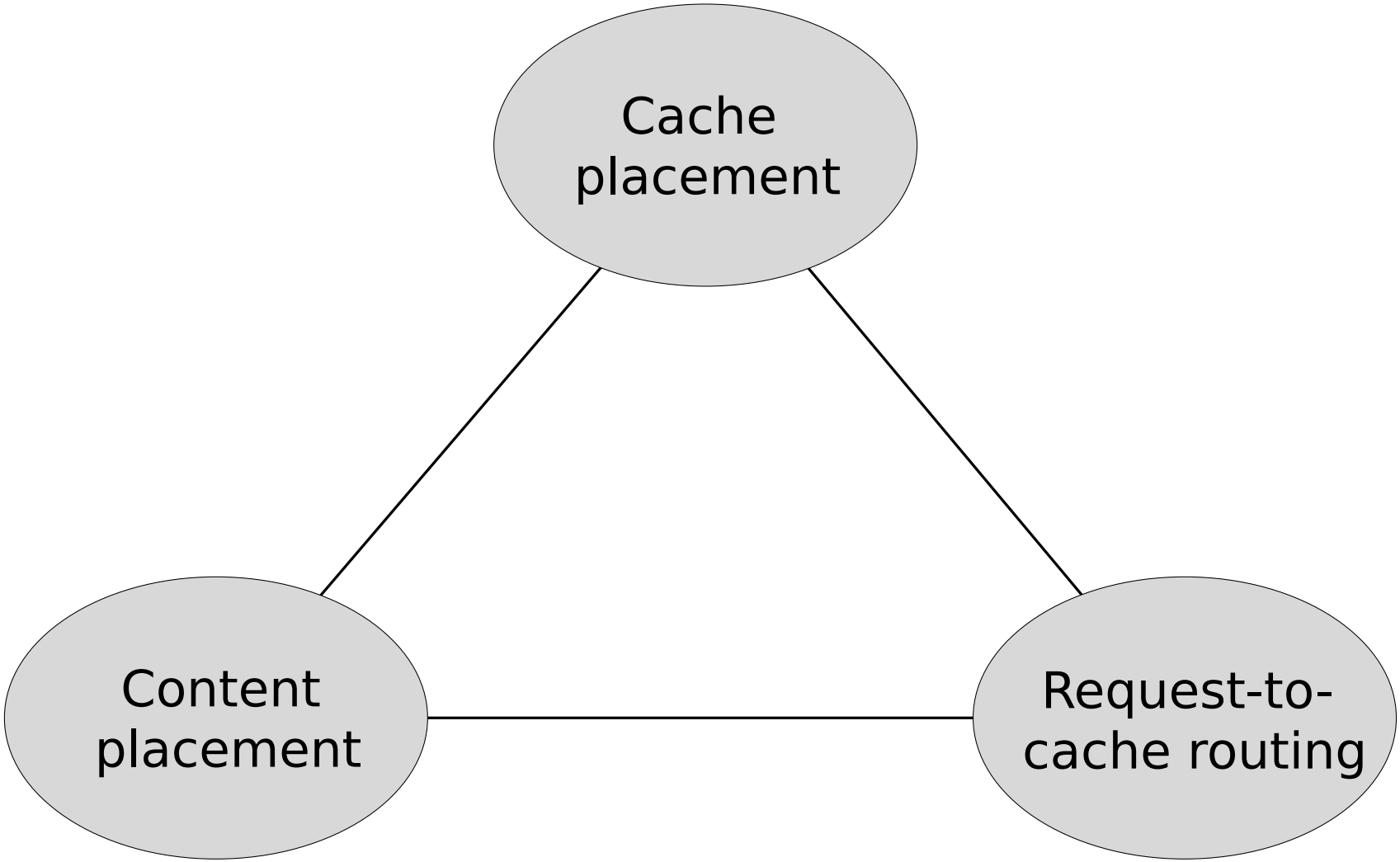
Hash-routing Schemes for Information Centric Networking

Lorenzo Saino, Ioannis Psaras, George Pavlou

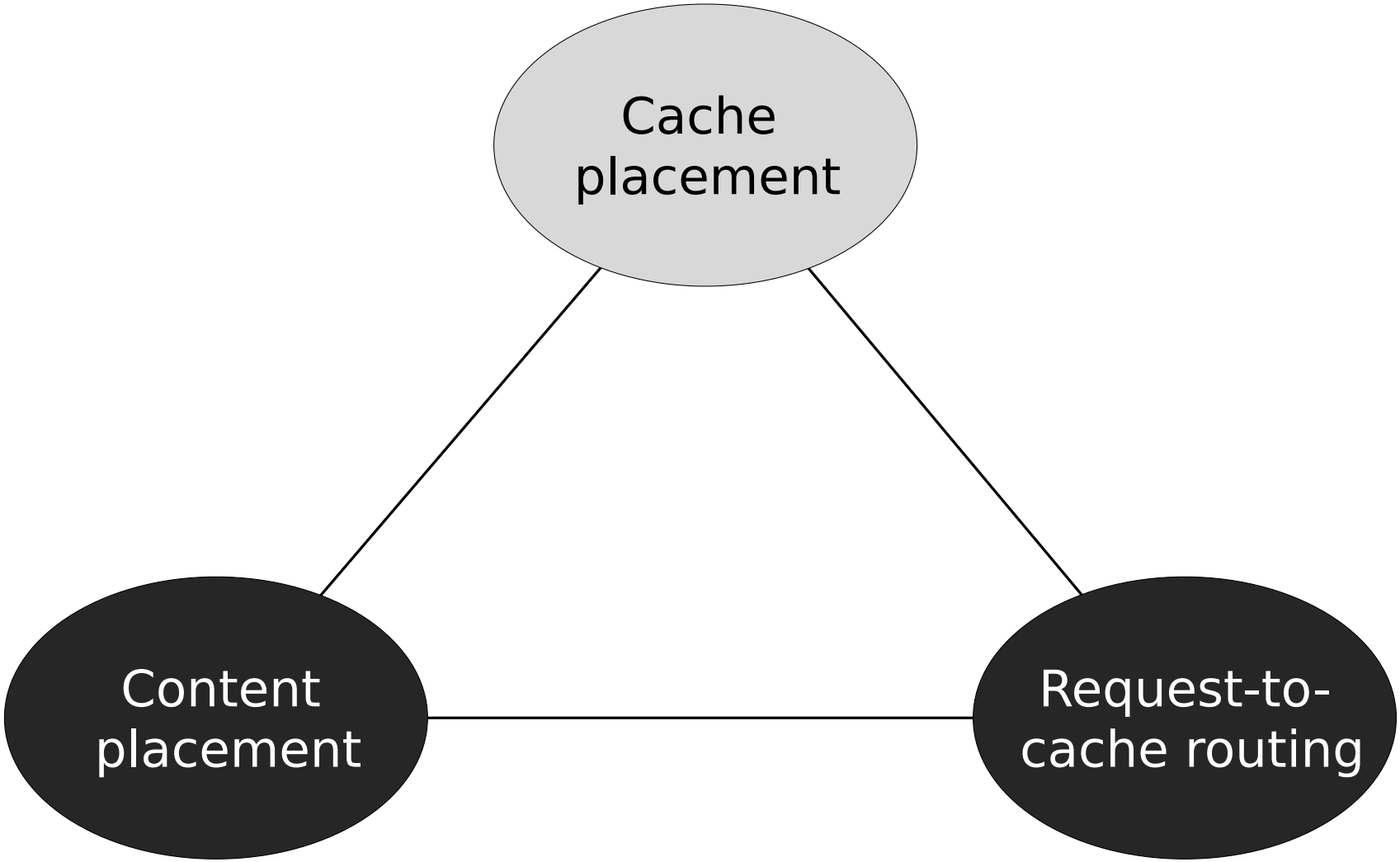
Communications and Information Systems Group
Department of Electrical and Electronics Engineering
University College London
{l.saino,i.psaras,g.pavlou}@ucl.ac.uk

In-network Caching Challenges[1]



[1] D. Kutscher and et al. ICN Research Challenges. IRTF draft *draft-kutscher-icnrg-challenges-01*, July 2013.

In-network Caching Challenges[1]



[1] D. Kutscher and et al. ICN Research Challenges. IRTF draft *draft-kutscher-icnrg-challenges-01*, July 2013.

Content Placement and Request-to-cache Routing

- **On-path caching with opportunistic request-to-cache routing**
  - Very scalable but limited cache hits due to redundant caching of contents
  - Examples: LCE, *ProbCache*, centrality-based caching
- **Off-path caching with co-ordinated request-to-cache routing**
  - High cache hits but limited scalability due to per-content state required for routing
- **Hybrid Techniques**
  - Mix features of on-path and off-path techniques
  - E.g. SCAN

Content Placement and Request-to-cache Routing

- **On-path caching with opportunistic request-to-cache routing**
  - Very scalable but limited cache hits due to redundant caching of contents
  - Examples: LCE, *ProbCache*, centrality-based caching
- **Off-path caching with co-ordinated request-to-cache routing**
  - High cache hits but limited scalability due to per-content state required for routing
- **Hybrid Techniques**
  - Mix features of on-path and off-path techniques
  - E.g. SCAN

Content Placement and Request-to-cache Routing

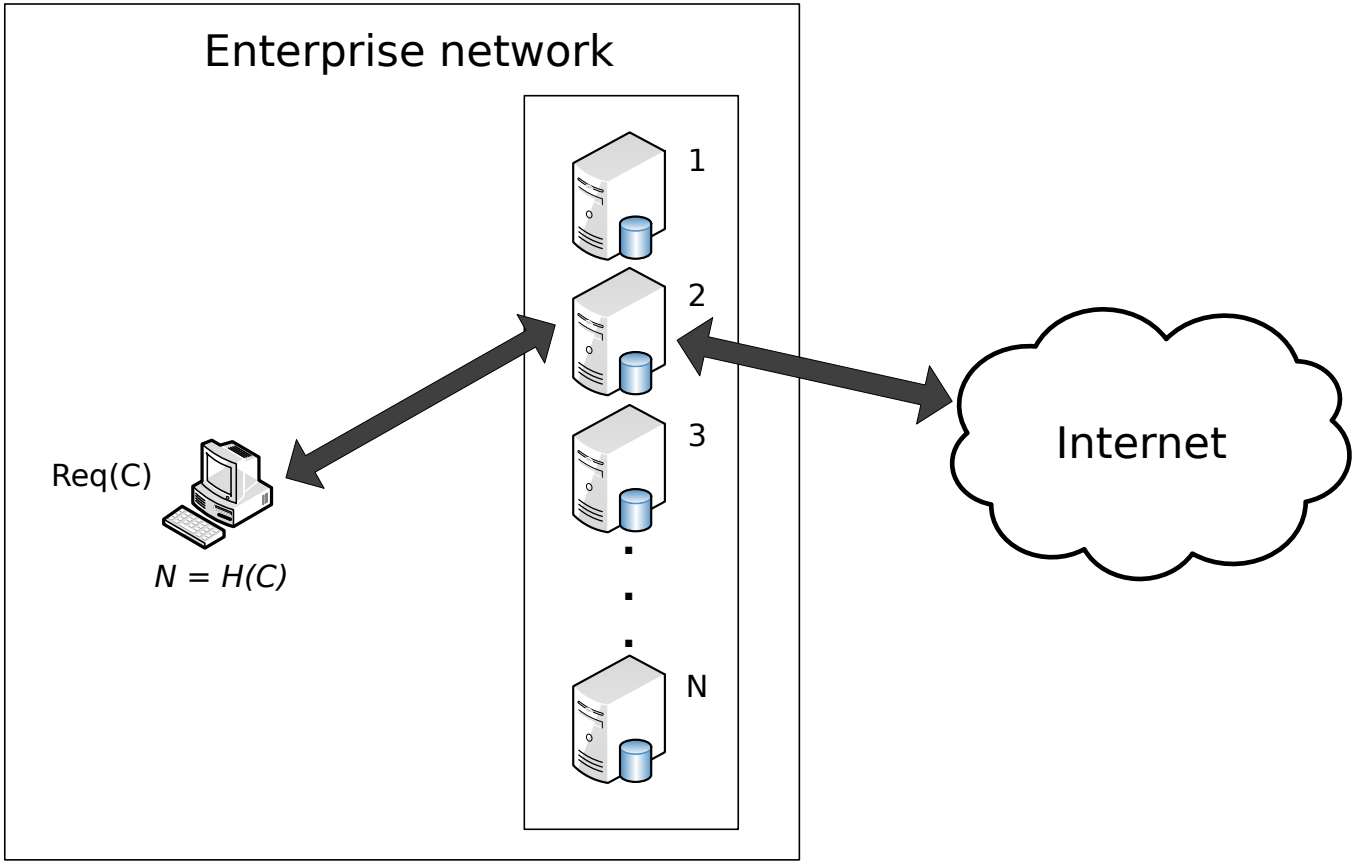
- **On-path caching with opportunistic request-to-cache routing**
  - Very scalable but limited cache hits due to redundant caching of contents
  - Examples: LCE, *ProbCache*, centrality-based caching
- **Off-path caching with co-ordinated request-to-cache routing**
  - High cache hits but limited scalability due to per-content state required for routing
- **Hybrid Techniques**
  - Mix features of on-path and off-path techniques
  - E.g. SCAN

Content Placement and Request-to-cache Routing

- **On-path caching with opportunistic request-to-cache routing**
  - Very scalable but limited cache hits due to redundant caching of contents
  - Examples: LCE, *ProbCache*, centrality-based caching
- **Off-path caching with co-ordinated request-to-cache routing**
  - High cache hits but limited scalability due to per-content state required for routing
- **Hybrid Techniques**
  - Mix features of on-path and off-path techniques
  - E.g. SCAN

Hash-routing

Hash-routing is a well-known Web caching technique to map content requests to nodes of a cache cluster using a hash function.



