Web*Express*

A System for Optimizing Web Browsing in a Wireless Environment

Paper by: Barron C. Housel and David B. Lindquist
IBM Corporation

Presentation by: Vipul Sawhney

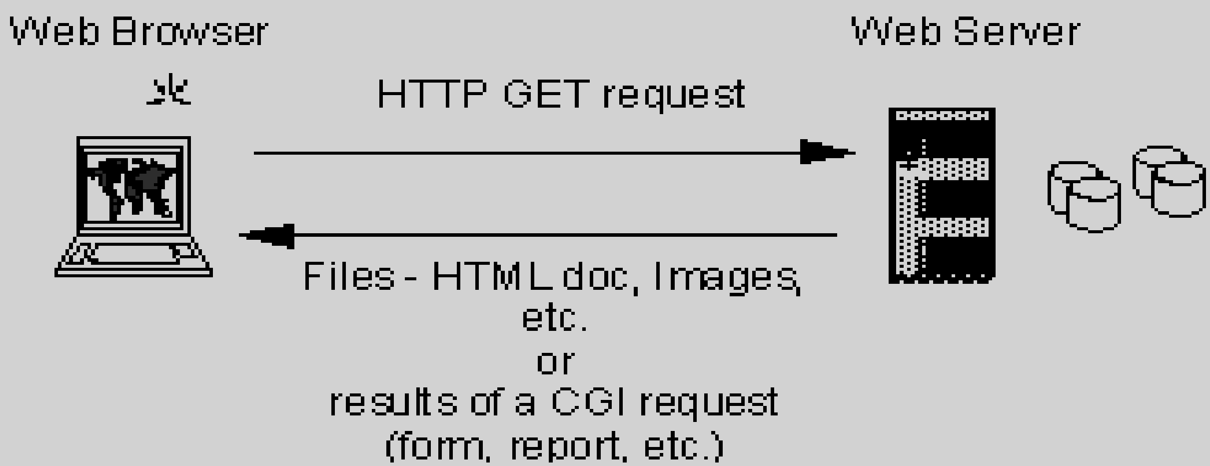
Presentation Overview

- Introduction
- HTTP Summary
- Inhibitors to Web Browsing over wireless networks
- Web*Express* software system
- Results achieved
- Conclusions

Introduction

- Software system which significantly reduces user cost and response time of wireless communications by intercepting the HTTP datastream and performing various optimizations on it.
- Aimed at routine commercial applications

