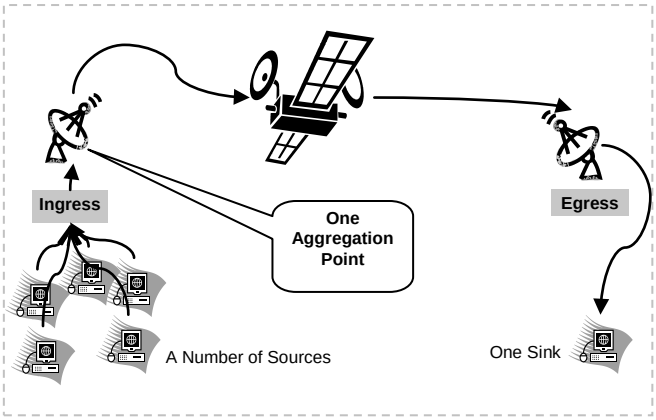

Figure 5: Single Aggregation Point Simulation Setup

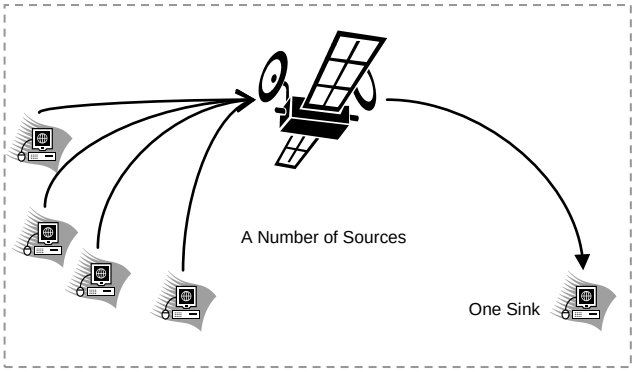

Figure 6: Multiple Uplink Points for Single Sink

The single aggregation point usually transmits the uplink in a time and frequency multiplexed manner, with a scheduling algorithm, to account for different types of streams. For purposes of the simulation, we use a single source traffic distribution pattern (Constant Bit Rate). Other traffic distributions could also be used to detect a QoS drop/pickup after a transmission mode change. A jammer (not shown in figures) is also introduced at the aggregation point (base station) to impede communications to the remote end. The satellite simulations are performed in OPNET [1], using the wireless module.

B. Jammer Characteristics

OPNET includes three types of Jammers for simulation purposes. They are:

- single band jammer
- pulsed jammer
- frequency swept jammer

A single band jammer, as the name suggests, transmits signals on a single predefined frequency band. A pulsed jammer is

usually used to target a frequency hopping system by targeting its power to a narrow spectrum, thereby hitting a minimal set of frequency hops of the target system. Like the single band jammer, it also transmits on a single fixed frequency band, but masks it with a periodic pulse. A frequency sweep jammer concentrates its power on a particular frequency range, which is constantly shifted in order to cover a wide range of operating frequencies. In this work, we use a single band jammer to impede satellite communications. This provides the best test case for the switch in the transmission mode; a pulsed or a frequency swept jammer would have the same (but limited) effect on the outgoing signal, and its effect on the transmission switch would be perceived only in the overlapping pluses or frequency ranges. We used a single band jammer with a similar frequency range and packet source as the SRS sources.

C. OPNET Setup

Figure 7 shows the OPNET setup, with a SRS and a receiver connected only by means of a satellite link. The SRS is modeled as a simple source with a packet multiplexing component that represents the scheduler for traditional satellite uplinks. The satellite receives the SRS signals (operating at the same frequency) and queue's them for processing and transmits them to the receiver. The processing is modeled as a simple delay, with a constant service rate at the queue.

When evaluating resource assignment algorithms, the processing takes form of multiplexing the received streams for broadcast, and in some cases, priority processing. However, in this situation, since we use a single stream, a service delay is appropriate to abstract the normal processing time. The QoS is measured at the receiver, and is currently the end-to-end packet delay in seconds. Note that this delay is at least 250 ms for one trip from the SRS to the remote receiver. The jammer is also similarly set, with frequencies matching the SRS transmitter (which in turn matches the satellite receiver). The jammer is a single band jammer, but with a similar source rate as that of the SRS, instead of the constant rate of 1 per second provided by the default OPNET template.

