Challenges in Distributed Adaptation

Mark Yarvis

UCLA

March 2, 2000

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

Outline

- Intro to Adaptive Networking
- Distributed Adaptation
- Conductor Architecture
- Challenges

Intro to Adaptive Networking

- Applications: increasingly network dependent
  - Internet radio/movies, gaming, MS Office
  - Thin clients
  - Internet appliances
- Minimum level of service assumed

Intro to Adaptive Networking

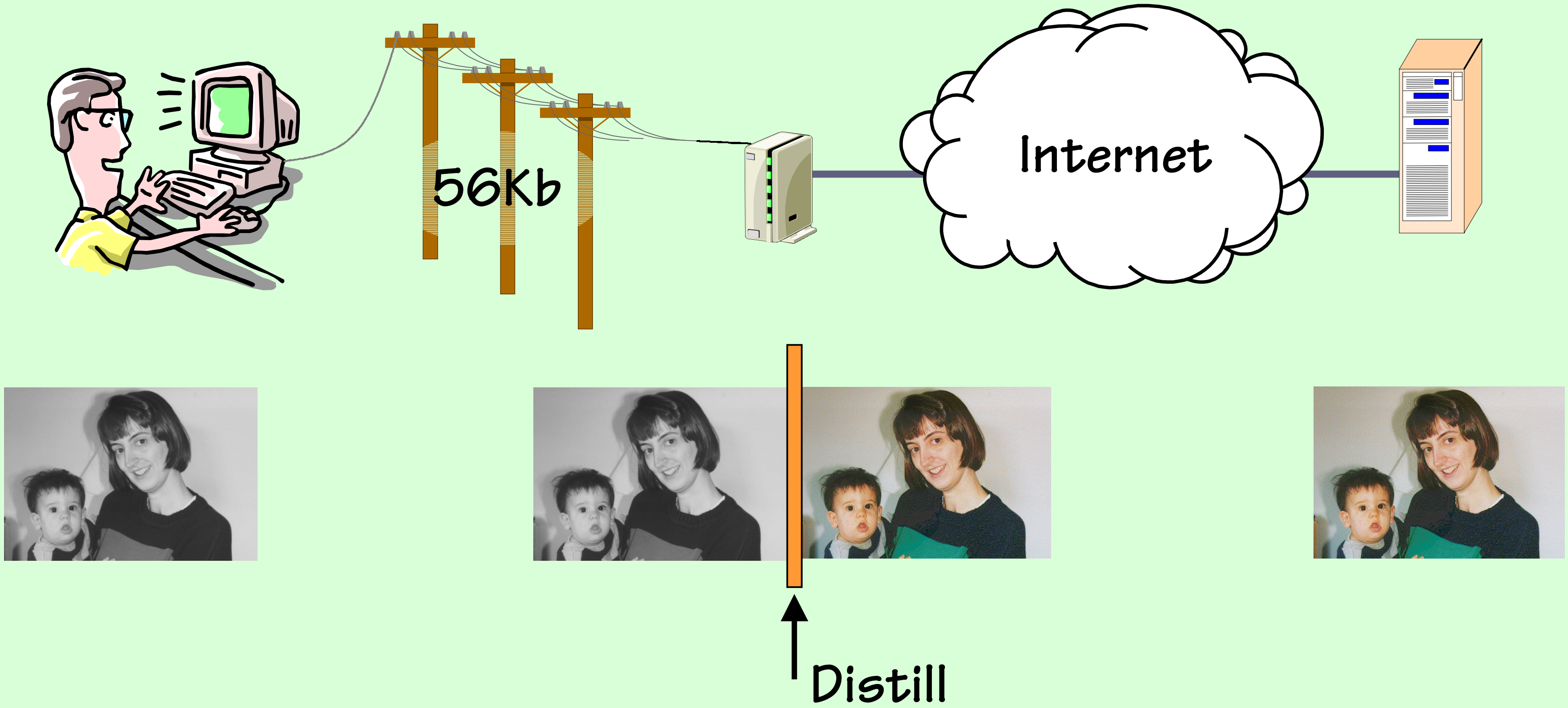
- **Networks: not always fast and free**
  - **Bandwidth, latency, jitter, security, $$, reliability**
- **Applications should provide gracefully degraded service**
  - **Research focus: last mile**

Achieving Graceful Degradation

- Use different applications
  - PalmOS clipping applications
- Write adaptive applications
  - Odyssey [Nob97], Rover[Jos95]
  - RealPlayer
- Adapt protocols within the network