Hyper Text Transport Protocol



The Problems With Browsing Over Wireless Networks

- High Cost
- High Latency
- Low Bandwidth
- Unreliability

Additional Inefficiencies of HTTP

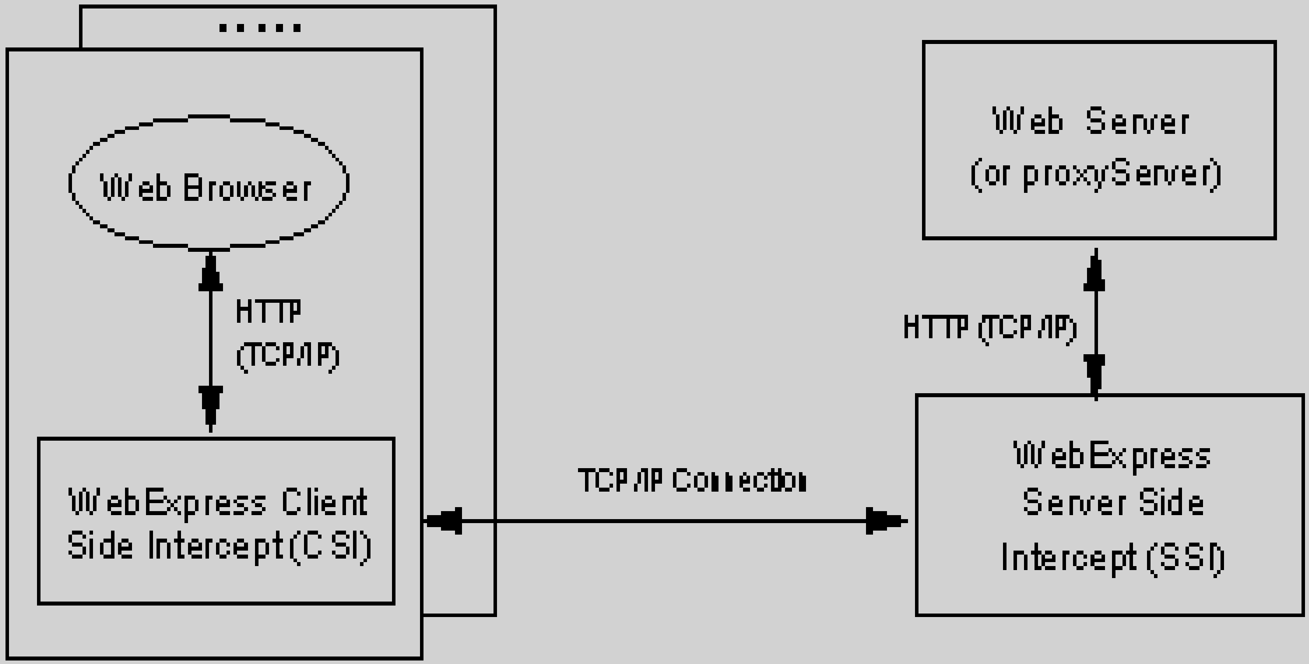
- Connection overhead
- Redundant Transmission of capabilities
- Verbose protocol

Introducing Web*Express*

- Utilizes an intercept technique which allows it to be used with any *Web Browser* and any *Web Server*
- Enables Web*Express* to intercept and control communications over the wireless link for the purpose of reducing traffic volume and optimizing the communications protocol to reduce data transmission

Web*Express* Components

- Two components are inserted into the data path between the Web client and Web server
- Client Side Intercept (CSI) : process which runs in the end user client mobile device
- Server Side Intercept (SSI) : process which runs within the wireline network



.....
Web Browser
HTTP
(TCP/IP)
WebExpress Client
Side Intercept (CSI)
Web Server
(or proxyServer)
HTTP (TCP/IP)
TCP/IP Connection
WebExpress
Server Side
Intercept (SSI)

