The Time-Correlated Update Problem

Joshua Reich, Vishal Misra, and Dan Rubenstein
{reich, misra, danr}@cs.columbia.edu

1. INTRODUCTION

Recent advances in the fields of sensor networks and mobile robotics have provided the means to place monitoring/sensing equipment in an increasingly wide variety of environments - a significant proportion of which can reasonably be expected to lack traditional network connectivity characteristics [5] [8]. Challenged networks, operating under significant sets of constraints in which disconnected paths and long delays are normal events, have come to be known as Delay/Disruption Tolerant Networks (DTN) [2]. Some examples of environments in which DTN techniques may be required include remote or vast domains such as underground, underwater, outer-space, Arctic, and mountainous environments.

Most often DTN are used to address data collection and monitoring tasks [6]. Among other possibilities, the end user of the data produced by DTN nodes may be interested in either (1) retrieving the maximal amount of data (long-term system monitoring - e.g., carbon dioxide consumption in temperate rain-forest) or (2) retrieving the most up-to-date data (situational awareness - e.g., current weather conditions at a mountain peak). While there has been significant examination of the former case, our work is the first of which we are aware to examine the latter [7] [9] [1]. Instead of total throughput, our metric of interest is the expected lag between the data available to the end user and the most current data produced by the monitoring equipment. In other words, we examine the problem of engineering a DTN to provide minimal-lag situational awareness.

The class of DTN we examine are defined by four essential characteristics of their end-to-end connections. Specifically the end-to-end connections experience high latency, low bandwidth, high loss rates, and are only intermittently connected. As a direct consequence of these characteristics, the RTT of connections in the DTN may often be too long to allow for the effective use of a feedback loop in the form of an acknowledgment-retransmission scheme. By the time retransmissions sent from the source are received, the retransmitted information is in all likelihood too old to be of significant utility. Consequently, we examine techniques that improve the throughput of up-to-date data without use of feedback from the recipient. This being achieved through the combination of forward error correction (FEC) techniques and exploitation of time correlations in the underlying data stream. Note, each of these characteristics is considered in light of the underlying application needs and data stream, which implies even networks with fairly high performance may be challenged domains for certain classes of applications.

In this study, we define an end-to-end DTN model and an accompanying data production model. We then consider the performance of patterned memory-clearing sending policies. Subsequently we examine an oracle-based policy, deriving the theoretical optimum performance bound of such a system. We compare this bound with the class of α parameterized stochastic sending policies and show a convergence of this policy class with the theoretic bound as node memory windows increases towards infinity.

2. MODEL

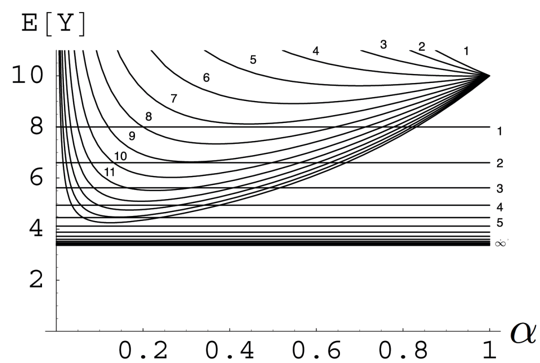

Figure 1: E[Y] for optimal α approaches oracle policy as memory window increases from 1 to 15 ($p_{tg} = 0.1, p_{tu} = 0.3$)

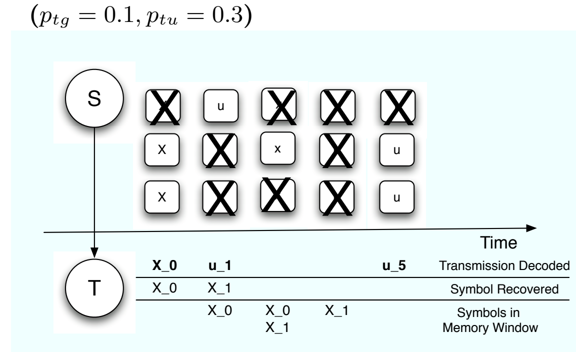

Figure 2: graphical depiction of model with $m = 2, m' = 1, l = 2$

We model the communication between a source S and a sink T as transmission over a channel with Bernoulli losses, perhaps the most basic channel loss model. While this model can be most straightforwardly applied to DTN in which S and T directly connect to one another, it may also be applied to connections in an overlay network.