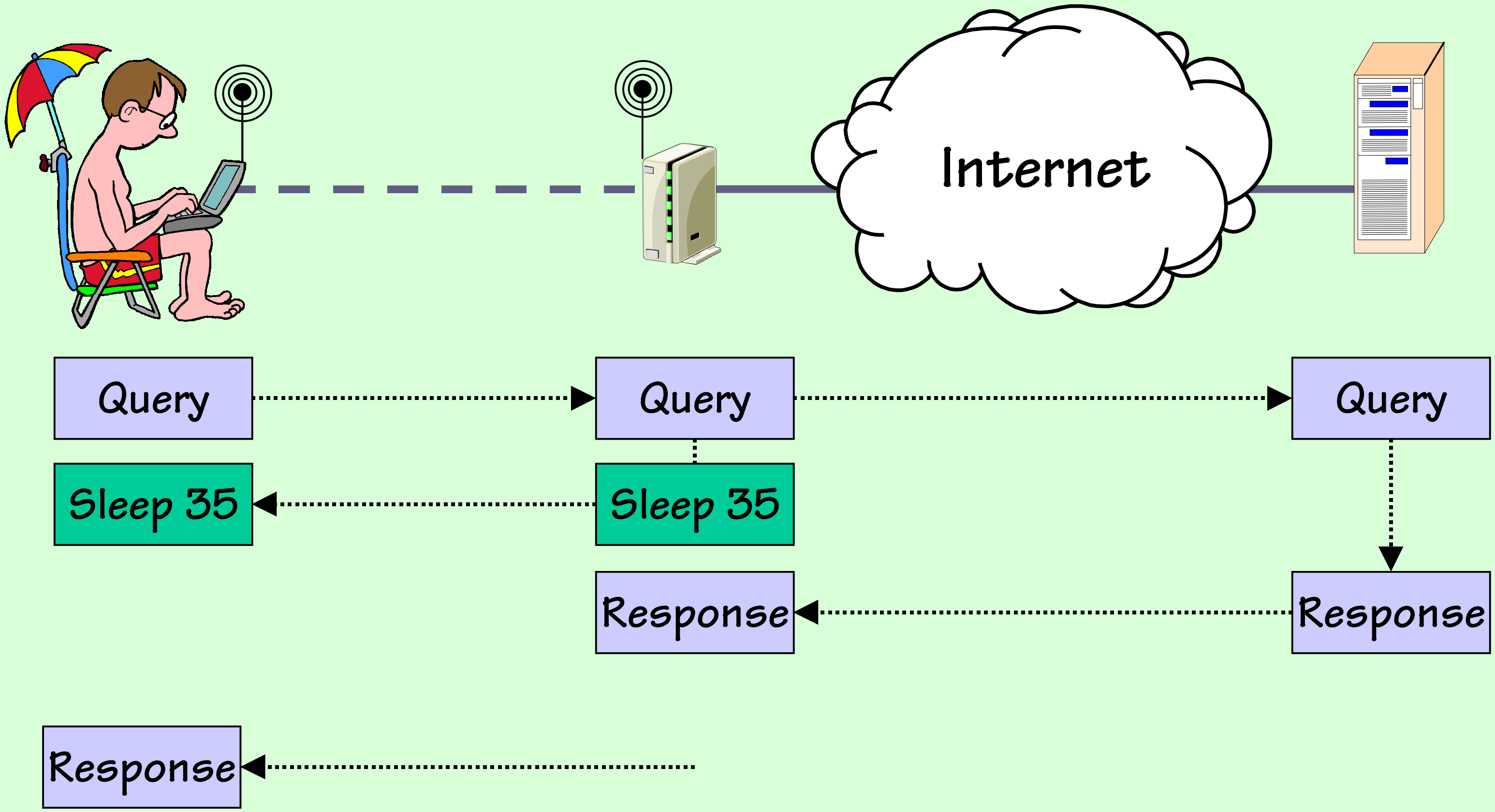
Distilling the Web

Trade: Quality for transfer time



Link Scheduling

Trade: Latency for battery power



Other Forms of Adaptation

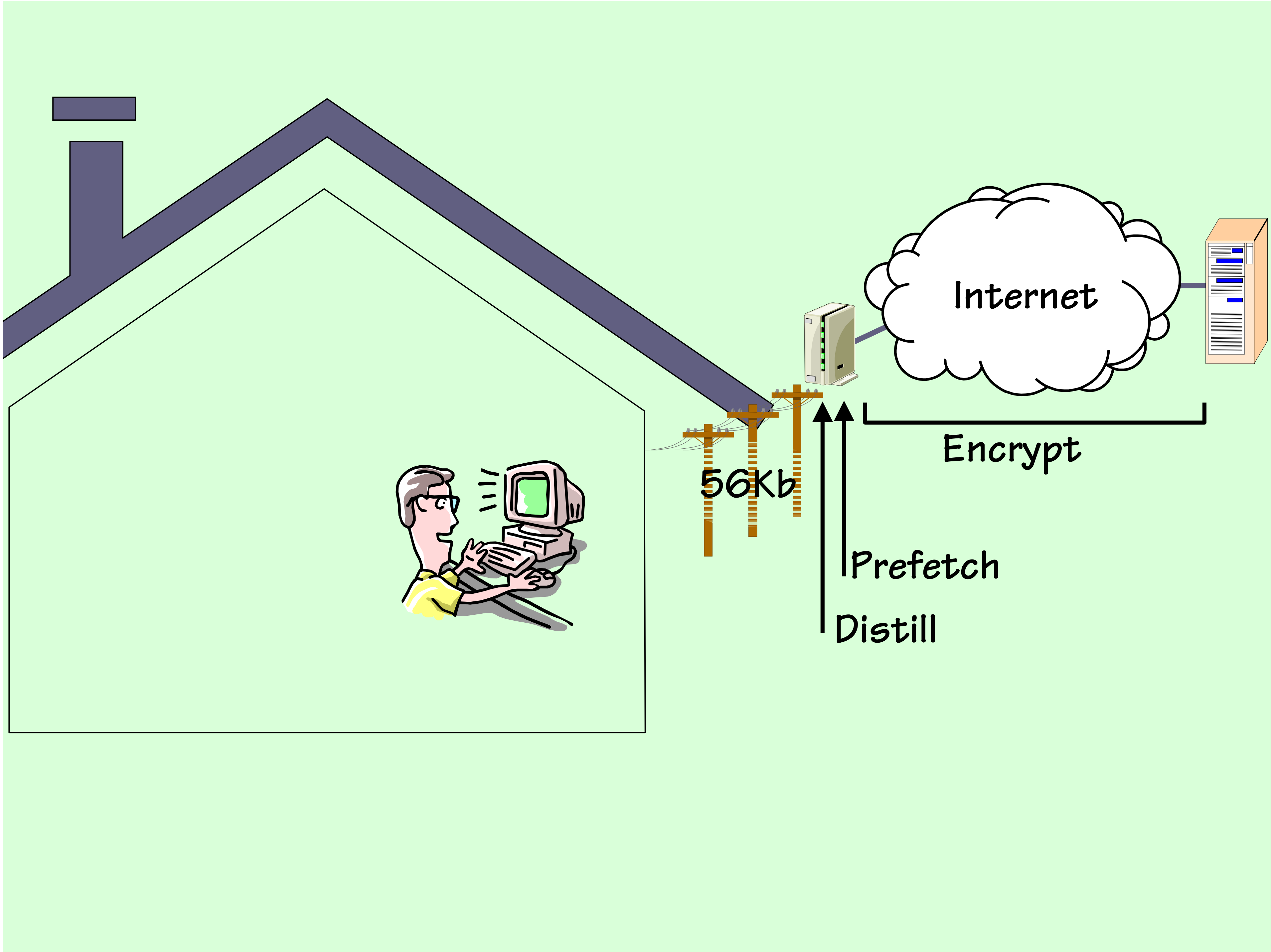
- Application layer
  - Distill, compress, encrypt, cache
- Network and transport layers
  - Link scheduling, prioritization, FEC
  - Snoop [Bal95]

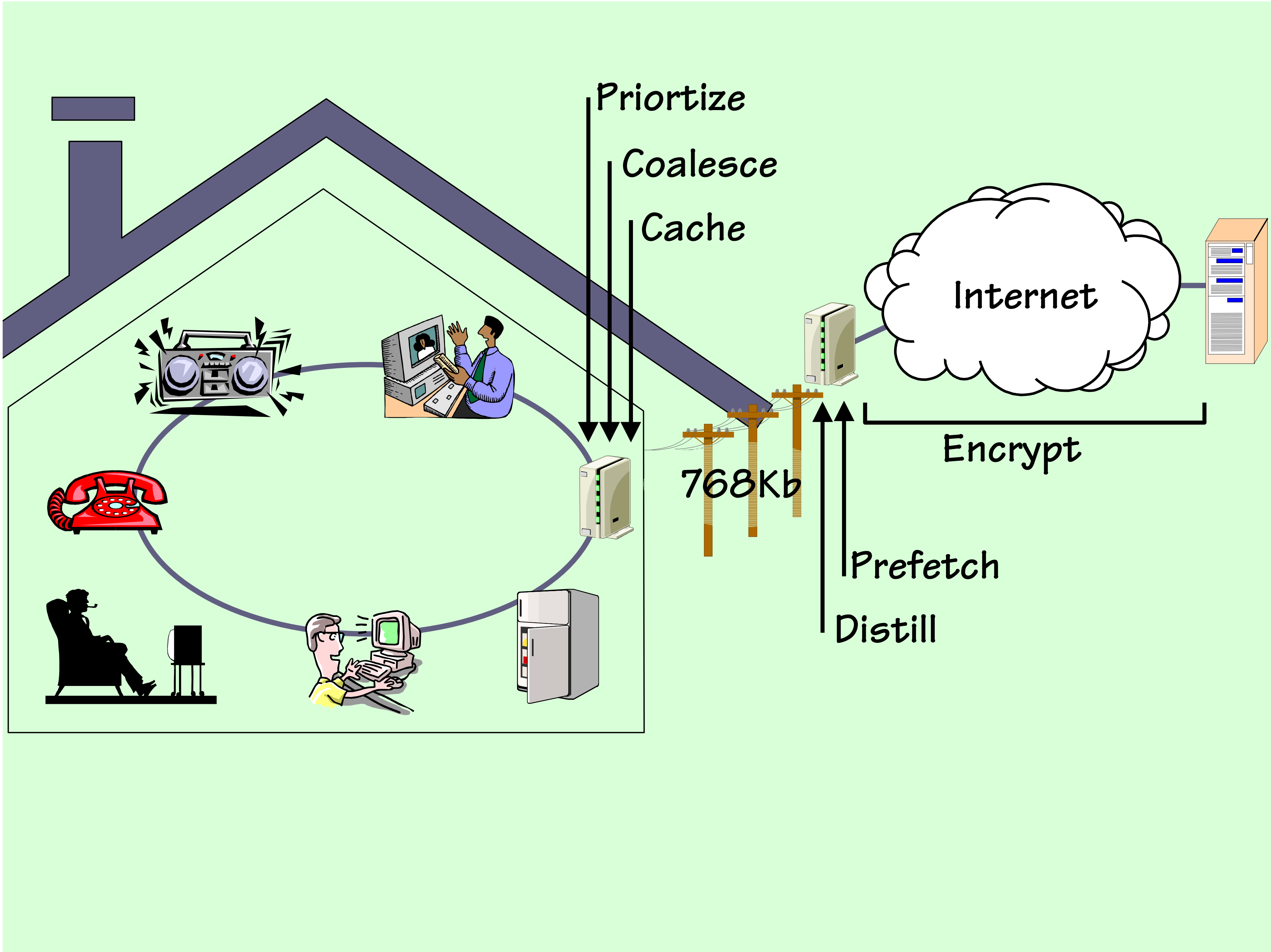
State of the Art

- Daedalus [Fox98]
- Protocol Boosters [Mal97]
- Transformer Tunnels [Sud98]
- Focus:
  - Last mile
  - Independent adaptation

Beyond the "Last Mile"

- Leaf nodes become leaf networks
  - Home/office wired nets
  - Home/office *wireless* nets
  - Personal area nets
- User-to-user services
- Multi-hop networks
- Network/server congestion


Internet
Encrypt
56Kb
Prefetch
Distill


Priortize
Coalesce
Cache
Internet
Encrypt
768Kb
Prefetch
Distill

Possible Approaches

- Solve end-to-end ✗
- Single proxy node ✗
- Independent solutions ✗
- Distributed adaptation ✓

Factors in Adaptor Placement

- Placement of adaptation is restricted by
  - Access to link status/control
  - Adaptation conflicts
  - Topology
  - Trust
  - Node resources
  - Load balancing

