

Overload Control Agent Use Cases

(draft-donovan-doic-agent-cases-00)

Steve Donovan
Ben Campbell
IETF90, Toronto

Agent Related Open Issues in DOIC

- 23 - DOIC behavior for realm overload
- 25 - Section 3.1.5 Diameter Agent Behavior
- 26 - Overload Control Endpoints under specified
- 27 - Behavior of agent acting on behalf of Client that does not support DOIC
- 49 - capabilities announcement mechanism needs to be rethought
- 57 - Handling of "Realm-Routed" Overload report type
- 60 - Agent Overload Report Handling considerations
- 61 - Agent Capability Announcement Considerations

New Terms

- Transaction Client (TC) – Node that originates a request.
- Transaction Server (TS) – Node that acts on a request and sends an answer
 - Different from “Client” and “Server” as defined in RFC 6733.
 - Defined in the context of a given transaction
 - When a server sends a request to a client, the server becomes the TC and the client the TS.
- “Client” and “Server” in these diagrams refer to TC and TS.

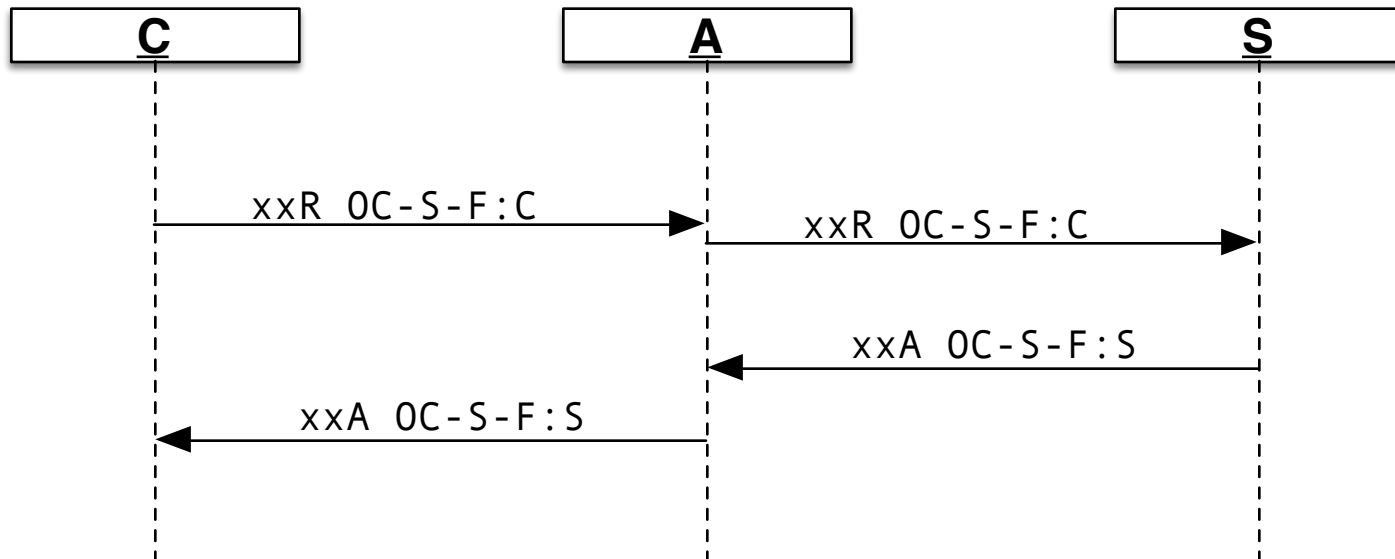
Abatement Techniques

- Throttling – rejecting some percentage of requests
- Diversion – forwarding of some fraction of requests to other servers with available capacity.
- Agents can delegate abatement downstream, or perform it locally
 - Choice varies for different abatement techniques, report-types, and abatement algorithms.

