A Reliability Model for Distributed Adaptation

Mark Yarvis, Peter Reiher, and Gerald J. Popek

UCLA

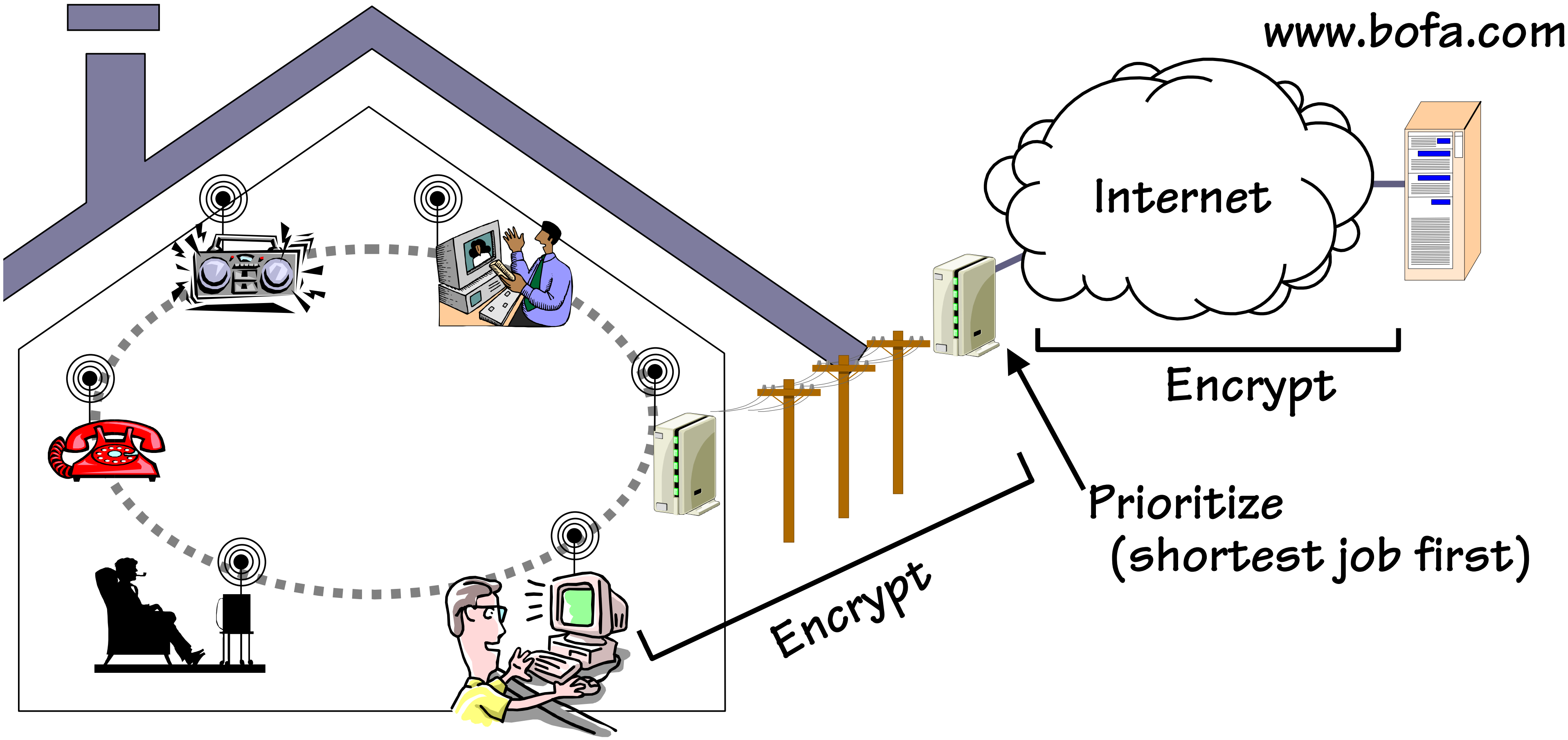
OpenARCH 2000

http://fmg-www.cs.ucla.edu/Conductor

Introduction

- Open architectures allow graceful degradation of applications
- Adaptation requires a new model of reliability
  - Semantic Segmentation
- Prototype implementation in Conductor adaptation framework

Motivation



www.bofa.com
Internet
Encrypt
Prioritize
(shortest job first)
Encrypt

Motivation

- Multi-point adaptation
  - Multiple problems can require multiple solutions
  - Adaptation location is not always flexible
- Protect end-to-end connection from failure
- Allow adaptations that alter data

Motivation

- Other approaches
  - Restrict the set of adaptations
    - Protocol Boosters, SNOOP
  - Assume reliability of adapting nodes
    - WTCP, MOWGLI
  - Provide a robust architecture
    - TACC

Reliability Infrastructure

- Provide adaptation for applications that expect reliable delivery
  - TCP, exactly once delivery
- End-to-end connection built using multi-split-TCP
  - Reliability between points of adaptation
  - Leverage existing technology
  - Adaptation at each node independent of TCP
- Must still address split-TCP issues