Hash-routing for Information Centric Networking

Functional entities:

- **Edge nodes:** Compute hash function and forward request and content packets to the responsible cache nodes
- **Cache nodes:** Store content objects for which they are responsible

Proposed routing schemes:

- Base schemes: Symmetric, Asymmetric, Multicast
- Hybrid schemes: Asymmetric-Multicast, Symmetric-Multicast

Hash-routing for Information Centric Networking

Functional entities:

- **Edge nodes:** Compute hash function and forward request and content packets to the responsible cache nodes
- **Cache nodes:** Store content objects for which they are responsible

Proposed routing schemes:

- Base schemes: Symmetric, Asymmetric, Multicast
- Hybrid schemes: Asymmetric-Multicast, Symmetric-Multicast

Hash-routing for Information Centric Networking

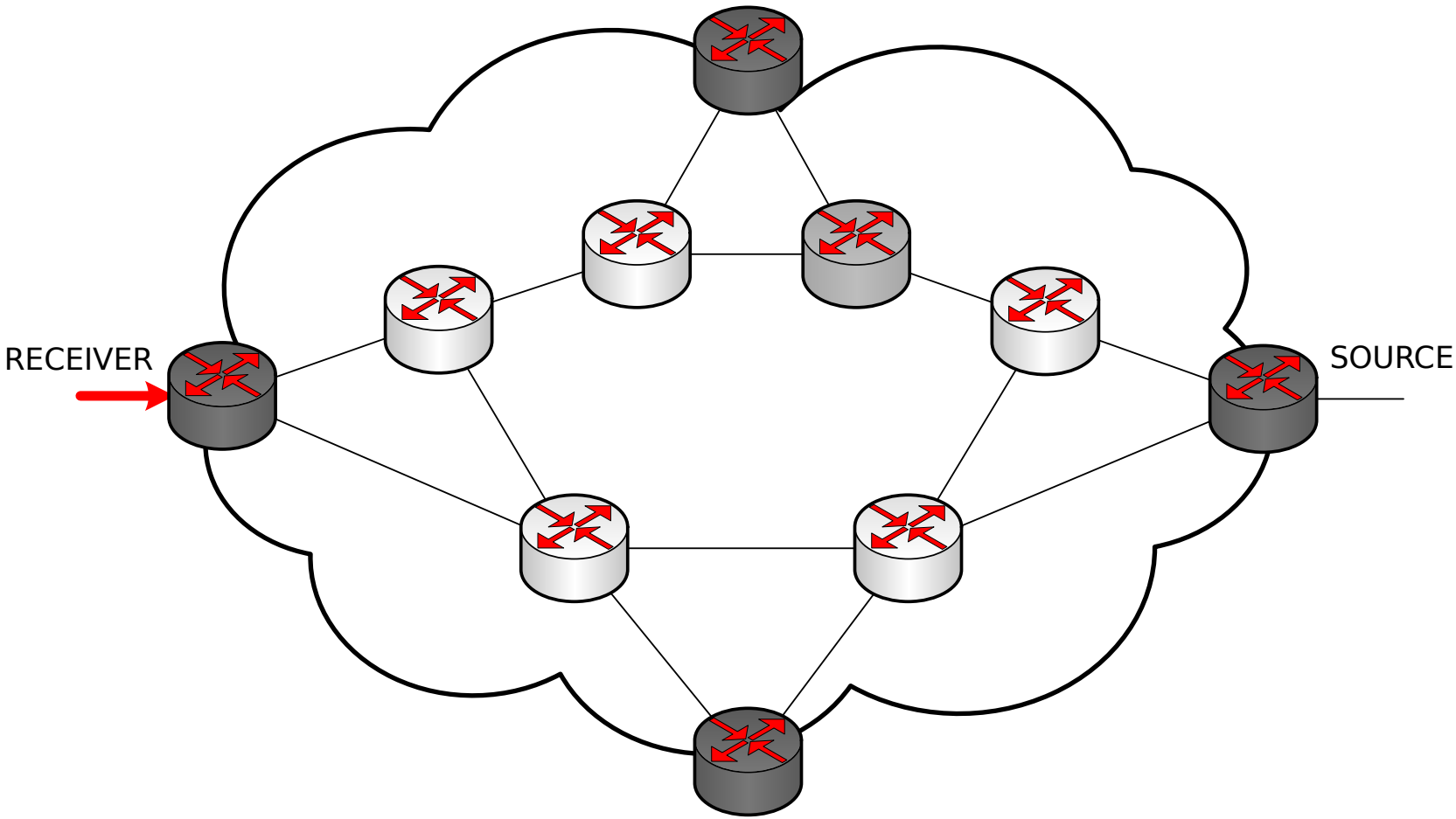
Functional entities:

- **Edge nodes:** Compute hash function and forward request and content packets to the responsible cache nodes
- **Cache nodes:** Store content objects for which they are responsible

Proposed routing schemes:

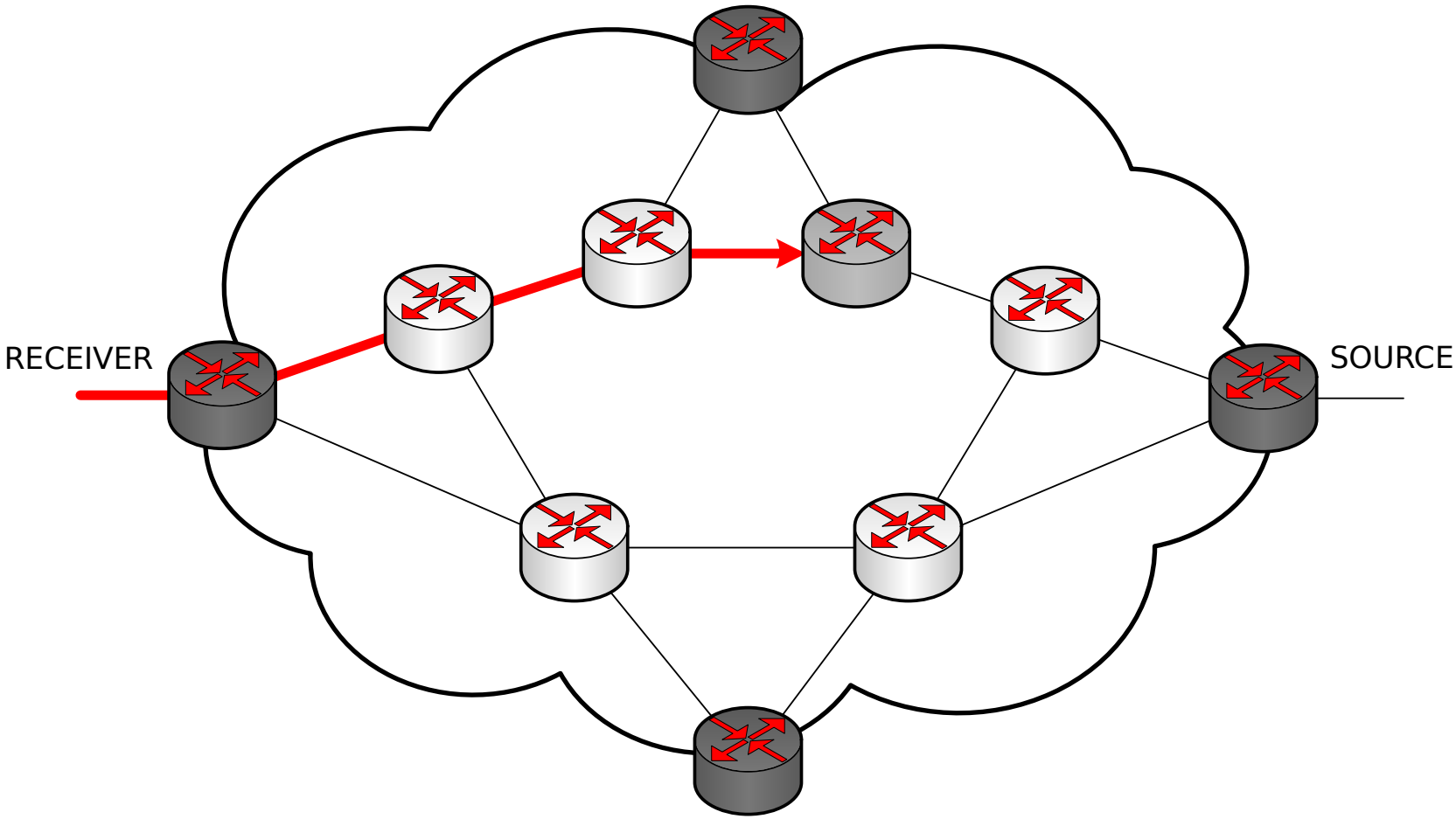
- Base schemes: Symmetric, Asymmetric, Multicast
- Hybrid schemes: Asymmetric-Multicast, Symmetric-Multicast

Request routing



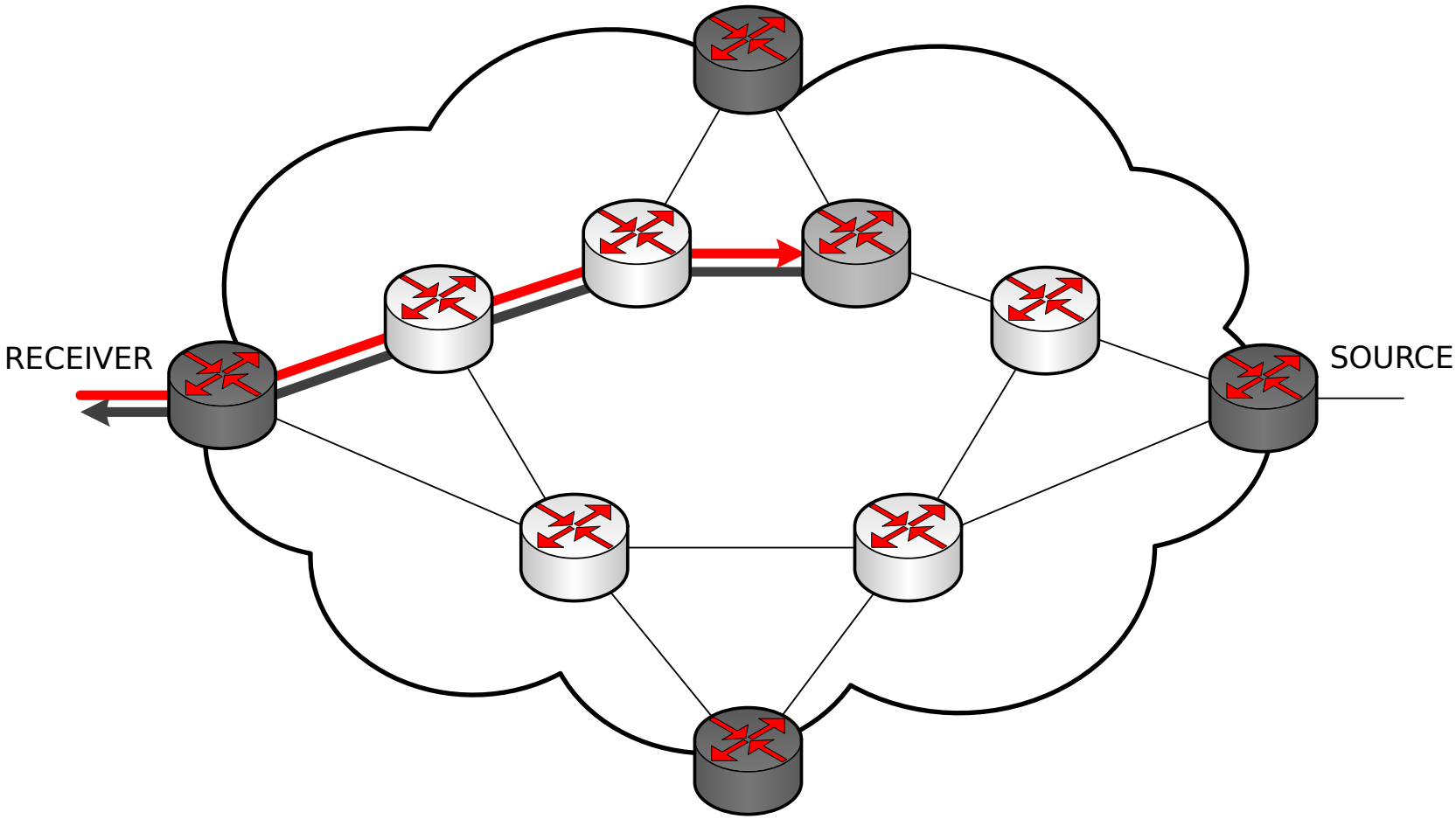
The ingress edge node computes hash function to map the content identifier to the responsible cache node

Request routing



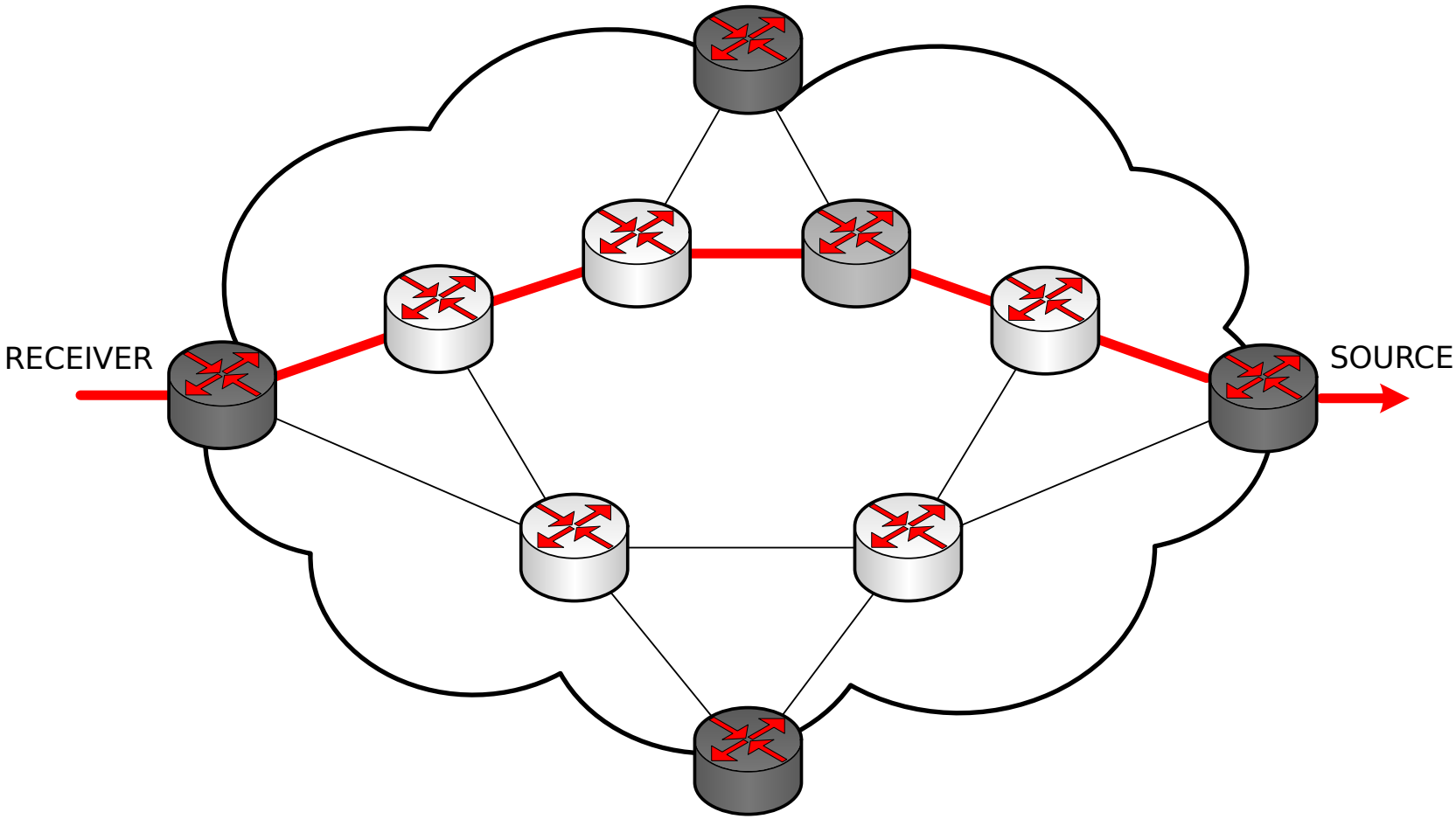
The ingress edge node forwards request to resolved cache node