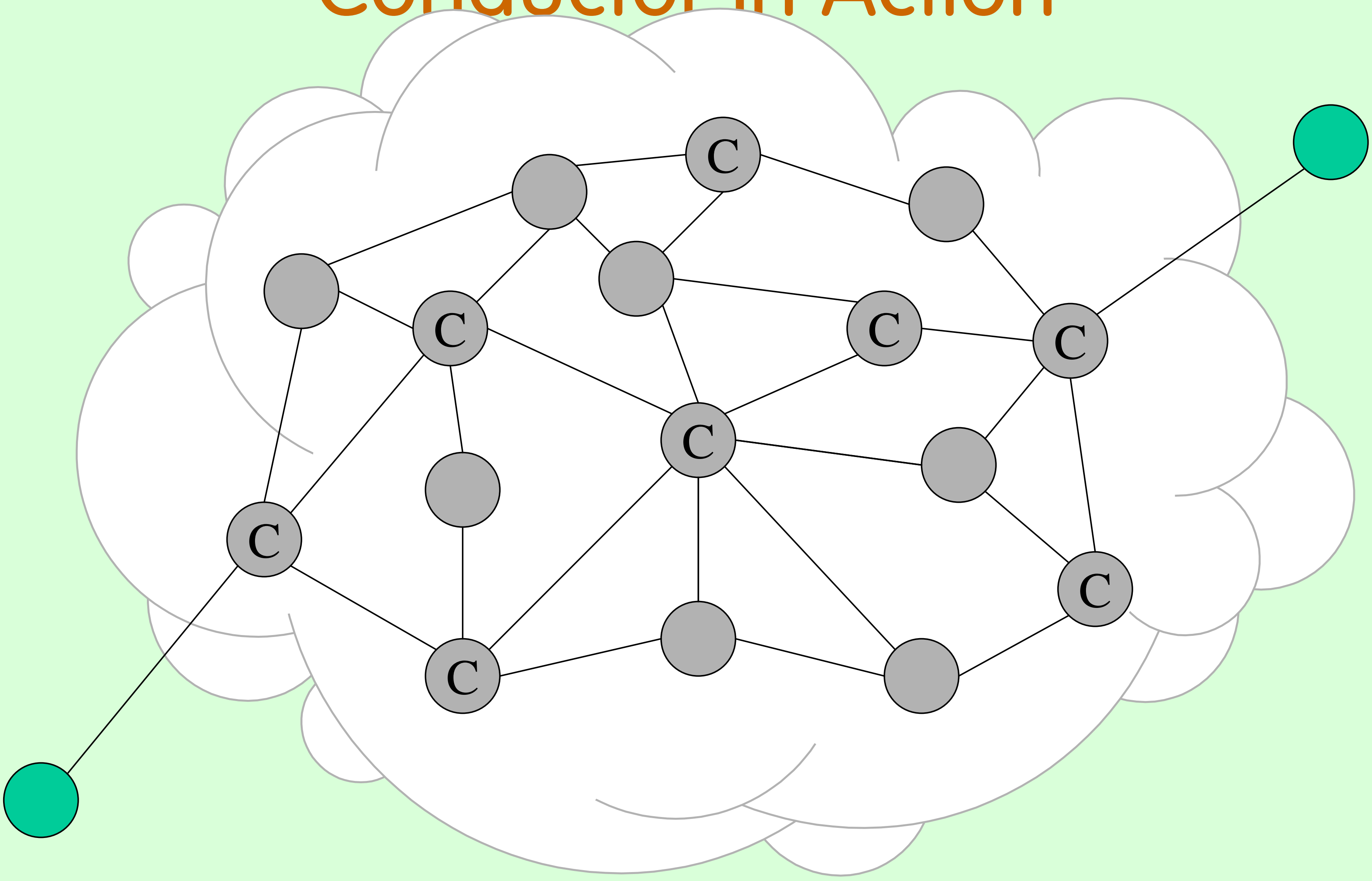
Conductor Design Goals

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

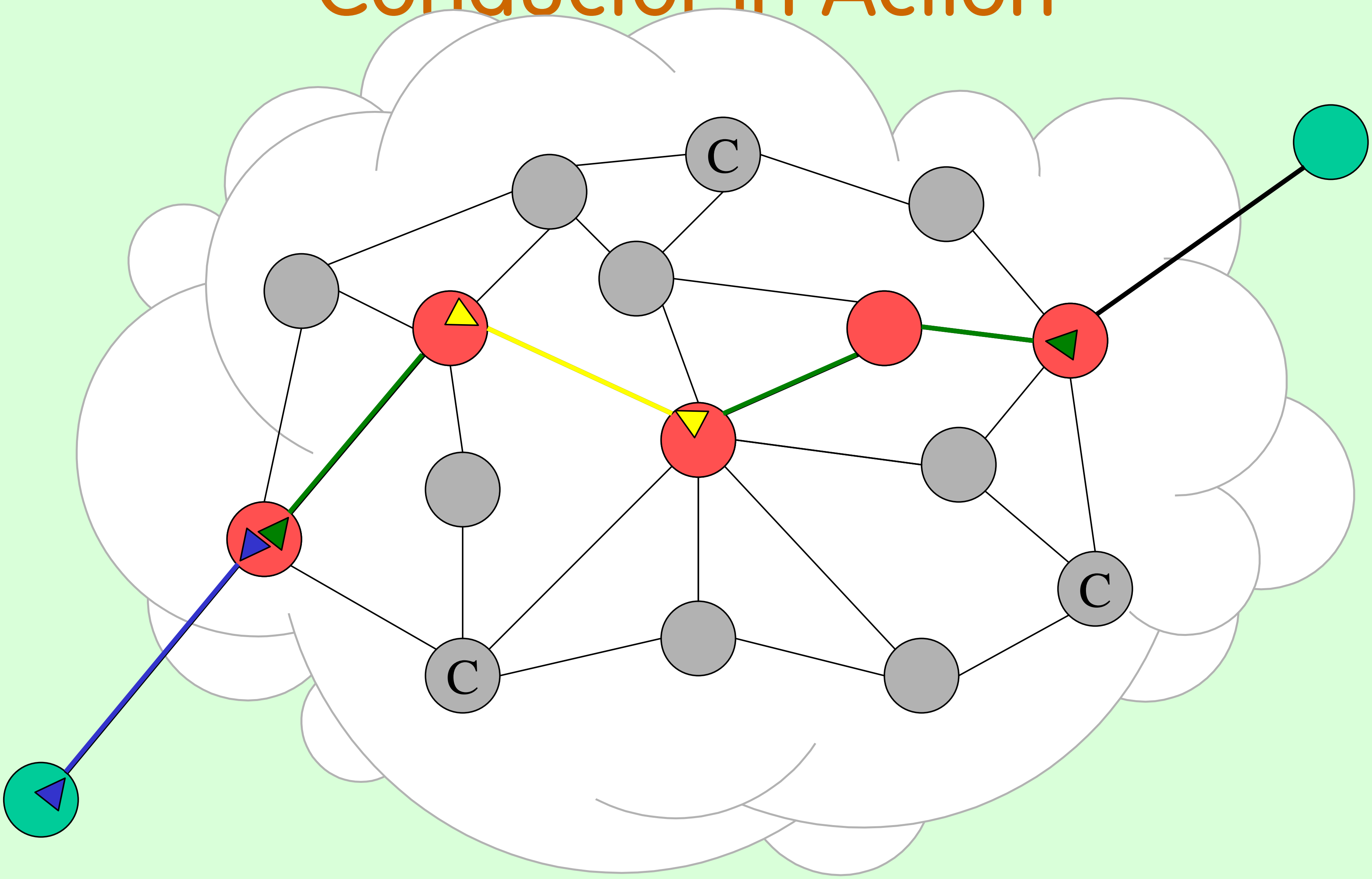
Conductor Architecture

- Adaptation framework
  - Transparent interception and routing
  - Node/link status monitoring
  - Distributed planning and deployment
  - Adaptor runtime environment
- Adaptor modules
  - Operate on data stream
  - Frequently paired

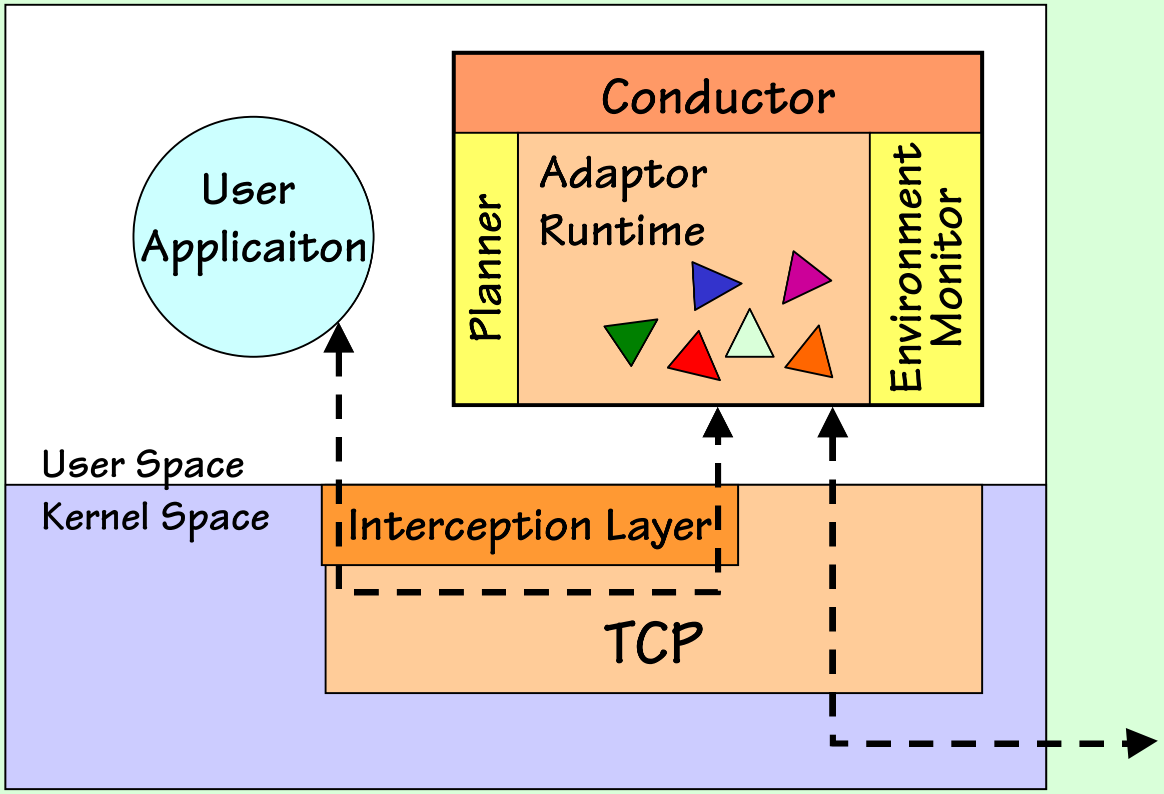
Conductor in Action



Conductor in Action



A Conductor-Enabled Node



Challenges in Distributed Adaptation

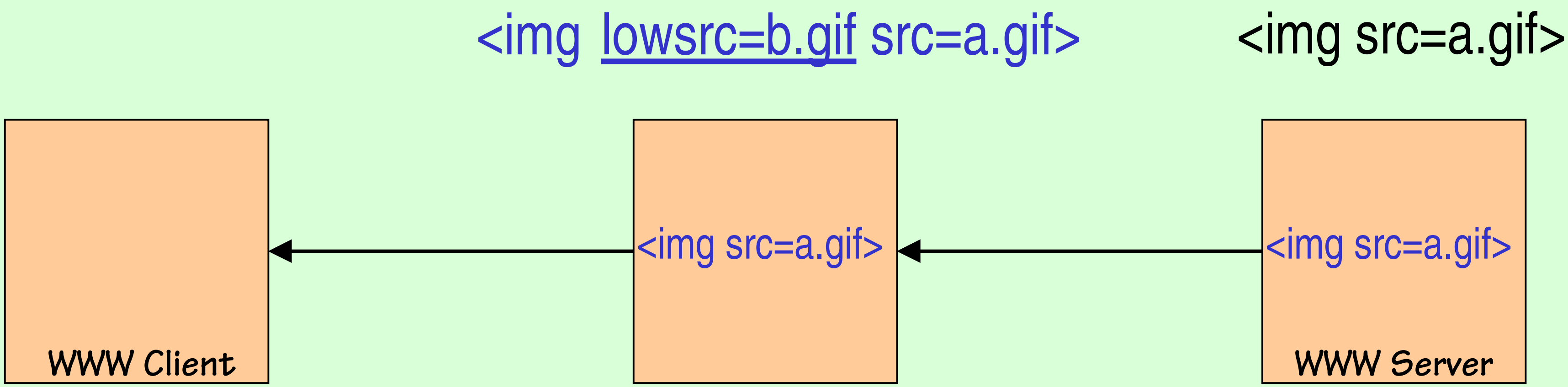
- Reliable Transmission
- Automated Planning
- Secure Adaptation