Outstanding Issues

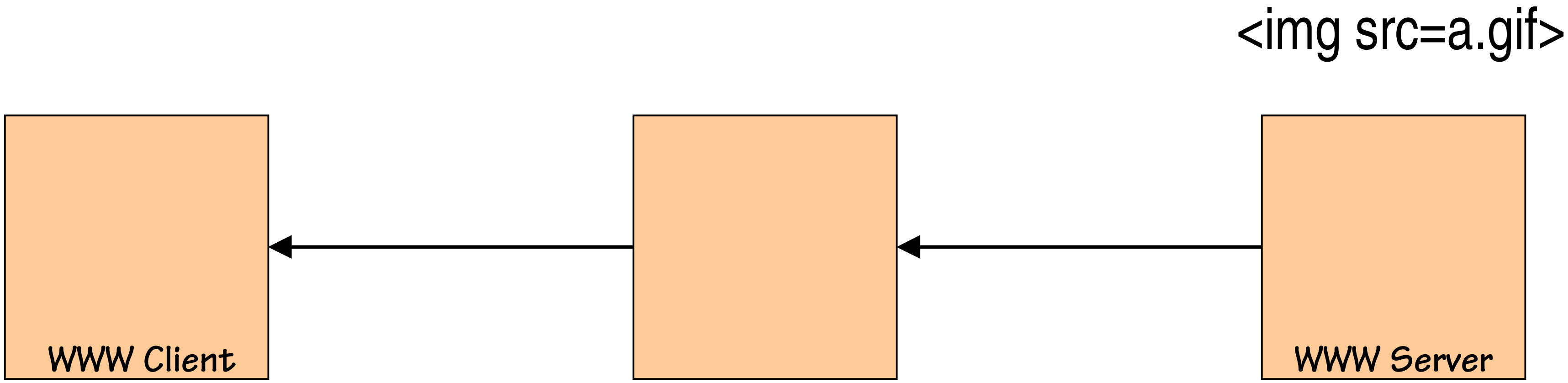
- Failure detection and recovery
- Reliability vs. Adaptation

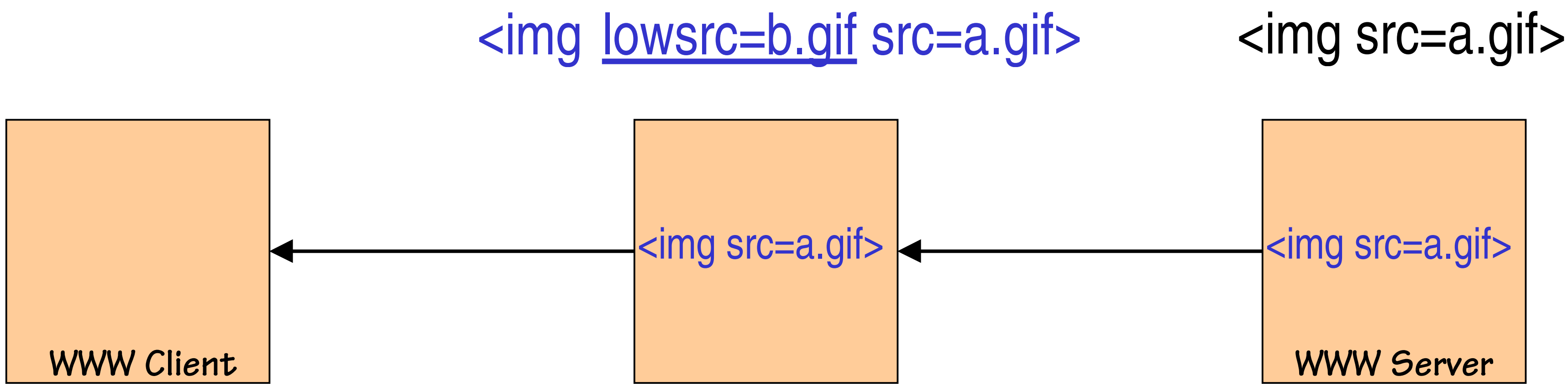
Failure Detection and Recovery

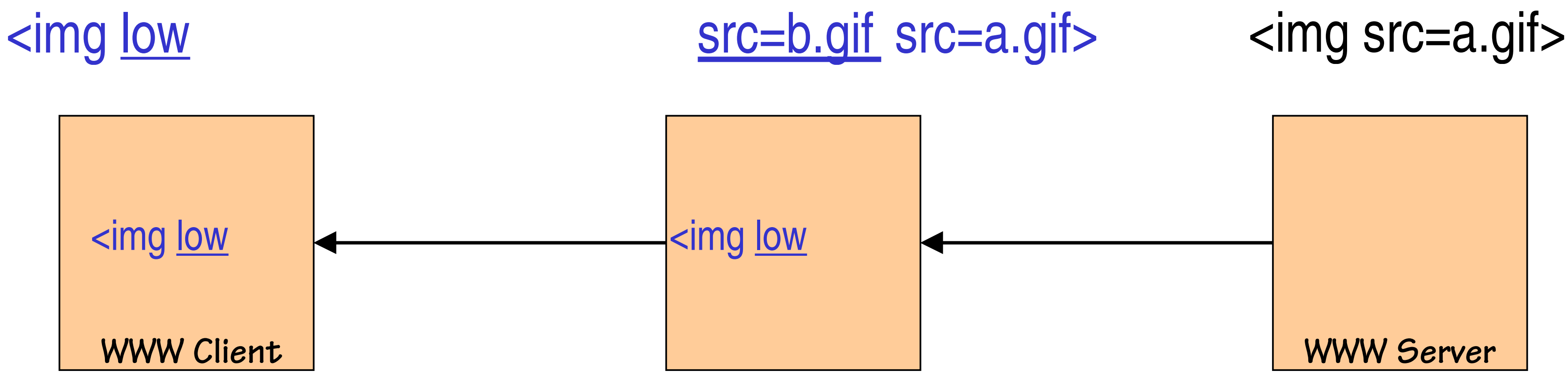
- Possible failures: adaptors, nodes, links
- Node and link failures detected as TCP connection failures
- Failure modes
  - Potential data loss
  - Partial adaptation of data
  - Lost adaptor state
  - Adaptor consistency