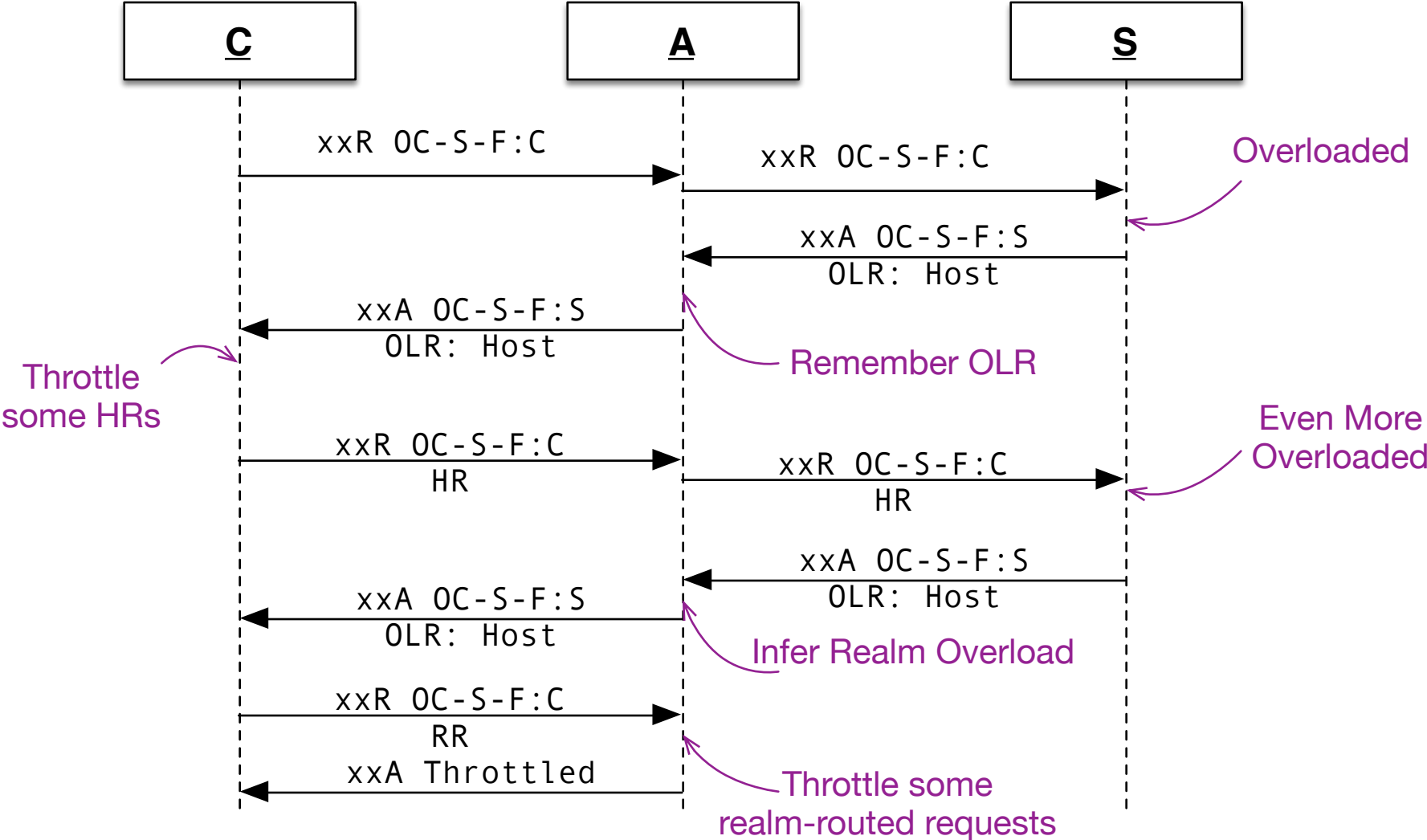
Architectural Principles

- Some basic principles
 - Throttling should be done at the TC, or as close as possible.
 - For varying definitions of “as close as possible”
 - TC has the greatest knowledge of how throttling decisions will affect the client application
 - The closer to the TC a request is throttled, the less it uses network resources
 - Diversion should be done at the last hop before the TS, or as close as possible
 - For varying definitions ...
 - Typically only the last hop can control TS selection.

