An Example of Persistent HTTP Connections
Persistent HTTP

Advantages and Disadvantages

Advantages for User:
- Faster response time
- Fewer TCP handshakes
- Can pipeline multiple requests

Advantages for Server:
- Less computation
- Fewer simultaneous TCP control blocks
- Less CPU time opening and closing TCP connections
- TCP can better determine network congestion
- Can pipeline multiple requests
- Fewer TCP handshakes
- TCP connections require memory

Disadvantages:
- More idle open TCP connections
- Many idle connections slow servers [Bangia, Druschel, Mogul 1999]
Holding Time Model

TCP and Persistent HTTP
Utility Model

Goal: Maximize server's utility.

Utility Equation

- idle connection time
- open connection hits
- server
- HTTP requests
- set $S$ of holding time
- conversion rates

utility
Utility Equation

Utility = Revenue - Cost

Conversion Rates

Cost = \( \text{hits} \cdot \text{open-hits} + \text{idle-time} \cdot \text{TCP setup} \cdot \text{TCP establishment cost} \)

Revenue = \( \text{hits} \cdot \text{open-hits} \)

Utility = Revenue - Cost
utility \approx \frac{3}{4} \cdot \text{idle-time} \cdot \text{open-hits} \cdot (c^0 + c \cdot c^0)$
Utility Equation

\[ \text{utility} \approx C + C^0 \cdot \text{open-hits} = C^0 \cdot \text{idle-time} \]

equivalent to

\[
\begin{align*}
\text{cost} & = (\text{hits} - \text{open-hits}) \cdot C + \text{TCP setup} \cdot C + \text{TCP idle-time} \\
\text{revenue} & = \text{hits} \cdot C + \text{open-hits} \cdot C^0 \\
\text{utility} & = \text{revenue} - \text{cost}
\end{align*}
\]

Which terms depend on holding time function?
Goal: Maximize server's utility.
TCP and Persistent HTTP 13
Basing Holding Times on HTTP Request Attributes

**Client Predictability:**
- request stock market information every 15 minutes
- request Toronto weather information
- probable request within few seconds
- short holding time

Current Page has Embedded Images
Types of Attribute-Based Functions

**Types of Attribute-Based Functions**

**FIXED:** Ignores all parameters
- Unimplementable
- Best possible function
- Closes connection if idle time more costly than next open hit

**OPT:** Uses FUTURE
- Future: use knowledge of future requests
- Current: use current client
- Referrer: use source for current URL's address
- Resource: use current URL

**RESOURCE:**
- Use current URL

**REFERER:**
- Use source for current URL's address

**HTTP:**
- HTTP/1.1 suggests this function
- Simple to implement
- Every holding time equals $t$ seconds

**FUTURE:**
- Future: use knowledge of future requests

**CLIENT:**
- Use current client

**Apache v. 1.3:**
- Uses 15 seconds

**HTTP/1.1:**
- Suggests this function
- Simple to implement
- Every holding time equals $t$ seconds

**BEST:**
- Uses FUTURE
- Closes connection if idle time more costly than next open hit

**OPT:**
- Uses FUTURE
Computation utility: for each resource

Server logs: Future requests may reflect past requests

Utility \( \approx C + C \cdot \text{open-hits} - C \cdot \text{idle-time} \)

Computing Most Beneficial Resource
\[ \frac{\text{connection} - C^o \text{idle-time}}{\text{open-hits}} = \frac{C^o \text{connection}}{\text{utility}} \]
Comparing RESOURCE, REFERRER, RES-REF.
Predicting using CLIENT

TCP and Persistent HTTP2
Conclusions

- Introduced model for evaluating persistent HTTP connections.
- Presented technique to produce beneficial holding time functions.
- Experimentally showed our holding time functions beneficial.
- Experimentally evaluated relevance of attributes to holding times.
- Reduced idle connection time up to 50%.

Use talk outline slide.

Conclusions
Future Work

- Are conversion rates constant?
- What is the effect of holding times on number of hits?
- What is the effect of client closing connections?
- What is the effect of holding times on number of hits?

Connection management functions for proxies:•