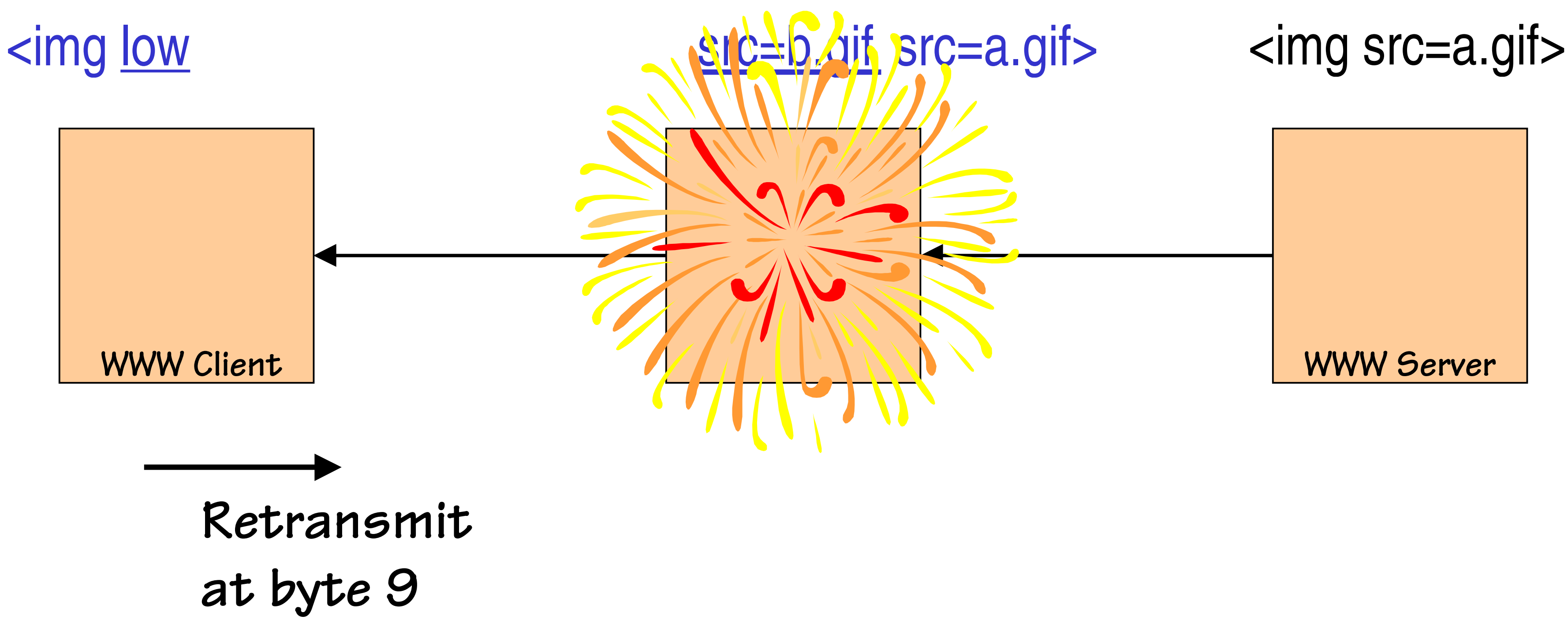
Adaptation vs. Reliability

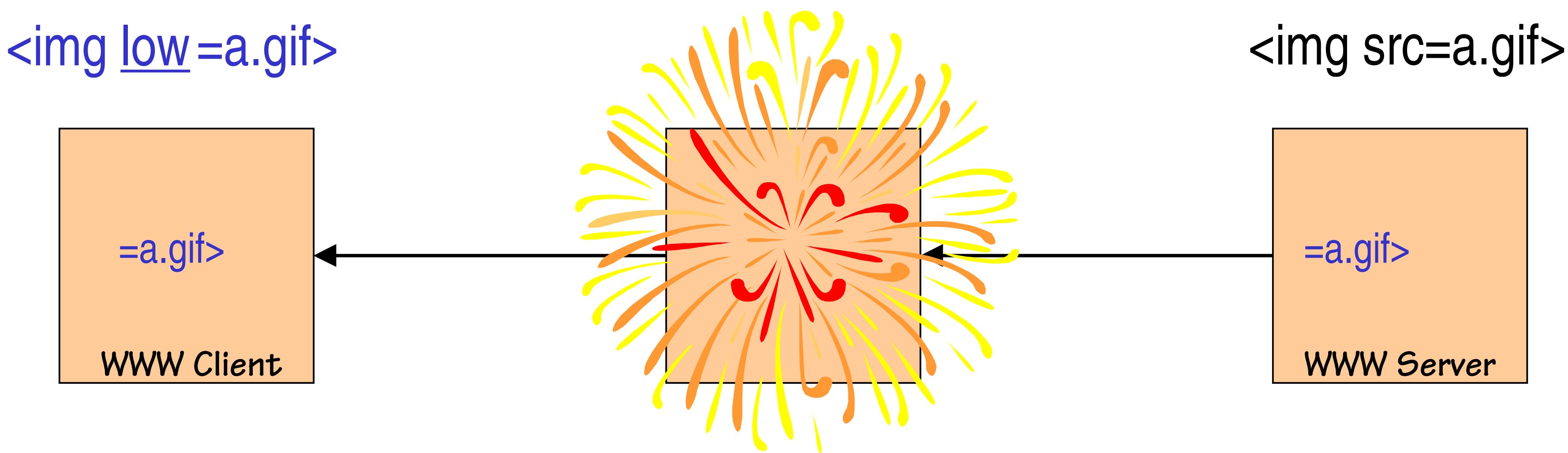
- Adaptation violates end-to-end reliability semantics
- End-to-end reliability typically assumes data immutability
  - Retransmission by byte or packet count
- Adaptation modifies data in transit
  - Need a new unit of retransmission


<img src=a.gif>
WWW Client
WWW Server


<img lowsrc=b.gif src=a.gif>
<img src=a.gif>
<img src=a.gif>
<img src=a.gif>
WWW Client
WWW Server


<img low
src=b.gif src=a.gif>
<img src=a.gif>
<img low
WWW Client
<img low
WWW Server


<img low
src=b.gif, src=a.gif>
<img src=a.gif>
WWW Client
WWW Server
Retransmit
at byte 9