1) Simulation Parameters and Assumptions

The transmitting nodes' propagation delay is set to 250 ms; the satellite orbit is set to a geosynchronous orbit (hence no orbital paths are defined in OPNET). The `channel-match` and `closure` properties of the transmitting and receiver pair are set appropriately to ensure that the SRS signal does reach the satellite (and the Jammer, when operational, interferes with the SRS-Satellite link). All terminals are assumed to transmit at full power. Since the `channel-match` and `closure` properties are set to ensure reception, the power variation does not change the simulation results in this context.

The OPNET simulations make the following assumptions with respect to the traffic streams and the Jammer.

1. The traffic from the source to the destination is assumed to be a Constant Bit Rate (CBR) stream. In a typical satellite uplink, multiple sources 'submit' their streams for transmission; a FTP server might submit a file upload, a VOIP/multimedia server would submit a real-time stream. These streams are frequency and time multiplexed, based on a reservation strategy that attempts to deliver real-time traffic with QoS guarantees while maintaining fairness for the non-real-time (FTP) traffic. Although we do not introduce multiple streams, the simulation results hold true since the QoS metric would include the summation of all streams (which would be affected by a Jammer).
2. We assume that the Jammer cannot predict the transmission mode switching pattern; if an adversary were able to do so, then the Jammer transmission mode could be synced with the satellite transmission, thereby affecting the QoS in a more granular level. In this case, although we would (theoretically) be assured of a optimal QoS convergence [22], the adversary, by suitably manipulating the QoS, could convince the remote end of an attack on the SRS (as opposed to a Jammer operation).
3. The simulation uses only one terminal for testing all the burst rates; in practice, no single terminal will be capable of handling all the burst rates and power requirements. However, a typical satellite ground station will have multiple terminals capable of handling the required burst rates: this is a standard assumption for satellite simulations (unless the simulation involves path fading and/or directional antenna characteristics).

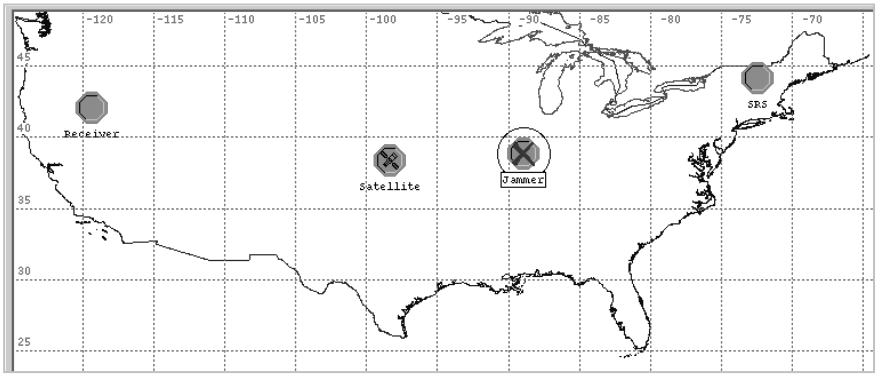

Figure 7: OPNET Simulation Setup

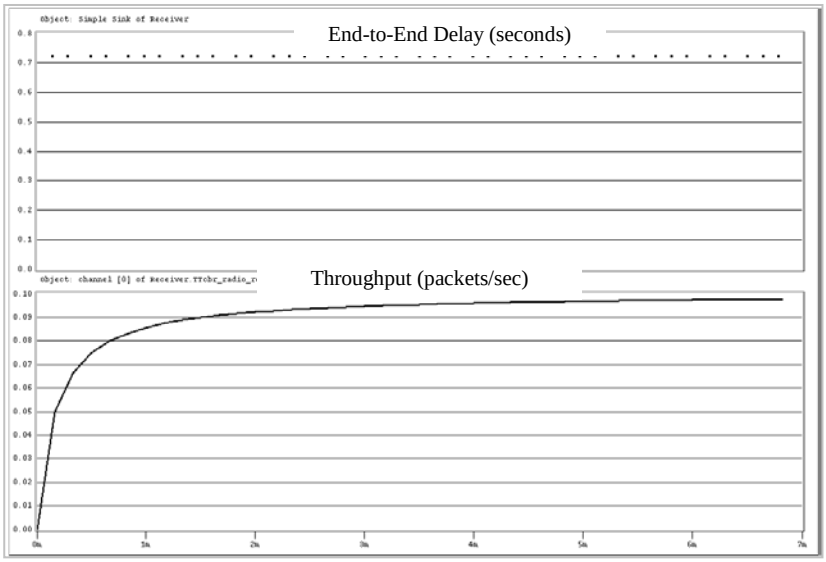

Figure 8: Throughput and End-to-End Delay curves: Uplink (BPSK, 1Mhz, 1000 kbps); Downlink (BPSK, 4MHz, 8000 kbps)

