Lecture 18: April 8

Lecturer: Danny Raz *Scribe: Chris Umans*

In this lecture we continue our discussion of "A Modular Analysis of Network Transmission Protocols," presenting an almost optimal randomized data selection policy.

In the previous lecture, we presented a deterministic data selection policy with a competitive ratio that approaches 1. However no deterministic policy can obtain a bounded Δ, which is what we achieve (with high probability) with the randomized policy presented here. This closely follows the work in [1].

18.1 The Algorithm

We use the following terms with respect to packets in order to describe the algorithm. A packet is *completed* when the sender has received an ack for it; a packet is *locked* when it is in transit and not yet acknowledged; otherwise a packet is *available*. We begin using the term "backlog" in place of "window" in the previous lecture, and note that the backlog at time t, denoted B_t, is exactly the number of locked packets at time t.

The algorithm is simple: at time t it sends one of the smallest $S(B_t)$ available packets, chosen uniformly at random. (The "smallest" packets are those with smallest index). The function S is chosen so that $\sum_i 1/S(i) \leq 1/2$; we define $S(x) = \lceil dx \log(x+1) \log\log^{1+\epsilon}(x+1) \rceil + 1$ where d and ϵ are appropriately chosen constants. The set of packets from which the random selection is made at time t, is the set of *valid* packets, denoted V_t.

18.2 Analysis

We prove the following claim that then implies the main theorem:

Claim 18.1 *For any choice of θ, any constant c, and any time τ, let $\widehat{B}_\tau = \max_{t \leq \tau} B_t$ and let $\delta = \lceil 4cS(\widehat{B}_\tau) \ln 2S(\widehat{B}_\tau) \rceil$. Then*

$$\Pr[\Delta(\tau, \theta) \geq \delta] \leq (2S(\widehat{B}_\tau))^{-2c+2}.$$

Proof: We fix θ and a time τ, and we consider the time period during which the δ successful sends prior to time τ occurred. That is, let τ_δ be the first time at which $P^*(\tau_delta, \theta) = P^*(\tau, \theta) - \delta$. Let $x = P^*(\tau_delta, \theta)$. The optimal algorithm delivers a prefix of length x by time τ_delta; we need to show that our algorithm delivers the prefix of length x by time τ with high probability.

We say that a packet is *eligible* at time t if it is either valid or locked at time t. Since our algorithm does not use redundancy, there is some packet y with index greater than x that is transmitted by time τ_δ. By the description of the algorithm, there are at most $\widehat{B}_\tau$ locked words and at most $S(\widehat{B}_\tau)$ valid words when y is transmitted. Letting Ψ be the set of eligible packets with indices smaller than that of x at the time y is transmitted, then, we have $|\Psi| \leq \widehat{B}_\tau + S(\widehat{B}_\tau) \leq 2S(\widehat{B}_\tau)$.

We want to show that the elements of Ψ are delivered by our algorithm by time τ with high probability, which will ensure that a prefix of length at least x is delivered by time τ. The next claim is the first step in this proof.

Claim 18.2 *Let t be the time of a successful "send" event between τ_δ and τ and let y be an element of Ψ. Then*

$$\Pr[y \textit{ sent at } t \mid y \textit{ is eligible at } t] \geq \frac{1}{2S(B_t)}.$$

Proof: Fix a suitable time t, and let $y \in \Psi$ be eligible at time t. If y is unlocked at time t, it is sent with probability $\frac{1}{S(B_t)}$, and the claim is satisfied. Otherwise, let $t_1, t_2, t_3, \ldots, t_{B_t}$ be the times of the most recent B_t unacknowledged transmissions prior to time t. The probability that y is locked at t is no more than the sum of the probabilities that y was sent in one of these transmissions. Recall that in our algorithm, the probability of y being chosen at time t_i is $\frac{1}{S(B_{t_i})}$, so we have that:

$$\Pr[y \text{ locked at } t] \leq \sum_{i=1}^{B_t} \frac{1}{S(B_{t_i})}.$$

We note that for all i, $B_{t_i} \geq i$, since the transmission of packet i at time t_i is not acknowledged until after time t. Also, recall two properties of the function S: (1) S is monotone increasing, so $\frac{1}{S(B_{t_i})} \leq \frac{1}{S(i)}$ and (2) S was chosen so that $\sum_i 1/S(i) \leq 1/2$. So

$$\sum_{i=1}^{B_t} \frac{1}{S(B_{t_i})} \leq \sum_{i=1}^{B_t} \frac{1}{S(i)} \leq \frac{1}{2}.$$

Therefore the probability of y being sent at time t is at least $\frac{1}{2} \times \frac{1}{S(B_t)}$. ∎

Consider $y \in \Psi$. If y is neither completed nor eligible at a successful send event in the time period between τ_delta and τ, then it must be that some other $y' \in \Psi$ is successfully delivered at that time. This can happen at most $|\Psi|$ times, and we showed above that $|\Psi| \leq 2S(\widehat{B}_\tau)$. Our τ_δ was chosen so that there are δ successful transmissions in the time period $[\tau_delta, \tau]$, so y must be eligible in at least $\delta - 2S(\widehat{B}_\tau)$ of them, or it must have been successfully delivered by time τ.

By our previous claim, the probability that y is *not* successfully delivered at time t in $[\tau_delta, \tau]$ when it was eligible is at most $(1 - 1/2S(B_t)) \leq (1 - 1/2S(\widehat{B}_t))$. So the probability that y is not delivered by time τ is bounded by:

$$\left(1 - \frac{1}{2S(\widehat{B}_\tau)}\right)^{\delta - 2S(\widehat{B}_\tau)} \leq e^{-(2c \ln 2S(\widehat{B}_\tau) - 1)}.$$

Finally, taking a union bound over at most $2S(\widehat{B}_\tau)$ elements of Ψ, the probability that our algorithm does not complete a prefix of length at least x packets by time τ is bounded by $(2S(\widehat{B}_\tau))^{-2c+2}$. ∎

Theorem 18.3 *The randomized algorithm of the previous section achieves the following with high probability:*

$$\Delta^A(t,\theta) = \tilde{O}(B \log^2 B).$$

Proof: The proof follows from our first claim, and the choice of S. ∎

18.2.1 Lower bound

Theorem 18.4 *For any randomized algorithm A and for sufficiently large B, there exists θ and some t for which the following with constant probability:*

$$\Delta^A(t,\theta) = \Omega(B \log B).$$

The proof was not given. The basic idea is to divide time into blocks of size B, phases of $\log B/2$ blocks, and a total of $k = 2B \log B$ phases. The proof depends on the correct choice of t, which depends on whether the algorithm uses redundancy or not.

References

[1] Micah Adler, Yair Bartal, John W. Byers, Michael Luby, Danny Raz. *"A Modular Analysis of Network Transmission Protocols"*, To appear in Proc. of the 5th Israeli Symp. on Theory of Computing and Systems, June 1997.