Request routing



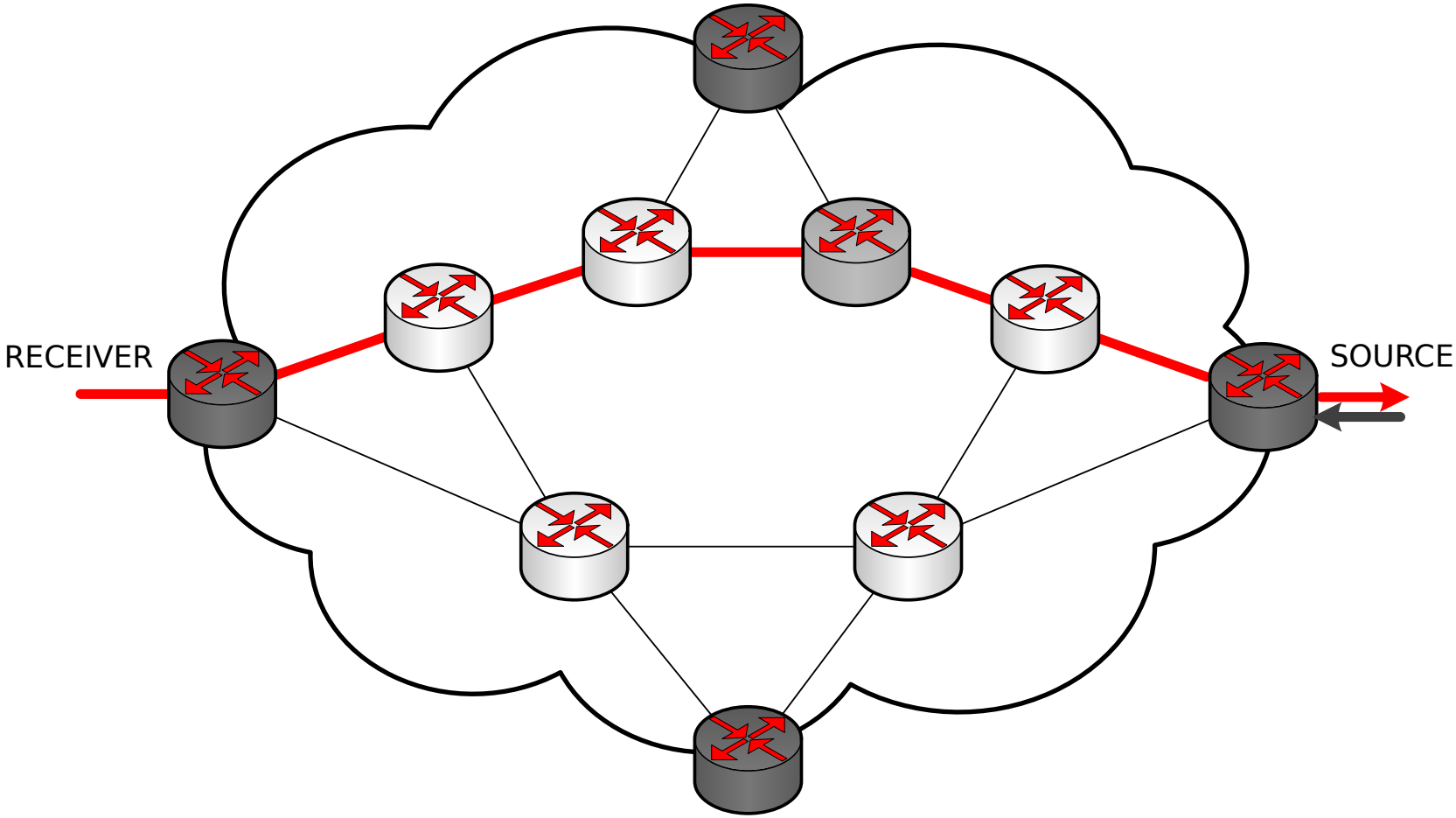
If the responsible cache node has a copy of the requested content, it serves it to receiver

Request routing

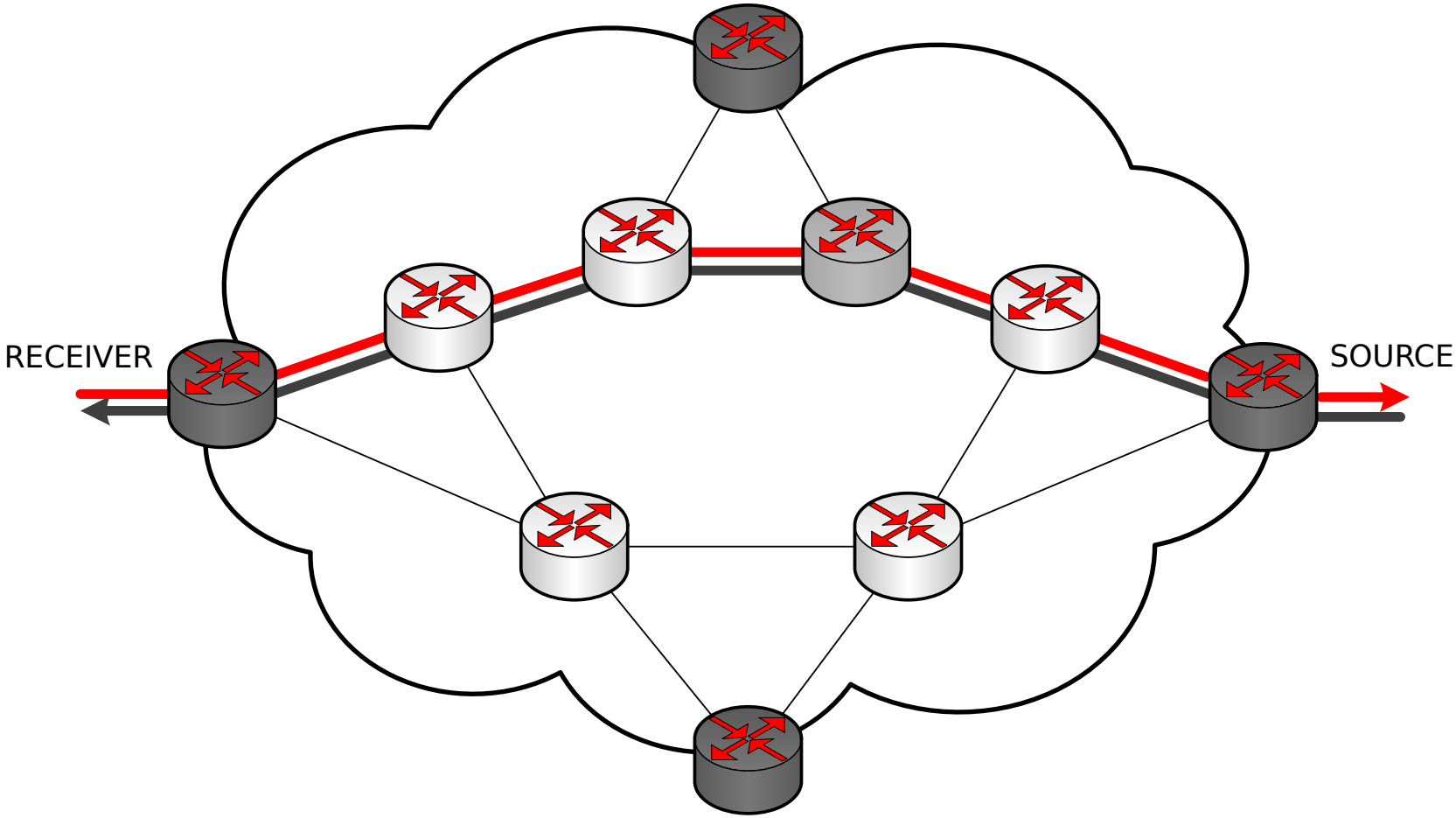


Otherwise, it forwards the request towards the original content source

Content routing - Symmetric

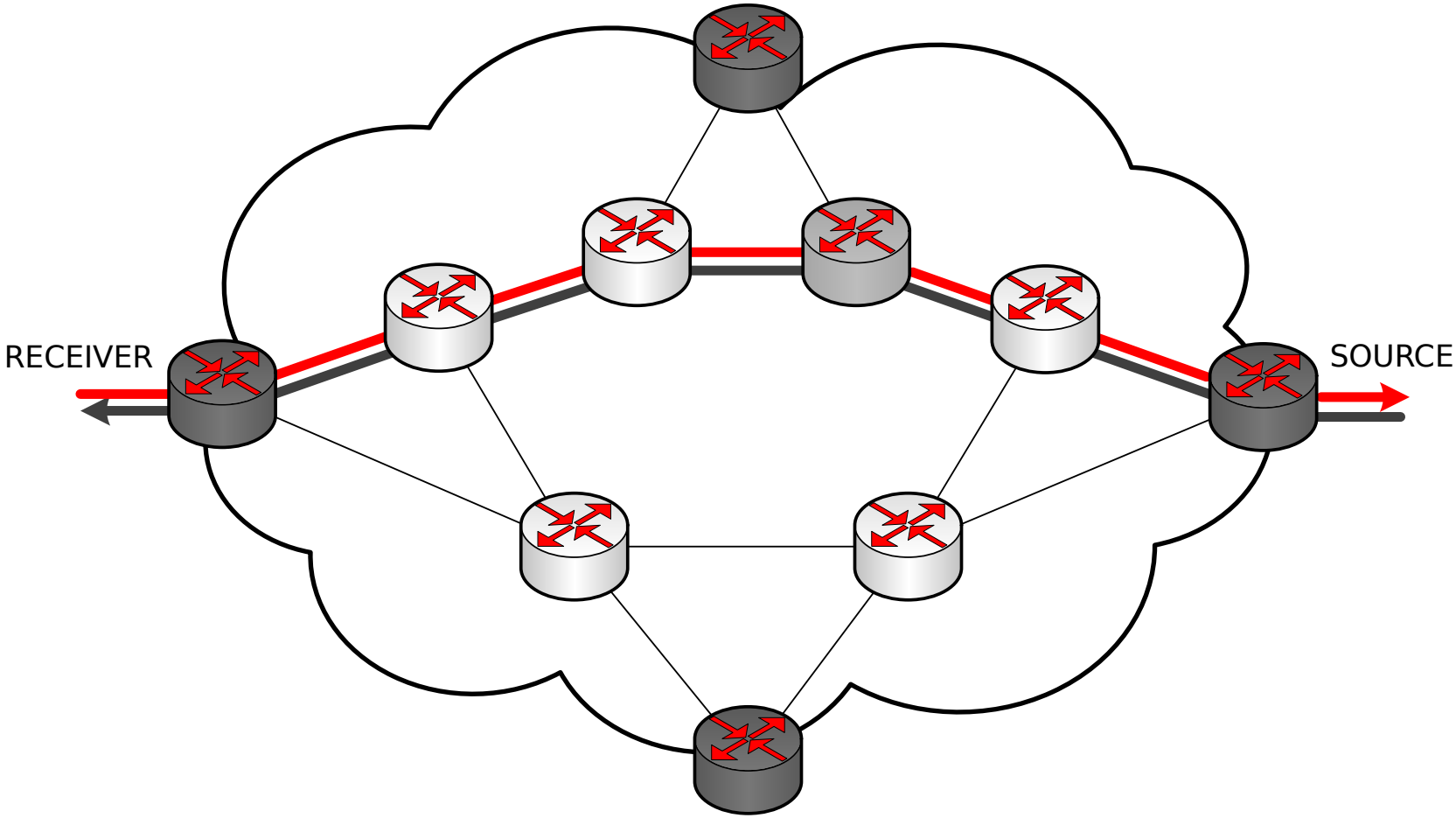


Content routing - Symmetric



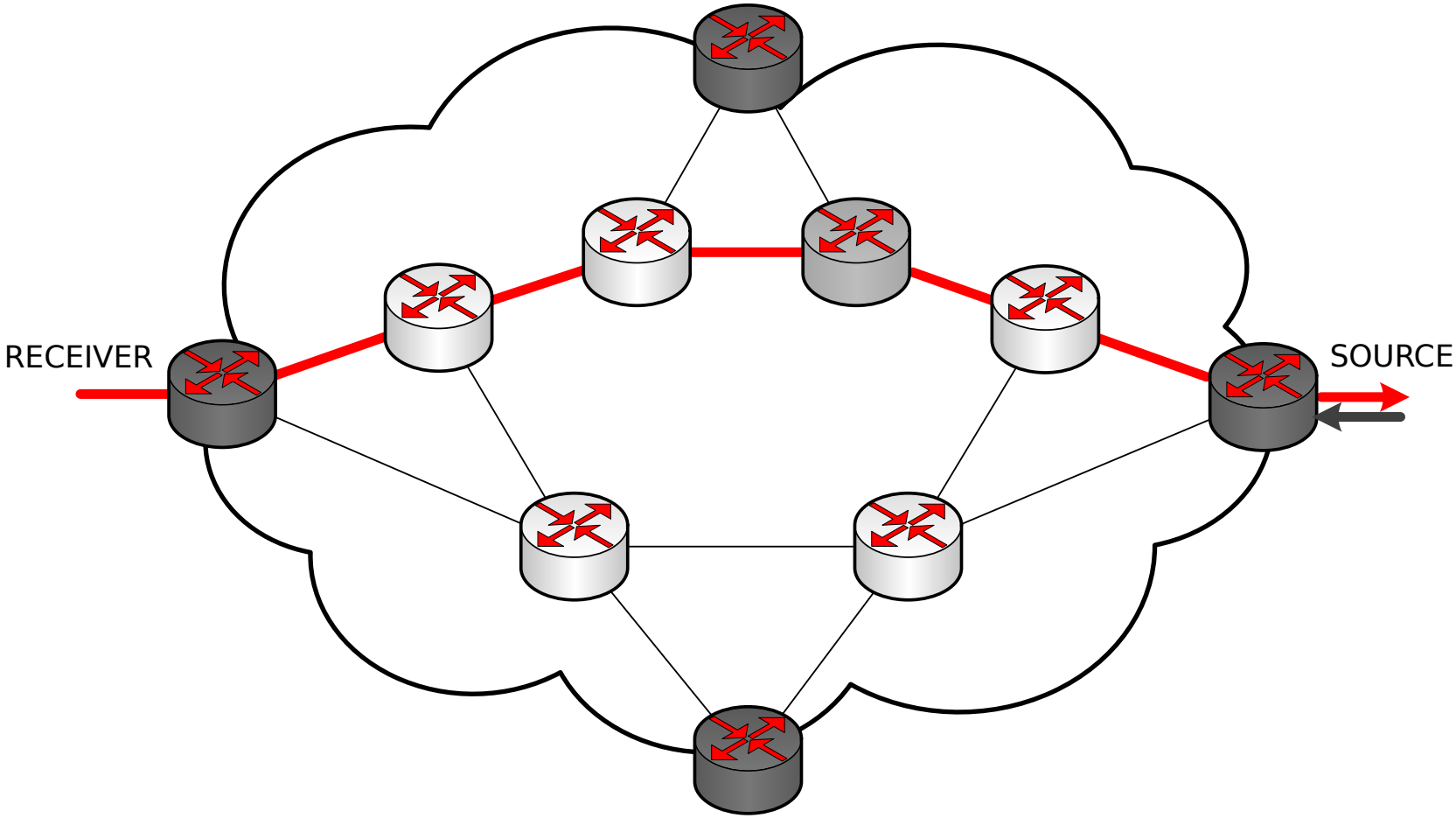
- Content packets follow the same path of the request