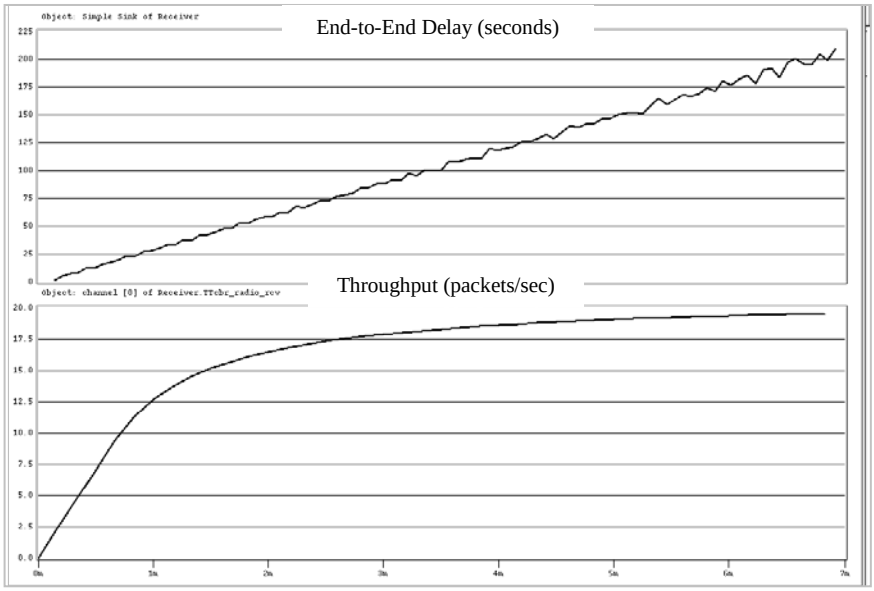

Figure 9: Throughput and End-to-End Delay curves: Uplink (BPSK, 4Mhz, 8000 kbps); Downlink (BPSK, 4MHz, 8000 kbps)

2) Simulation Results

The uplink and downlink are characterized by the bandwidth, the burst rate and the modulation. As derived from [21], the combination of the typical values for satellite operation are form around 20 transmission modes, each with a different power requirement. The values for the Bandwidth are 4/16/64/256 MHz and the data burst rates are 2/8/32/128 Kb/time slot, where a time slot is 500 ms (the round trip time). For normal operation without a Jammer, with uplink (BPSK, 1Mhz, 1000 kbps), downlink (BPSK, 4MHz, 8000 kbps) with a Constant Bit Rate (CBR) source, the throughput and end-to-end delay curves are shown in Figure 8. The throughput levels off and the end-to-end delay remains constant, around 0.7 seconds. Similarly, the throughput and end-to-end delay curves for Uplink (BPSK, 4Mhz, 8000 kbps), downlink (BPSK, 4MHz, 8000 kbps) with CBR source is shown in Figure 9. As expected, the end-to-end delay is rises linearly since the data burst rates on the uplink and downlink are equal (the processing time on the satellite keeps packets in the service queue). With a limited buffer size on the satellite, this value would level off soon. With an uplink (BPSK, 4Mhz, 8000 kbps), and downlink (BPSK, 8MHz, 16000 kbps) with CBR source, the curves shown in Figure 10 show a structure similar to Figure 8, but for an improved end-to-end delay (since the downlink bandwidth and burst rates are both higher). The throughput is also seen

to approach its asymptotic value around 1 minute and 31 seconds.

Thus, as the transmission mode is changed, the effective QoS (measured in terms of end-to-end delay) is also seen to change. This provides validation that different transmission modes can serve as the logical equivalent of multiple links over the same channel. A Jammer that is successful in lowering the QoS in one transmission mode will not be as effective in another (switched) transmission mode. Figure 11 shows the end-to-end delay and throughput curves with a jammer operating after 210 seconds of SRS/satellite operation. As expected, the end-to-end delay immediately drops and the throughput gradually drops after the jammer starts operating (the jammer operates on the same frequency bandwidth as the SRS-satellite link).The hypothetical throughput curve for the SRS-satellite link after a transmission mode switch is shown in Figure 12 (this is Figure 11 spliced with figure 8; they are operating in a different transmission modes). With a transmission mode switch, the single band Jammer is ineffective (in an absolutist sense; a pulsed or frequency sweep jammer would have some effect even after a transmission switch, but the effective observed QoS would still rise, albeit at a slower pace): in this case, the frequencies do not overlap; in other situations, the data burst rate / modulation also play a role. The jammer effectiveness, unlike the hypothetical situation in Figure 12, may not be completely lowered, but the remote end would perceive a raise in the effective QoS, thereby leading to the inference of a Jammer at the backend (as opposed to an attack on the SRS). To verify the jammer ineffectiveness after a transmission mode switch, we changed the transmission modes of the SRS to satellite link (without making the corresponding change for the jammer transmission mode) and found the throughput to be unaffected (with no frequency overlap).

