



Protecting Public Servers from DDoS Attacks Using Drifting Overlays

Venkata Pingali & Joe Touch
Agile Tunneling Project
USC/ISI

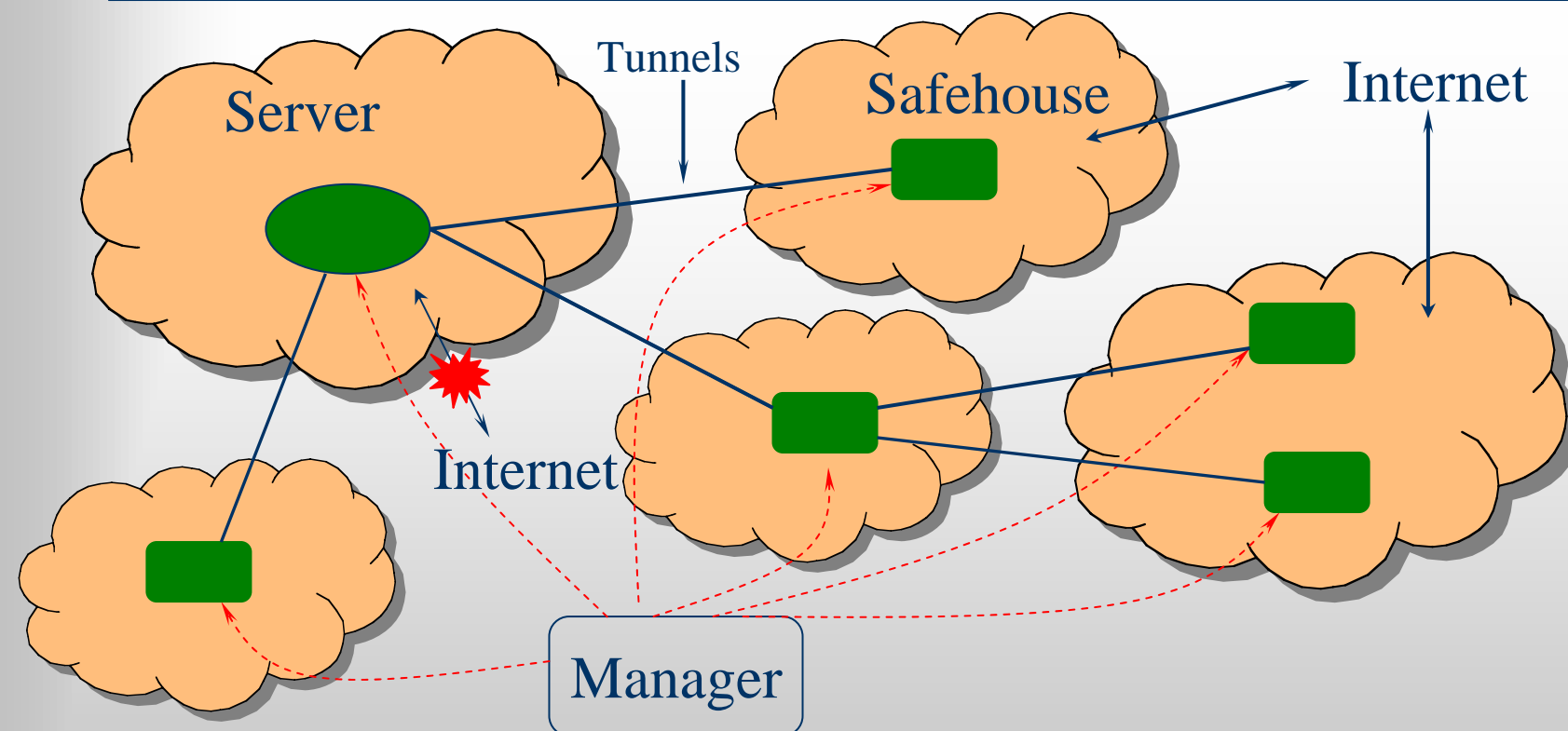




Overview

- Problem: Critical public servers (e.g., SAP)
 - Server replication is hard
 - Server downtime is expensive
- Approach: Use network-level overlay to control paths to the server
 - Control: reachability and predictability
 - Customizability: rapid, dynamic deployments