Content routing - Symmetric

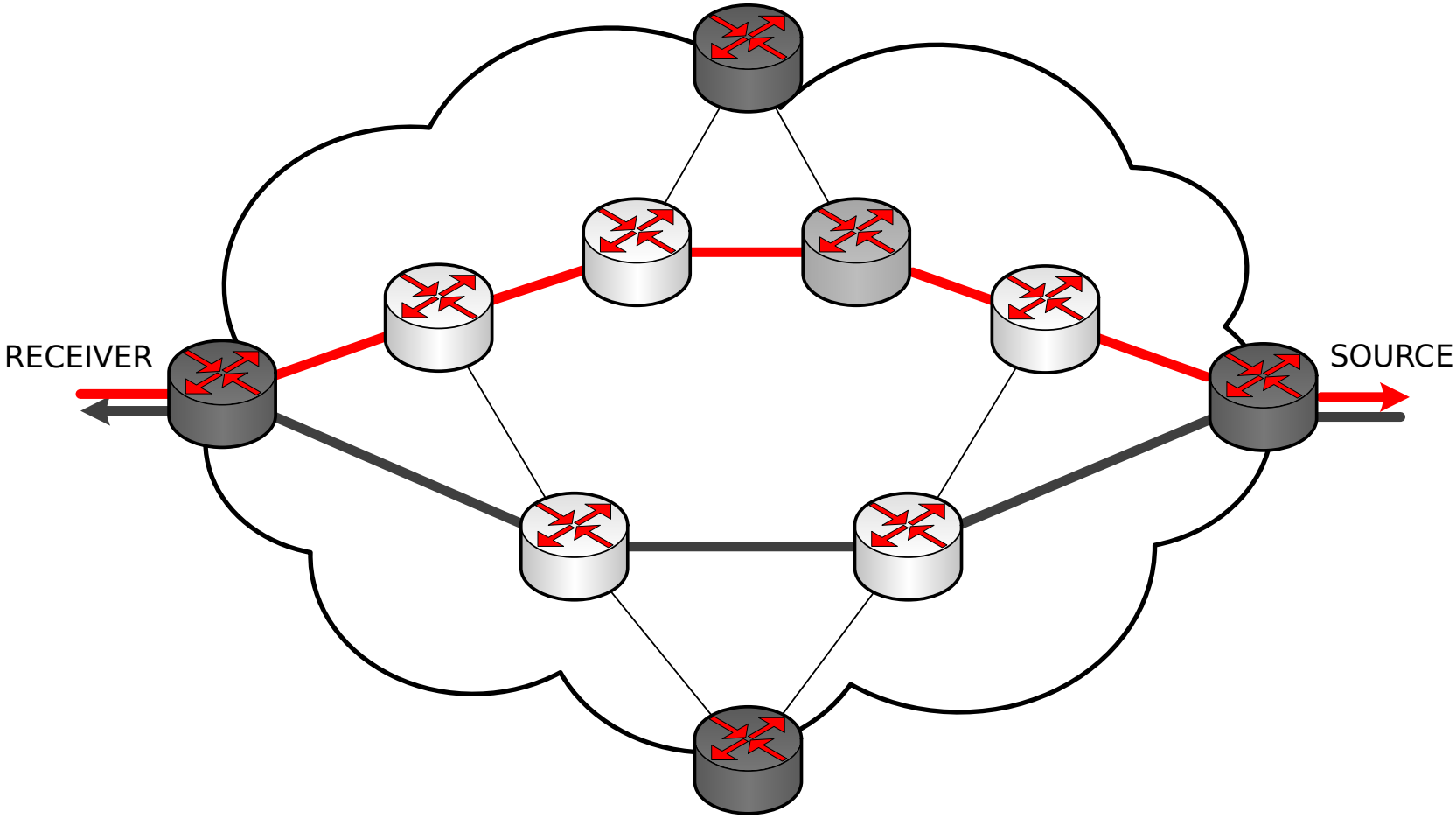


- Content packets follow the same path of the request
- This approach can achieve high cache hit rate but at the cost of possibly increasing intradomain link load

Content routing - Asymmetric

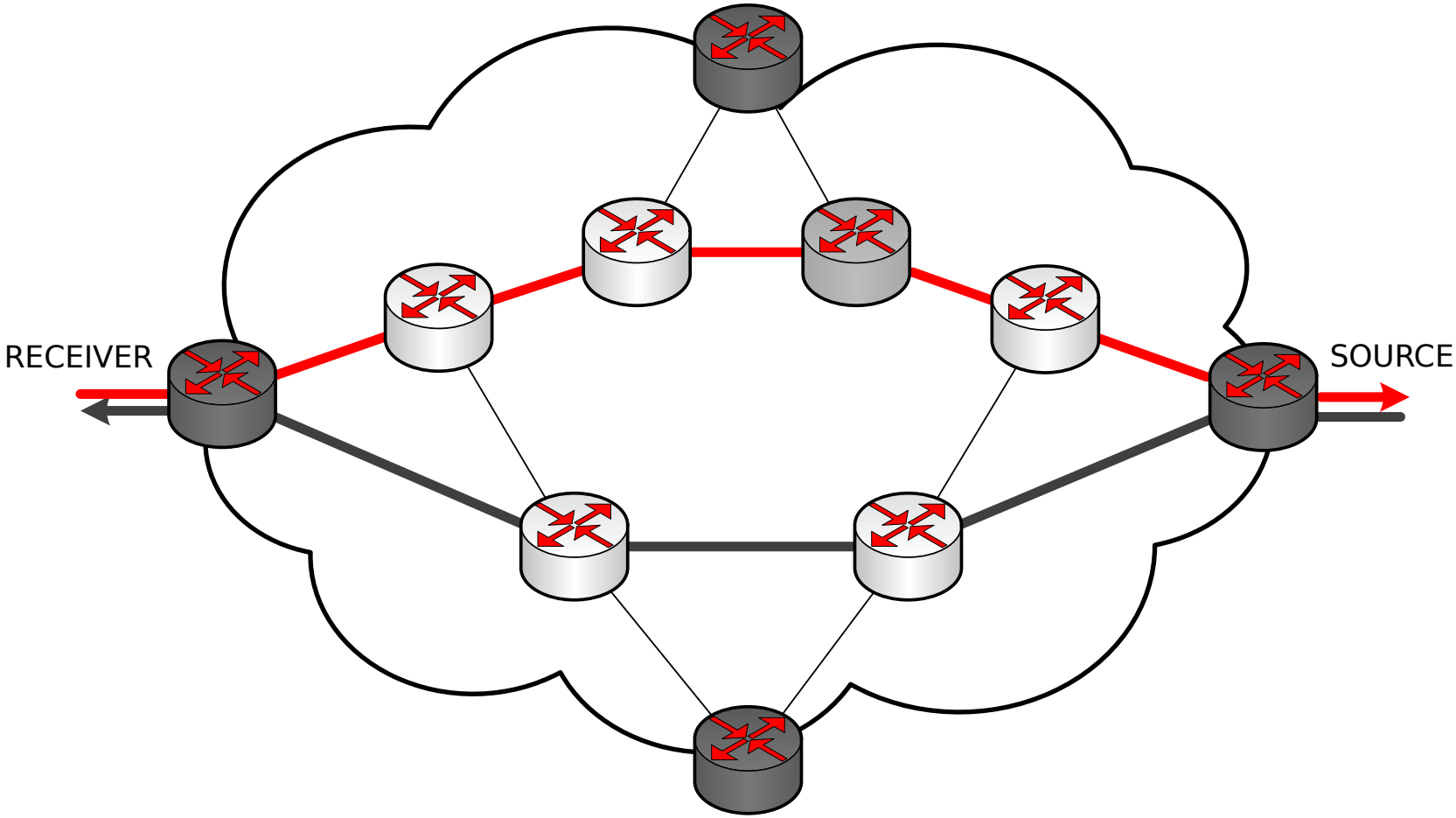


Content routing - Asymmetric



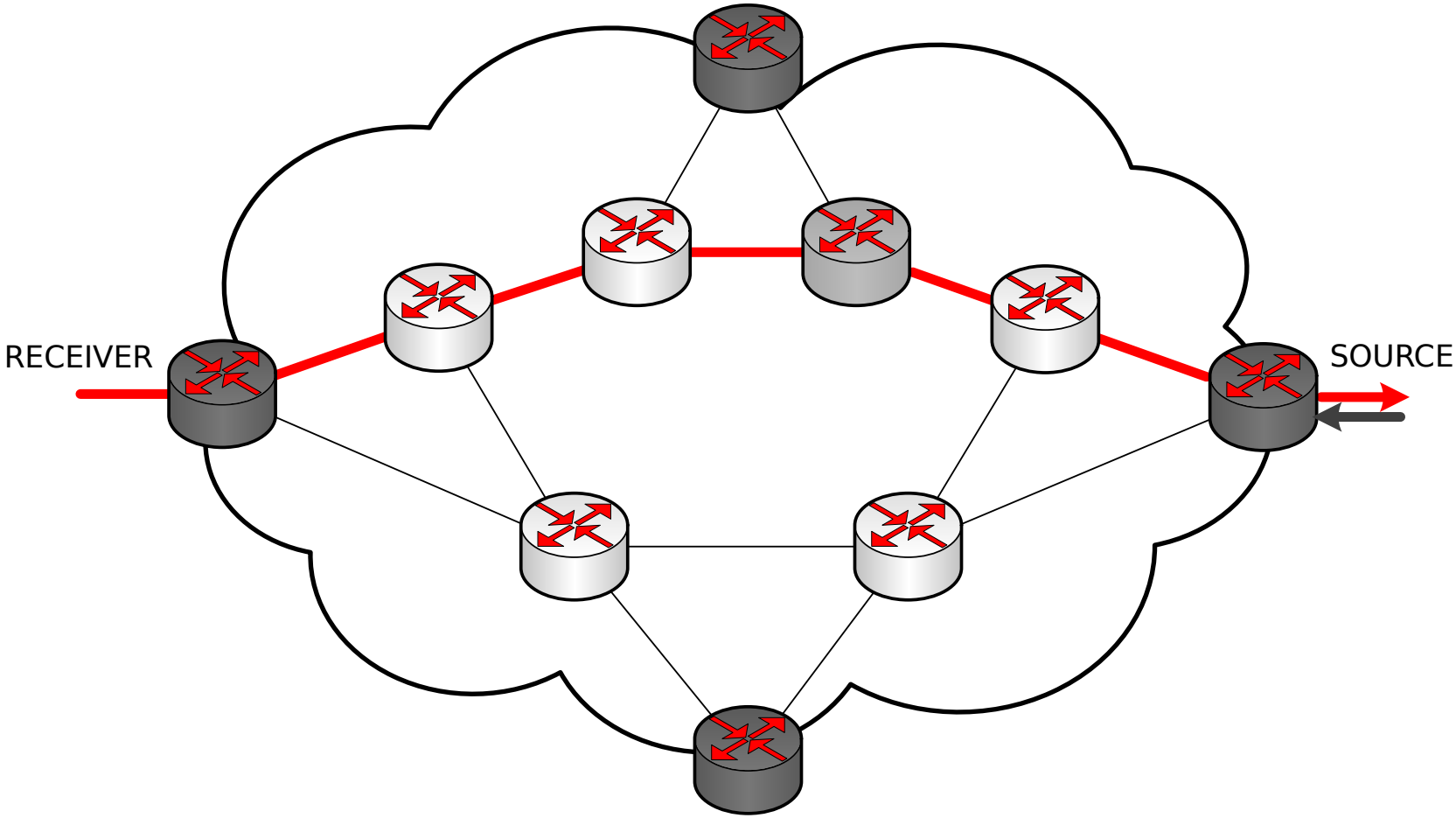
- Content packets are always forwarded over the shortest path

