

A blue wireframe globe with a grid of latitude and longitude lines, set against a dark blue background with a subtle pattern of wavy lines.

Improving Access to Web Applications

Bill Wehl
Chief Technology Officer
Akamai Technologies, Inc.

May 13, 2004



The Problem:

- Getting information to users to:
 - Increase adoption of portals
 - Improve revenue on commerce sites
 - Drive corporate efficiencies

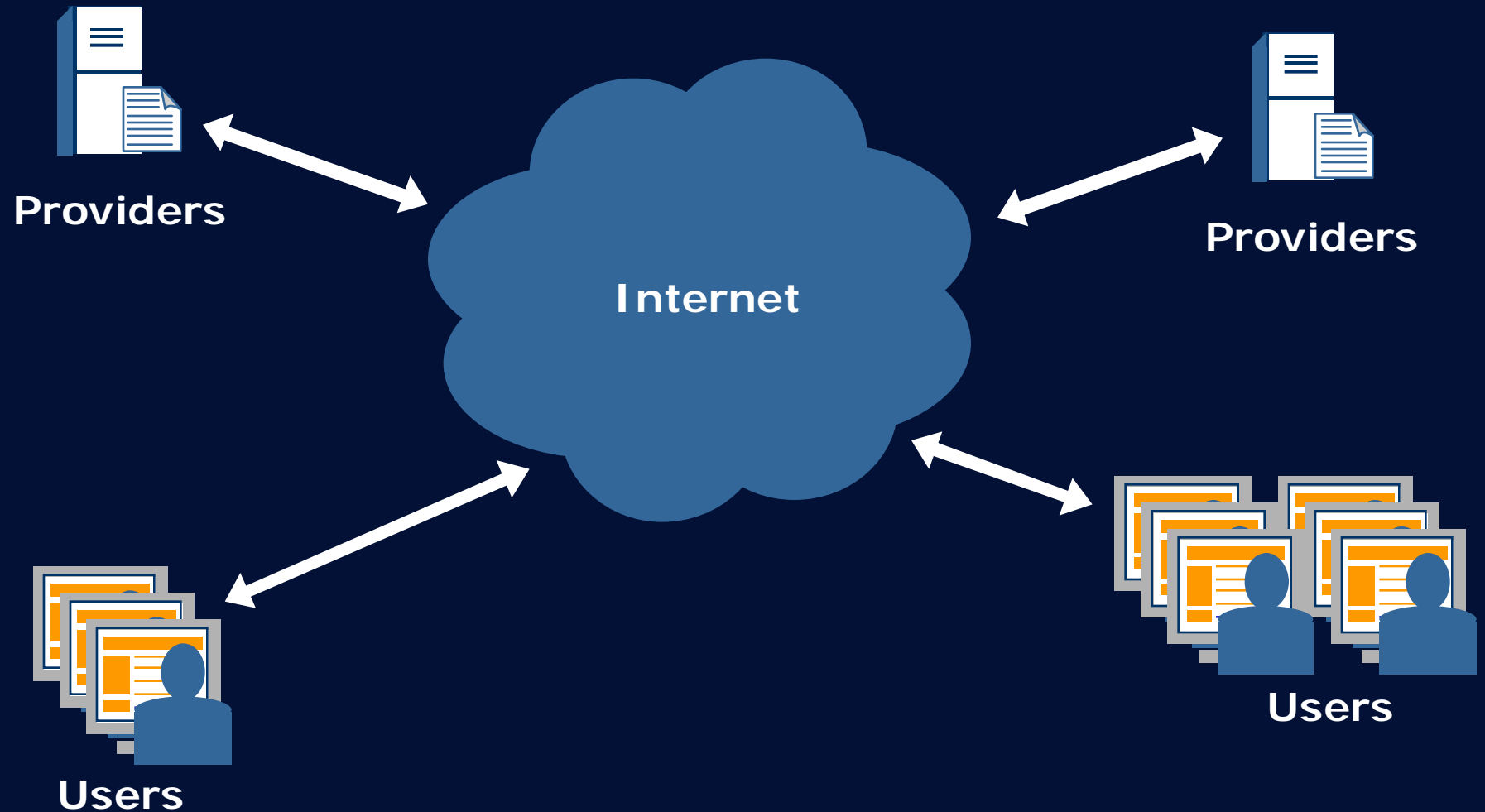
Some examples:

- Order Submission
- Account Management
- Supply Chain Management
- Reservations

Goal: *subsecond* response time for interactive applications for all users



The Web: Simple Abstraction







What's Hard?