Drifting Overlays



- Safehouses terminate connections
- Some safehouses are clients



Related Work

	Server Reloc.	Client Coop.	Overlay Layer	Overlay Routing	Addressing	Topology Control	Connection Validation
Drifting Overlays	No	Yes	Network	RIP/OSPF	Unicast	Custom	No
MOVE	Yes	Yes	Application	DHT	Unicast	DHT	No
SOS	No	No	Application	DHT	Unicast	DHT	No
WebSOS	No	Yes	Application	DHT	Unicast	DHT	Yes
Mayday	No	No	Application	Any	Unicast	Unspec.	Maybe
Firebreak	No	No	Network	N/A	Anycast	Static	No
FONet	No	No	Network	BGP	Unicast	Static	No
Roaming Servers	Yes	Yes	N/A	N/A	Unicast	None	No
IP Hopping	No	Yes	N/A	N/A	Unicast	None	No
OverDoSe	No	Yes	Any	Any	Unicast	N/A	Yes



Overlay Advantages

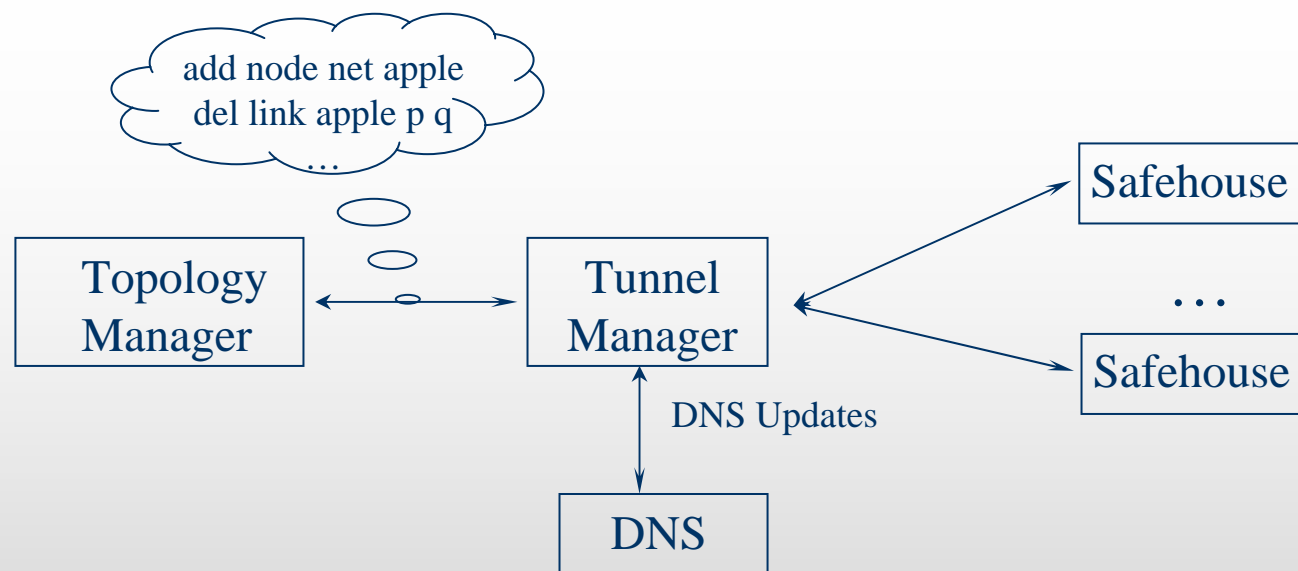
- Adds a layer of indirection
 - Alter reachability assumption of the server
 - Allows non-uniform treatment
- Improves configurability
 - Customizable to need and reusability
 - Low complexity: Tunneling+Routing+DNS
 - Allows quick deployment
- Alters nature of problem
 - Predictability and reachability of servers
 - Turns resource attack into information attack



Challenges

- Applicable only when:
 - Network is multihomed
 - Server replication is hard
 - Adaption timeframes of the order of DNS TTL is acceptable
 - Alternative hosts available for use
- Potentially shifts the attacker targets
 - Examples: Routers on path, DNS
 - Possibly to more light weight systems

Architecture



- **Issues**
 - Topology management
 - Abstraction provided to topology manager



Topology Mgmt Issues

- Nodes
 - Discovery: host registry
 - Selection: Predictability, connectivity
- Address space and routing
- Topology
 - Single root vs. multiple roots
 - Single or multiple overlays per server
 - Incremental modification vs destroy/create
- Drift strategy
 - Topology change ordering and timing
 - Topology primitives



Status

- Implementation
 - Perl + FreeBSD
 - IP-IP tunnels+ static routing
 - Kernel optimization to handle large number of tunnels
- Architecture
 - Unified topology and tunnel manager
 - Centralized implementation
- Topology
 - Simple strategies using randomization
 - Tree