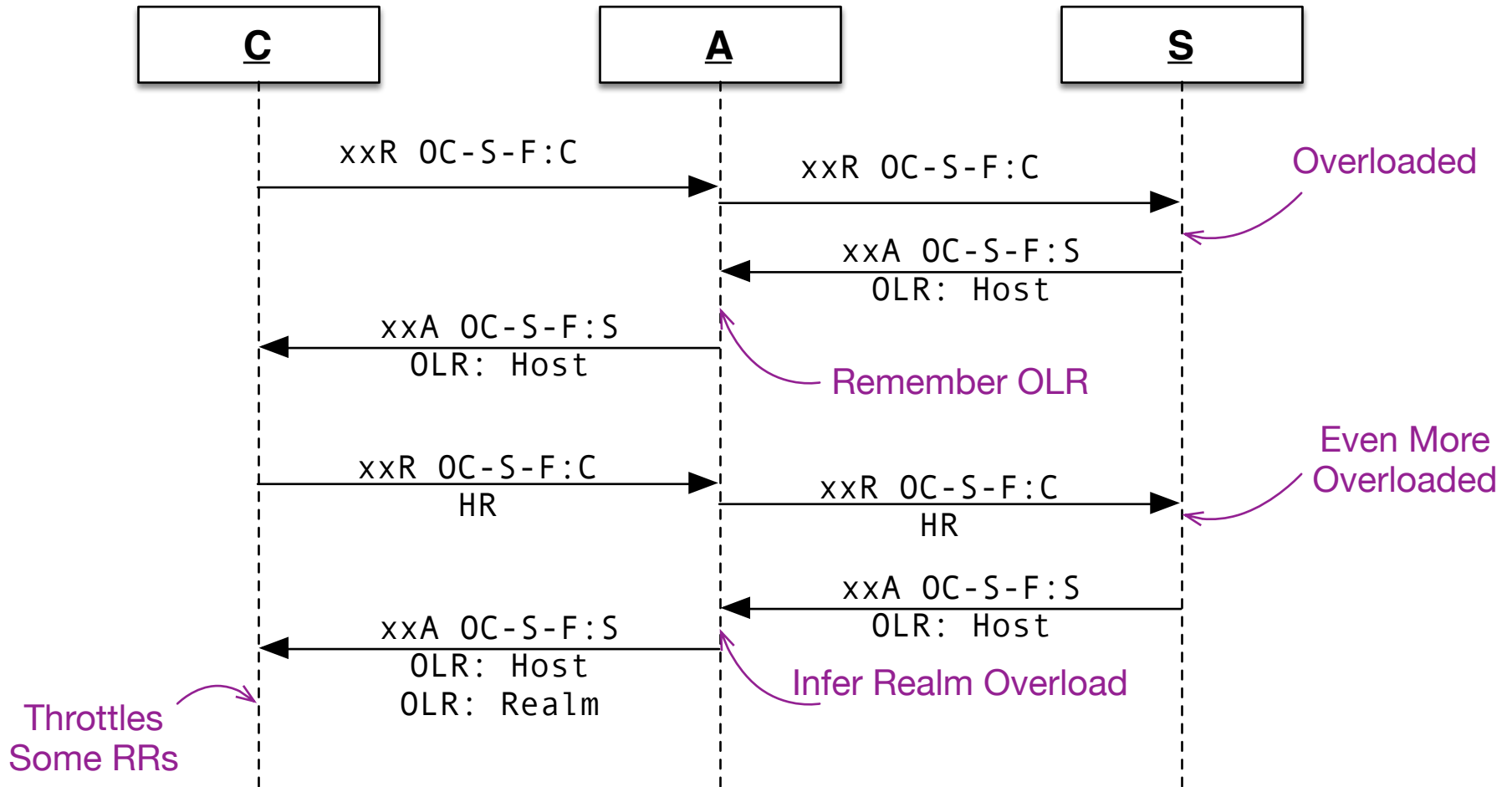
Simple Agent: Capabilities Announcement



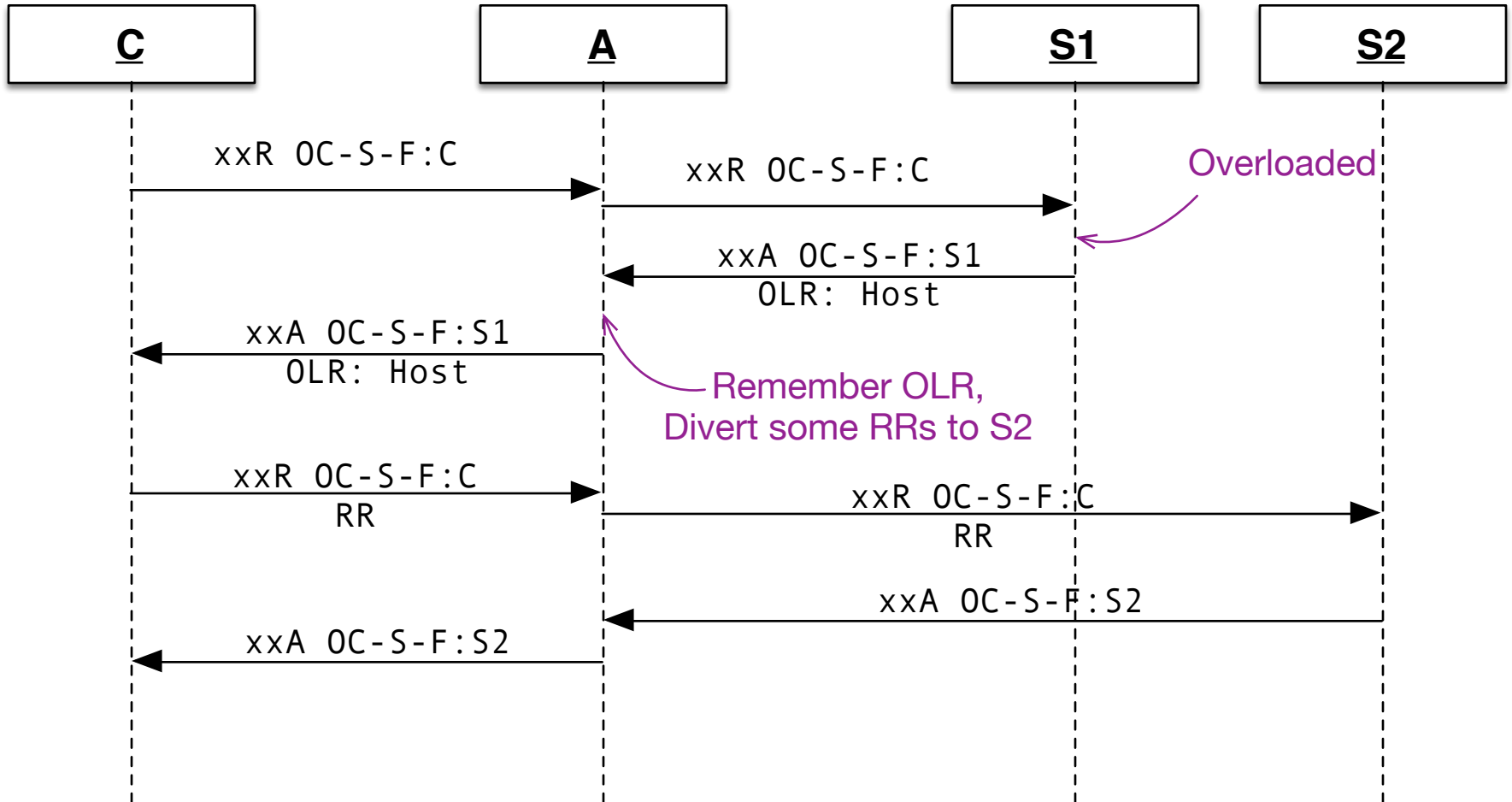
Simple Agent: Report Handling



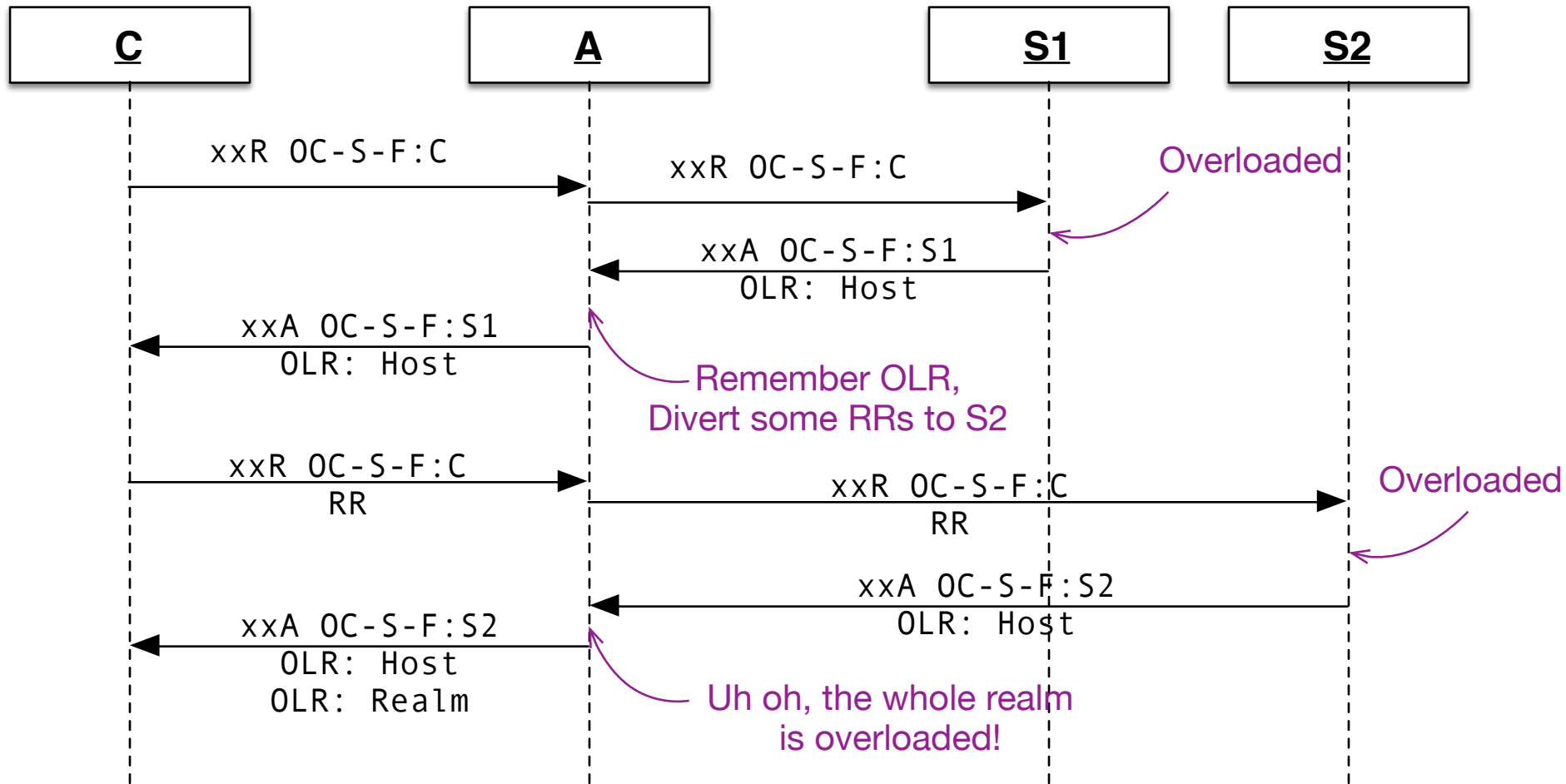
Simple Agent: Realm Report



Multiple Servers: Diversion