Strengths of the Intercept Model

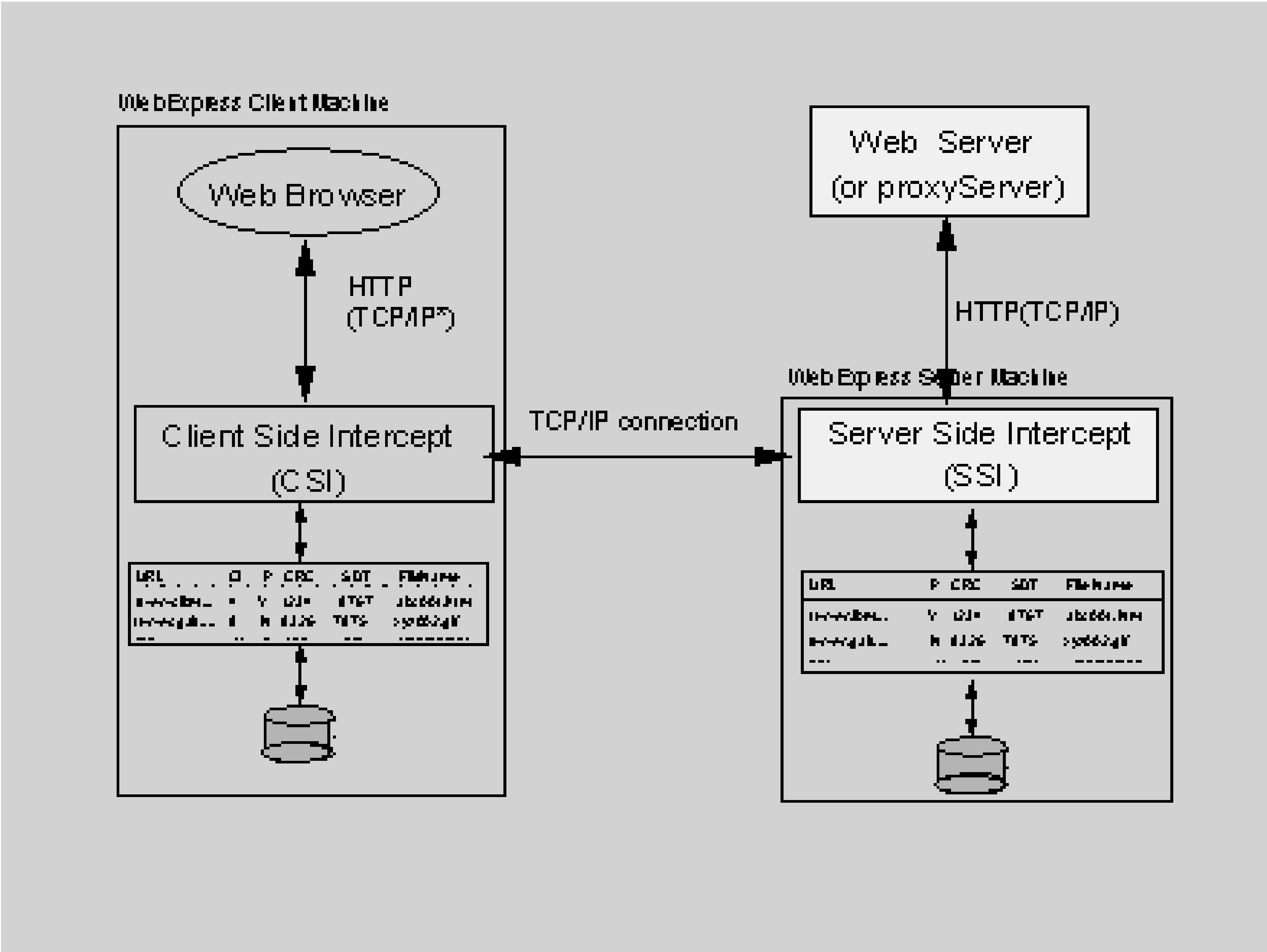
- Transparency to both Web browsers and servers
- Insesitive to the development of the rapidly maturing HTML/HTTP technology
- Highly effective data reduction and protocol optimization without limiting browser functionality or interoperability

Web*Express* Optimizations

- Caching
- Differencing
- Protocol Reduction
- Header Reduction

Caching

- Web*Express* supports both client and server caching
- Least Recently Used (LRU) algorithm
- Persistance of cache objects across browser sessions
- Each cached object has a *digital signature* (CRC) and a *coherency interval* (CI)


WebExpress Client Machine
Web Browser
HTTP
(TCP/IP*)
Client Side Intercept
(CSI)
TCP/IP connection
Web Server
(or proxyServer)
HTTP(TCP/IP)
Server Side Intercept
(SSI)