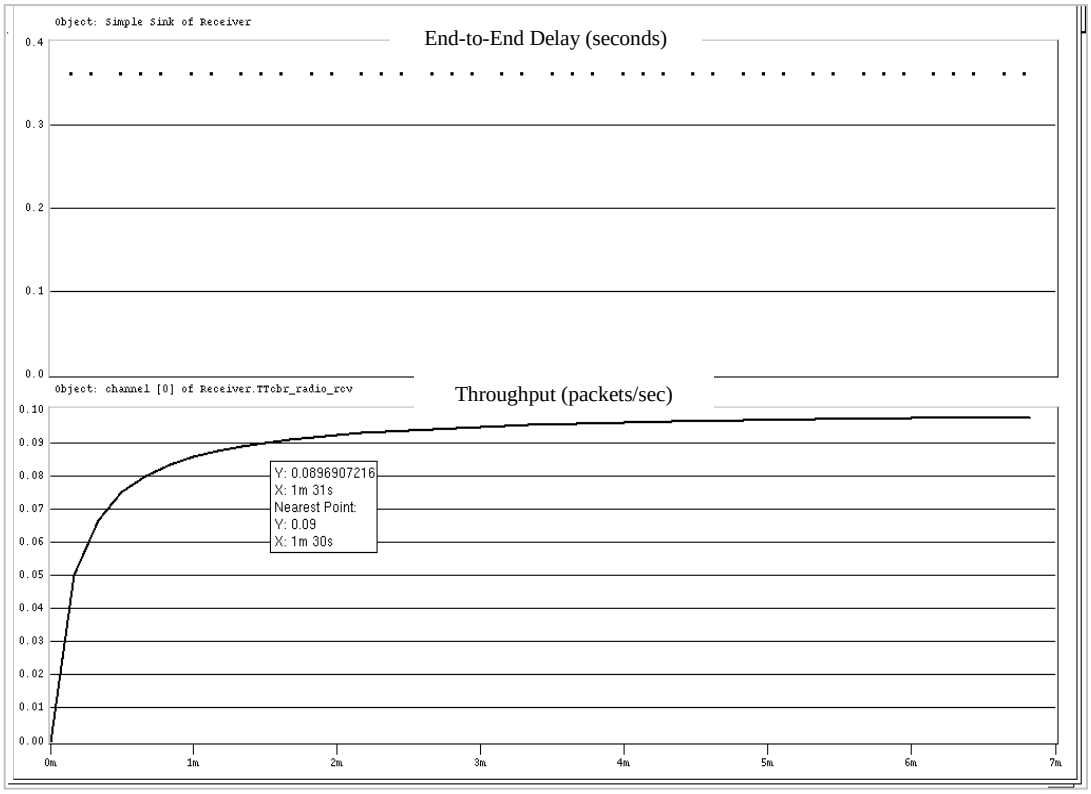

Figure 10: Throughput and End-to-End Delay curves: Uplink (BPSK, 4Mhz, 8000 kbps); Downlink (BPSK, 8MHz, 16000 kbps)