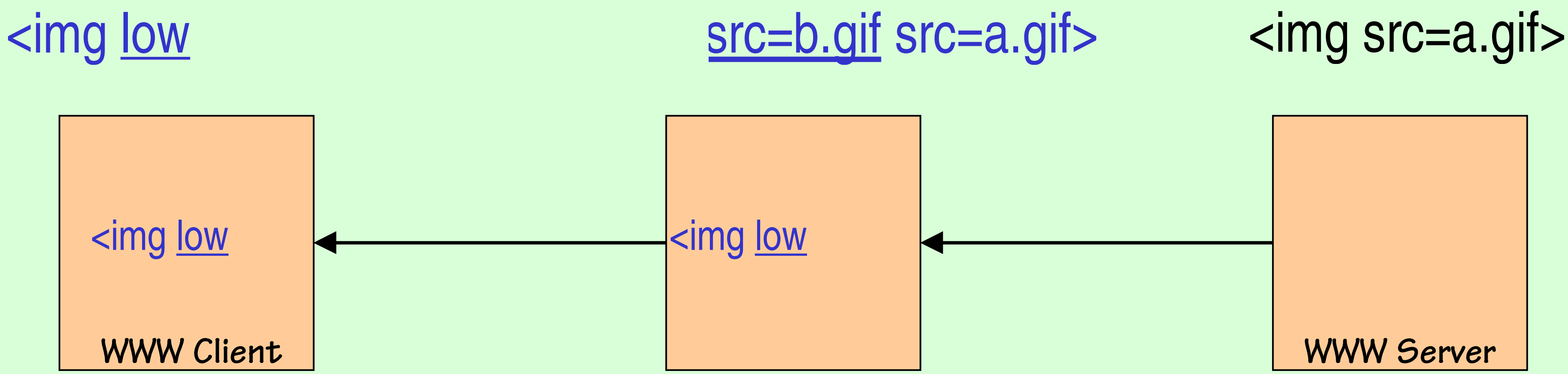
“A distributed system is one in which the failure of a computer you didn’t even know existed can render your own computer unusable”

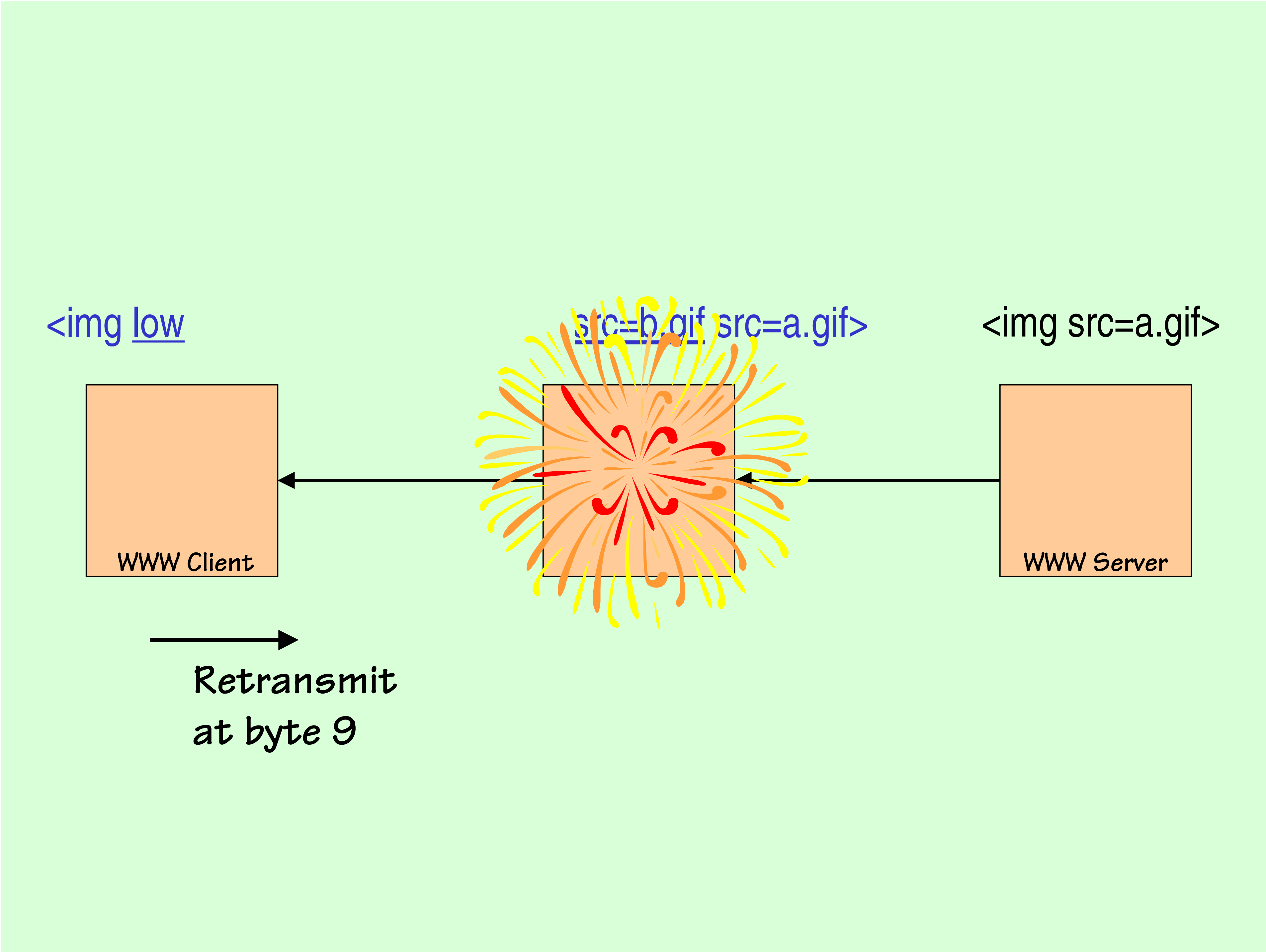
— Leslie Lamport, May 1987

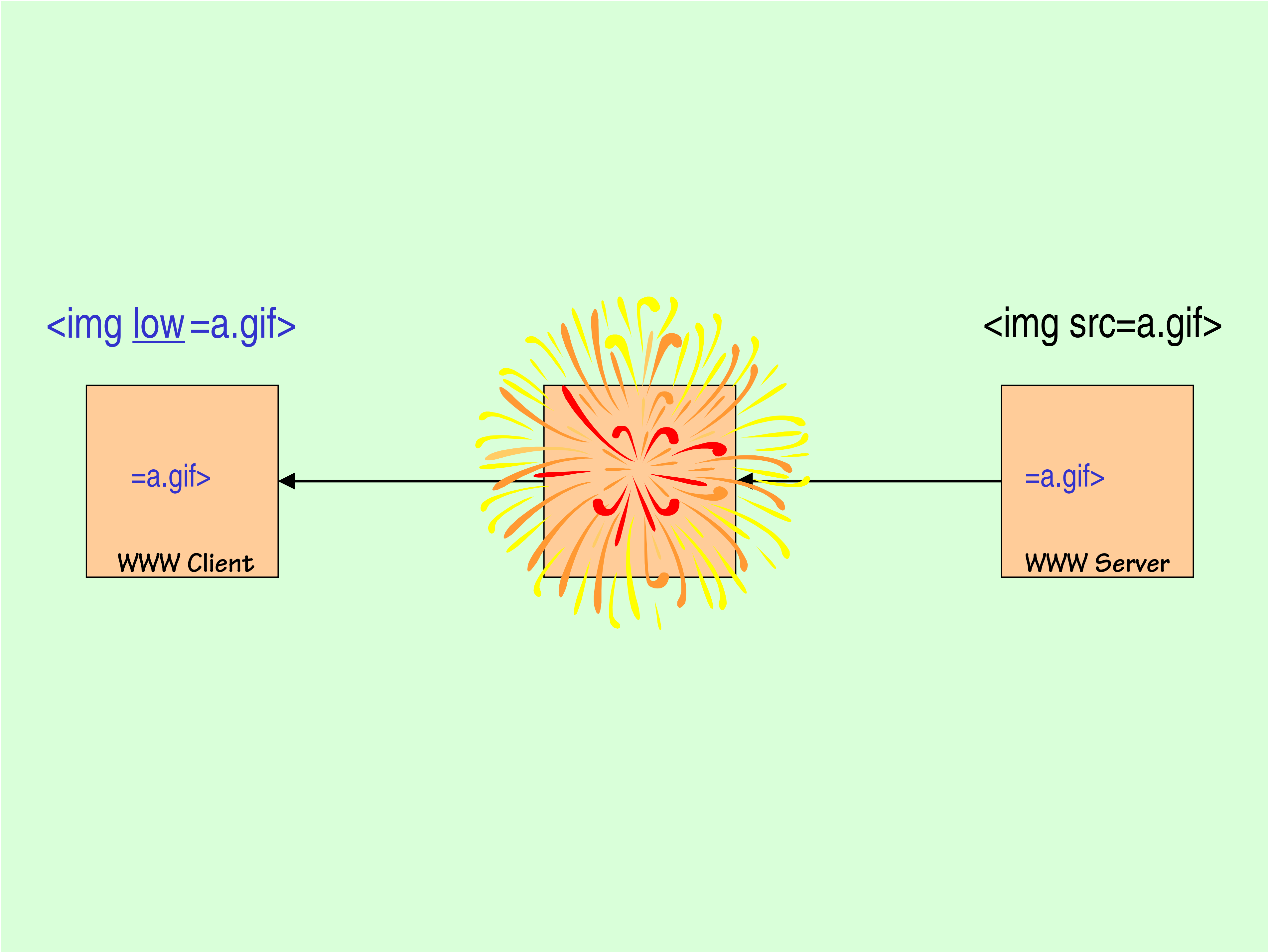
Reliable Transmission

- Distribution introduces new points of failure
- 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 lowsrc=b.gif src=a.gif>
<img src=a.gif>
<img src=a.gif>
<img src=a.gif>
WWW Client
WWW Server

