# Diameter Overload Control

Requirements

# Overload, that happens?

Yes!

# What happened?

- Lost messages
  - High speed signaling
- Can you hear me now?  Now?  Now? Now? ….
- Avalanche

# Whose fault is it anyways?

- Inadequate Capacity
- Dependency Failures
- Component Failures
- Network Initiated Traffic Flood
- Subscriber Initiated Traffic Flood
- Mars Attacks!
  - Or DoS attacks

# Is this a Diameter problem?

- Network congestion vs. overload
- Diameter servers and other protocols
- Diameter signaling in wireless networks

# Does Diameter handle this now?

- DIAMETER_TOO_BUSY
  - Duration open to interpretation
  - Applies only to server scope
- Message dropping
  - Retries can exacerbate issue
- No mechanism to avoid or respond to overload

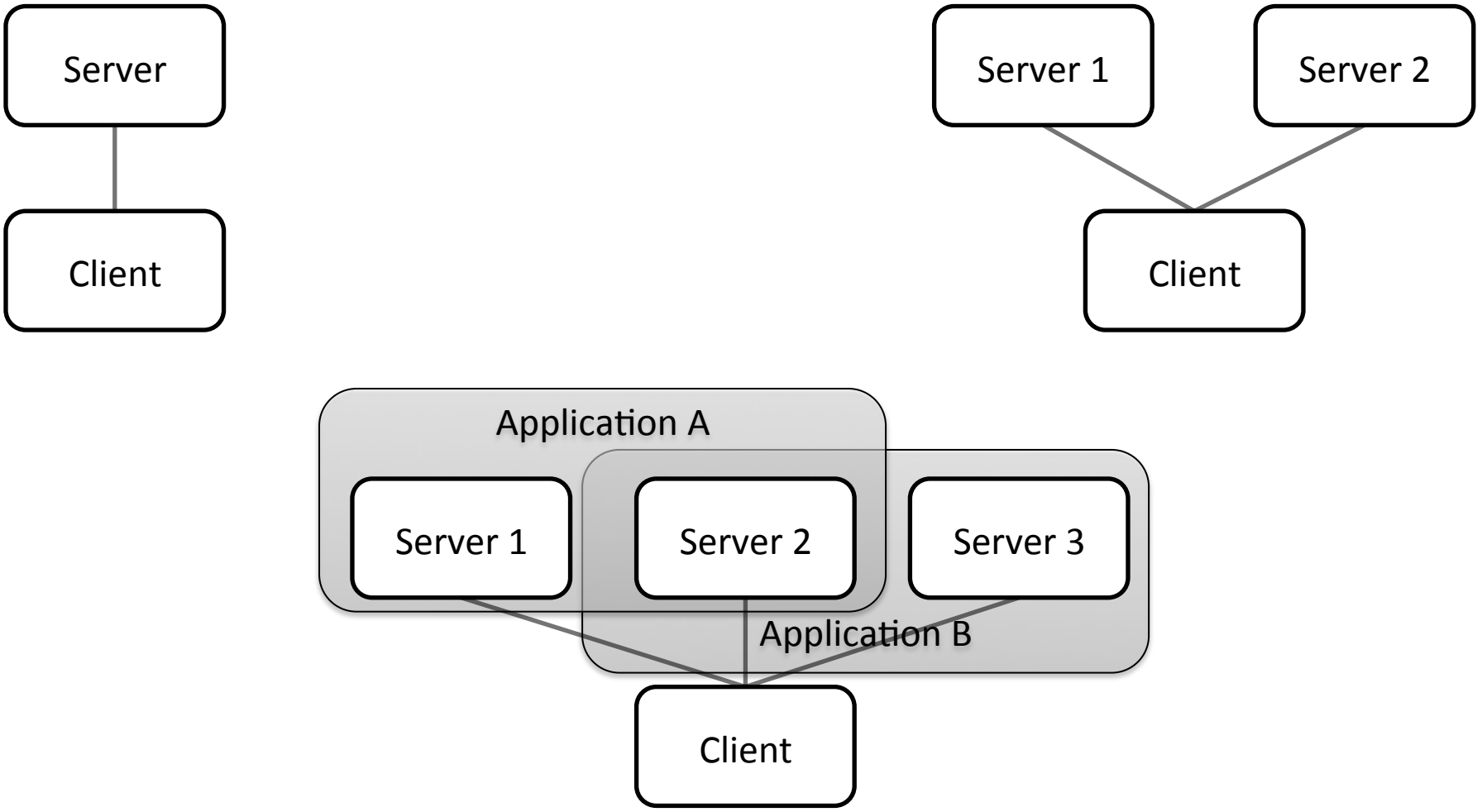
# But is only affects a couple of Diameter applications, right?