Caching Strengths

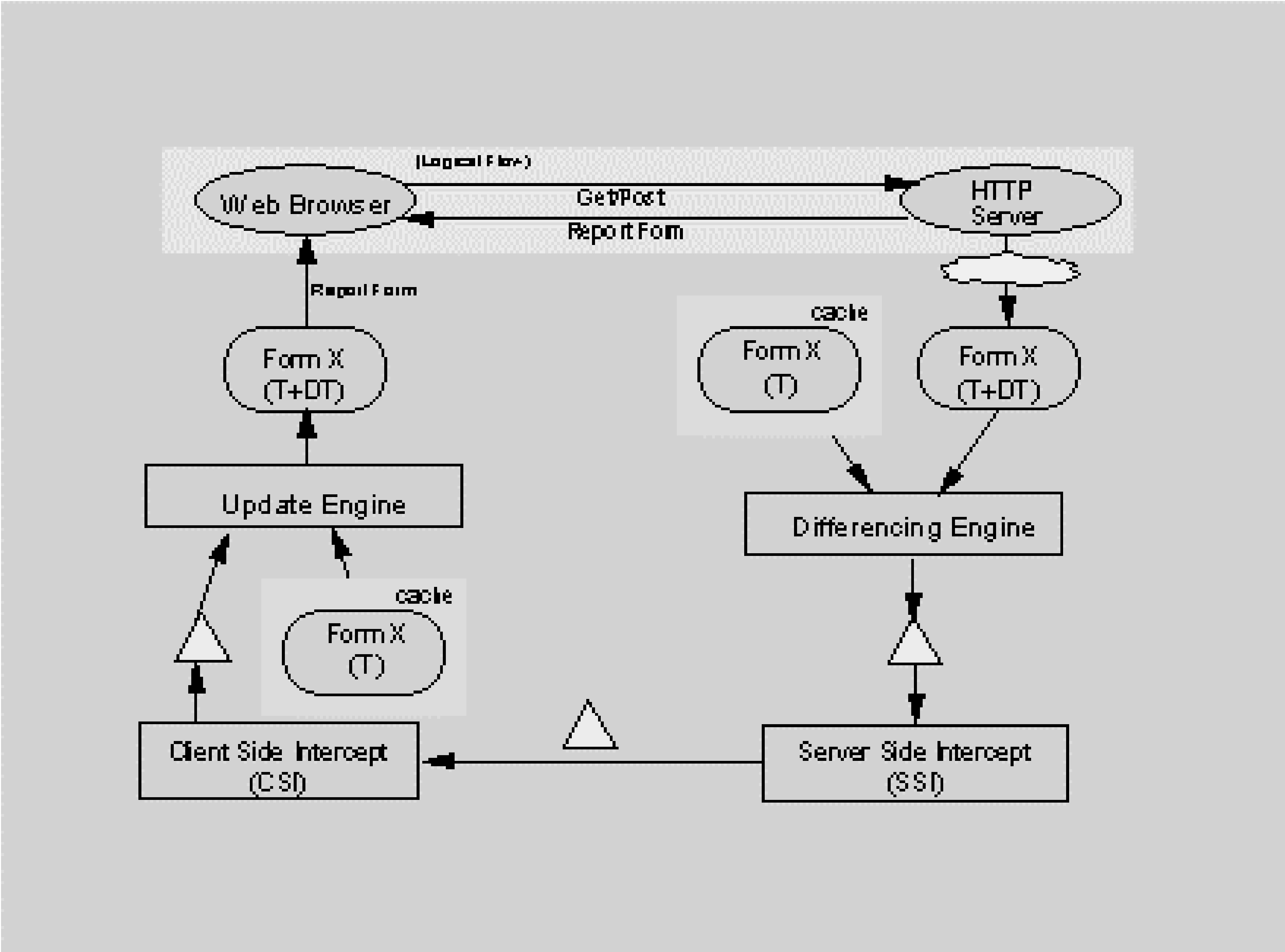
- Reduce volume of application data trannsmitted over:
  - Wireless link
  - Through cross browser sessions
  - Age basecd coherency algorithms
  - CRC
  - User options

Differencing

- Caching techniques do not help in CGI processing where each request returns a different result, i.e. *stock-quote server*
- *Different replies from the same program (appication server) are usually very similar*
- Example: Replies from a stock-quote server contain lots of unchanging data such as graphics, etc.

How Does It Work ?

- A *base object* is intially established for a new CGI URL at the CSI
- CSI checks to see if URL is cached on further requests
- CSI forwards the request to the SSI along with the CRC value of the base object
- SSI determines differencing possible and computes a *difference stream* between the received report and the base object



Web Browser
Get/Post
Report Form
HTTP
Server
cache
Form X
(T)
Form X
(T+DT)
Form X
(T+DT)
Update Engine
Differencing Engine
cache
Form X
(T)
Client Side Intercept
(CSI)
Server Side Intercept
(SSI)