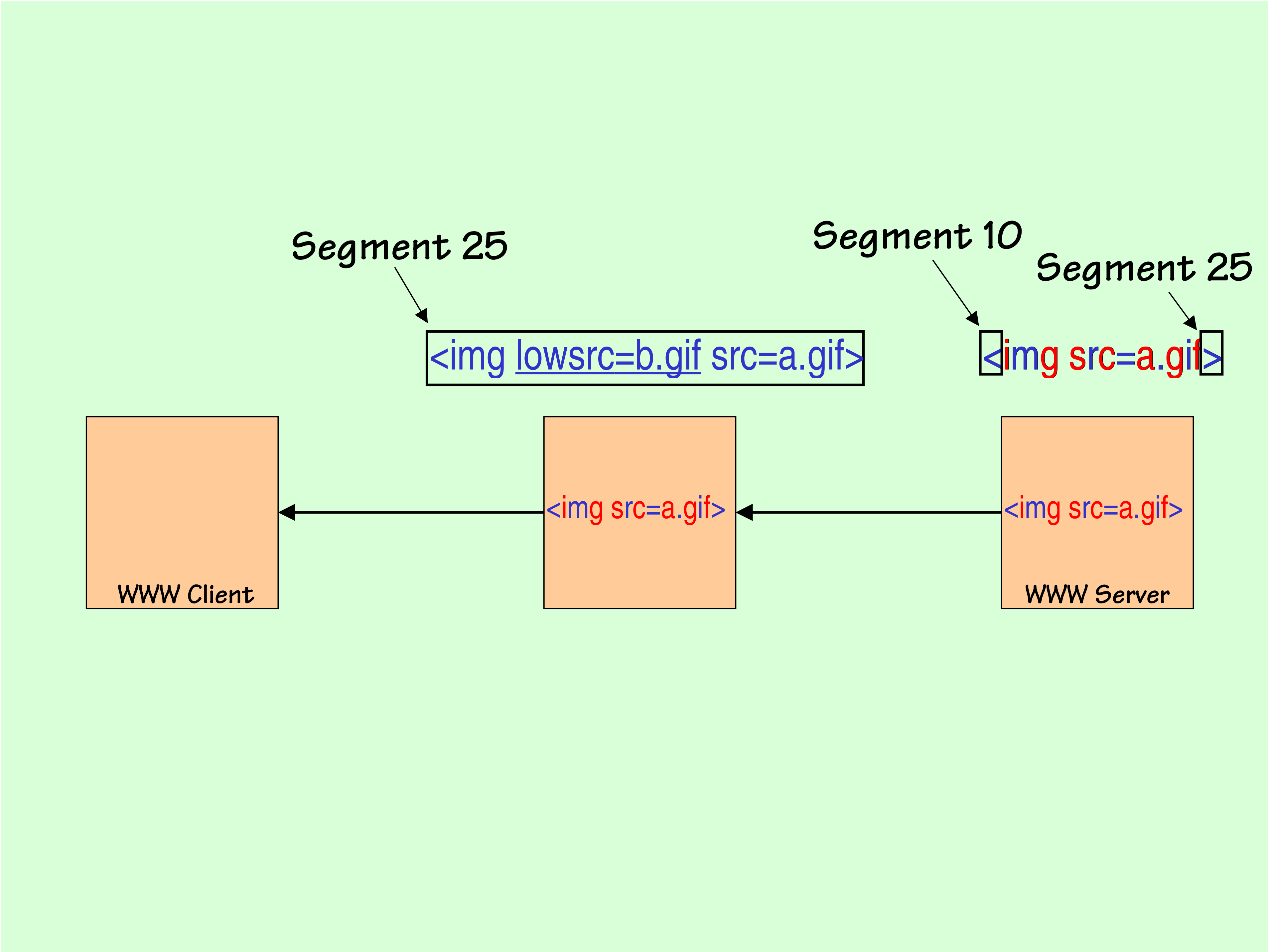

<img low
src=b.gif src=a.gif>
<img src=a.gif>
<img low
<img low
WWW Client
WWW Server

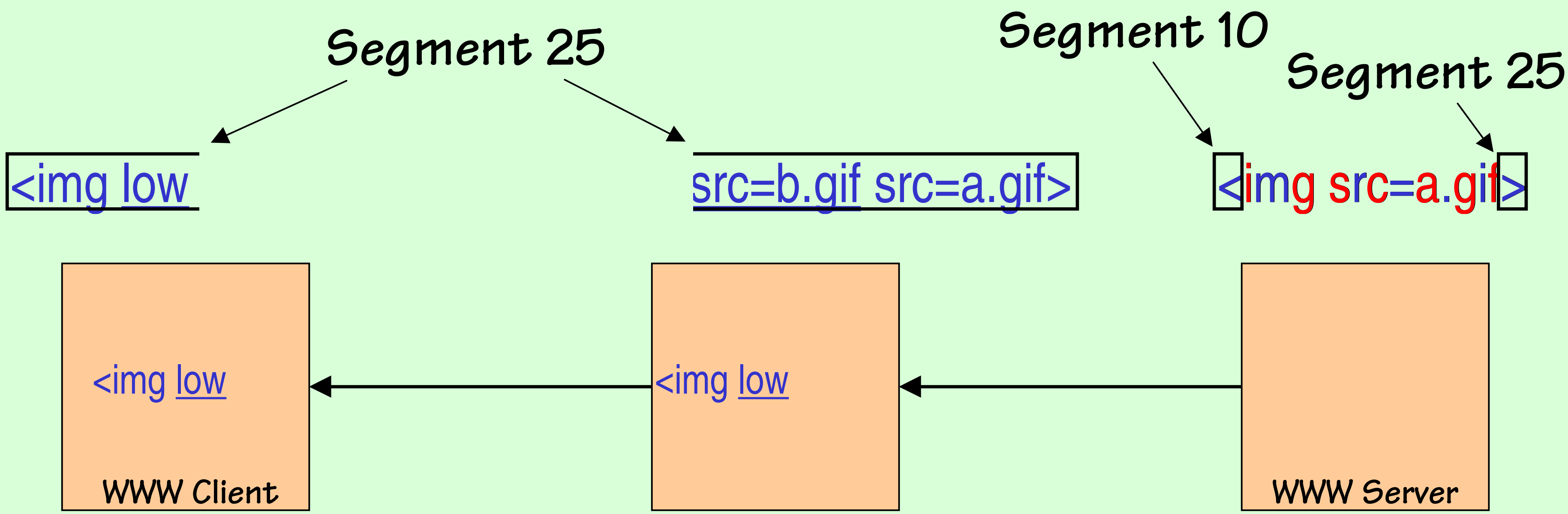

<img low
src=b.gif src=a.gif>
<img src=a.gif>
WWW Client
WWW Server
Retransmit
at byte 9