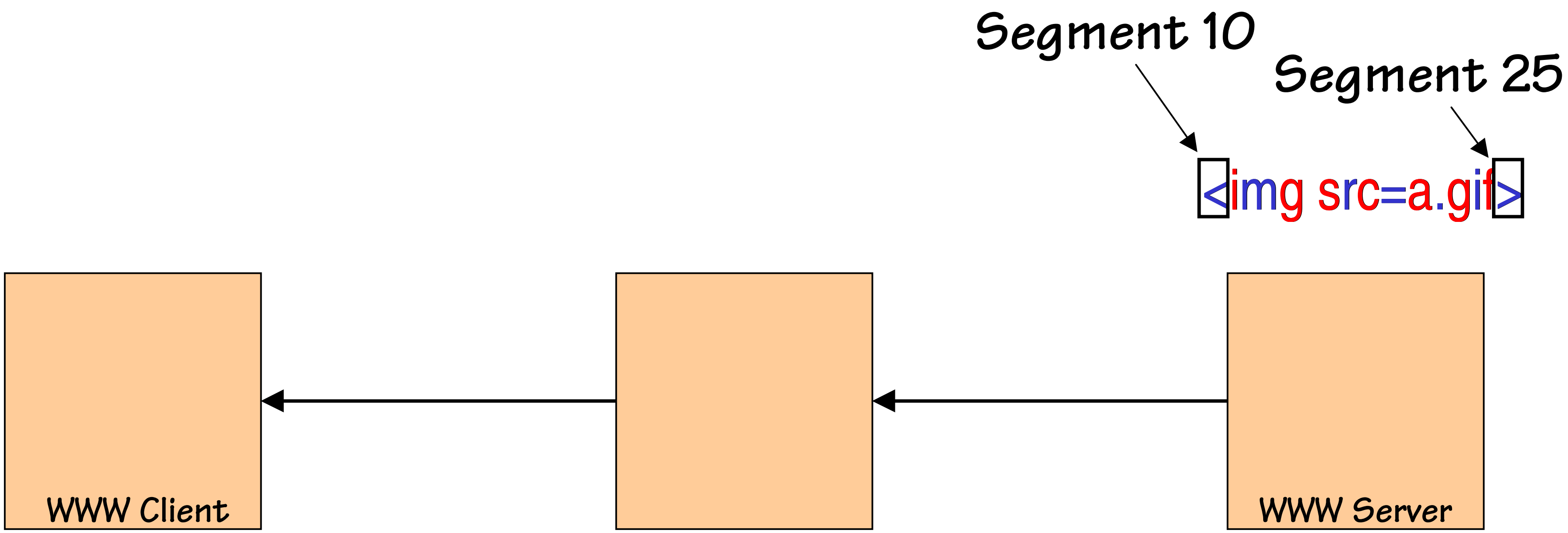

<img low=a.gif>
<img src=a.gif>
=a.gif>
WWW Client
=a.gif>
WWW Server

Solution Components

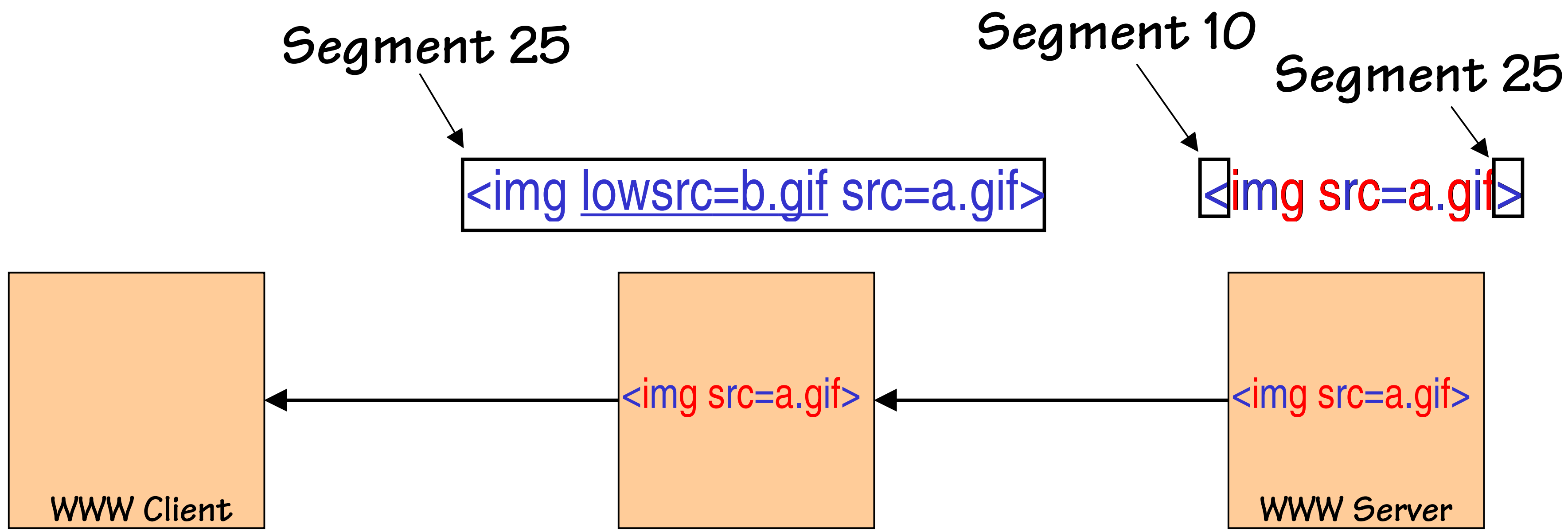
- Semantic Segmentation
- Adaptor consistency maintenance