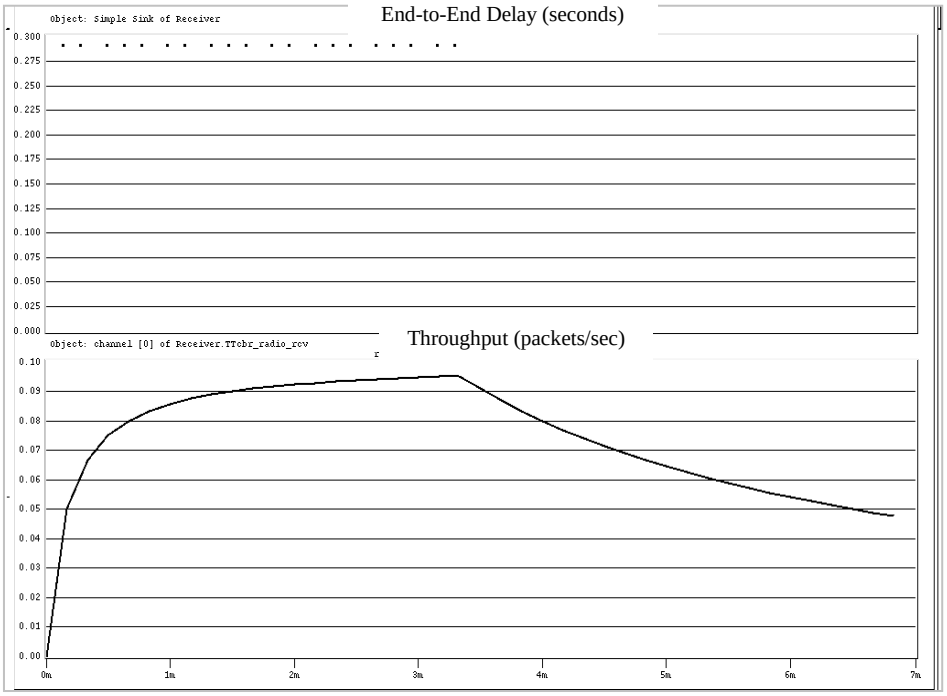

Figure 11: Throughput and End-to-End Delay curves: Jammer operating with exaggerated data rate of 512 kbps

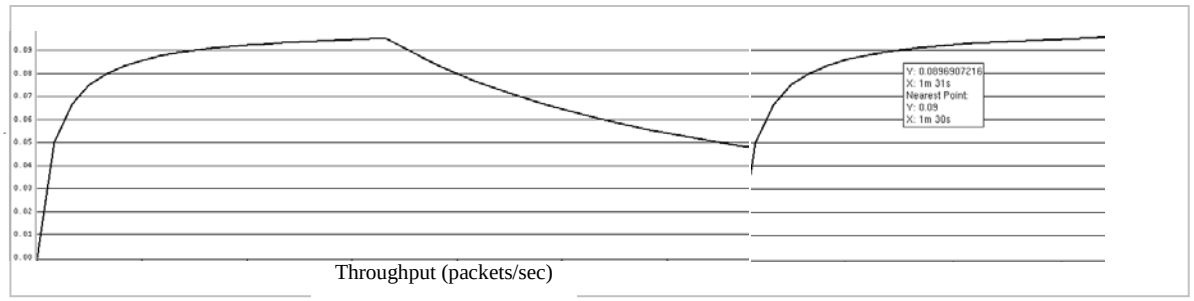

Figure 12: Hypothetical Operation after transmission mode switch

The metric for the QoS must be chosen depending on the system under consideration. In a simulation with an exponential source, in the presence of a Jammer, the throughput was found to be relatively constant (due to packets arriving at an exponential rate), but the end-to-end delay showed a gradual decrease. Finally, we modified the SRS source to have different sources, namely a Constant Bit Rate (CBR) source, an exponential distribution source and a Poisson source. With transmissions at a burst rate of 512 kbps and BPSK modulation, Figures 13 and 14 show the throughput and end-to-end delay curves without and with a jammer respectively. The throughput decreases (at a lower rate, due to the burst rate of 512 kbps), though not at smoothly as in Figure 11, where we had a single source. Thus, even with multiple sources, a drop in QoS in terms of observed throughput (not end-to-end delay, as in previous cases) can be observed, and a transmission switch should result in an improved QoS.

3) Applicability

In a typical satellite system, there are many other mechanisms in place for ensuring QoS. Dynamic Resource Allocation (DRA) algorithms like the one proposed in [21] change the transmission mode every epoch. These allocation mechanisms can proceed independently of the scheme proposed in this work. Once a drop in the QoS is observed, the remote system can initiate a deterministic (or dynamically evaluated) transmission mode switching strategy to infer the nature of the QoS loss. This argument applies to other link layer optimizations for QoS. The remote system is typically a individual terminal [2] that are marketed by commercial entities. The *QoS-LI* module can be implemented as a software component on the remote system or integrated as a plug-in to existing QoS optimization mechanisms. Depending on the rate of the QoS drop/rise, the inference can be made in real-time, in as low as 10 epochs (5 seconds). The actual time convergence, however, depends on the particular application context and traffic characteristics at the transmission station. The existence of PEPs [3] does not affect the *QoS-LI* module; the placement options for the *QoS-LI*

module with respect to PEPs have been discussed in our prior work [26].