- Overload can happen to any application
- So…
- Extend the base protocol

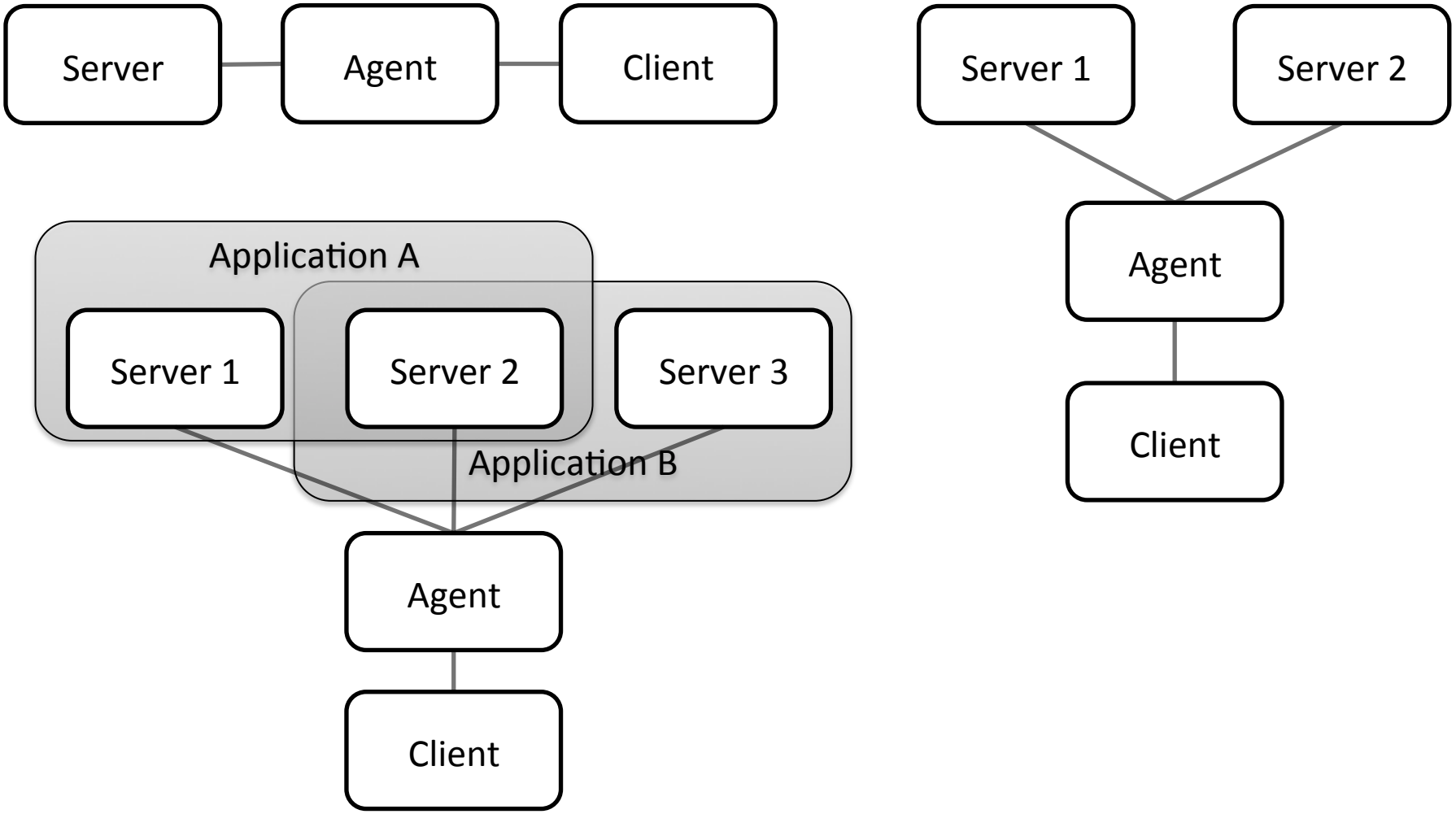
# Is overload all or nothing?

- A Diameter node can be overloaded for one purpose but not another!
  - They may serve multiple applications with different capacities
  - … or different realms
  - … or different back-end dependencies such as databases or other resources

# Basic Scenarios



# Basic Scenarios with an Agent



# Solution Requirements Approach

- Informed by
  - SIP overload control
  - 3GPP study on core network overload
  - Other mechanisms and congestion control principles

# Solution Requirement Highlights …

- Allow nodes to communicate overload information
- Help prevent overload conditions in the first place
- Improve behavior when overloaded
  - Recovery after overload ends
  - Mitigate cascades
  - Doesn't make things worse
- Don't block use of available capacity
  - Support all the listed scenarios
    - With or without agents

# … More Requirements Highlights …

- Work with any existing or new application
  - Including bidirectional applications
- Scalable to both large and small deployments
- Minimal new configuration burden
- Incremental Deployments
  - Still works if not all nodes support it.
- Works across domains and realms

# … Still More Highlights

- Don’t add new vulnerabilities
- Allow administrative control over who can see overload info
- Extensible

# Next Steps

- Does the draft sufficiently describe the problem
  - Is it real?
  - Is it worth working on?
- Does this work belong in DIME?
  - Seems to fit the ongoing maintenance aspect of the DIME charter