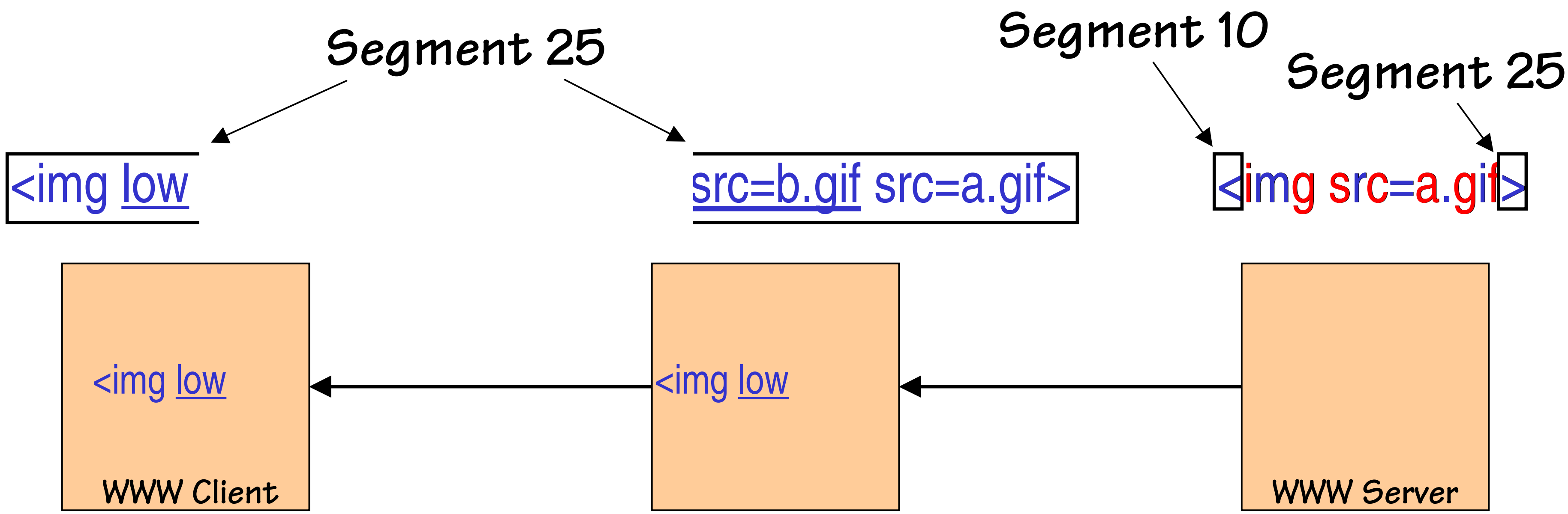
Semantic Segmentation

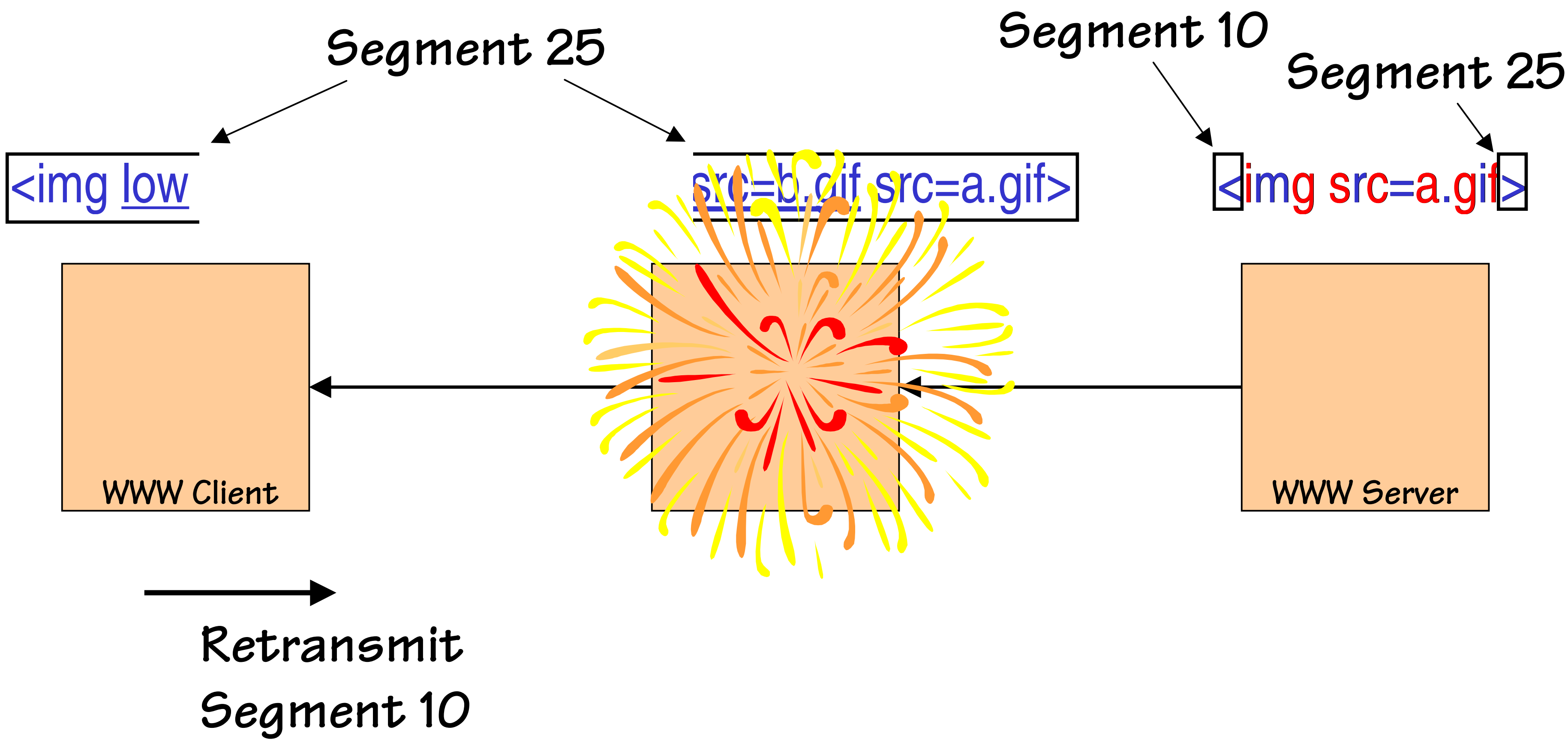
- *Semantic Segmentation*: a semantically meaningful unit of retransmission
  - Divide stream into semantic units
    - Dynamically and automatically, by adaptors
  - Preserve semantic meaning of each segment end-to-end
    - Maintained by segment combination
  - Allows adaptors to express recovery constraints