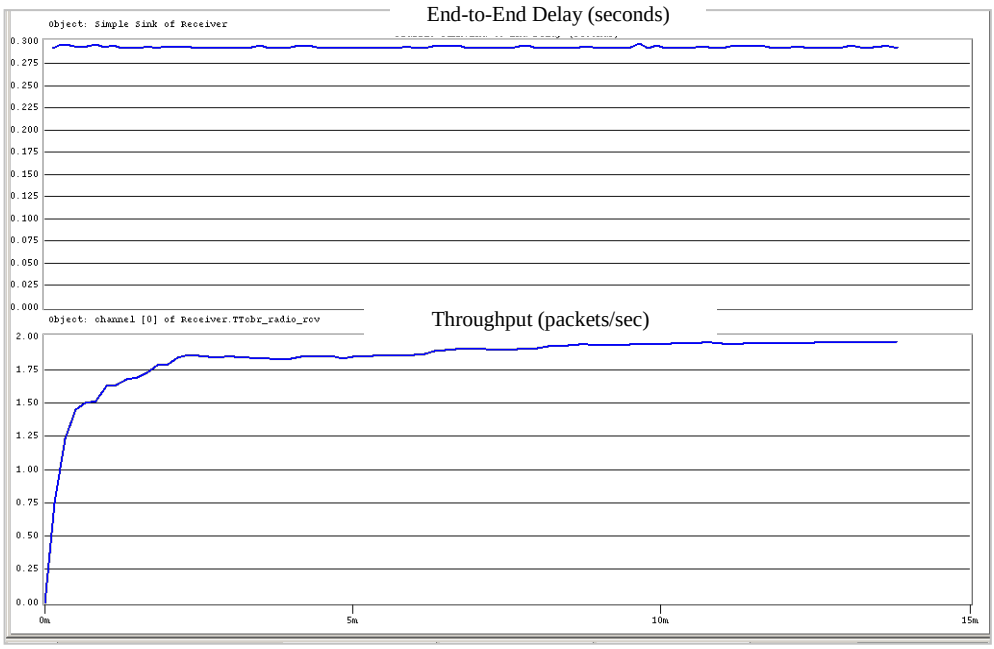

Figure 13: SRS with multiple sources, without Jammer

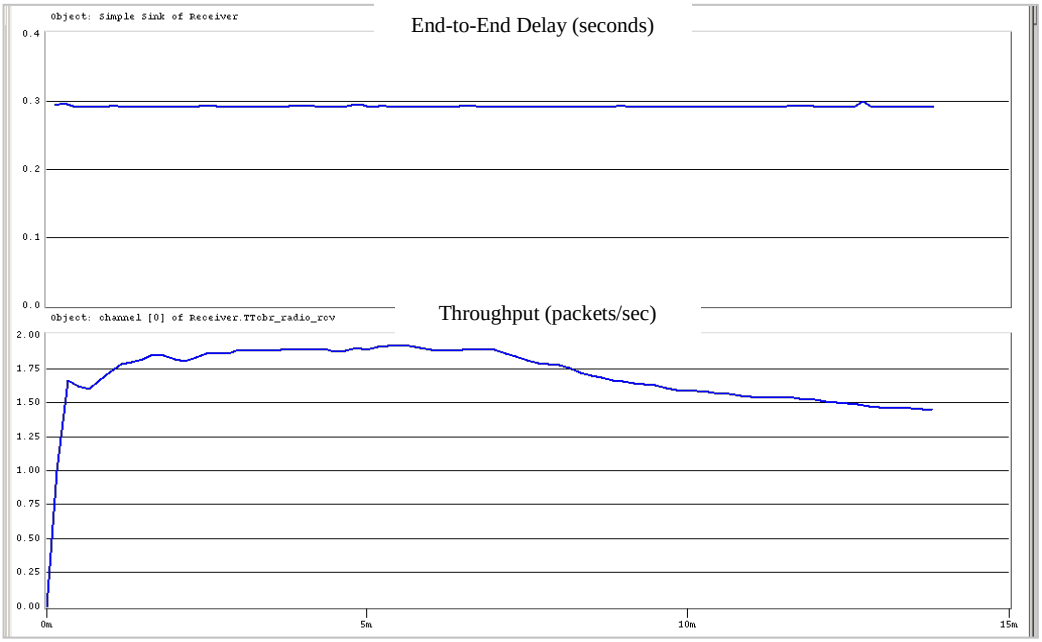

Figure 14: SRS with multiple sources, with Jammer

V. COMPARISON WITH RELATED WORK

The QoS of disadvantaged networks has been investigated from different angles. The focus on improving QoS in disadvantaged networks began with the proposal of Performance Enhancing Proxies [3]. PEPs have been used in wireless and mobile networks [20] and in satellite networks [29] with latency greater than typical 802.11 wireless networks. Most of the PEPs have been made application transparent to ensure compatibility with existing infrastructures. In situations where end-to-end security is required, PEPs are not useful, unless the deploying agency trusts the service provider or uses the security solutions provided by the PEP manufacturer). All these efforts, however, have been focused at mitigating link related degradation. Similarly, other protocols are tuned towards the characteristics of the particular link layer characteristics. For example, the protocol STP [28] is tuned for satellite networks. A common underly-

ing theme is all these works is the consideration of QoS only from the network viewpoint.

In specialized situations where the remote network receives service from a complex backend system such as a Self-Regenerative System (SRS) [16], the notion of QoS is larger than just the network. The receiving system would be greatly benefited by the knowledge of the actuators behind the QoS loss, if any. In such systems, QoS loss may occur due to a variety of reasons, including adversarial network control. Since all previous works do not consider security while building up their metrics of QoS, they are unable to handle any malicious intervention at any part of the system, be it at the network or at the backend. Our work takes a step further in this line of thought, providing a non-intrusive mechanism of QoS Loss Inference. The basis of our framework is derived from a game theoretic model, called the *K-Armed Bandit problem* [11, 18]. This problem has been well studied with respect to the rewards obtained in optimal resource selection. While other resource allocation algorithms have been proposed in the field of satellite networks [10], our work is different in two aspects; first, we are concerned with resource selection, not allocation. Secondly, the performance of other schemes is measured with respect to the QoS gained or lost in the end. We, on the other hand, using the timing estimates to provide an inference of the nature of the QoS loss. Thus while ensuring the optimal resource selection algorithm, we also use it to detect possible violations, and also the nature of such violations. The *K-Armed Bandit* problem has been widely studied, with different algorithms proposed for different conditions and restrictions on the input parameters. These situations can easily be mapped to the link layer characteristics of the network. The simulations of these algorithms [14] indicate the viability of the approach.

