Robust Heuristics: Attacks and Defenses on Job Size Estimation for WSJF Systems

Erica Chiang, Nirav Atre, Hugo Sadok
Carnegie Mellon University

A Proofs for DF Analysis

A.1 Proof of Theorem 1 (Perfect Heuristic)

Proof. We first prove that a heuristic with strictly monotonically increasing ratios must be a perfect heuristic. Consider two arbitrary packets p_A and p_B, and a heuristic h with strictly monotonically increasing ratios. $\frac{c(p_A)}{s(p_A)} < \frac{c(p_B)}{s(p_B)} \implies \frac{h(p_A)}{s(p_A)} < \frac{h(p_B)}{s(p_B)}$, which means that if p_A has a smaller true z-ratio then it must be served first in a WSJF system, and thus the heuristic is optimal.

Next, we show that if a heuristic does not estimate z-ratios to be strictly monotonically increasing, then the heuristic is *not* optimal, meaning that it is not guaranteed to schedule all packets correctly. If a function h is not strictly monotonically increasing in ratio estimates, there must exist packets p_A,p_B such that $\frac{c(p_A)}{s(p_A)} < \frac{c(p_B)}{s(p_B)}$ and $\frac{h(p_A)}{s(p_A)} \geq \frac{h(p_B)}{s(p_B)}$. Then it is possible for the system to serve p_B before p_A, an exploitable point that makes the heuristic imperfect.

□

We also provide an example of a class of perfect heuristic: Any heuristic of the form $h(p) = k \cdot c(p)$ for some $k \in \mathbb{R}^+$ will ensure perfect scheduling under WSJF.

We first prove that a heuristic of this form is, in fact, perfect. Given that the expected service order (priority under WSJF scheduling) is in increasing value of $\frac{c(p)}{s(p)}$, while the heuristic's service order will be in increasing value of $\frac{h(p)}{s(p)}$, we want to show that these orderings will always be the same. Consider 2 packets p_A and p_B. It holds that

$$\frac{c(p_A)}{s(p_A)} < \frac{c(p_B)}{s(p_B)} \iff k \cdot \frac{c(p_A)}{s(p_A)} < k \cdot \frac{c(p_B)}{s(p_B)} \iff \frac{h(p_A)}{s(p_A)} < \frac{h(p_B)}{s(p_B)}$$

A perfect scheduling order ensures that all of the assumptions from SurgeProtector are maintained, allowing WSJF to provide the same DF upper bound of 1.

A.2 Proof of Theorem 2 (Step Function Heuristic)

Proof. In order for an adversary to achieve a DF greater than k, it must be able to displace innocent packets such that for every byte of data transmitted by the adversary, k times the amount of innocent data is displaced. Equivalently, the innocent packet job size to packet size ratio is a factor of k less than the adversary's. We will prove that the step function guarantees this property by showing the contrapositive to be true.

We first consider an adversarial packet p_A and innocent packet p_I, and assume that $\frac{c(p_A)}{s(p_A)} > k \cdot \frac{c(p_I)}{s(p_I)}$, and we want to show that then the packets cannot be swapped by the scheduler, *i.e.* $\frac{h(p_A)}{s(p_A)} > \frac{h(p_I)}{s(p_I)}$.

$$\begin{aligned}
h(p_A) &= a \cdot k^{\lfloor \log_k c(p)) \rfloor} \\
&\geq a \cdot k^{\lfloor \log_k (k \cdot c(p_I) \cdot \frac{s(p_A)}{s(p_I)}) \rfloor} && \text{(By assumption)} \\
&= a \cdot k^{\lfloor 1 + \log_k c(p_I) + \log_k (\frac{s(p_A)}{s(p_I)}) \rfloor} \\
&\geq a \cdot k^{1 + \lfloor \log_k c(p_I) \rfloor + \log_k (\frac{s(p_A)}{s(p_I)})} \\
&= a \cdot k \cdot k^{\lfloor \log_k c(p_I) \rfloor} \cdot k^{\log_k (\frac{s(p_A)}{s(p_I)})} \\
&= k \cdot h(p_I) \cdot \frac{s(p_A)}{s(p_I)} \\
\implies \frac{h(p_A)}{s(p_A)} &\geq k \cdot \frac{h(p_I)}{s(p_I)} \implies \frac{h(p_A)}{s(p_A)} > \frac{h(p_I)}{s(p_I)}
\end{aligned}$$

Given that an adversarial packet cannot displace innocent packets with a job size to packet size ratio more than a factor of k smaller than the adversarial packet's, we see that the step function heuristic upper bounds the DF at the fixed value of k.

□

A.3 Proof of Theorem 3 (Preemption)

Proof. We consider a system that starts with estimated job size $J_p = \epsilon$ for all packets, in order to complete analysis with no assumptions about the quality of job size estimates. We also assume no preemption cost. We first look at a period of T seconds, during which N innocent packets arrive. We can represent the true job sizes j_i of incoming packets in scheduled order, as $S = [j_1, j_2, ..., j_N]$ where $j_i \leq j_{i+1} \forall i$. The main insight here is that an attacker can exploit the system by injecting adversarial packets such that all innocent packets are *partially* served but displaced (preempted and then never fully served). An adversary's goal thus becomes to "weaponize" innocent packet work by causing the system to serve each innocent packet for some amount of time that is *less* than its true job size.

Since job estimates for all packets increase by factors of 2 of the initial estimate ϵ, for a packet of job size c, the maximum work the adversary can weaponize is $w = \max_{k \in \mathbb{Z}^+}(\epsilon \cdot 2^k)$ such that $w < c$. We see worst-case behavior when the value of $c - w$ is minimized for all innocent packets, such that practically all of the innocent work can be weaponized. So, for a fixed $k \in \mathbb{Z}^+$ (and thus a fixed $w = \epsilon \cdot 2^k$), we consider a scenario where all innocent packets have the same job size $c = w + \delta$, with $\delta \to 0$. For simplicity, we assume that all packets have a packet size of 1.

Assume that the attacker pushes the system to capacity by using l adversarial packets, each of packet size 1 and a true job size of J_A. In order to minimize l, it is in the attacker's interest to encode as much work as possible in each attack packet. Observe that, as long as service capacity is available, each adversarial packet is guaranteed at lease w service time (equivalent to the work served in each innocent packet). Thus, we choose $J_A \geq w$. Since the system must be at capacity in order to displace any traffic, we have:

$$\overbrace{w \cdot N}^{\text{Weaponized work}} + \overbrace{w \cdot l}^{\text{Adversarial work}} = T$$

$$l = \frac{T - w \cdot N}{w} = \lim_{\delta \to 0} \frac{T - (c - \delta)N}{c - \delta} = \frac{T - cN}{c} = \frac{(T - t)N}{t}$$

where $t = cN$ is the cumulative true service time for innocent traffic alone. We also note that given the uniform-sized packets in innocent traffic, we can express the load due to innocent traffic as $\rho = \frac{t}{T}$. We are now able to bound the DF for preemptive WSJF, the innocent traffic displaced relative to the adversarial traffic sent:

$$DF = \lim_{T \to \infty} \frac{N}{\frac{(T-t)N}{t}} = \lim_{T \to \infty} \frac{t}{T - t} = \lim_{T \to \infty} \frac{\frac{t}{T}}{1 - \frac{t}{T}} = \frac{\rho}{1 - \rho},$$

which becomes unbounded as $\rho \to 1$.

□