We examine a discrete-time model in which at every timestep S produces a new collection of data items, referred to as a *snapshot*. The snapshot produced by S at time i is denoted by the symbol x_i. All x_i are assumed to have the same size and can be sent in m packets where $m = \frac{|x_i|}{MSS}$ and MSS is the maximum segment size of the packets. In each time-step, S can send a total of k packets to T. Packet loss across the channel is Bernoulli with probability q (probability of successful transmission being $p = 1 - q$) [3]. Through the use of standard FEC/coding approaches S may encode x_i in $n \geq m$ total packets, any m of which when received at T will successfully decode to produce x_i. We call such a success a *recovery*.

If the underlying data is time-correlated (i.e., the underlying data stream is drawn from some underlying physical phenomenon such as a vector of physical measurements taken at a weather observatory), an update based on previous snapshots may be sent in lieu of an entire snapshot. We assume that symbol x_i is related to symbol x_{i-j} in such a way that x_i can be produced given x_{i-j} and a subsequent update symbol u_i where $|u_i| < |x_i|$. Specifically we consider the case where l updates can fit in $m' < m$ packets. Consequently, if a collection of m' packets containing $\{u_{i-l+1}, \ldots, u_i\}$ is received at T, x_i can be decoded if any of $\{x_{i-l+1}, \ldots, x_{i-1}\}$ has been previously recovered by T.

Let $Y_t = t - i$ be the data delay (delay the data experiences in arriving at T) at time t, where $i = \max i'$ for all $x_{i'}$ that have been received at T by time t. Assuming no feedback channel is available, successful transmissions take approximately constant time, and that a node is limited to either encoding a snapshot or an update in a given time-step - what sending policy should S adopt in order to minimize $E(Y_t)$?

2.1 Update and Snapshot Transmission Probabilities

Let: p_{tx} be the probability successful snapshot transmission, p_{tu} be the probability successful update transmission, and X_i be the r.v. corresponding to whether snapshot i is recovered.

Following immediately from these definitions, we see that

$$p_x = p_x(i) = p_{tx} = P\{X_i = 1\} = \sum_{j=m}^{k} \binom{k}{j} p^j q^{k-j}$$

while for an update

$$\begin{aligned} p_u(i) &= P\{X_i = 1\} \\ &= P\{X_i = 1 | X_{i-1} = 1 \vee \ldots X_{i-l+1} = 1\} \\ &\quad * P\{X_{i-1} = 1 \vee \ldots X_{i-l+1} = 1\} \\ &= p_{tu} P\{X_{i-1} = 1 \vee \ldots X_{i-l+1} = 1\} \\ &= \sum_{j=m'}^{k} \binom{k}{j} p^j q^{k-j} P\{X_{i-1} = 1 \vee \ldots X_{i-l+1} = 1\} \end{aligned}$$

When not discussing a particular time i we simply refer to the probability of successful recovery from transmission of an update as p_u.

While specific values for k, l, m, m', q will determine the values of the transmission probabilities p_{tx} and p_{tu}, for the remainder of our discussion we will examine the relationship of a policy to the transmission probability values. This will provide our results with maximal generality, applicable also to observed values of p_{tx} and p_{tu} across a channel structure potentially more complex than the one used to motivate our model.

2.2 Non-optimality of Snapshot-only Sending Policies

Clearly, the first thing worth proving is that utilizing updates will help improve the sending policy's performance, which we now show. Consider a policy ϕ sending a sequence of two snapshot packets followed by an update packet where $l = 2$, $p_{tx} = 0.1$, and $p_{tu} = 0.9$. Then $p_u = p_{tu}(1 - (1 - p_{tx})^2) = (0.9) * (1 - (1 - 0.1)^2) = 0.171$ and $E(p_\phi) = \frac{p_x + p_x + p_u}{3} = \frac{0.1+0.1+0.171}{3} = 0.124$, proving the snapshot-only policies may prove non-optimal (in fact, this holds true in almost all realistic scenarios).

2.3 Update-Memory Clearing Policies

Initially one might wonder if there exist values of p_{tu} for which an update-only policy might conceivably converge to some $p_u(i) = p_u(i-1)$. But this is clearly not the case as the p_u for a sequence of packets comprised solely of updates that starts with a known success $X_0 = 1$ will be p_{tu} times 1 minus the probability of having at least l consecutive failures in $i-1$ Bernoulli trials where each trial has probability q of failure, a product which converges to zero as l approaches infinity.