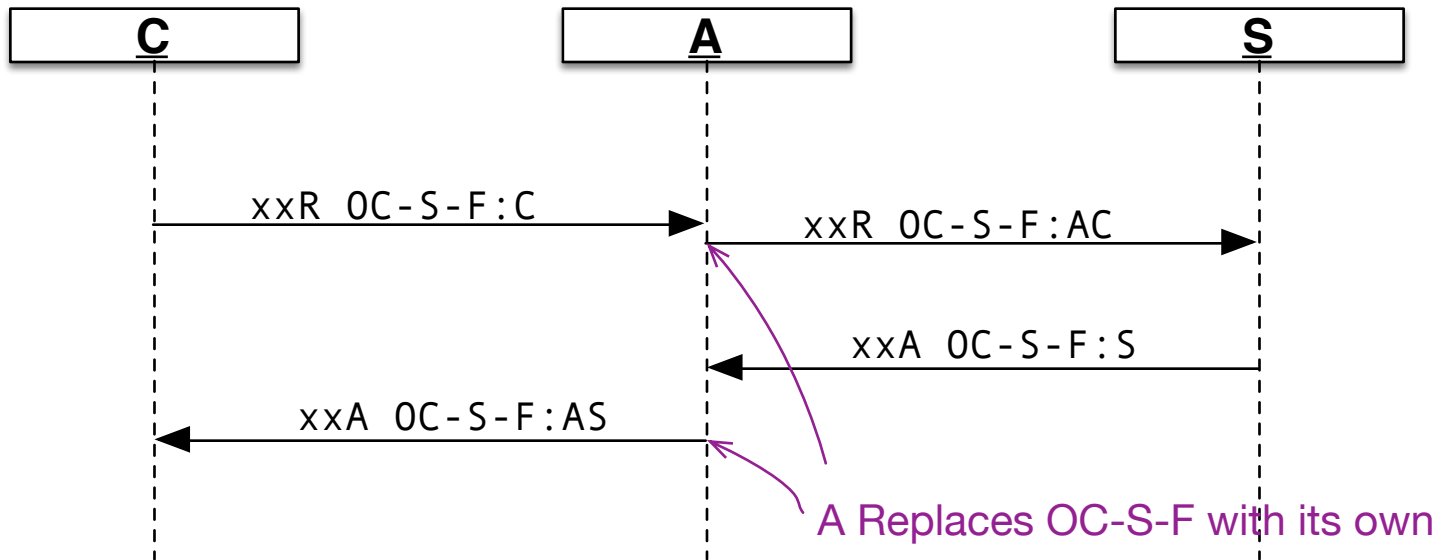
Multiple Servers: Realm Reports



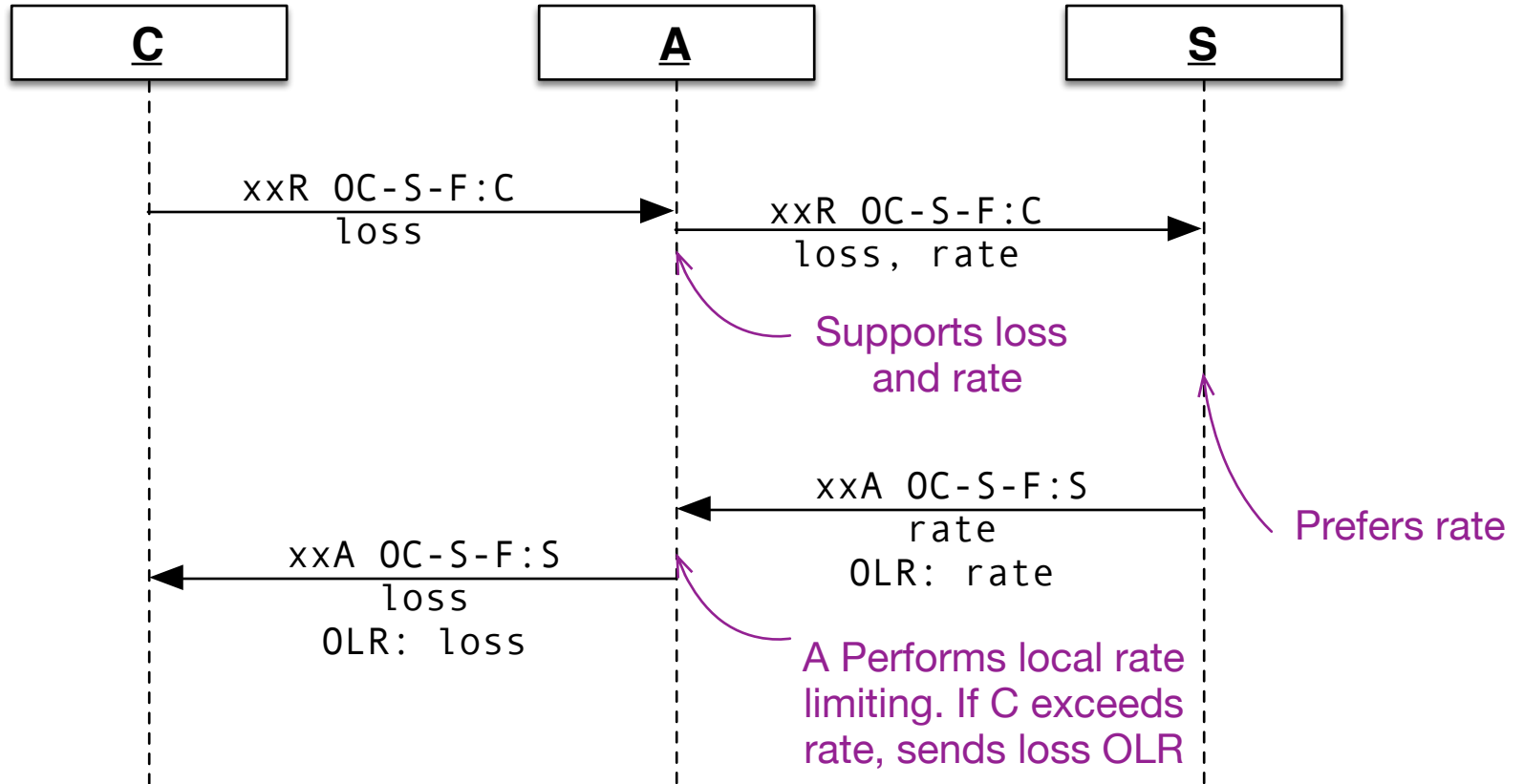
Simple Agent: DOIC impacts

- Single OLR can result in multiple abating nodes
 - e.g., client abates HR requests, agent abates RR requests.
- Multiple OLRs may occur in same answer
 - Supported in current spec
 - May need constraints (e.g. must be different report types?)
- Agents need to be able to insert Realm-type OLRs into answers.
- Need an error code to say a request was throttled by an agent (or server.)