- **International Performance & Remote Mobile Users**
 - Interactive applications need to make many roundtrips
 - Peering is optimized more for economics than for QoS
 - Routing is insensitive to congestion
 - Routes take time to stabilize after changes
- **One-off events** that affect availability
 - Congestion and failures cause outages and slow response
 - Single points of failure with centralized infrastructure
 - DDoS attacks directed at your infrastructure and at networks
- **Capacity Concerns**
 - Time to manage growing infrastructure
 - Expensive to scale – unusual peaks can overload a site



The Solution

- **Distribute and share** the infrastructure
- **Avoid long trips** through the network
- When long trips are unavoidable...
 - Send data by the **best possible path**
 - **Minimize data** sent
 - Optimize at the **protocol level**
- And provide **visibility and control** into this extended infrastructure

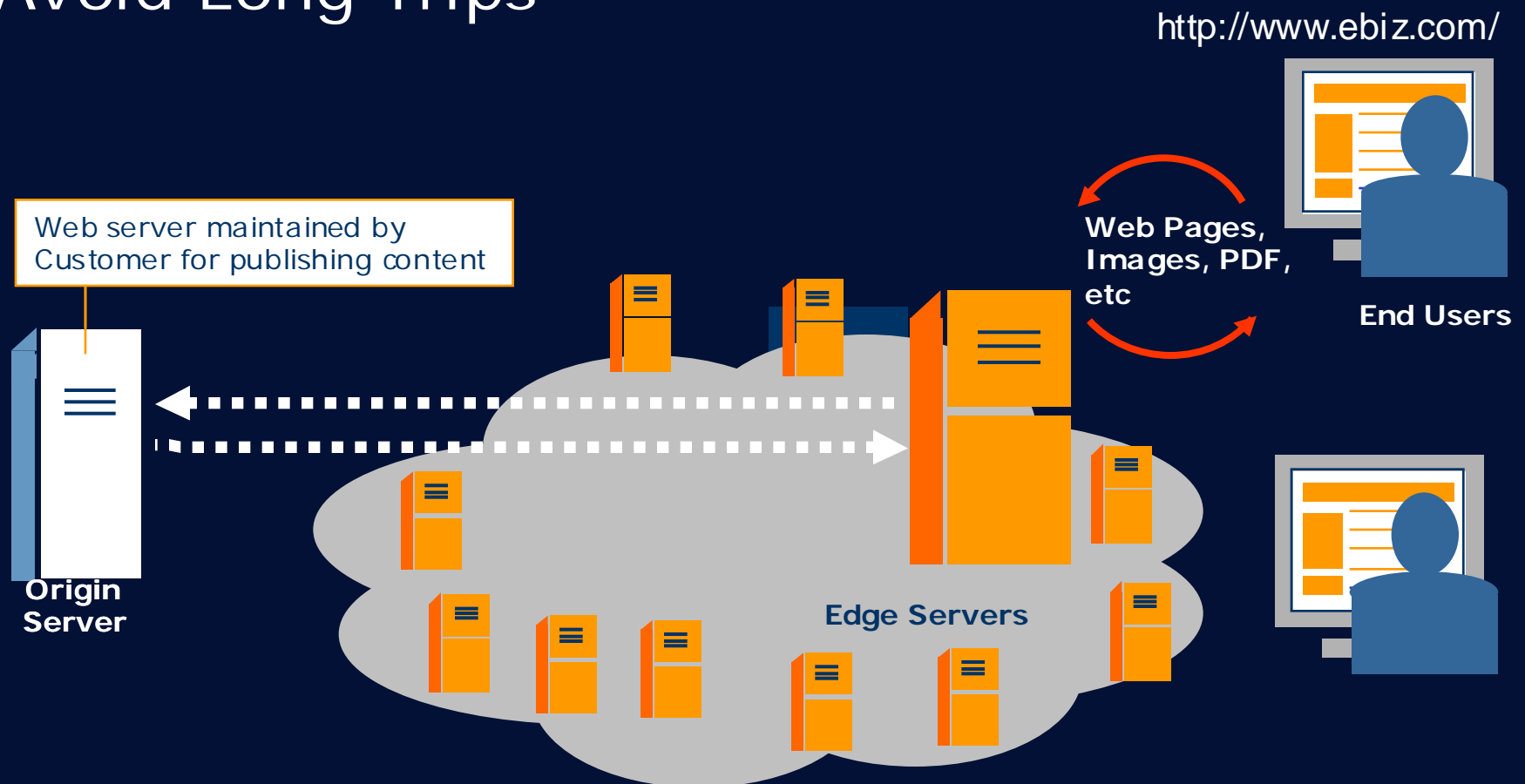


Distribute and Share

- Virtualize – use a shared distributed infrastructure
- Move delivery into the network
 - Caching and delivery from the edge provides giant shock-absorber for peak traffic transparently
- Move processing into the network
 - Simple template structure (ESI, XSLT)
 - Minimal integration can render many dynamic sites (e.g., portals) completely cacheable
 - Other sites can reduce long-haul bandwidth by 10X-100X
 - Edge computing
 - Move part of application to the edge
- Result: much less central infrastructure to manage, scale, provision for peaks, replicate for fault-tolerance