This provides an alternate formulation of $p_u(i)$ where we first solve for $P\{run(i)\}$: the probability of at least l consecutive successes in i Bernoulli trials with probability q. $P\{run_l(i)\}$ is equal to the probability of at least l consecutive successes in $i - 1$ trials plus the probability the l^{th} consecutive success was achieved on the i^{th} trial. The former quantity is simply $P\{run(i-1)\}$ while the latter is q times the probability of a run ending in exactly $l - 1$ consecutive successes which contains no larger run in $i-1$ trials. But this probability is the probability of one failure followed by $l - 1$ successes with a sequence of $i-1-[(l-1)+1] = i-l-1$ trials in which fewer than l consecutive successes occur - which is simply $1 - P\{run(i-l-1)\}$. Hence:

$$\begin{aligned} &P\{run(i)\} \\ &\quad = P\{run(i-1)\} + q * q^{l-1} p(1 - P\{run(i-l-1)\}) \\ &\quad = P\{run(i-1)\} + q^l p(1 - P\{run(i-l-1)\}) \end{aligned}$$

and

$$p_u(i) = p'(1 - P\{run(i-1)\})$$

As it turns out this recursion was first noted in 1718 by De Moivre and has not yielded an analytical solution yet [4]. However it can be numerically approximated with relative ease, making the above analysis a useful one.

2.4 Stochastic Policies

In empirical comparison with several classes of non-memory clearing policies, the probability of successful recovery for the best memory clearing policy is almost always dominated by that of policies which do not clear the memory window . This is because memory clearing policies fail to take full advantage of updates (although they do provide a useful lower bound on the best achievable policy). As patterned policies that do not clear the update-memory window are significantly more difficult to analyze, we examine the set of stochastic policies.

Under an α-snapshot sending policy the source will choose to send a full snapshot with probability α and an update with probability $1 - \alpha$.

We wish to determine $P\{X_i = 1\}$ in the steady-state. In other words, we want to calculate the probability of successful snapshot recovery under an α-snapshot sending policy.

Let $\{N(i), i \geq 0\}$ be a counting process and Y_n denote the time between the $(n-1)^{st}$ and n^{th} events of this process. Initially the only way for a success to occur is if a snapshot is successfully sent. Thus the time taken until the first success at Y_1 can be considered the start-up phase. However, once the first success has occurred all subsequent inter-arrival periods Y_n have identical and independent distributions. This is because the probability of the successful recovery of a snapshot sent more than j time-steps after the last success is:

$$P\{Y > i\} = \left\{ \begin{array}{cc} q_a^i & i \le l \\ q_a^l q_b^{i-l} & i > l \end{array} \right\}$$

letting $p_a = \alpha p_{tx} + (1-\alpha)p_{tu}, q_a = 1 - p_a, p_b = \alpha p_{tx}, q_b = 1 - p_b$ - which is identical and independent for all $Y_n, n > 1$.

Consequently, after the start-up phase, $N(i)$ is a renewal process and the probability of success at any point in time i during this process is:

$$P\{X_i = 1\} = \lim_{t\to\infty} \frac{N_t}{t} = \frac{1}{\lim_{t\to\infty} \frac{t}{N_t}} = \frac{1}{\lim_{t\to\infty} \frac{\sum_{i=1}^{N_t} Y_i}{N_t}} = \frac{1}{E[Y_n]}$$

We now solve for $E[Y_n]$

$$\begin{aligned} E[Y_n] &= \sum_{i=1}^{\infty} iP\{Y = i\}) \\ &= \sum_{i=0}^{l} P\{Y > i\} + \sum_{i=l+1}^{\infty} P\{Y > i\} \\ &= \sum_{i=0}^{l} q_a^i + \sum_{i=l+1}^{\infty} q_a^l q_b^{i-l} \\ &= \sum_{i=0}^{l} q_a^i + q_a^l \sum_{i=0}^{\infty} q_b^i \\ &= 1 + \frac{q_a - q_a^{l+1}}{1 - q_a} + q_a^l \left(\frac{1}{p_b} - l\right) \end{aligned}$$