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

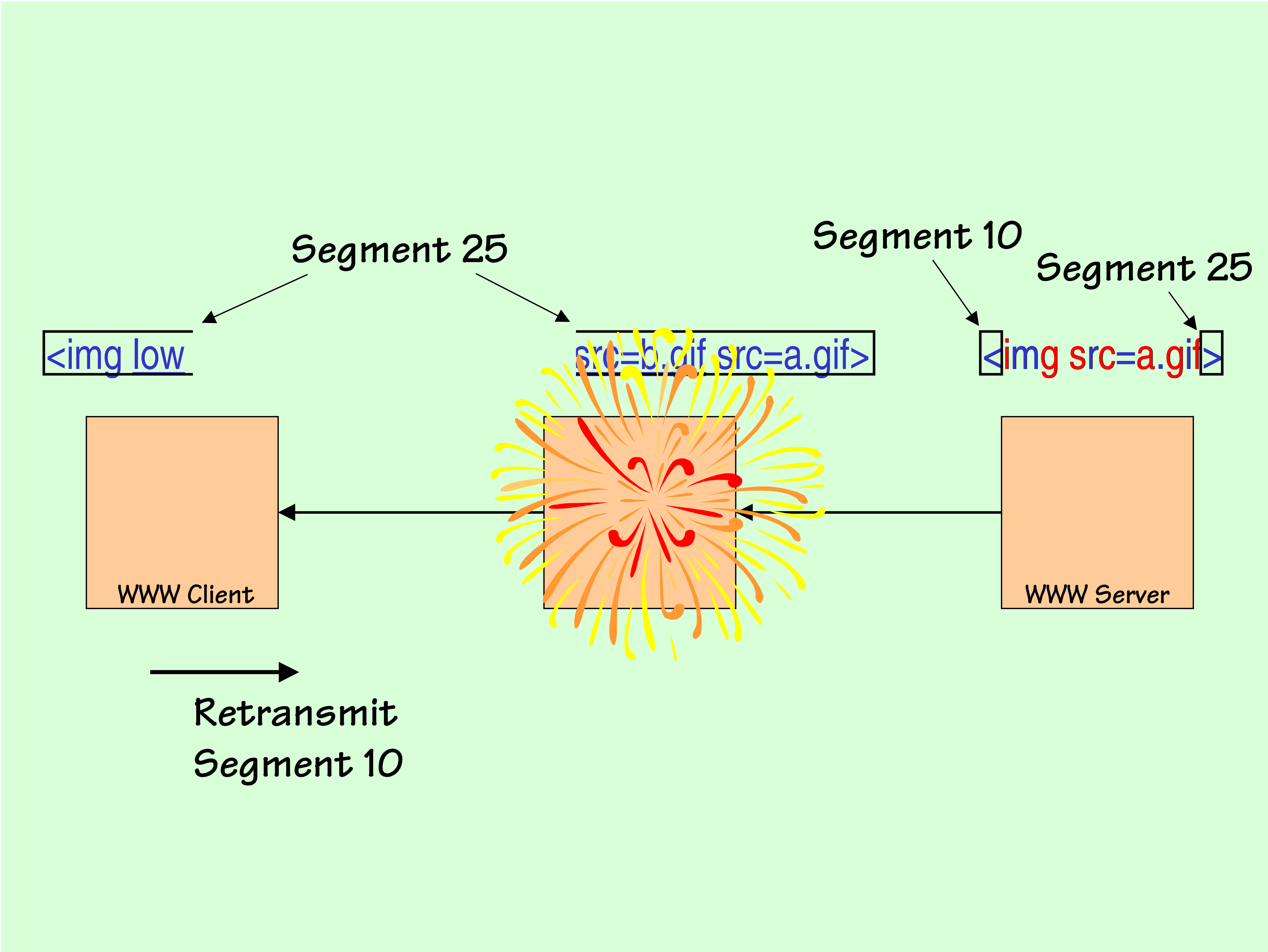
Reliable Transmission

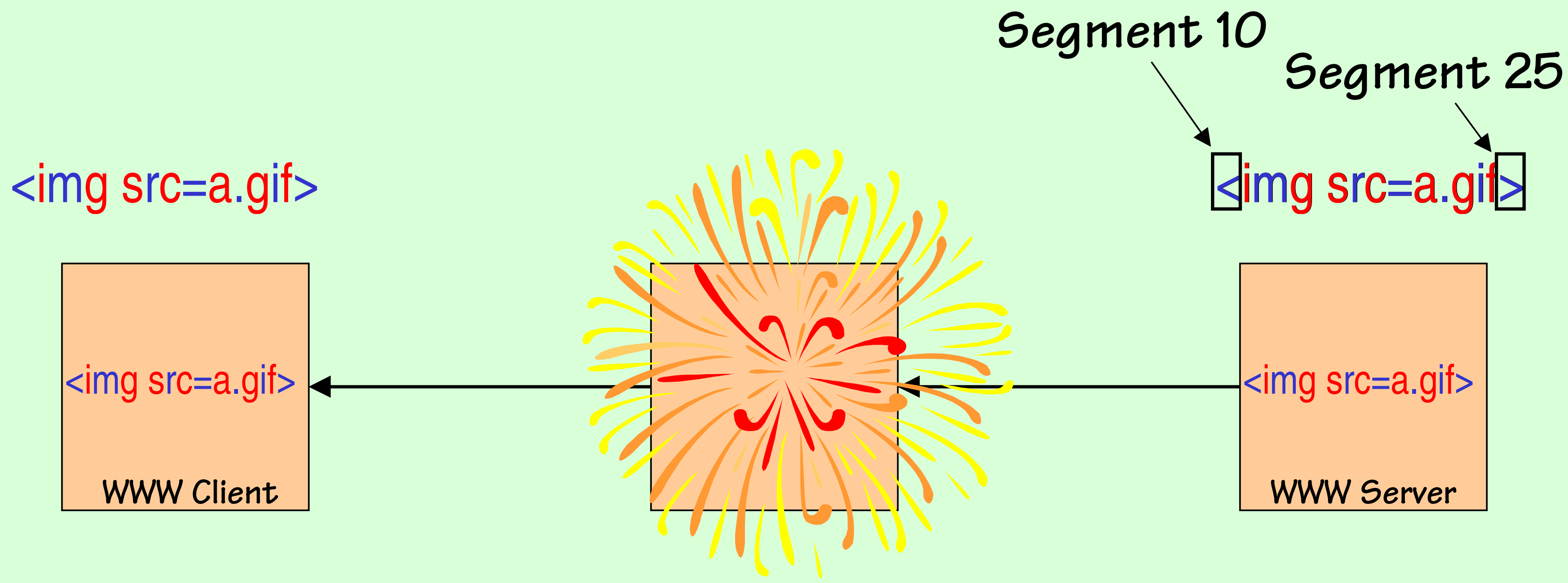
- *Semantic Segmentation*: a semantically meaningful unit of retransmission
  - Divide stream into semantic units
    - Dynamically, based on data type and adaptation
  - Preserve semantic meaning of each segment end-to-end
    - Maintained by segment combination
  - Allows adaptors to express recovery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


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



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

Reliable Transmission

- Rules of segmentation
  - 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

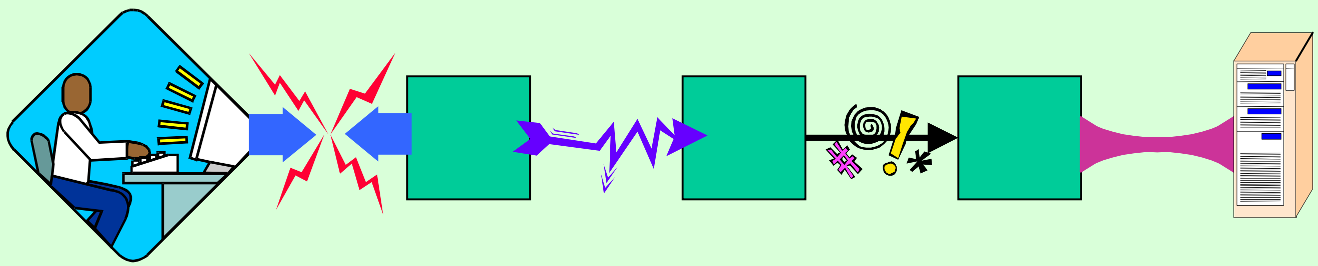
Reliable Transmission

- Service guarantees:
  - Transaction-like adaptation (all or nothing)
  - Exactly-once delivery of some form of each semantic element
- Other reliability models are possible

Challenges in Distributed Adaptation

✔Reliable Transmission
- Automated Planning
- Secure Adaptation

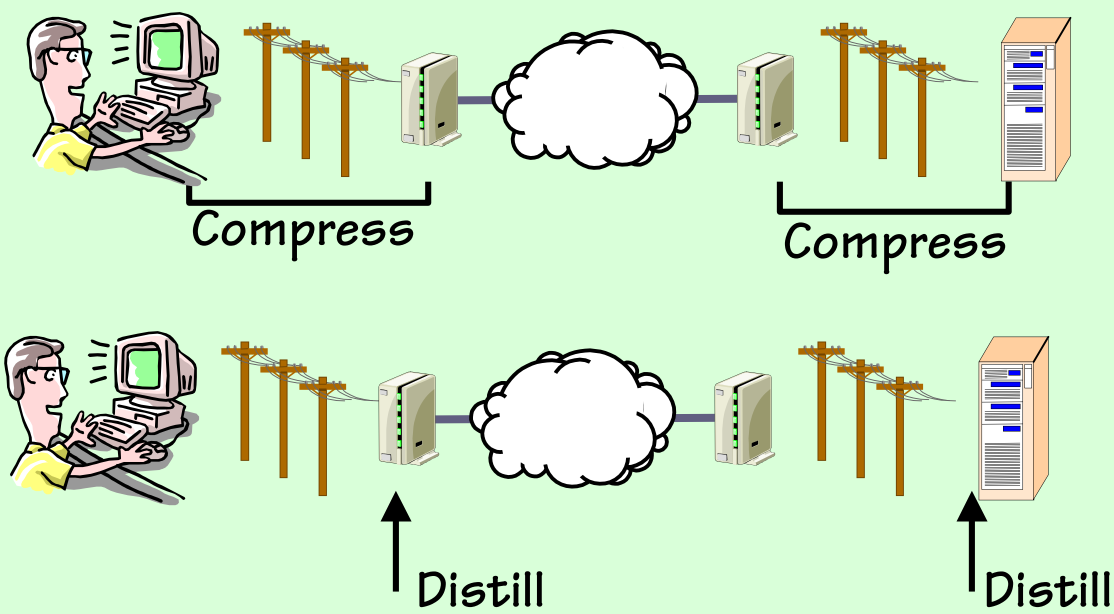
Automated Planning



- **Goal:**
  - Select which adaptors and where to put them
- **Based on:**
  - Link characteristics
  - Node resources
  - Available adaptors

Automated Planning

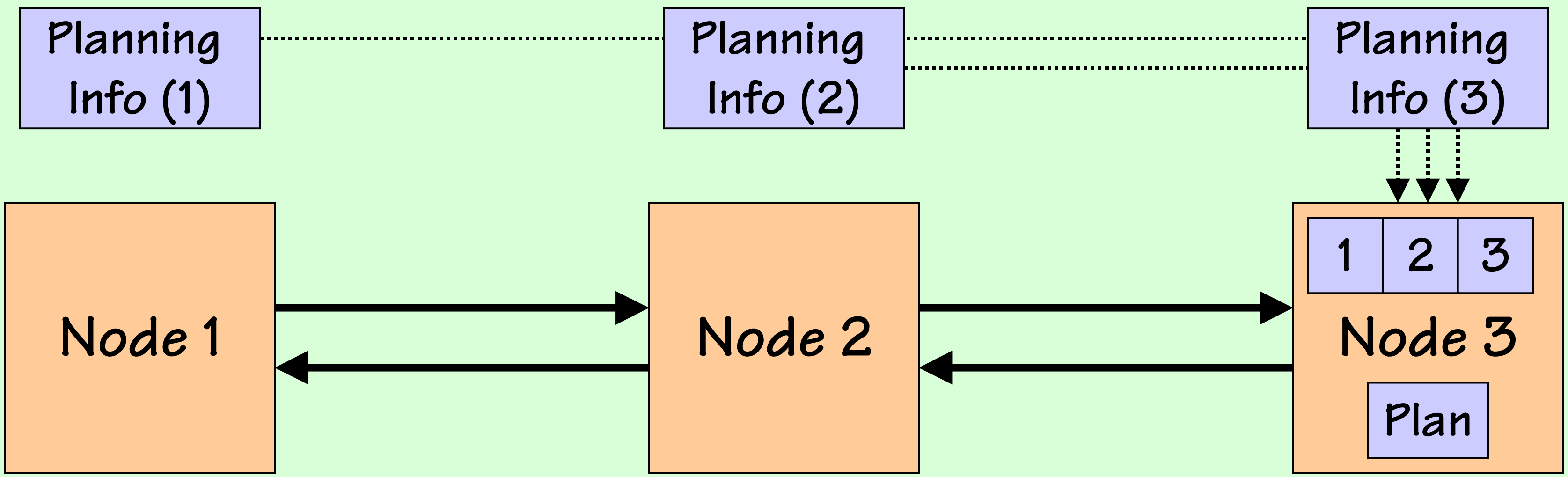
- Distributed planning
  - fast
  - non-optimal results