Mixed Capabilities Announcement



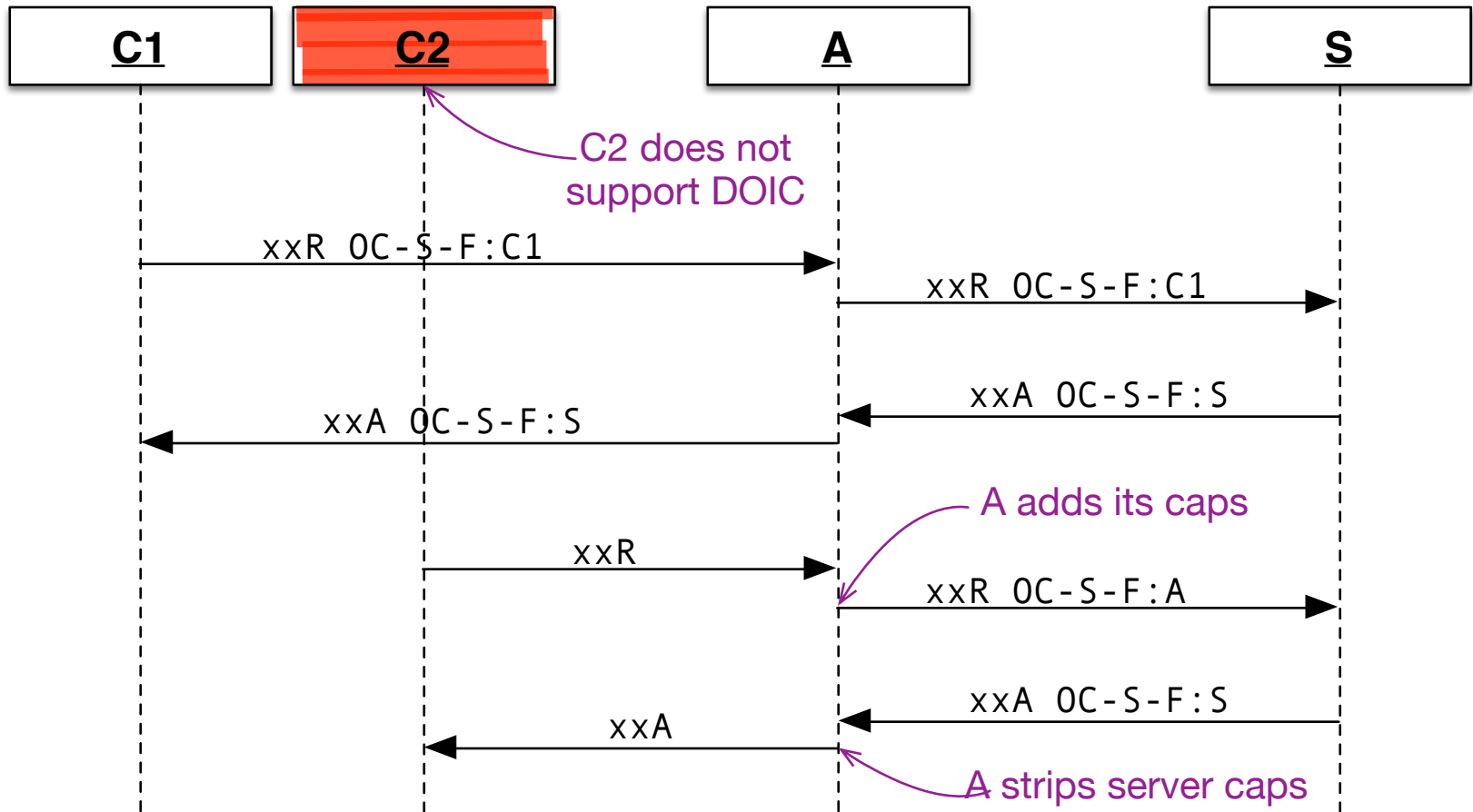
Mixed Algorithms



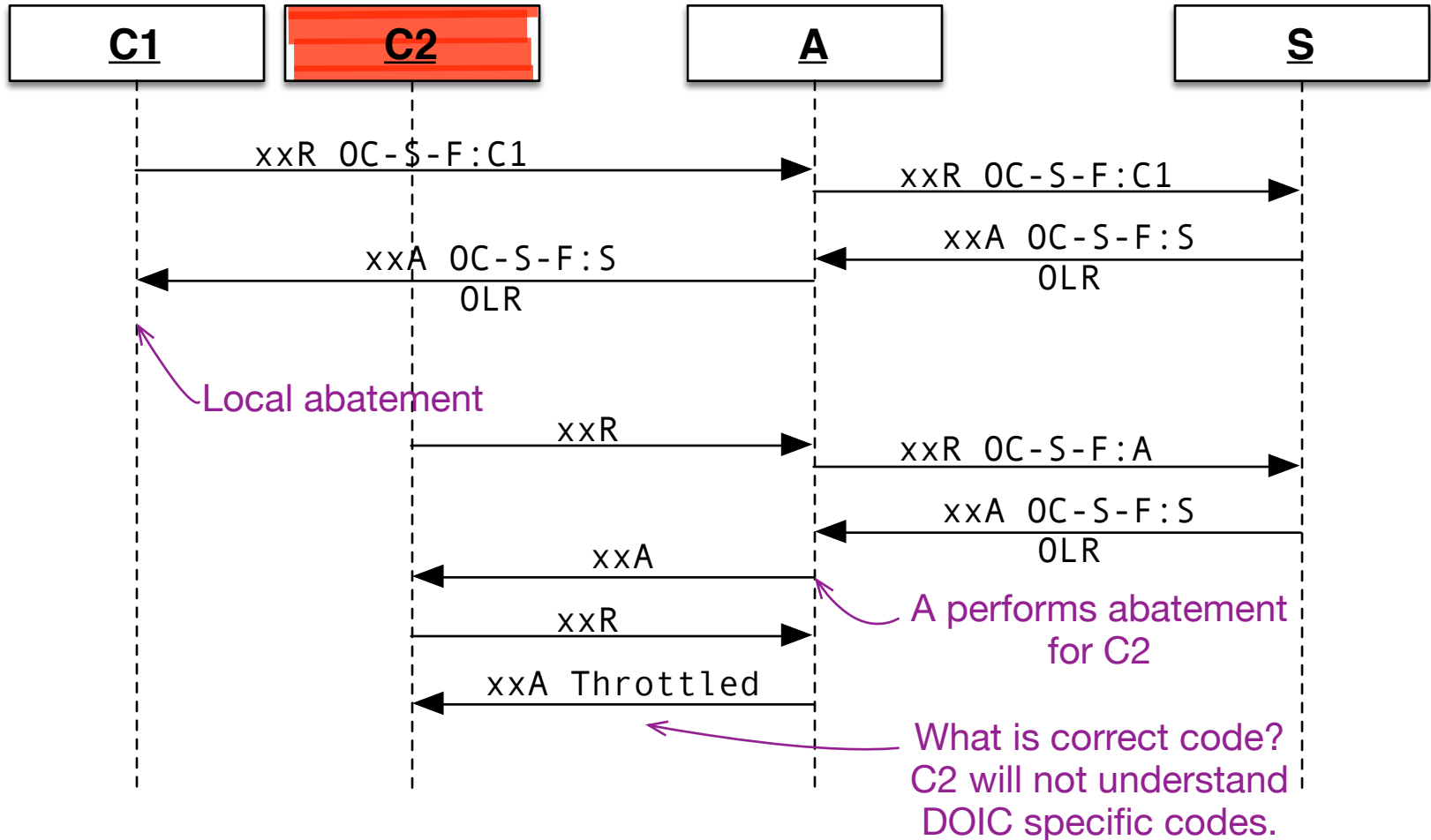
Mixed Capabilities: DOIC Impacts

- Agents need to be able to replace OC-S-F in requests
- Agents need to be able to replace OC-S-F in answers
- Agents need to be able to remove or replace OLRs
- Agents need to present a coherent view of DOIC to each “side”.