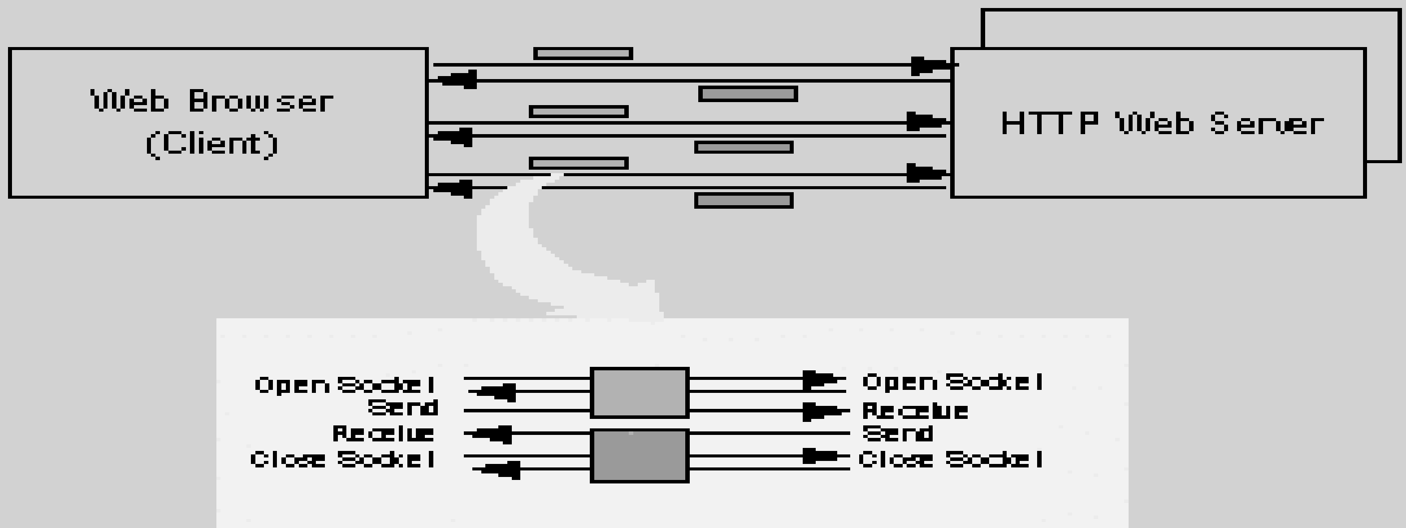
Basing and Rebasing

- If SSI receives a response from the CGI server and the CRC received **does not match** the CRC of the base object, SSI returns **both** difference stream and base object
- *Rebasing* is carried out in the same fashion when the SSI detects that the difference stream has grown beyond a certain threshold