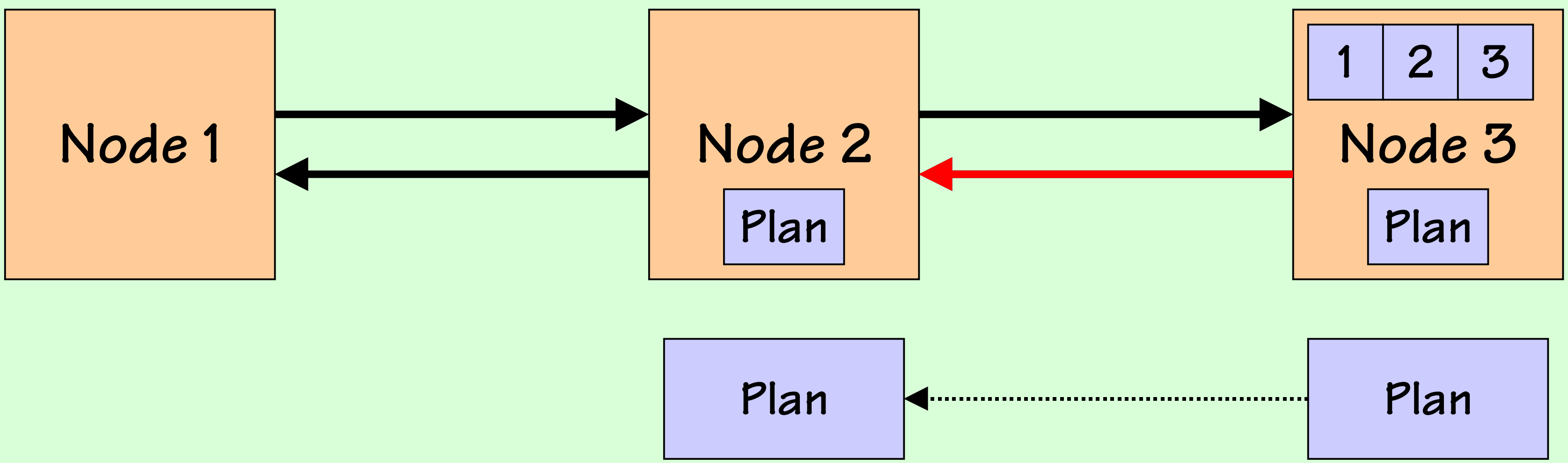
Automated Planning

- **Distributed planning**
- **Distributed planning with incremental refinement**
  - **How constraining is the initial plan?**
- **Centralized planning**
  - **Round trip for information gathering and plan distribution**

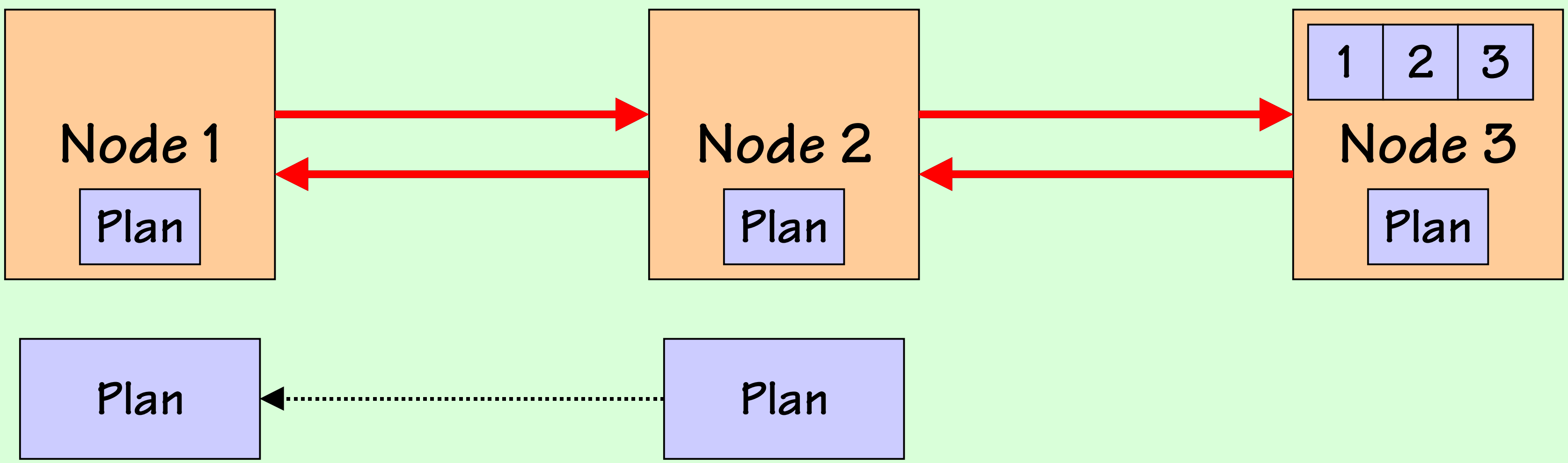
Automated Planning



Automated Planning



Automated Planning



Automated Planning

- Feasible plans may be hard to find
- Large search space
  - # of problems, # of adaptors, # of nodes
  - Adaptor ordering and composition
- Many constraints
  - Node resources and trust
  - Adaptor composition
- Limited time!

Challenges in Distributed Adaptation

- ✔ Reliable Transmission
- ✔ Automated Planning
- Secure Adaptation

Secure Adaptation

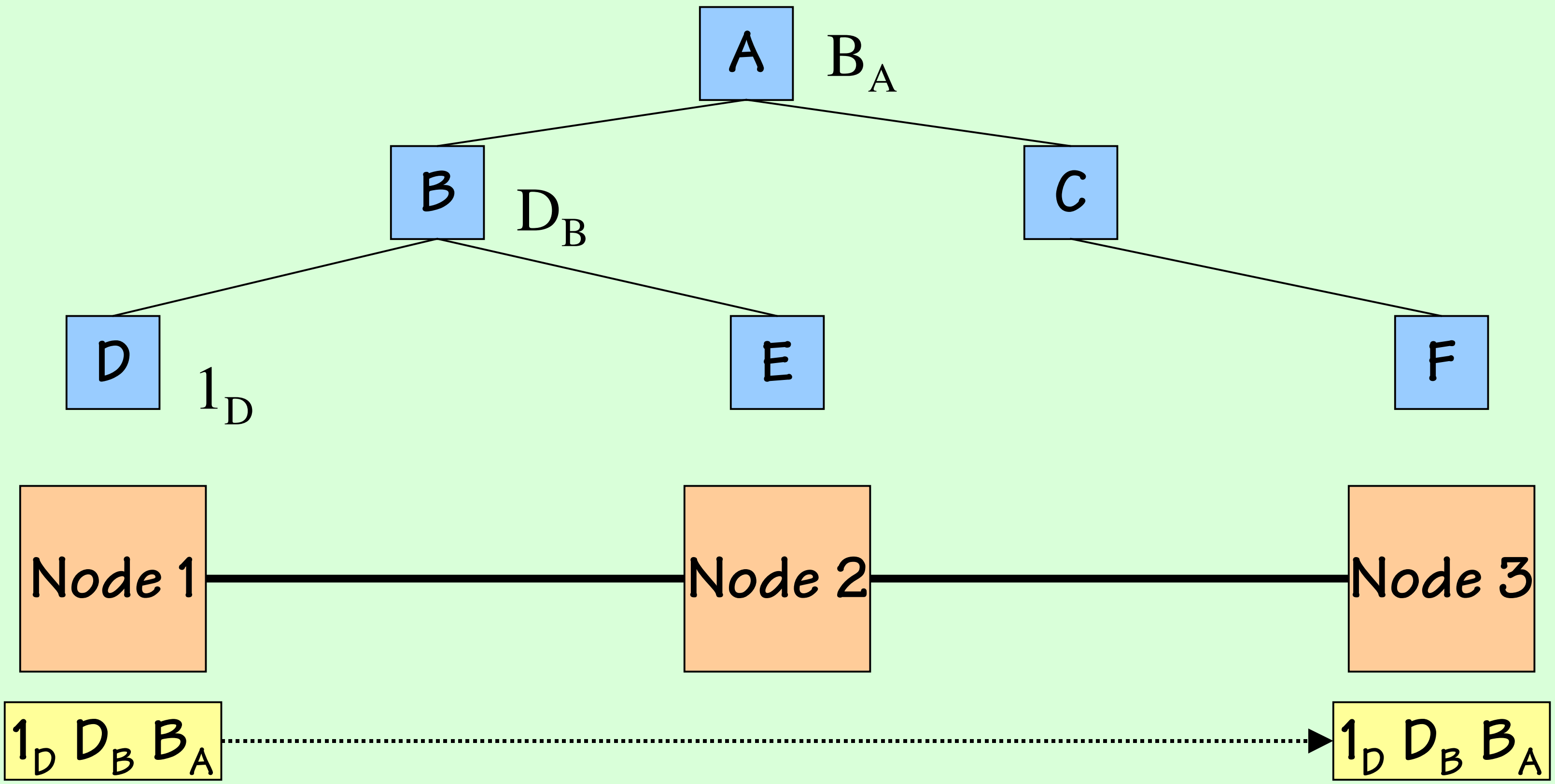
- **Protect the infrastructure**
  - The usual mobile code issues
  - Java is good enough
- **Protect the data**
  - Integrity and secrecy (when needed)
  - Allow adaptation, but only authorized adaptation

Secure Adaptation

- Mechanisms
  - Select trusted nodes
    - Implicitly trust endpoints
    - Endpoints select other trusted nodes
  - Protect planning
    - Digitally sign planning messages
  - Protect data
    - Distribute session keys to trusted nodes