VI. Conclusion and Future Work

With the operation of Self-Regenerative Systems [16] over disadvantaged networks as the motivation, the efforts in this paper have been directed towards formulating the QoS loss inference problem and presenting a solution methodology in terms of resource selection algorithms. The primary utility of the QoS-LI model proposed in this paper lies in disadvantaged networks, where the remote system must not only know about the QoS loss, but also its nature, particularly if the backend system is under attack. In this work, we have:

- presented a game theoretic framework based on the *K-Armed Bandit* problem.
- used the notion of transmission modes as the logical equivalent of multiple links in satellite networks
- verified through OPNET simulations that switching between multiple transmission modes in the presence of a jammer does cause a change in the observed QoS

The QoS metric used was end-to-end delay in the simulations; the metric used for a transmission station depends on the traffic characteristics for that particular station and must be chosen during setup time.

Future extensions to this work include fixing the topology of an SRS and evaluating the QoS variations that are expected to occur if the SRS were under attack. Such studies need to be made for specific scenarios with known traffic characteristics. This work also has a rich set of future extensions to be investigated. First and foremost is the extension of the framework to other types of disadvantaged links (narrowband users, dial-up connections, cellular communications, etc.). In each of these links, adversarial intervention may have different connotations, depending on the context of the application. The existence of multiple logical links is an issue that would need further investigation in each area. The base game theoretic framework used in this work would be a good starting point for these different communications.

Acknowledgments

Approved for Public Release; Distribution Unlimited: WPAFB 08-0350, 12-Feb-2008.

References

[1] *OPNET Technologies, Inc.: http://www.opnet.com*, 2007.

[2] AOS SkyPipe, *Secure Communications Link Optimization from AOS: http://www.aosusa.com/scrunch.html*, 2004.

[3] J. Border, M. Kojo, J. Griner, G. Montenegro and Z. Shelby, *Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations*, RFC Editor, 2001.

[4] C. J. C. H. Watkins, *Learning from Delayed Rewards*, Cambridge University, 1989.

[5] B. Carlo, D. A. Paolo, S. Alfredo De and L. Massimiliano, *Design of Self-Healing Key Distribution Schemes*, Designs, Codes and Cryptography, 32 (2004), pp. 15-44.