Protocol Reduction

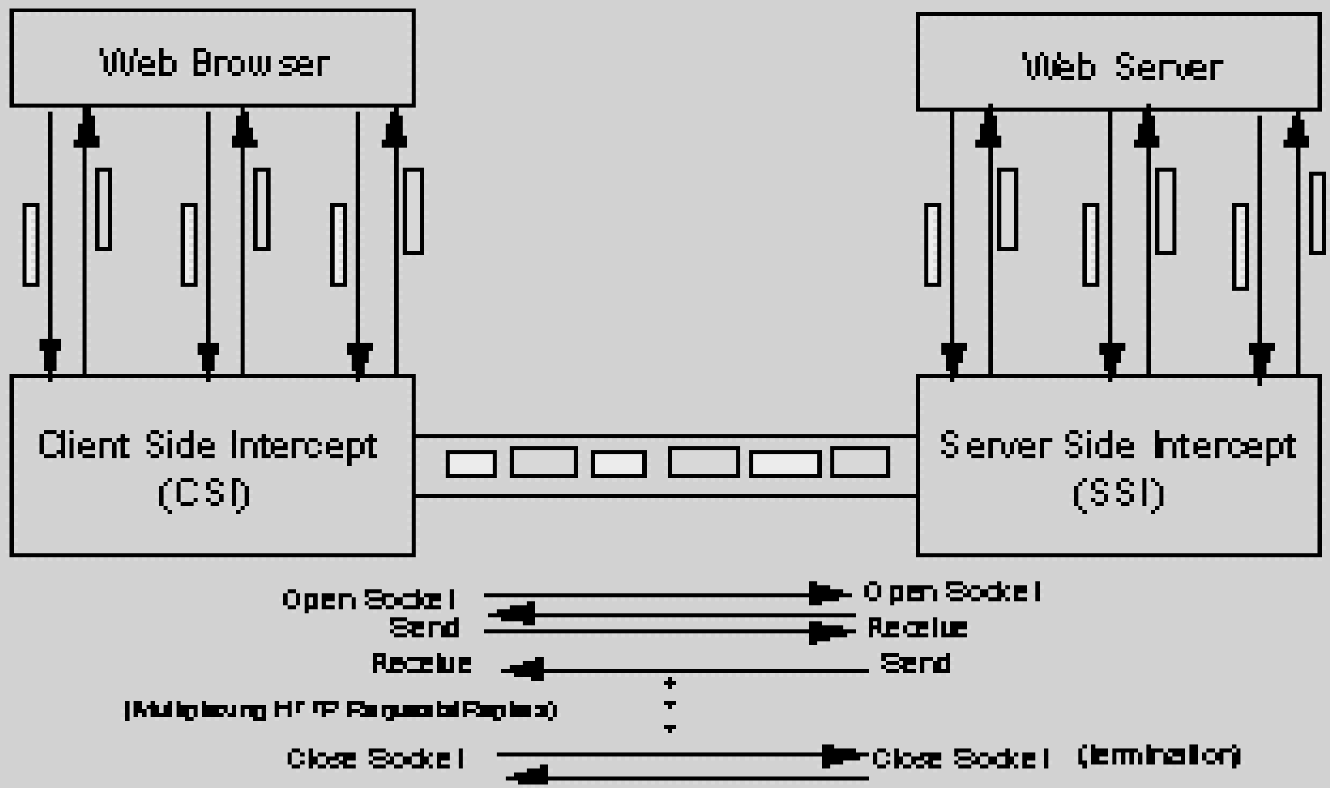
- Repeated TCP/IP connections and redundant header transmissions present additional overhead
- Web*Express* system utilizes two main techniques to reduce this overhead and optimize browsing in a wireless environment

Reduction of TCP/IP Connection Overhead



Virtual Sockets

- Web*Express* establishes a single TCP/IP connection between the CSI and SSI
- CSI sends requests over this connection
- SSI establishes a connection with the destination server and forwards the request
- Thus overhead is incurred between SSI and Web server but not over the wireless link
- *Virtual Sockets* used to provide multiplexing support


Web Browser
Web Server
Client Side Intercept
(CSI)
Server Side Intercept
(SSI)
Open Socket
Open Socket
Send
Receive
Receive
Send
Close Socket
Close Socket
(termination)

Virtual Sockets cont...........

- Data sent is prefixed by a virtual socket id, command byte, and a length field
- At CSI the virtual id associated witha real socket to the browser
- At SSI the virtual socket id mapped to a socket cnnection to a HTTP server
- Mechanism permits effiecient transport of HTTP requests and response while maintaining protocol transparency

Reduction of HTTP Headers

- HTTP *request headers* containing lists of MIME content-types can be hundreds of bytes in length
- CSI allows this information to flow in the first request and saves the list
- For further requests CSI compares the list received with the saved one
- On match list is deleted from request
- SSI inserts the saved one if none else saves

Results Achieved

	Data Volume Comparison (bytes)			Response Time Comparison (seconds)		
Test	Base	WebExpress	%Reduction	Base	WebExpress	%Reduction
1	56779	2302	96	1260	166	87
2	9635	2643	73	343	96	72
3	4853	1272	74	114	59	48
4	137649	456	99.7	1079	30	97
5	2234	533	76	59	37	37
6	1326	515	61	19	11	42

Conclusions

- Web*Express* allows the operation of commercial Web application of wireless networks
- Predictability and Repetiveness allow this success
- *Differencing* and *Virtual Sockets* offer the most critical optimizations