Secure Adaptation

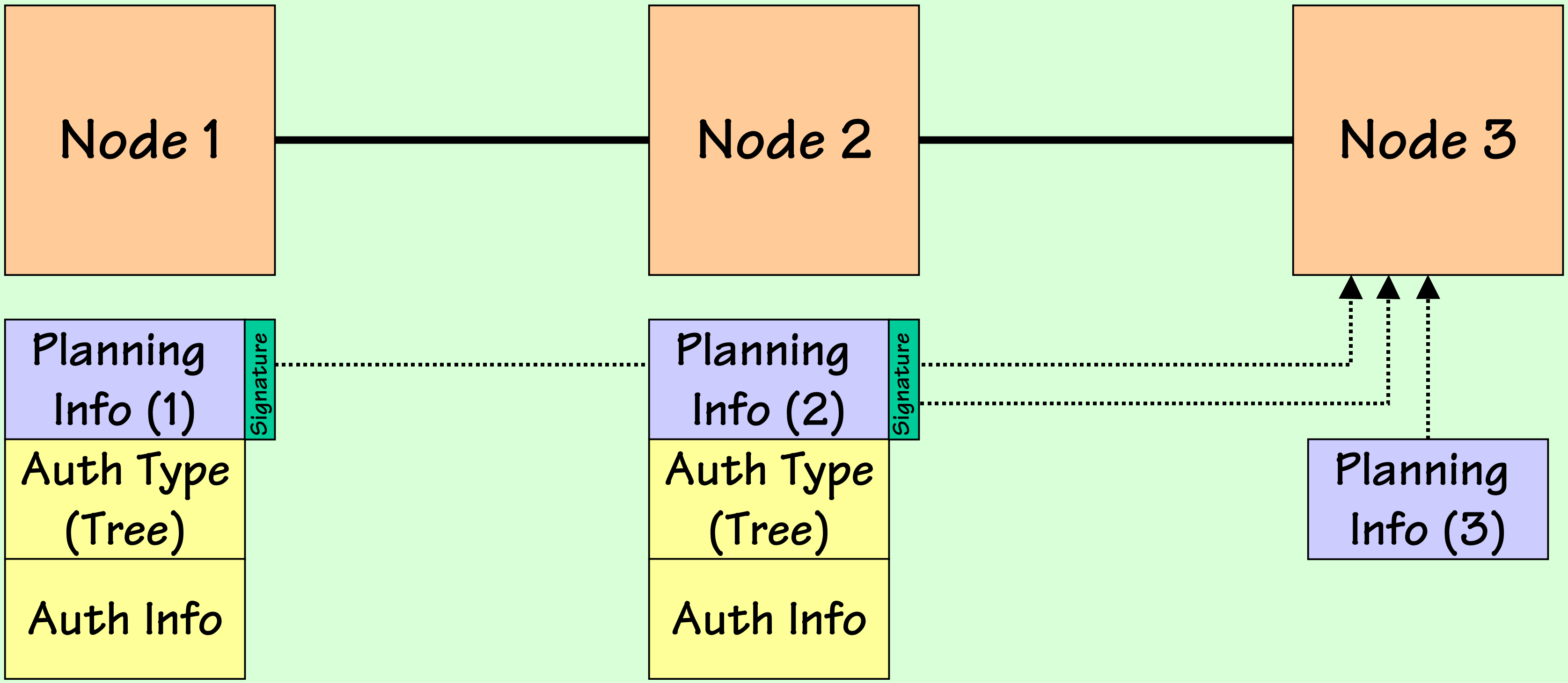
- Need a verified public key for each node



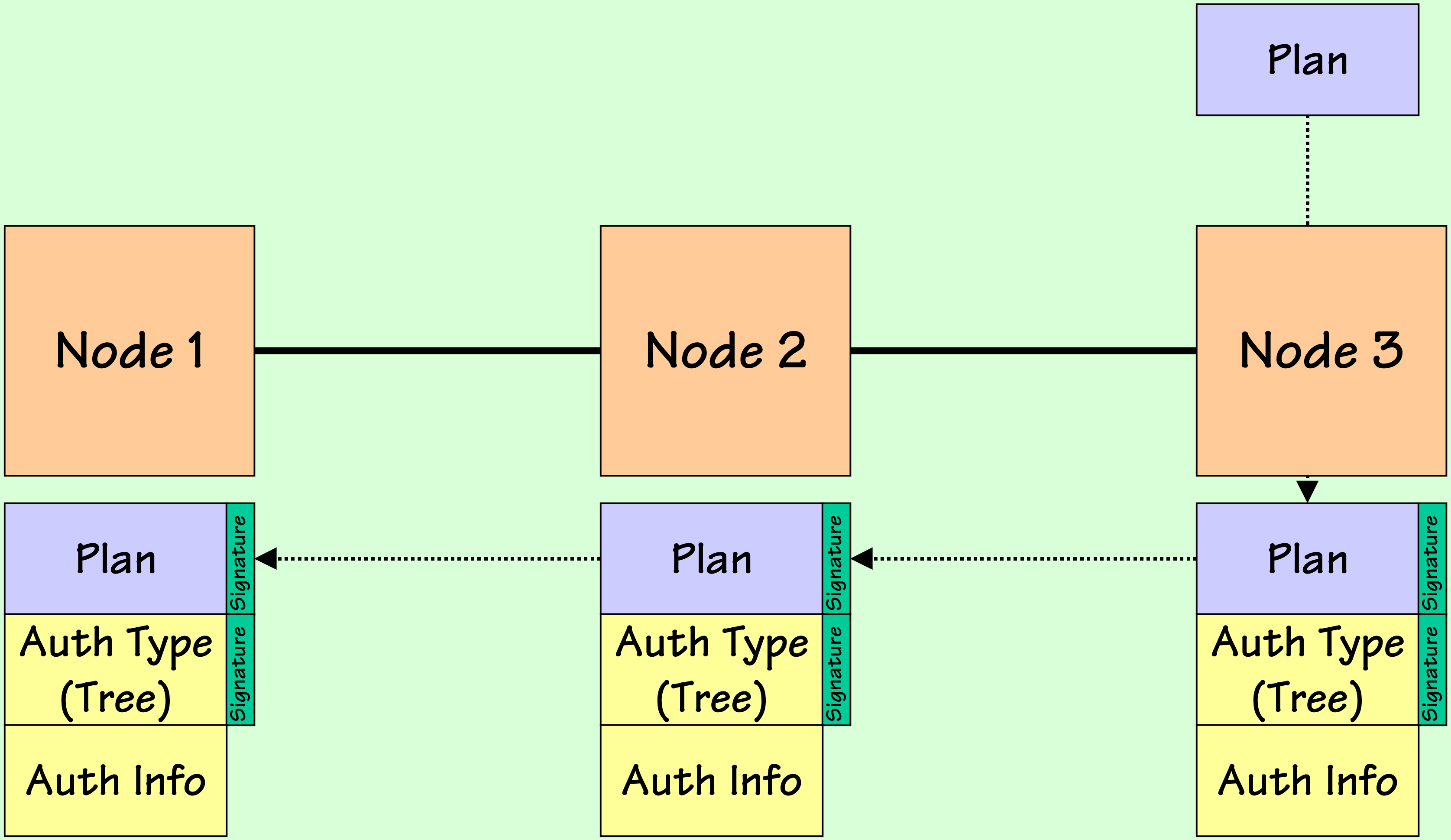
Secure Adaptation

- Complications
  - Connections span administrative domains
  - No ubiquitous public key architecture
  - Each connection may require a different level of trust
- Pluggable authentication mechanism
  - Requires secure agreement of mechanism

Secure Adaptation



Secure Adaptation



Secure Adaptation

- Authentication scheme proposed in plaintext and verified via signature
  - No node can change the authentication type without notice
- Public key encryption is used for session key distribution
- Additional mechanisms are needed to prevent replay

Challenges in Distributed Adaptation

- ✔ Reliable Transmission
- ✔ Automated Planning
- ✔ Secure Adaptation

Concluding Remarks

- Applications must be adaptive
- In heterogeneous networks applications benefit from distributed adaptation
- Key issues
  - Reliability, automatic planning, security
  - Automatic component composition
- Conductor, a prototype of proposed solutions

References

[Bal95] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP Performance Over Wireless Networks," *Proceedings of the 1st ACM International Conference on Mobile Computing and Networking (MobiCom '95)*, Nov. 1995.

[Fox98] Armando Fox, Steven D. Gribble, Yatin Chawathe, and Eric Brewer, "Adapting to Network and Client Variations Using Infrastructural Proxies: Lessons and Perspectives," *IEEE Personal Communications*, Setember 1998, 5(4):10-19.

[Jos95] A. Joseph, A. deLespinasse, J. Tauber, D. Gifford, and F. Kaashoek, "Rover: A Toolkit for Mobile Information Access," *Proceedings of the 15th ACM Symposium on Operating Systems Principles*, December 1995.

[Mal97] A. Mallet, J. Chung, and J. Smith, "Operating Systems Support for Protocol Boosters," *HIPPARCH Workshop*, June 1997.

[Nob97] B. Noble, M. Saytanarayanan, D. Narayanan, J. Tilton, J. Flinn, and K. Walker, "Agile Application-Aware Adaptation for Mobility," *Proceedings of the 16th ACM Symposium on Operating Systems Principles*, October 1997.

References

[Rei00] Peter Reiher, Richard Guy, Mark Yarvis, and Alexey Rudenko, "Automated Planning for Open Architectures," Short paper to be presented at *OPENARCH 2000*, Tel-Aviv, Isreal, March 2000.

[Sud98] P. Sudame and B. Badrinath, "Transformer Tunnels: A Framework for Providing Route-Specific Adaptations," *Proceedings of the Usenix Technical Conference*, June 1998.

[Yar99a] Mark Yarvis, Peter Reiher, and Gerald J. Popek, "Conductor: A Framework for Distributed Adaptation," *Proceedings of the 7th Workshop on Hot Topics in Operating Systems (HotOS VII)*, Rio Rico, AZ, March 1999.

[Yar99b] Mark Yarvis, An-I A. Wang, Alexey Rudenko, Peter Reiher, Gerald J. Popek, "Conductor: Distributed Adaptation for Complex Networks," UCLA Tech Report CSD-TR-990042, August 1999.

[Yar00] Mark Yarvis, Peter Reiher, and Gerald J. Popek, "A Reliability Model for Distributed Adaptation," *Proceedings of the 3rd IEEE Conference on Openarchitectures and Network Programming (to appear)*, Tel-Aviv, Isreal, March 2000.