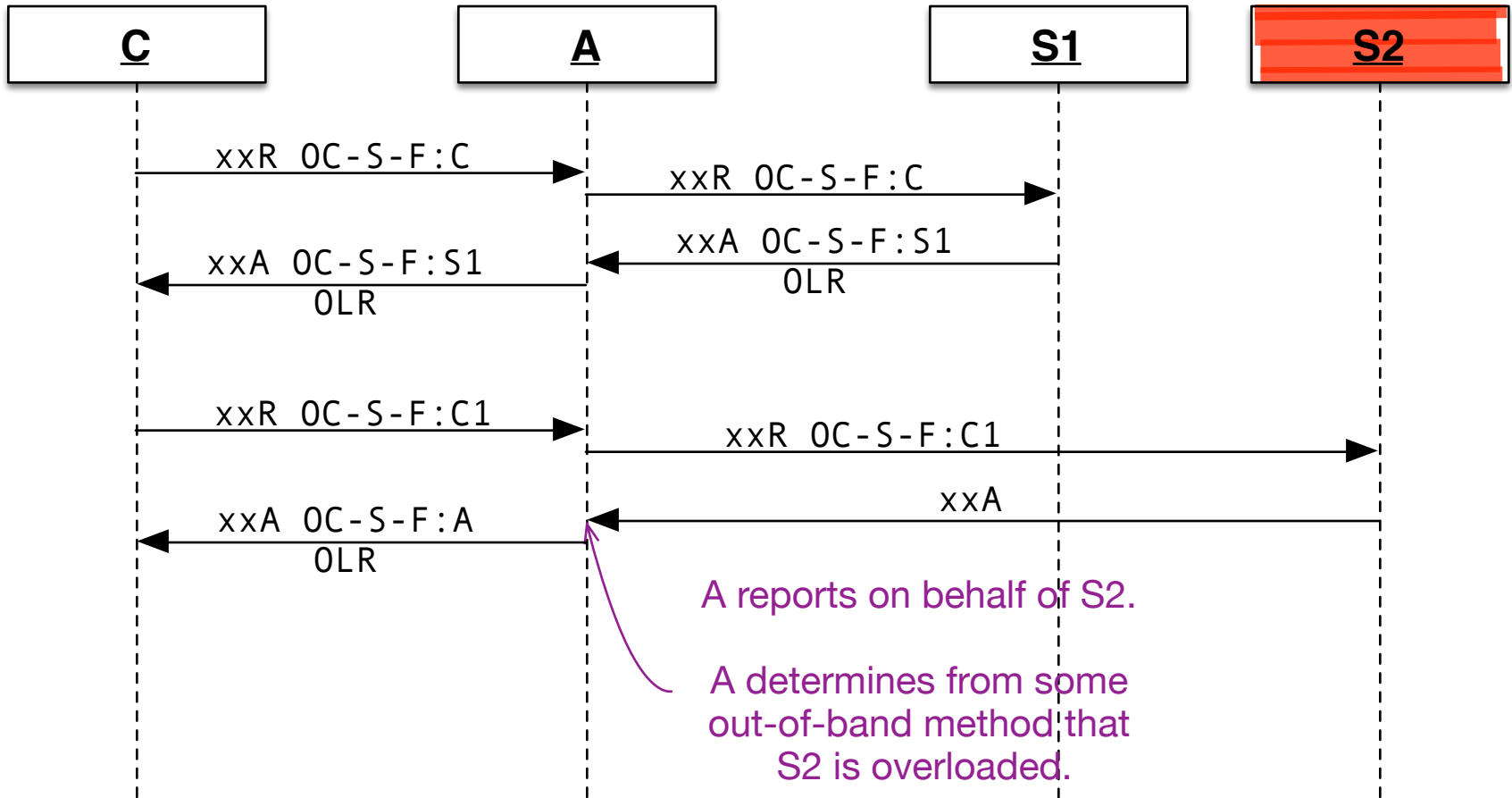
Non-Supporting Client: Capabilities



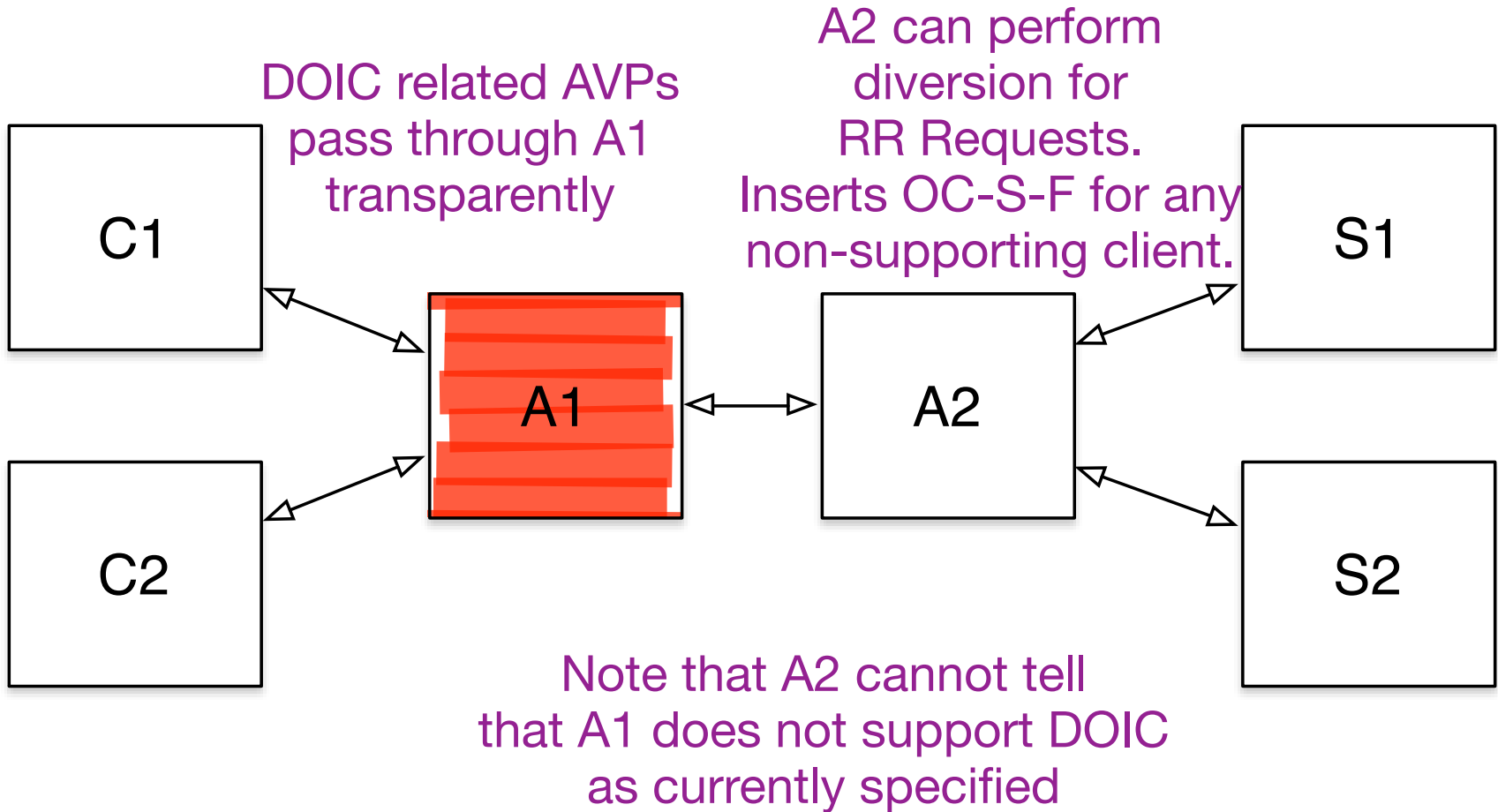
Non Supporting Client: OLRs



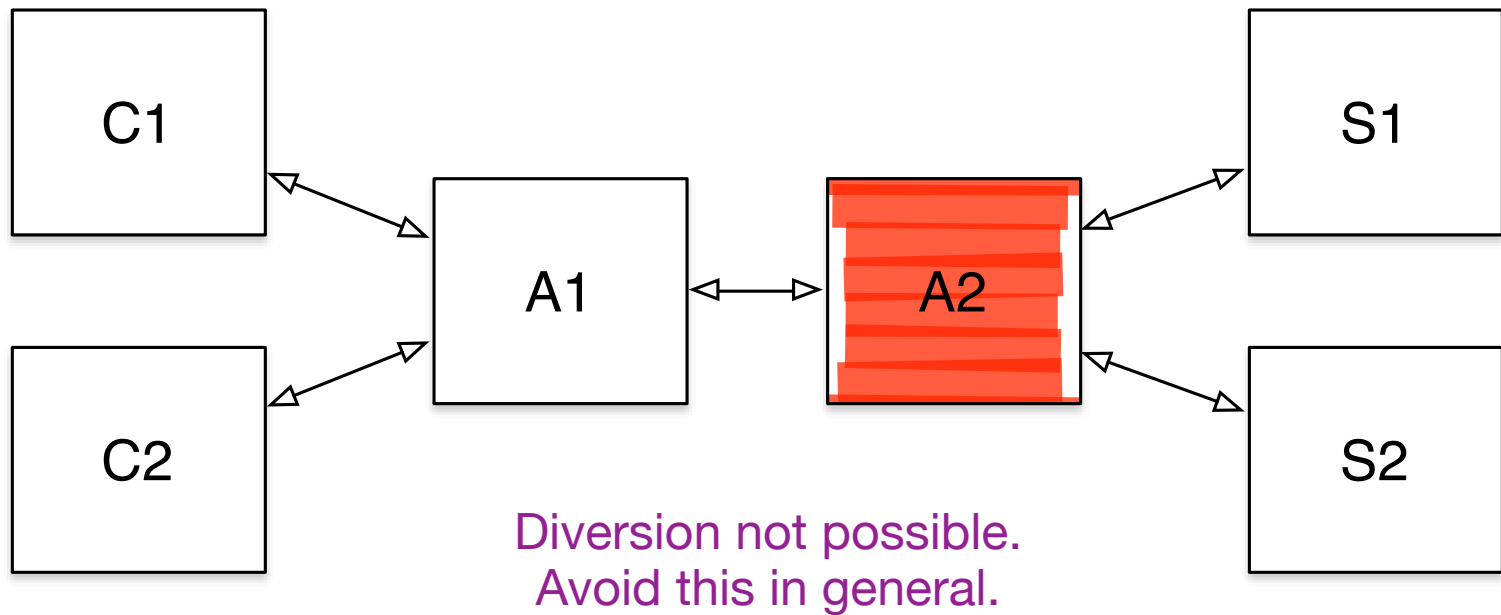
Non Supporting Server



Non-Supporting Agent 1



Non-Supporting Agent 2

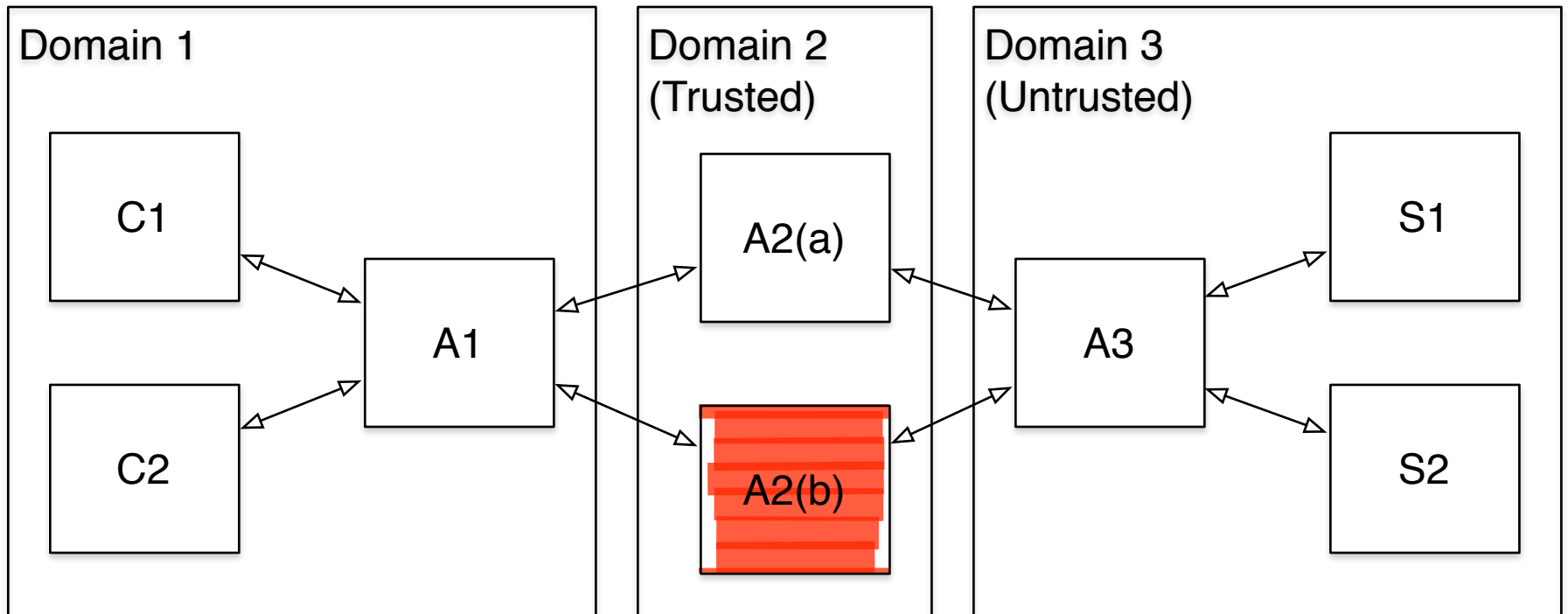


Non-Supporting Node: DOIC

Impacts

- Agents must be able to insert OC-S-F AVPs into both requests and answers.
- Agents must be able to remove OC-S-F AVPs from both requests and answers.
- Agents must be able to insert host reports.
- Agents must be able to remove OLRs
- Agents must be able to abate host-routed requests.

Inter Domain Trust



A2(a) will police DOIC AVPs from Domain 3. A2(b) will pass them transparently. Either way looks identical to Domain 1.

DOIC Impacts

- A DOIC node must be able to tell if a peer supports DOIC
 - Must be able to distinguish between a DOIC avp from a DOIC supporting peer, vs one passed transparently by a non-supporting peer.
- We may need to be able to attribute a DOIC AVP to the node that originated it.
 - This needs further study. If this is true, then there may be no such thing as transparent pass-through by a DOIC agent.

Agent Related Recommendations

- Mostly suggestions for non-normative guidance
 - ... but some may require normative changes

General

- Need a “Diameter Throttled” error code
 - Request should not be retried
 - What about non-supporting clients?
- Multiple OLRs allowed in a single message
 - Currently allowed, as long as report types do not conflict

Capabilities Exchange