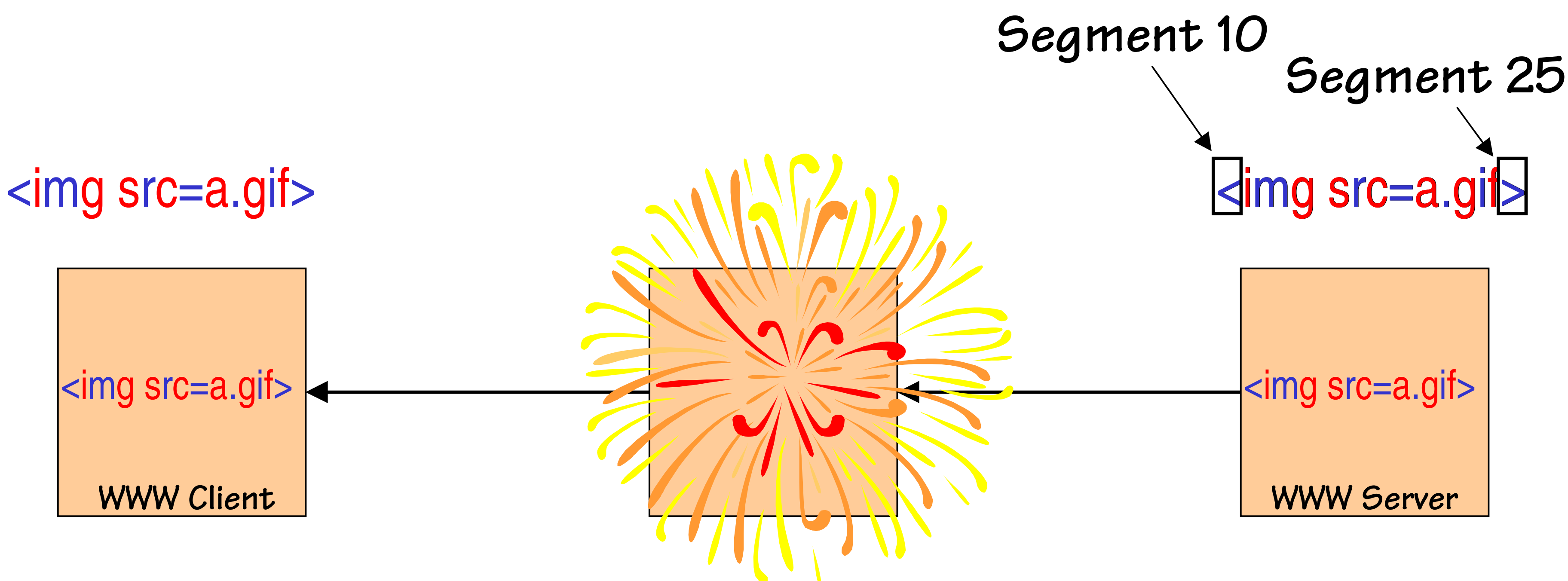


Segment 10
Segment 25
<img src=a.gif>
WWW Client
WWW Server


Segment 25
<img lowsrc=b.gif src=a.gif>
Segment 10
Segment 25
<img src=a.gif>
<img src=a.gif>
<img src=a.gif>
WWW Client
WWW Server


Segment 25
Segment 10
Segment 25
<img low
src=b.gif src=a.gif>
<img src=a.gif>
<img low
WWW Client
<img low
WWW Server


Segment 25
Segment 10
Segment 25
<img low
src=b.gif src=a.gif>
<img src=a.gif>
WWW Client