[6] C. Chao and E. Eylem, *A routing protocol for hierarchical LEO/MEO satellite IP networks*, Wirel. Netw., 11 (2005), pp. 507-521.

[7] D. Luce, *Individual Choice Behavior*, Wiley, 1959.

[8] DARPA, *Self Regenerating Systems: http://www.darpa.mil/ipto/programs/srs/*, 2004.

[9] Dean P. Foster and Rakesh Vohra, *Regret in the on-line decision problem*, Games and Economic Behavior, 29 (1999), pp. 7–36.

[10] C. Dennis, R. Bo, J. P. Gregory and D. Son, *A medium access control protocol for real time video over high latency satellite channels*, Mob. Netw. Appl., 7 (2002), pp. 9-20.

[11] H. Robbins, *Some aspects of the sequential design of experiments*, Bulletin American Mathematical Society, 55 (1952), pp. 527–535.

[12] H. Hlavacs, M. Haddad, C. Lafouge, D. Kaplan and J. Ribeiro, *The CODIS content delivery network*, Comput. Netw. ISDN Syst., 48 (2005), pp. 75-89.

[13] IEEE Working Group, *IEEE 802.11 standards: http://standards.ieee.org/getieee802/portfolio.html*, 1997.

[14] Joannès Vermorel and Mehryar Mohri, *Multi-Armed Bandit Algorithms and Empirical Evaluation*, *ECML*, 2005.

[15] P. K. Lala and B. K. Kumar, *An Architecture for Self-Healing Digital Systems*, J. Electron. Test., 19 (2003), pp. 523-535.

[16] Lee Badger, *Self Regenerative Systems: http://www.darpa.mil/ipto/Programs/srs/index.htm*, 2004.

[17] X. N. Liu and S. J. Baras, *Modelling multi-dimensional QoS: some fundamental constraints: Research Articles*, Int. J. Commun. Syst., 17 (2004), pp. 193-215.

[18] Michel Benaïm and Gerard Ben Arous, *A two armed bandit type problem*, International Journal of Game Theory, 32 (2003), pp. 3.

[19] N. Cesa-Bianchi and P. Fischer, *Finite-Time Regret Bounds for the Multiarmed Bandit Problem*, *15th International Conference on Machine Learning (ICML 1998)*, Morgan Kaufmann, San Francisco, CA, 1998, pp. 100-108.

[20] R. Pablo and F. Vitali, *Performance of peps in cellular wireless networks*, *Web content caching and distribution: proceedings of the 8th international workshop*, Kluwer Academic Publishers, 2004, pp. 19-38.

[21] J. Pandya, A. Narula-Tam, H. Yao and J. Wysocarski, *Network layer performance of a satellite network with dynamic link-layer resource allocation*, International Journal of Satellite Communications and Networking, 25 (2007), pp. 217-235.

[22] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund and Robert E. Schapire, *Gambling in a Rigged Casino: The adversarial multi-armed bandit problem In Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 1998.

[23] F. Rima Abi and T. Samir, *Connection admission control and comparison of two differentiated resources allocation schemes in a low earth orbit LEO satellite constellation*, Wirel. Netw., 10 (2004), pp. 245-258.

[24] M. Rui and I. Jacek, *Regenerating Nodes for Real-Time Transmissions in Multi-Hop Wireless Networks*, *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04) - Volume 00*, IEEE Computer Society, 2004.

[25] M. Rui and I. Jacek, *Reliable Multipath Routing with Fixed Delays in MANET Using Regenerating Nodes*, *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, IEEE Computer Society, 2003.

[26] S.Vidyaraman, S. Upadhyaya and K. Kwiat, *QoS-LI: QoS Loss Inference in Disadvantaged Networks*, *Proceedings of 2007 IEEE International Symposium on Ubisafe Computing (UbiSafe '07)*, Niagara Falls, Ontario, Canada., 2007.

[27] G. Selvin., D. Evans and S. Marchette, *A biological programming model for self-healing*, *Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems: in association with 10th ACM Conference on Computer and Communications Security*, ACM Press, Fairfax, VA, 2003.

[28] T. Henderson and R. Katz, *Satellite transport protocol (STP): An SSCOP-based transport protocol for datagram satellite networks*, *Workshop on Satellite-Based Information Services (WOSBIS)*, Budapest, Hungary, 1997.

[29] D. Velenis, D. Kalogeras and S. B. Maglaris, *SaTPEP: A TCP Performance Enhancing Proxy for Satellite Links*, *Proceedings of the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications*, Springer-Verlag, 2002.