Content routing - Asymmetric

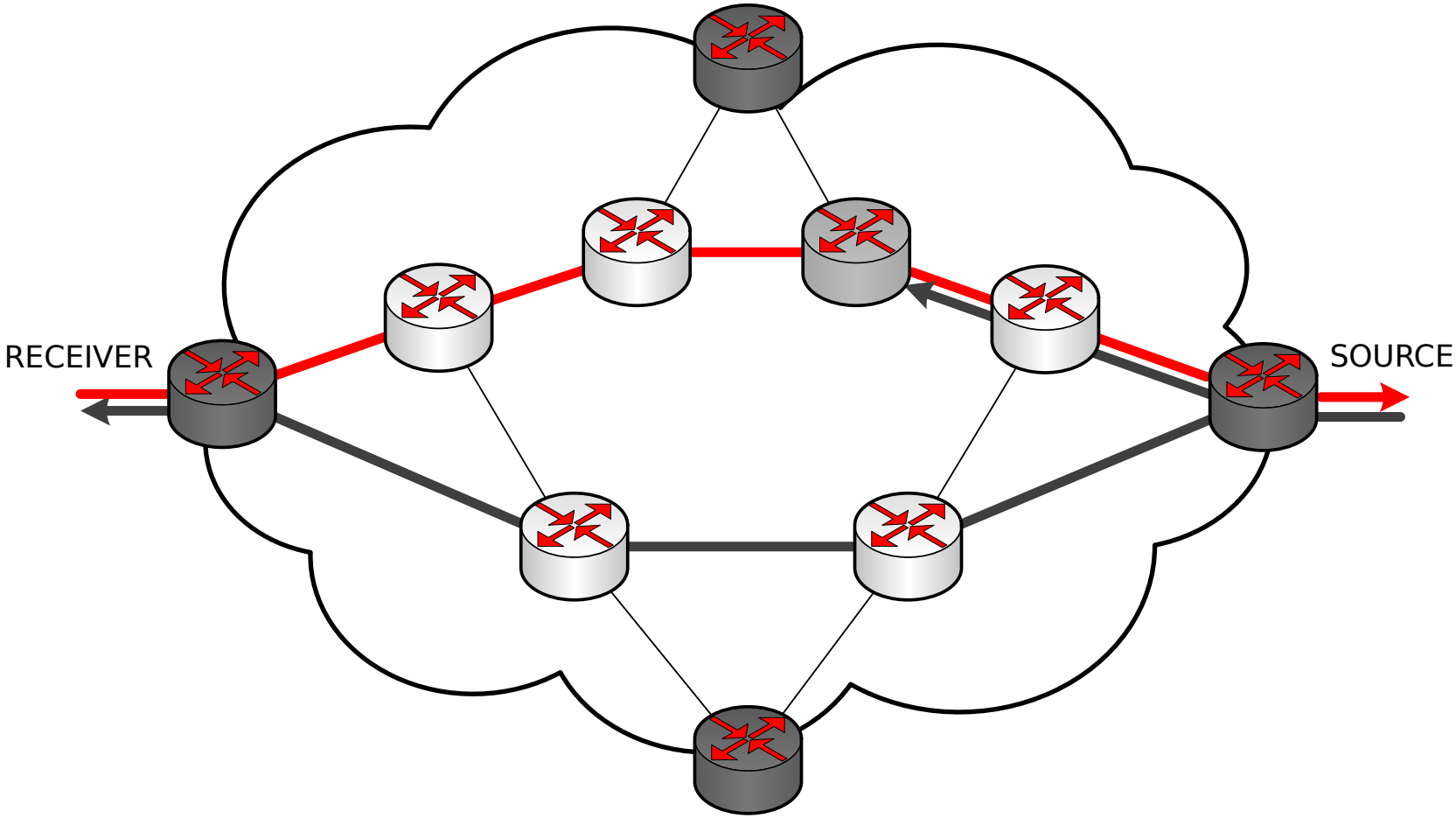


- Content packets are always forwarded over the shortest path
- This approach has minor impact on link load but cache nodes with small betweenness centrality may be underutilized

Content routing - Multicast

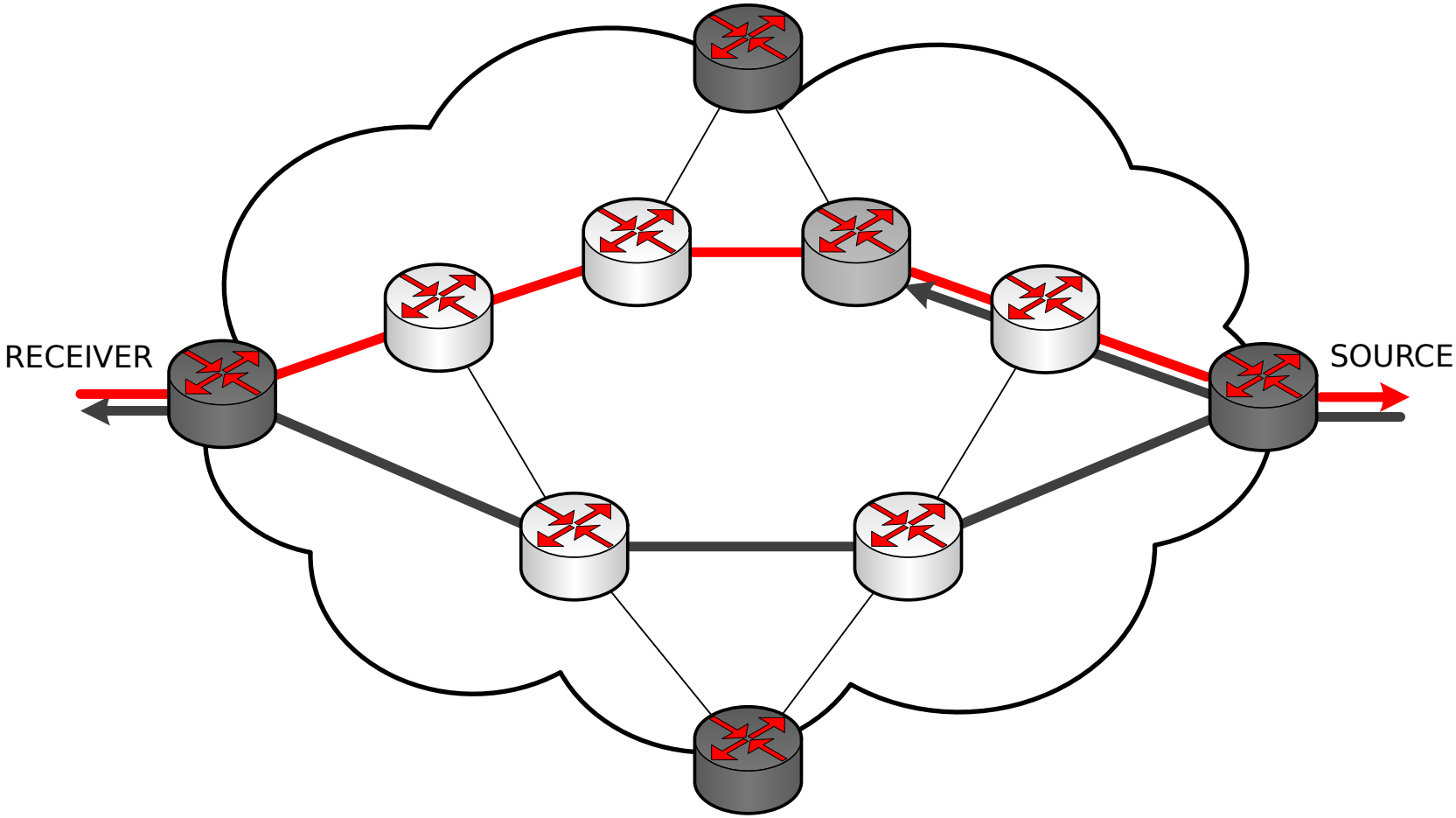


Content routing - Multicast



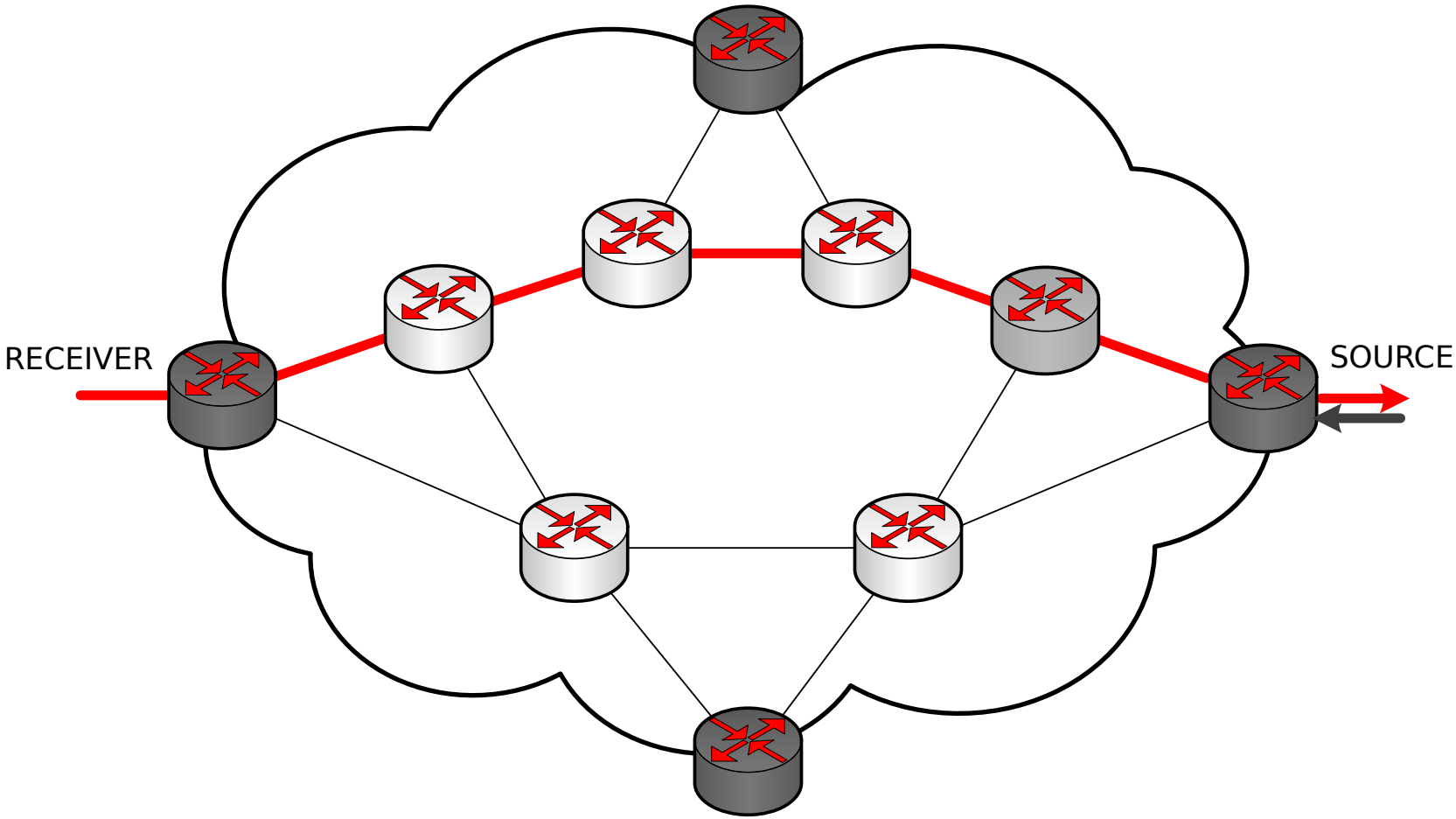
- Content packets are multicast to receiver and cache nodes

Content routing - Multicast



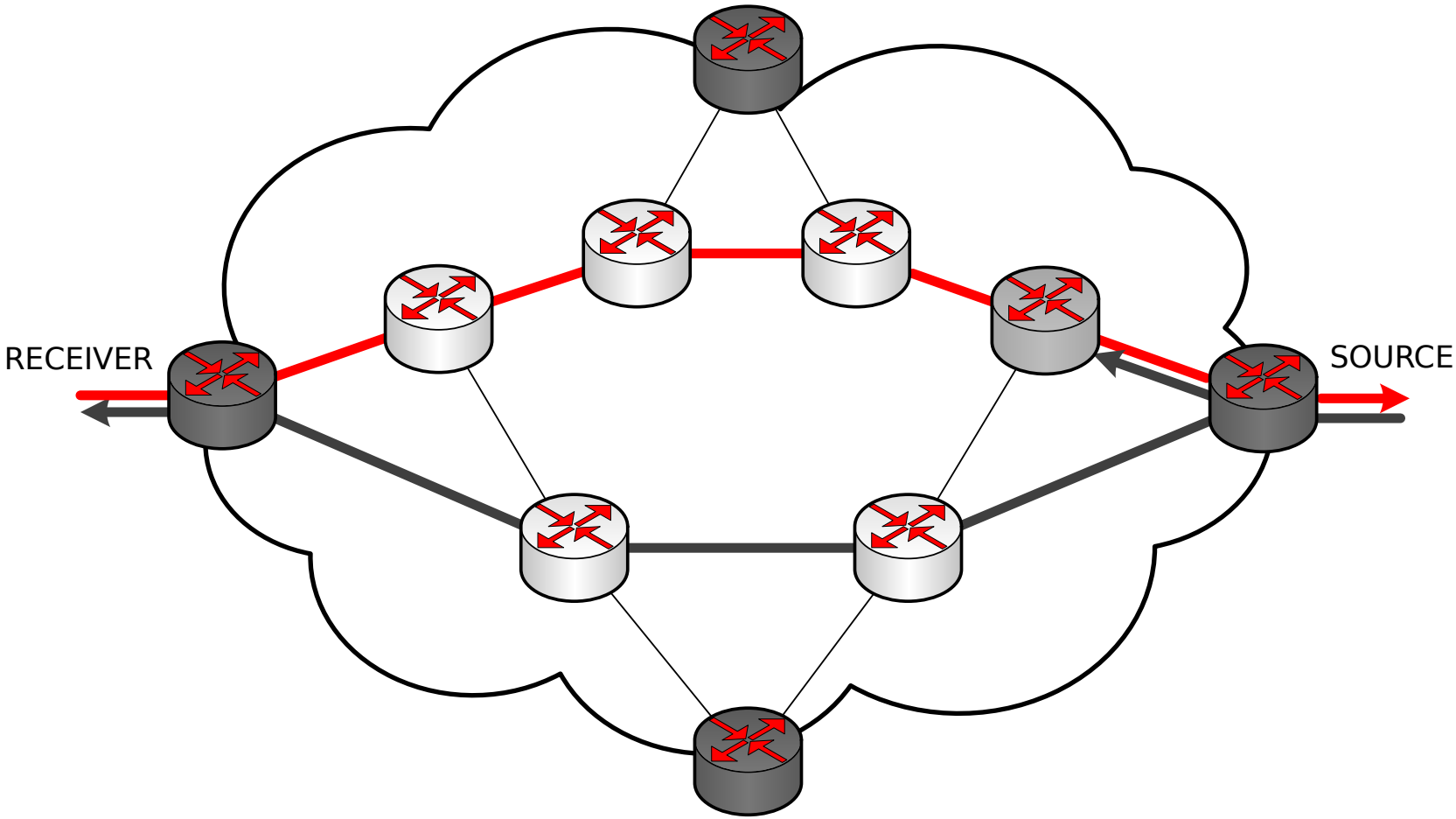
- Content packets are multicast to receiver and cache nodes
- This approach can achieve high cache hits and low latency, but may increase link load depending on network topology