WWW Server
Retransmit
Segment 10



<img src=a.gif>
Segment 10
Segment 25
<img src=a.gif>
<img src=a.gif>
WWW Client
<img src=a.gif>
WWW Server

Rules of Segmentation

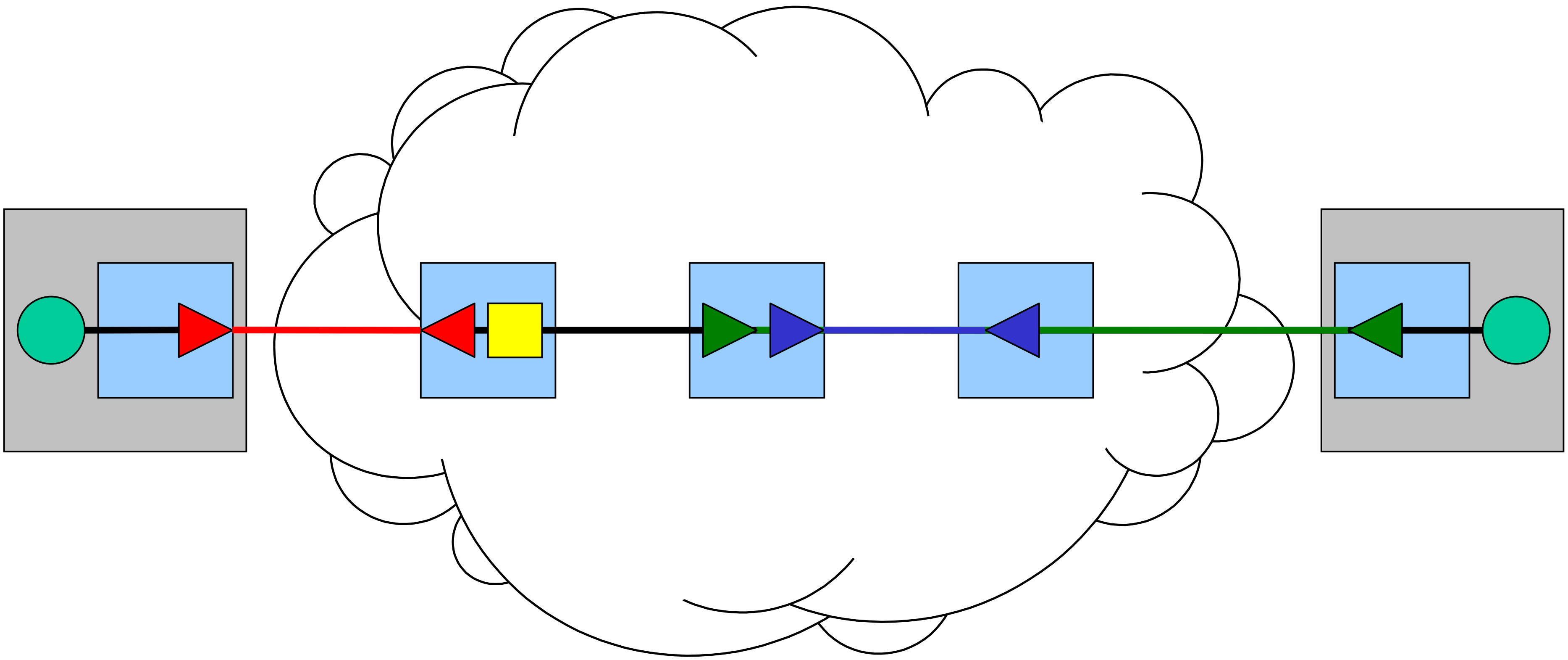
- Start with one byte segments
- Constrain each stream modification to one segment
- Combine segments where necessary
  - Not reversible
  - New segment contains combined semantic meaning
- Final delivery of complete segments only