2.5 Optimal Policy

We provide a theoretic upper bound on performance for any sending policy ϕ. Consider a situation in which perfect, instantaneous feedback is available to S. Formally, if at the beginning of time-step $i+1$, S can consult an oracle that answers TRUE if x_i was recovered by T and FALSE otherwise. Then by keeping track of the last l answers, S can choose the optimal action, defining a policy $\phi_{optimal}$: if any of the past l snapshots were successfully received, send an update; otherwise send a snapshot.

$$P\{Y^{\phi_{optimal}} > i\} = \left\{ \begin{array}{cc} (1-p_{tu})^i & i \le l \\ (1-p_{tu})^l (1-p_{tx})^{i-l} & i > l \end{array} \right\} \quad (1)$$

and by analogous logic to the derivation of $E[Y_n]$

$$\begin{aligned} &E[Y^{\phi_{optimal}}] \\ &= \sum_{i=0}^{l-1} (1-p_{tu})^i + \frac{(1-p_{tu})^l}{p_{tx}} \\ &= 1 + \frac{(1-p_{tu}) - (1-p_{tu})^{l+1}}{p_{tu}} + (1-p_{tu})^l \left(\frac{1}{p_{tx}} - l\right) \end{aligned}$$

2.6 Convergence

We now prove that for the optimal value of α, α-policies ϕ_α approach the oracle policy as memory window goes to infinity.

$$\begin{aligned} \lim_{l\to\infty} E[Y_{\phi_\alpha}] &= \lim_{l\to\infty} 1 + \frac{q_a - q_a^{l+1}}{1-q_a} + q_a^l \left(\frac{1}{p_b} - l\right) \\ = \lim_{l\to\infty} 1 + \frac{q_a}{1-q_a} &= \lim_{l\to\infty} 1 + \frac{1 - \alpha p_{tx} - (1-\alpha)p_{tu}}{\alpha p_{tx} + (1-\alpha)p_{tu}} \end{aligned}$$

which is clearly maximized by $\alpha = 0$. Consequently as $l \to \infty$ and $\alpha \to 0$

$$E[Y_{\phi_\alpha}] \to 1 + \frac{1 - p_{tu}}{p_{tu}} = E[Y_{\phi_{oracle}}]$$

This convergence has the intuitive explanation that as the memory window increases to a very large size there will almost always be a recovery within the memory window enabling the sender to send almost all updates. But this, of course is the best even an oracle policy can do. Consequently for very large memory windows, one would expect a stochastic policy that sends a very low proportion of snapshots to updates to achieve almost optimal performance. This trend can be seen in figure 1 as the expected inter-arrival time of recoveries for a given memory window size l (the curved lines) continually decrease their distance from the corresponding optimal bounds (the straight lines) for that l, as l increases from 1 to 15.

3. CONCLUSIONS AND FUTURE WORK

Ideally we would like to provide an analytic solution for determining the optimal α value of a stochastic policy, along with bounds on the difference between an optimal stochastic policy and the theoretically achievable limit. Unfortunately the fact that solution form of the stochastic policy is a $l+1$-degree polynomial in α does not bode well for analytic solution. It does appear clear from extensive empirical sampling that α takes only a single optimal value in the domain $[0, 1]$ for all values $0 < p_{tx} < p_{tu} < 1$. Moreover there are no other local minimal for α, as can be seen in figure 1. This observation implies that in practice a numerical solution could be found through binary search across the domain within ϵ accuracy in $O(\log(\frac{1}{\epsilon}))$ time.

4. REFERENCES

[1] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *IEEE INFOCOM*, 2006.

[2] K. Fall. A delay-tolerant network architecture for challenged internets. In *Applications, Technologies, Architectures, and Protocols for Computer Communication*. ACM / SIGCOMM, 2003.

[3] C. Fragouli, J.-Y. L. Boudec, and J. Wildmer. Network coding: An instant primer. Technical Report LCA-REPORT-2005-010, LCA, 2005.

[4] A. Hald. *History of Probability and Statistics and Their Applications before 1750*. John Wiley and Sons, Inc., 2005.

[5] K. Pister. `http://robotics.eecs.berkeley.edu/ pister/SmartDust/`.

[6] H. Qi, S. Iyengar, and K. Chakrabarty. Distributed sensor networks - a review of recent research. In *Journal of the Franklin Institute*. Elsevier, 2001.

[7] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modelling a three-tier architecture for sparse sensor network. In *Sensor Network Protocols and Applications*. IEEE, 2003.

[8] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University, 2000.

[9] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc*. ACM, 2004.