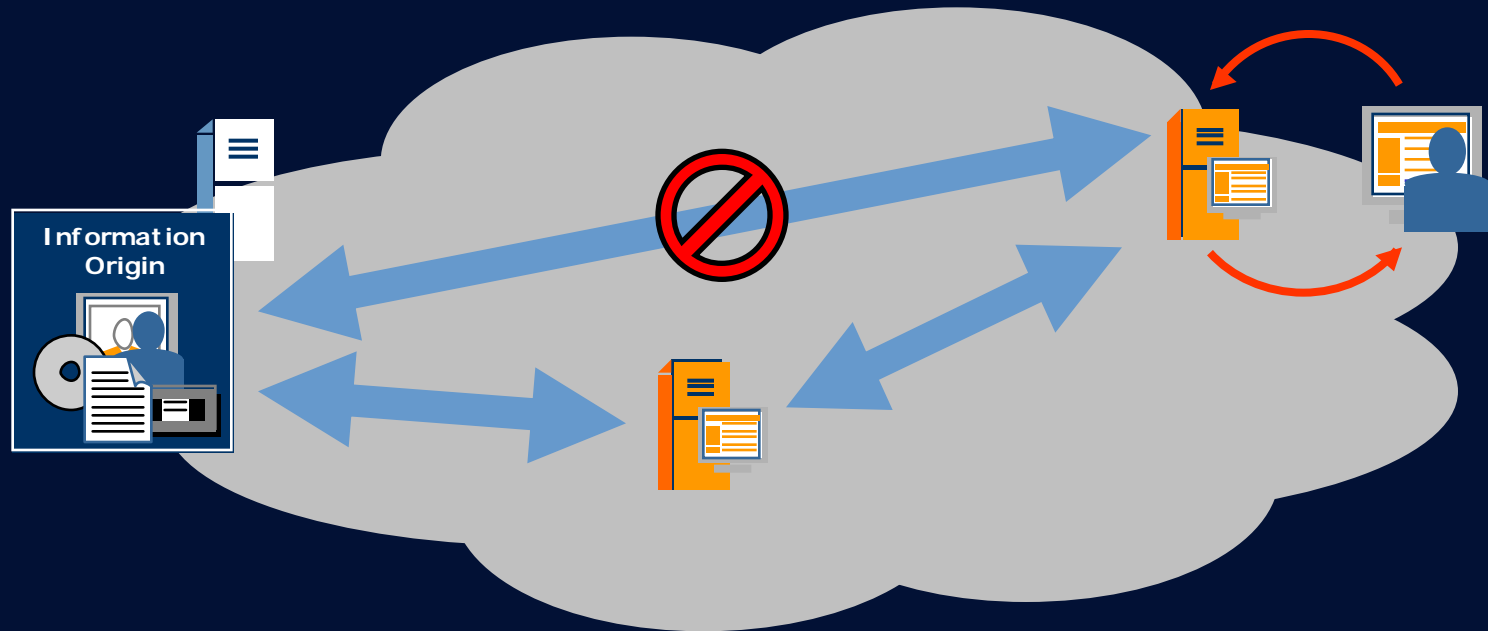


Avoid Long Trips



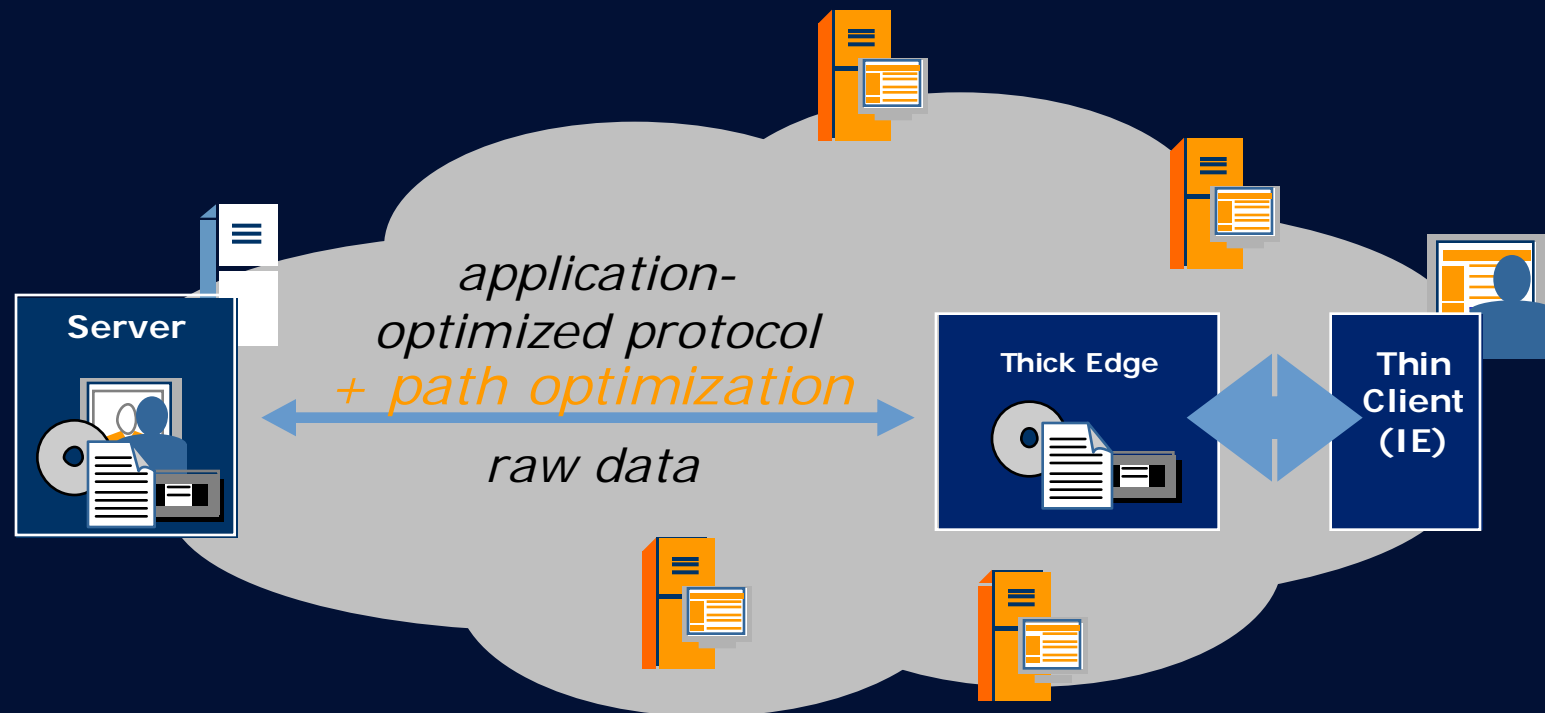
- Useful for images, downloads, whole site, streaming, secure sites
- NetStorage can be used as origin for static content or as failover

Send data by the best possible path



- BGP ignores congestion and latency; is slow to route around failures.
- Path-optimization technology finds alternate paths by tunneling traffic through intermediate global edge servers.
- *Plus* compression, delta encoding, persistent TCP and SSL connections, and TCP optimizations for even bigger impact.
- 2-4X average speedup; bigger impact on “tail” of distribution.

Minimize Data Sent



Additional resources are brought online automatically when and where needed

The best of client-server and web-based applications

- Central management
- Distributed performance

*A new deployment model,
not a new set of APIs*



Optimize at the protocol level

- Persistent connections
 - Avoid TCP setup and slow start over long-haul
 - Avoid SSL session negotiation
- Pre-fetch data not in cache
 - Stream data over long-haul
- Delta encoding
 - Send only the bytes that have changed
- Compression for the bytes that are left



Visibility and Control

- Gain insight into customer traffic, content, and users on the extended infrastructure
- Control the delivery of enterprise content and applications on the Internet
- Discover, diagnose, and resolve problems on the network – fast





Summary

- Virtualize for scale
- Distribute to avoid long trips
- Optimize remaining long trips:
 - Send only raw data
 - Choose the best path
 - Optimize the protocol
- Visibility and control into e-business infrastructure
- Each solves a piece of the problem – together they make a much bigger difference.
 - Extranets: international adoption went from 5% to 100%; no impact from Slammer
 - Online commerce: revenue increased 20%
 - Online commerce: cost decreased by 2X while handling 4X the load