TCP forking

TCP forking

- When a browser asks for a web page, one TCP connection can be started for each page component request (and another one for the web server's reply)
  - For simplicity, we will restrict to single-component pages
  - Such TCP connections are short in duration and may not reach the 'congestion avoidance' phase (conservative bandwidth usage)
- **TCP forking**: make parallel TCP requests for the same page to 'multiple' servers
  - TCP forking is possible if a web server runs in multiple instances or is multi-homed, and DNS replies with all mappings

TCP forking

- We argue that the minimum page download time can be achieved with TCP forking strategy
  - TCPm would need a few RTTs until the set of available locators to be learnt by the source
- But, if volume pricing is applied by ISPs and a source is responsible for all traffic triggered by it, then average cost can be higher with TCP forking
  - Browsers could discard all late web server replies, but any extra cost caused would be credited to the user

The single-attempt model

- Host i starts a request to web-server s for a simple page of size A (i.e. A= 40 KB)
- Web-server s is assumed to be reachable through a set of locators $L_s=\{s_1,\ldots,s_n\}$, that is available to browser
- A locator s_j is unreachable with some probability b_j
  - Interface being down, server unavailability, path failure, …
- The expected total response time of server s through network path (i, s_j) is: $E[T_j] = t_j (1- b_j)+ t_0 b_j$

RTT (i, s_j) + s processing time

a fixed threshold waiting for response

The single-attempt model

- End user's patience has duration T
  - a random variable following exponential distribution with a known mean
- End user's utility r_j from using s_j is a decreasing function of s_j responsiveness, i.e. $r_j = f(E[T_j])$
- Let I_j be distributed as a binomial $B(1,1-b_j)$ random variable, that is:

$$I_j = \begin{cases} 1, & s_j \text{ is available with probability } 1-b_j \\ 0, & s_j \text{ is blocked with probability } b_j \end{cases}$$

The single-attempt model

- ISP charges price p for each downloaded KB
  - Uploaded traffic is ignored for simplicity
- The set L_s is available to browser, which tries to find the set of locators L to send a request that maximises user's Net Benefit:

$$E[\max_{j \in L}\{I_j r_j\}] - Ap\sum_{j \in L} I_j$$