Content routing - Symmetric-Multicast Hybrid



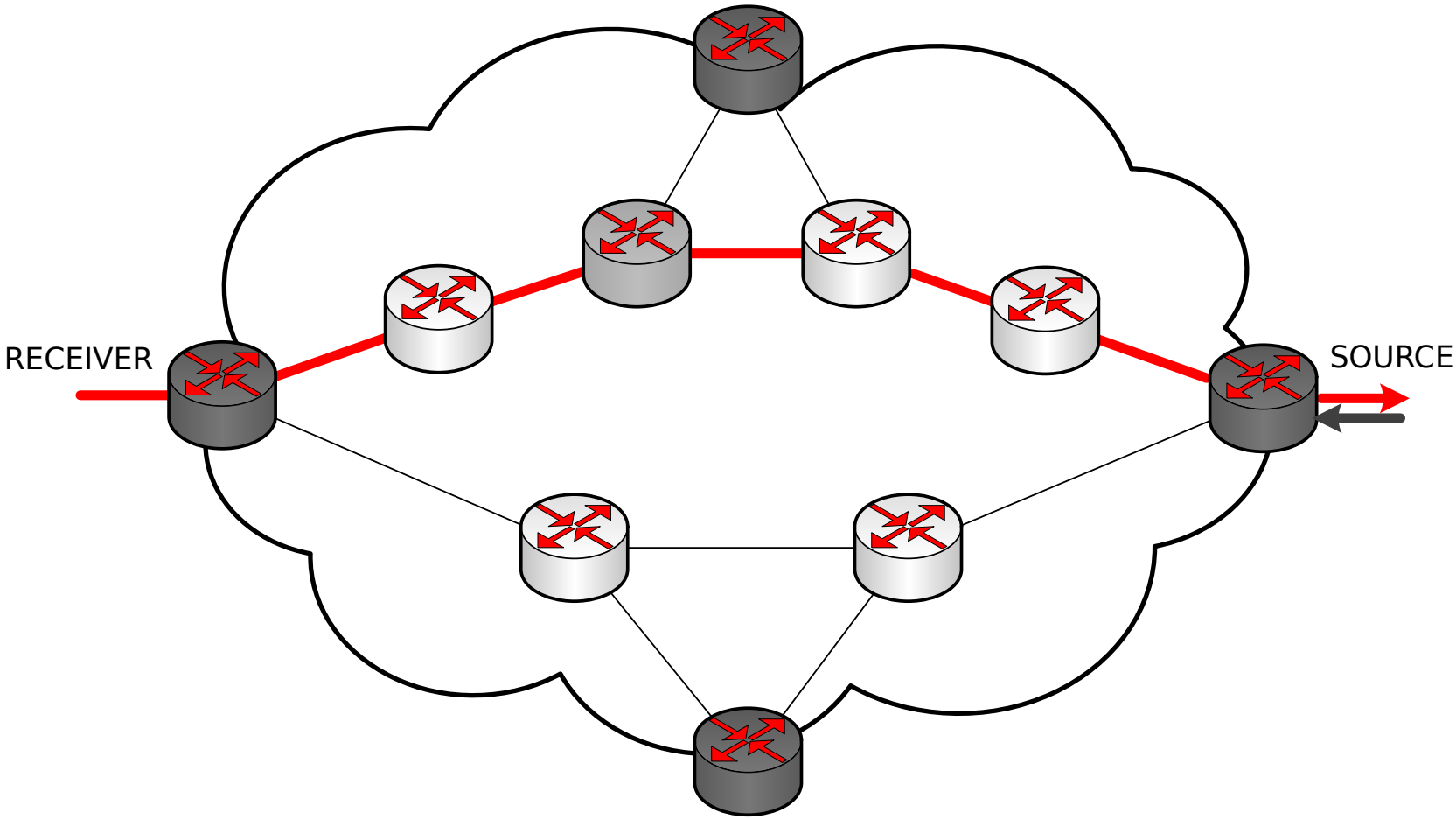
Edge nodes decide to forward content packets in a multicast or symmetric manner in order to minimize the total cost of the traversed path

Content routing - Symmetric-Multicast Hybrid



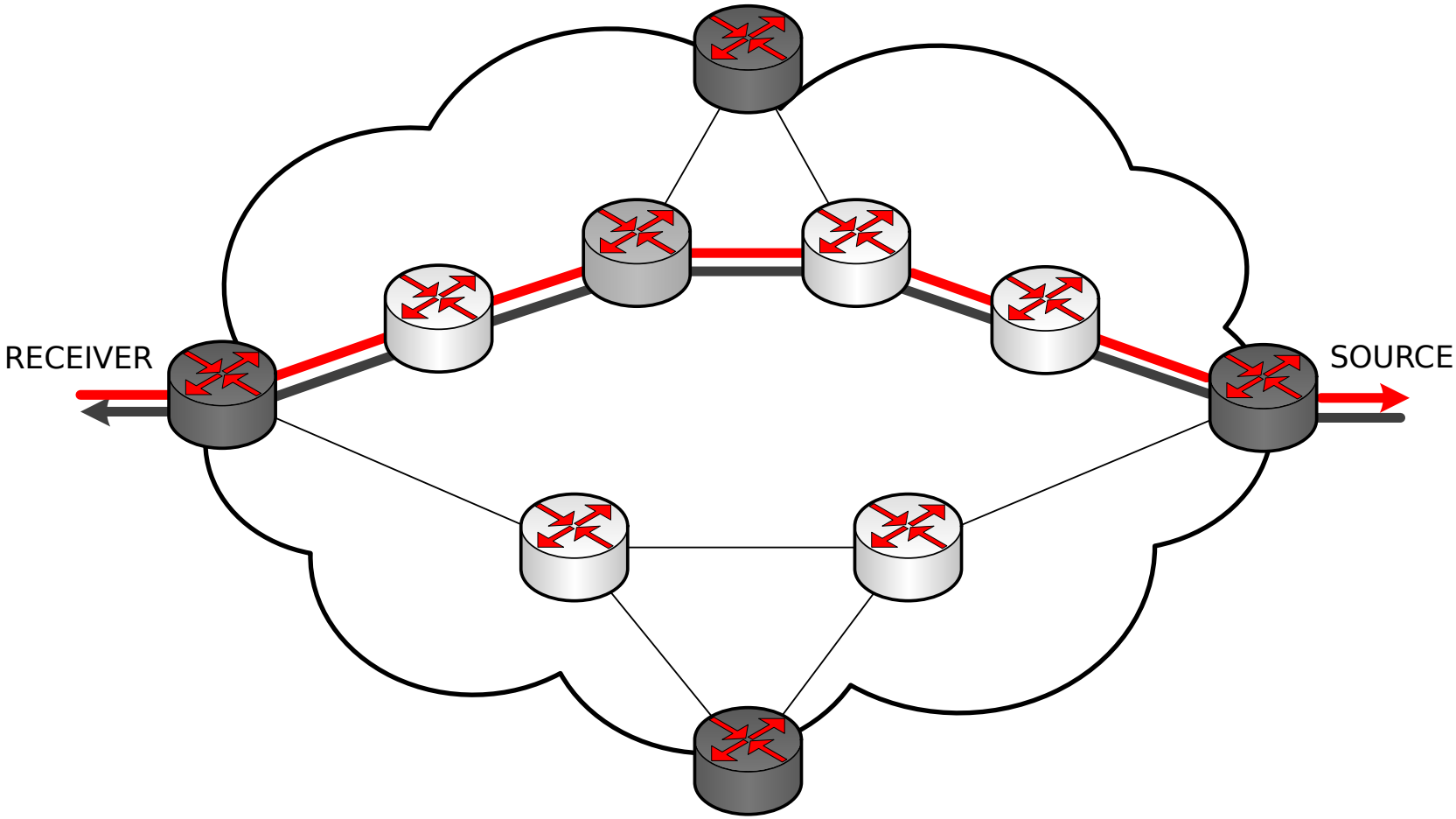
Edge nodes decide to forward content packets in a multicast or symmetric manner in order to minimize the total cost of the traversed path

Content routing - Symmetric-Multicast Hybrid



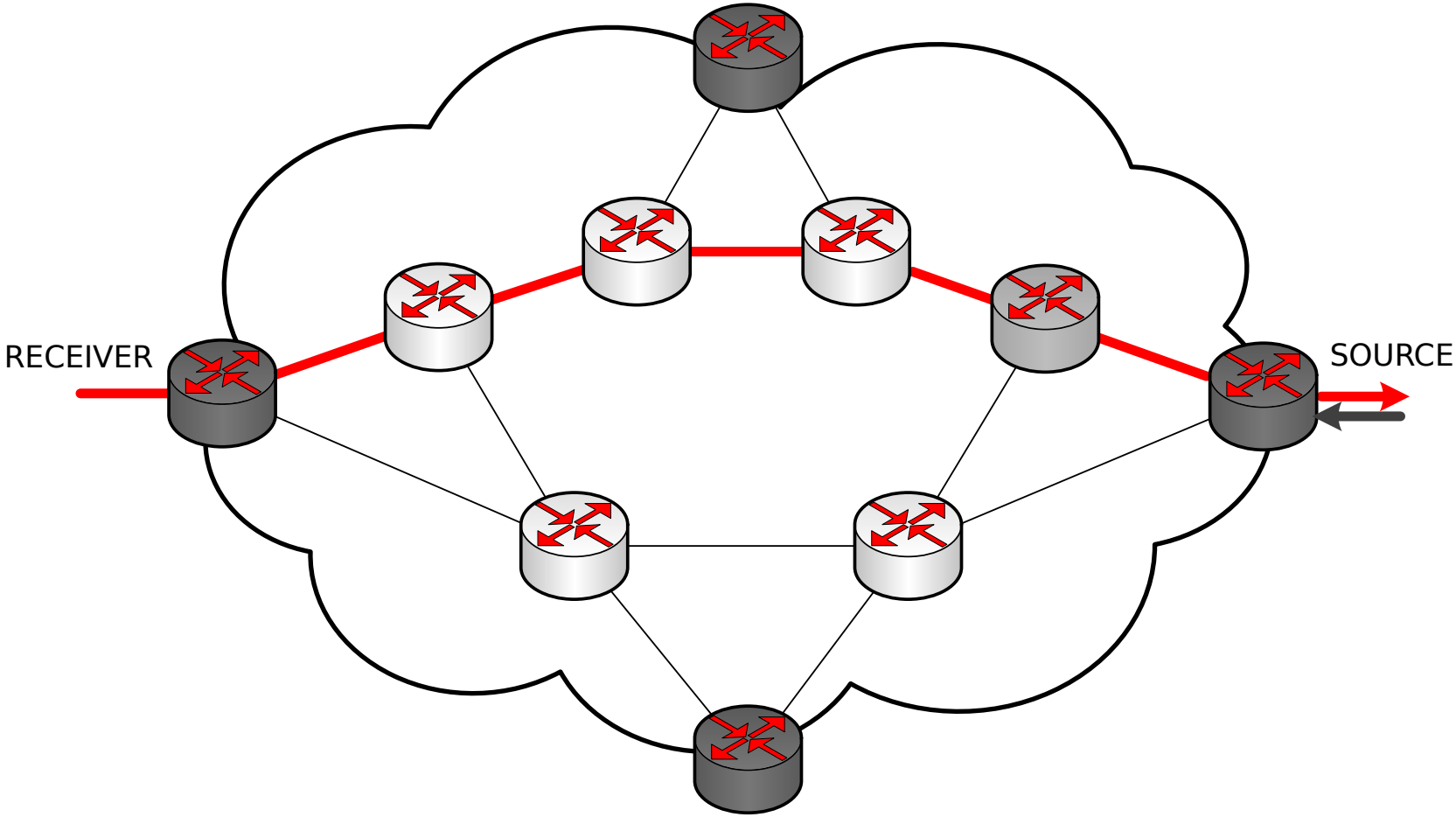
Edge nodes decide to forward content packets in a multicast or symmetric manner in order to minimize the total cost of the traversed path

Content routing - Symmetric-Multicast Hybrid



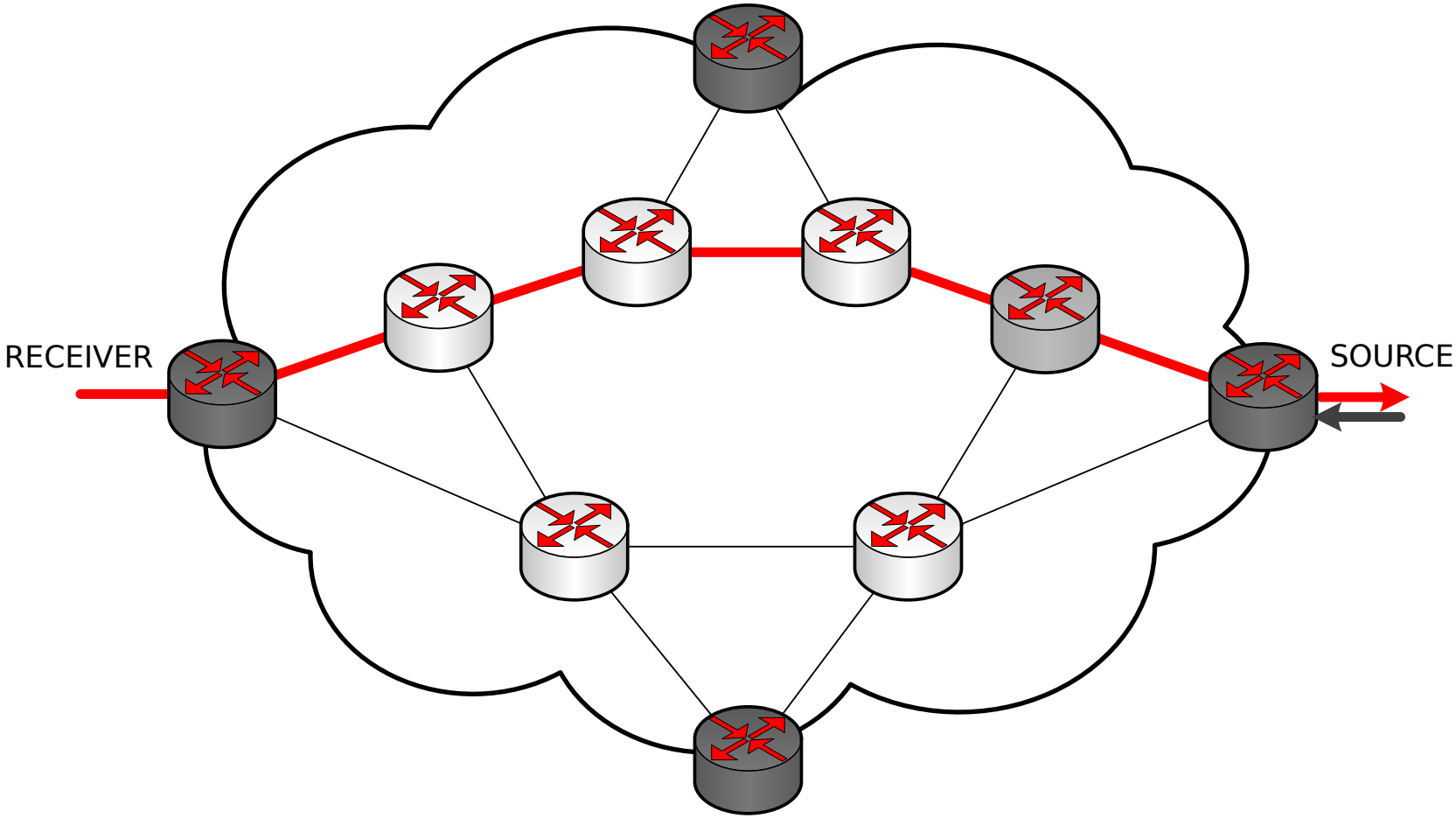
Edge nodes decide to forward content packets in a multicast or symmetric manner in order to minimize the total cost of the traversed path