Benefits of Segmentation

- Service guarantees:
  - Transaction-like adaptation (all or nothing)
  - Exactly-once delivery of an acceptable form of each semantic element

- Other reliability models are possible

Adaptor Consistency



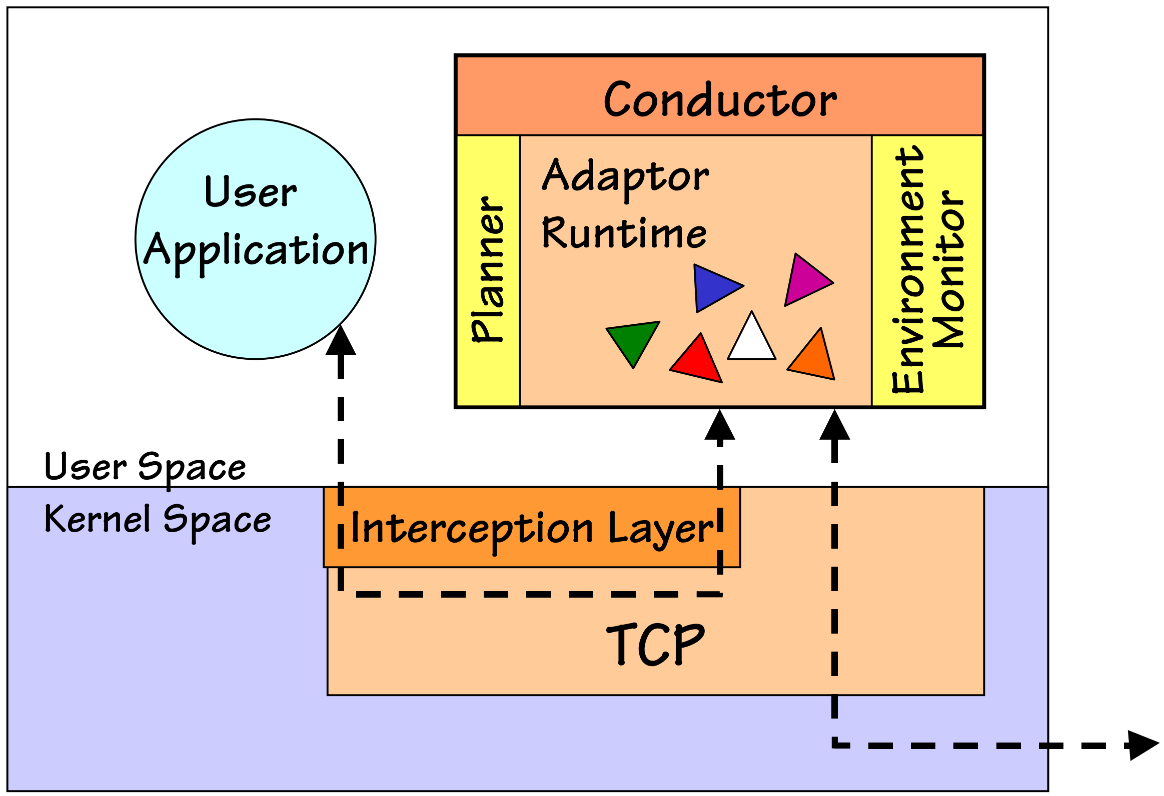
Adaptor Consistency

- Adaptor state not saved across failure
  - Ex: decompression dictionary, decryption key
  - Replacement adaptors will be out of sync
- Recovery
  - Locate/remove partners of failed adaptors
  - Remove adaptors that depended on the presence of a failed partner
  - Optionally re-deploy failed and removed adaptors

Conductor Design Goals

- Application-level, connection-oriented protocol adaptation
- Support heterogeneous networks
- Application transparent
- Automatic, but user controllable
- Arbitrary adaptations
- Easy-to-deploy adaptations
- Reliable

A Conductor-Enabled Node



Concluding Remarks

- Many open architectures allow distributed adaptation
- Adaptation need not and should not reduce the reliability of the system
  - Requires a new reliability model
- Conductor is a prototype of the proposed solution