- Agents can be reporting-nodes, reacting-nodes, or both.
- When acting as a Reacting-Node, the agent must insert OC-S-F into requests
 - May copy from inbound request or replace with its own.

Capabilities Exchange (cont)

- When acting as a Reporting-Node, the agent must insert OC-S-F into answers
 - If the agent changed the OC-S-F in a request, it needs to reverse the modification in the answer.
- Peers must be able to distinguish an DOIC supporting agent from a non-supporting agent
 - Even if a DOIC agent otherwise passes the AVPs through unchanged.

Overload Abatement

- A DOIC relay that includes (or passes through) an OC-S-F AVP must ensure any resulting OLRs are honored
 - Can be through local abatement or delegation
- Agent should use all available information to determine overload conditions
 - Not just explicit OLRs. Examples:
 - Errors, time outs, and slow response times
 - Infer realm overload from host-reports.
 - Out-of-band methods

Overload Abatement (cont)

- When possible, diversion is better than throttling.
 - But must be done with care!
 - Typically needs to be done by node that does final server selection
 - HR requests typically cannot be diverted
 - Emphasis on “typically”

Overload Abatement (cont)

- Throttling is best done as close to source as possible
 - But there may be good reasons a given deployment needs agent throttling.
- If an agent throttles a request, it must send an answer with an appropriate error code.
 - Bad things happen if you don't.
- Agent must be careful to avoid redundant abatement situations
 - This may be algorithm-specific