Content routing - Asymmetric-Multicast Hybrid



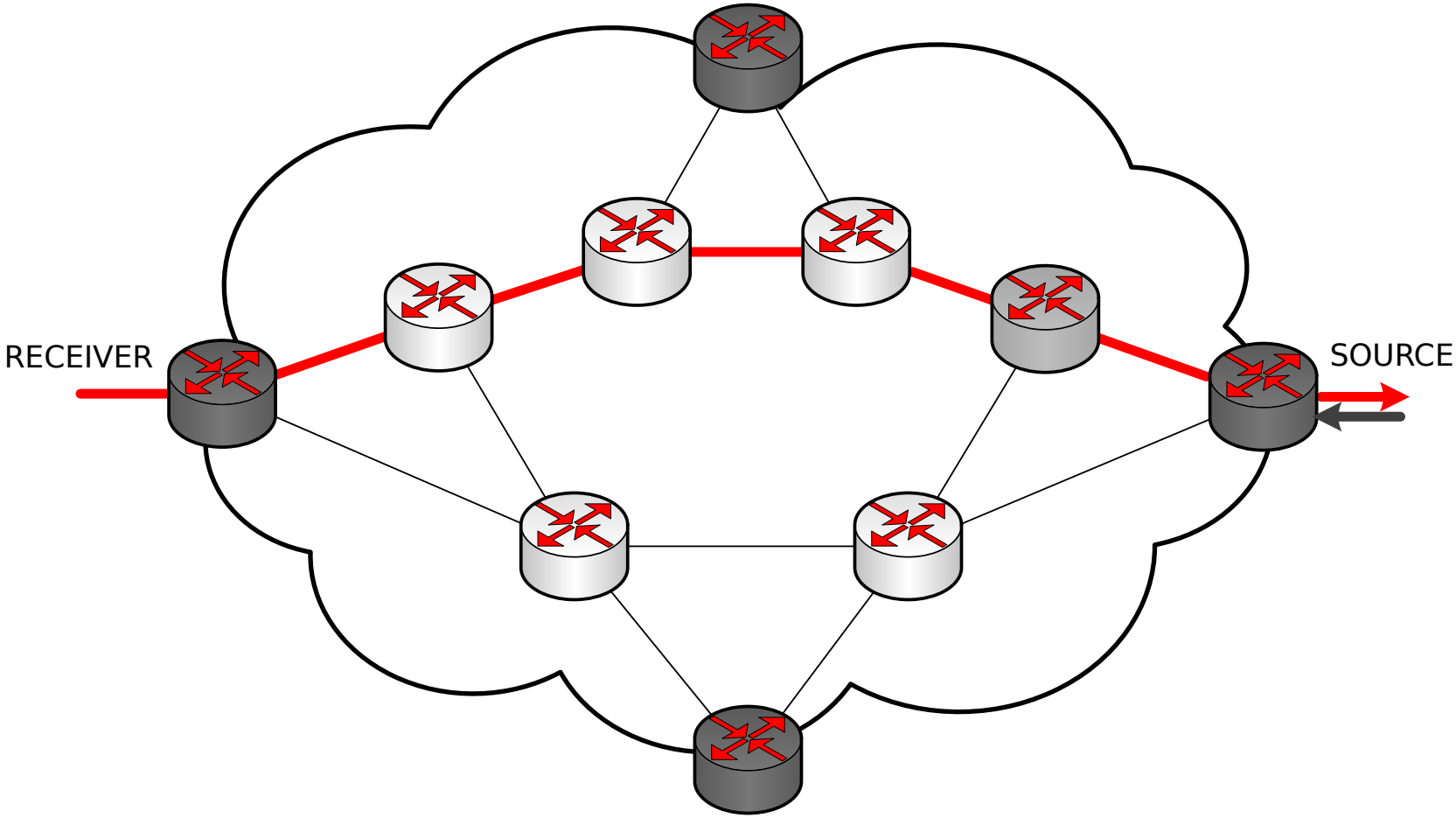
Edge nodes select multicast delivery only if the marginal cost of the multicast path with respect to the source-receiver shortest path is smaller than a predefined threshold.

Content routing - Asymmetric-Multicast Hybrid



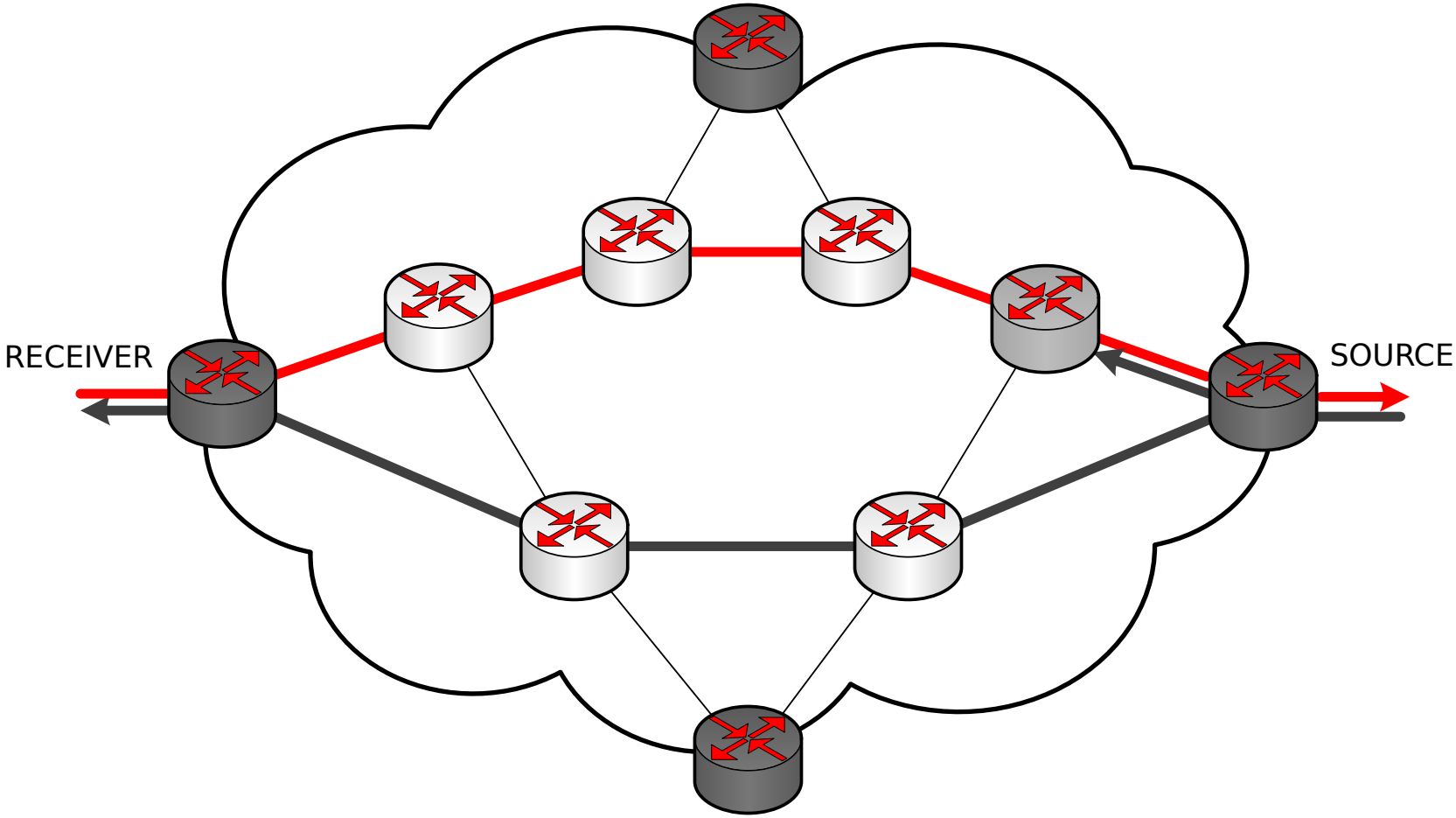
$$S = \begin{cases} MULTICAST & \text{if } C = \frac{C_{MCAST} - C_{ASYMM}}{C_{MAX}} < k_{MAX} \in (0,1) \\ ASYMM & \text{otherwise} \end{cases}$$

Content routing - Asymmetric-Multicast Hybrid



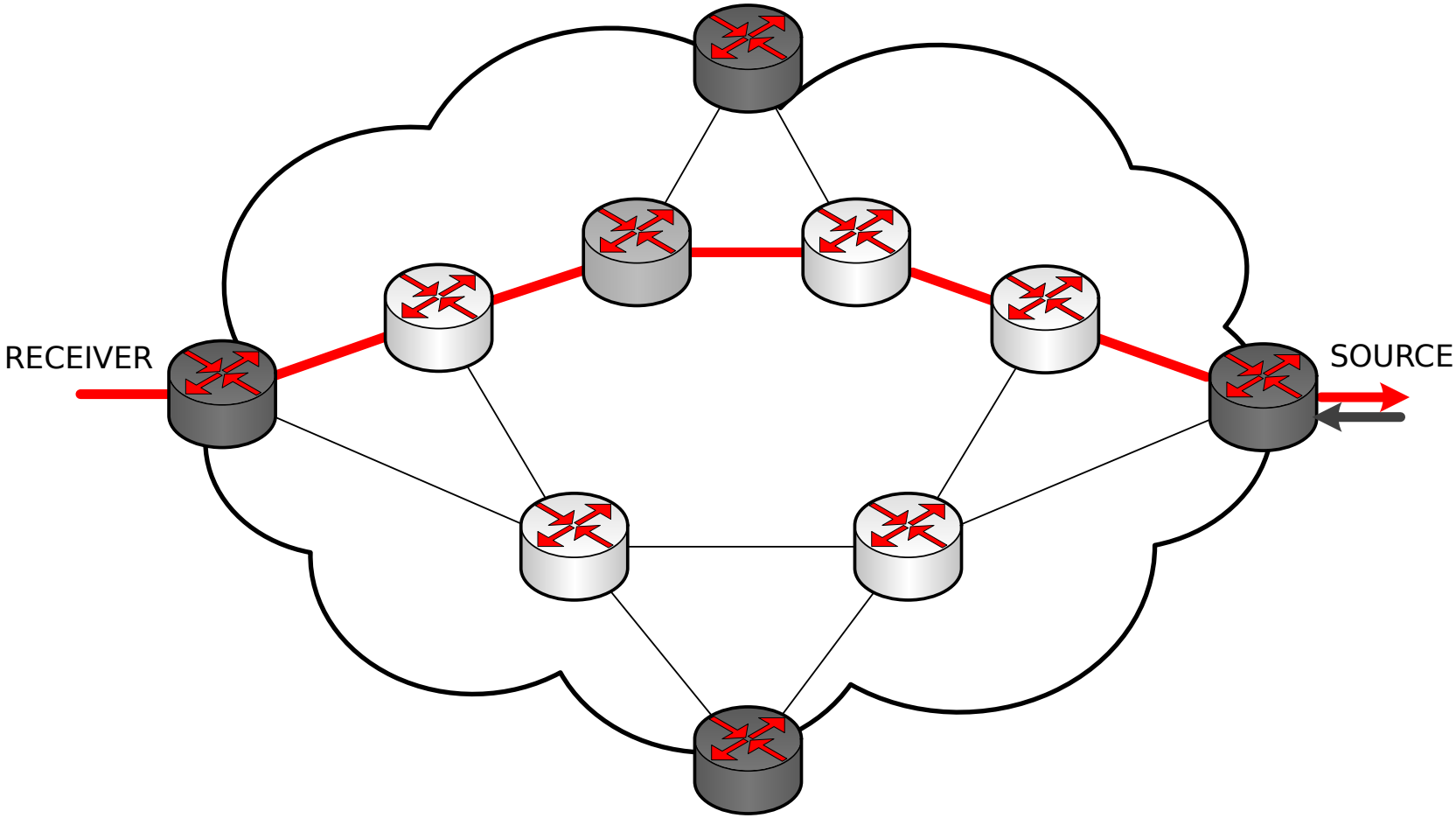
- We use unitary link weights to calculate path costs and $K_{MAX} = 0.3$
- $C_{ASYMM} = 3$, $C_{MCAST} = 4$, $C_{MAX} = 4$
- $C = 0.25 < K_{MAX} = 0.3$: multicast is used

Content routing - Asymmetric-Multicast Hybrid



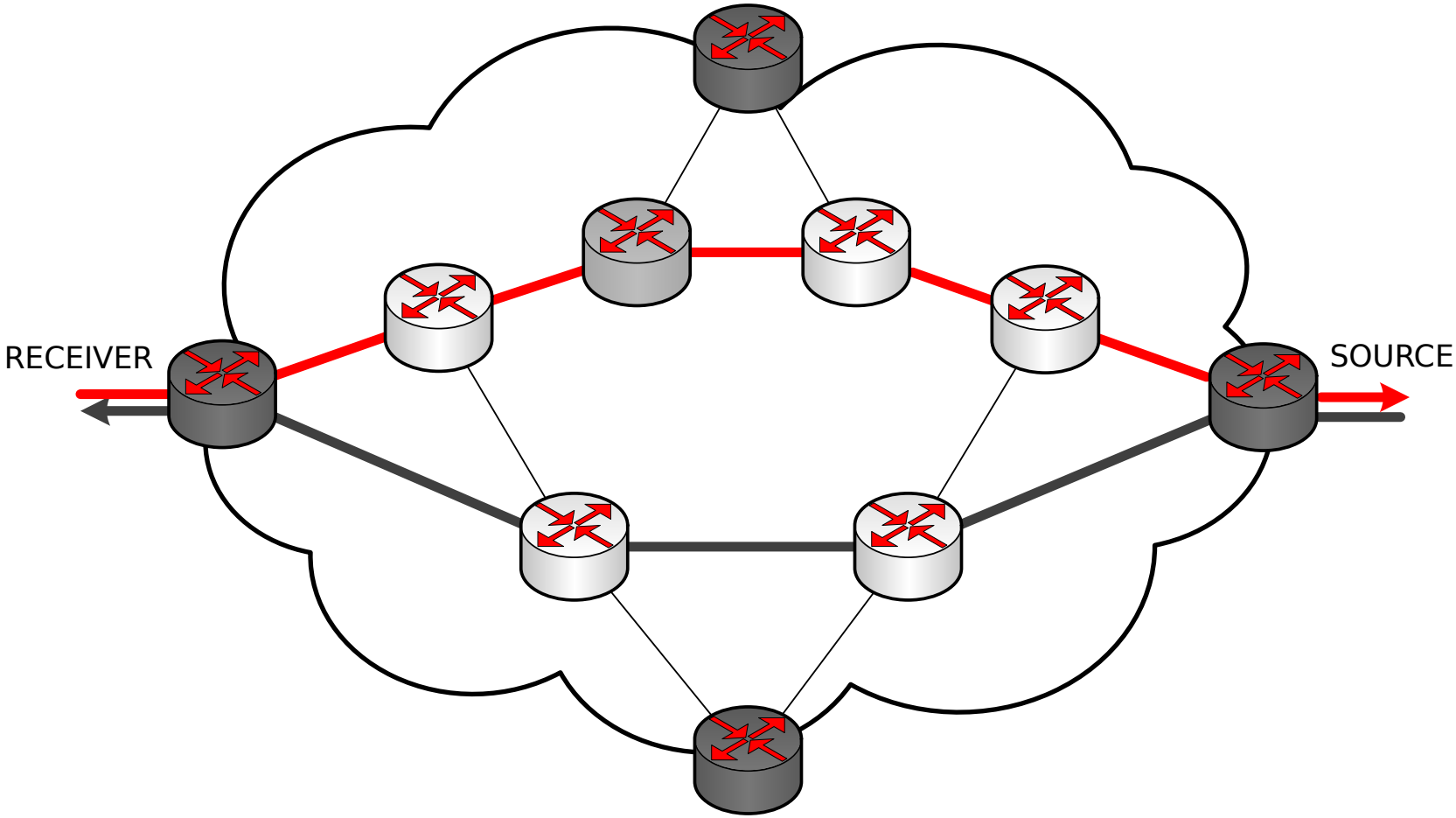
- We use unitary link weights to calculate path costs and $K_{MAX} = 0.3$
- $C_{ASYMM} = 3$, $C_{MCAST} = 4$, $C_{MAX} = 4$
- $C = 0.25 < K_{MAX} = 0.3$: multicast is used

Content routing - Asymmetric-Multicast Hybrid



- We use unitary link weights to calculate path costs and $K_{MAX} = 0.3$
- $C_{ASYMM} = 3$, $C_{MCAST} = 5$, $C_{MAX} = 4$
- $C = 0.5 > K_{MAX} = 0.3$: asymmetric is used

Content routing - Asymmetric-Multicast Hybrid



- We use unitary link weights to calculate path costs and $K_{MAX} = 0.3$
- $C_{ASYMM} = 3$, $C_{MCAST} = 5$, $C_{MAX} = 4$
- $C = 0.5 > K_{MAX} = 0.3$: asymmetric is used

Performance evaluation

Methodology

- Evaluation was carried out using using our *Icarus* simulator. We made all code required to reproduce this paper's results publicly available[2]
- We Investigate the performance of the proposed schemes in terms of *cache-hit ratio* and *link load* and analyse their sensitivity against:
  - *cache to content population ratio (C)*: 0.04% - 5%
  - content popularity skewness (α): 0.6 - 1.1
- Real network topologies:
  - GEANT: European academic network
  - GARR: Italian academic network
  - WIDE: Japanese academic network
  - Tiscali: pan-European commercial ISP

[2] http://www.ee.ucl.ac.uk/~lsaino/software/icarus/

Performance evaluation

Methodology

- Evaluation was carried out using using our *Icarus* simulator. We made all code required to reproduce this paper's results publicly available[2]
- We Investigate the performance of the proposed schemes in terms of *cache-hit ratio* and *link load* and analyse their sensitivity against:
  - *cache to content population ratio (C)*: 0.04% - 5%
  - content popularity skewness (α): 0.6 - 1.1
- Real network topologies:
  - GEANT: European academic network
  - GARR: Italian academic network
  - WIDE: Japanese academic network
  - Tiscali: pan-European commercial ISP

[2] `http://www.ee.ucl.ac.uk/~lsaino/software/icarus/`

Performance evaluation

Methodology

- Evaluation was carried out using using our *Icarus* simulator. We made all code required to reproduce this paper's results publicly available[2]
- We Investigate the performance of the proposed schemes in terms of *cache-hit ratio* and *link load* and analyse their sensitivity against:
  - *cache to content population ratio (C)*: 0.04% - 5%
  - content popularity skewness (α): 0.6 - 1.1
- Real network topologies:
  - GEANT: European academic network
  - GARR: Italian academic network
  - WIDE: Japanese academic network
  - Tiscali: pan-European commercial ISP

[2] `http://www.ee.ucl.ac.uk/~lsaino/software/icarus/`

Performance evaluation

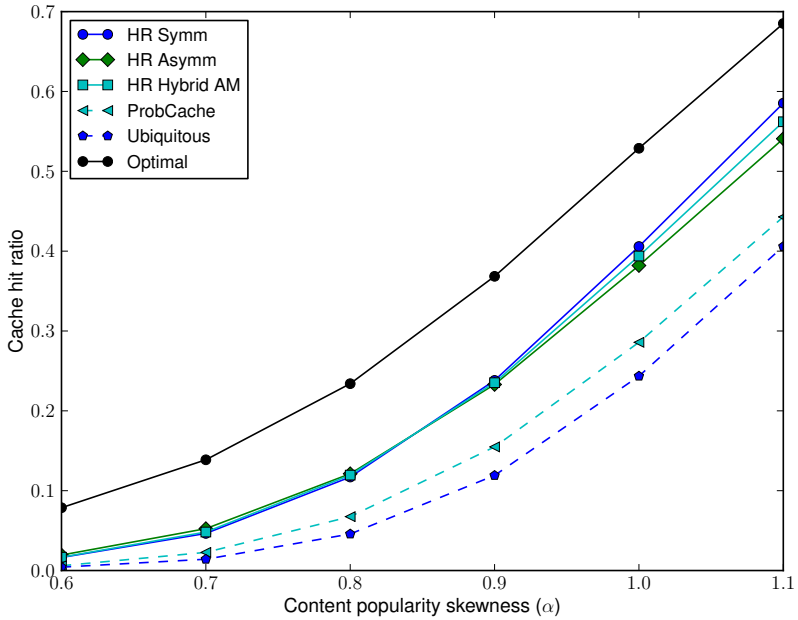
Methodology

- Evaluation was carried out using using our *Icarus* simulator. We made all code required to reproduce this paper's results publicly available[2]
- We Investigate the performance of the proposed schemes in terms of *cache-hit ratio* and *link load* and analyse their sensitivity against:
  - *cache to content population ratio (C)*: 0.04% - 5%
  - content popularity skewness (α): 0.6 - 1.1
- Real network topologies:
  - GEANT: European academic network
  - GARR: Italian academic network
  - WIDE: Japanese academic network
  - Tiscali: pan-European commercial ISP

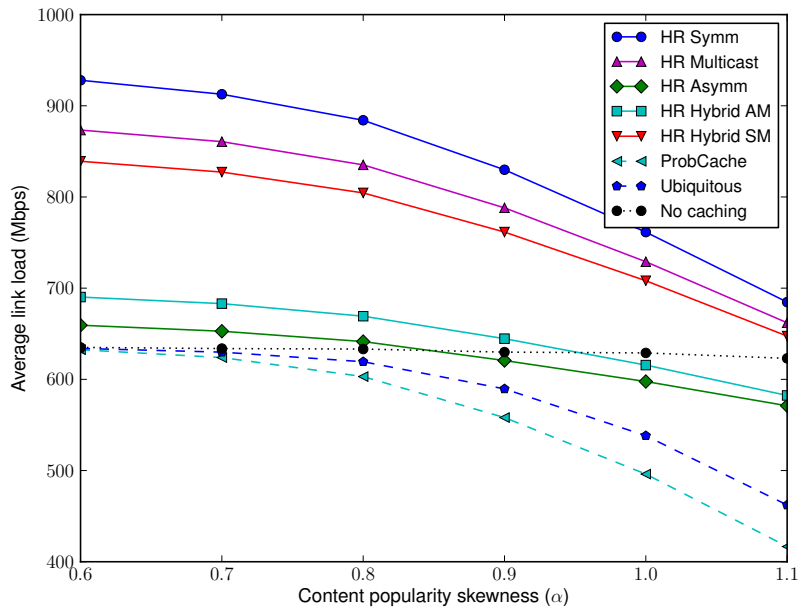
[2] `http://www.ee.ucl.ac.uk/~lsaino/software/icarus/`

Performance evaluation

Cache hits and intradomain link load vs α (GEANT, $C = 0.2\%$)



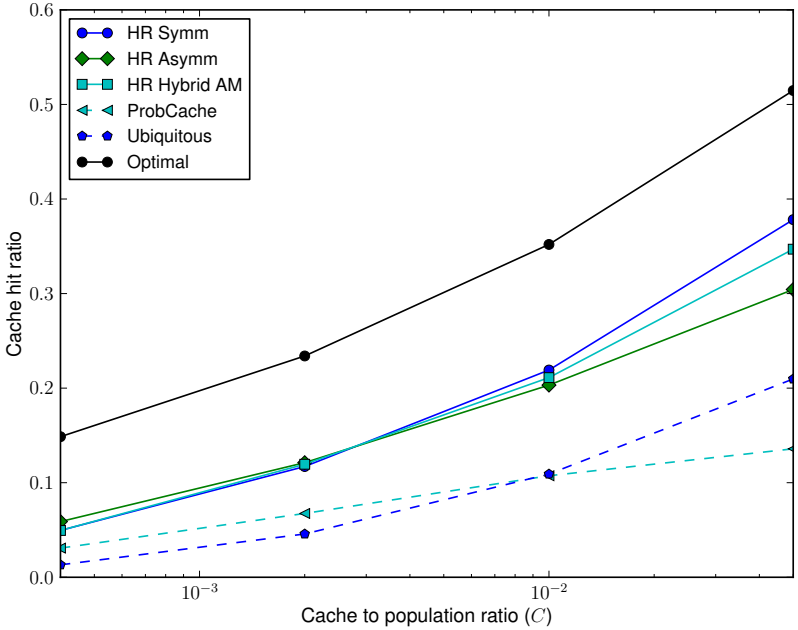
(a) Cache hits



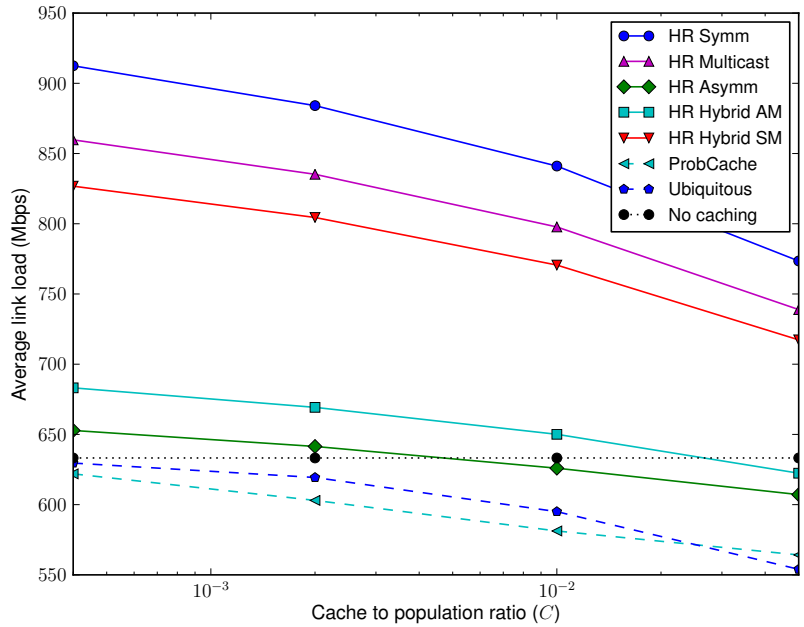
(b) Link load

Performance evaluation

Cache hits and intradomain link load vs C (GEANT, $\alpha = 0.8$)



(c) Cache hits



(d) Link load

Conclusions

- Hash-routing techniques are a viable solution for improving cache hits in a scalable and incrementally deployable manner in an ICN environment.
- The hash-routing schemes proposed provide different trade-offs between intradomain link load and cache hit ratios.
- In particular, asymmetric and symmetric-multicast schemes can provide substantial reduction in interdomain traffic (and average latency) at the cost of a very limited increase in intradomain traffic.

Conclusions

- Hash-routing techniques are a viable solution for improving cache hits in a scalable and incrementally deployable manner in an ICN environment.
- The hash-routing schemes proposed provide different trade-offs between intradomain link load and cache hit ratios.
- In particular, asymmetric and symmetric-multicast schemes can provide substantial reduction in interdomain traffic (and average latency) at the cost of a very limited increase in intradomain traffic.

Conclusions

- Hash-routing techniques are a viable solution for improving cache hits in a scalable and incrementally deployable manner in an ICN environment.
- The hash-routing schemes proposed provide different trade-offs between intradomain link load and cache hit ratios.
- In particular, asymmetric and symmetric-multicast schemes can provide substantial reduction in interdomain traffic (and average latency) at the cost of a very limited increase in intradomain traffic.

Conclusions

- Hash-routing techniques are a viable solution for improving cache hits in a scalable and incrementally deployable manner in an ICN environment.
- The hash-routing schemes proposed provide different trade-offs between intradomain link load and cache hit ratios.
- In particular, asymmetric and symmetric-multicast schemes can provide substantial reduction in interdomain traffic (and average latency) at the cost of a very limited increase in intradomain traffic.