Request Aggregation: The Good, The Bad, and The Ugly

Gaurav Panwar
New Mexico State University
gpanwar@cs.nmsu.edu

Reza Tourani
New Mexico State University
rtourani@cs.nmsu.edu

Satyajayant Misra
New Mexico State University
misra@cs.nmsu.edu

Abderrahmen Mtibaa[†]
New Mexico State University
amtibaa@cs.nmsu.edu

ABSTRACT

Request aggregation is a fundamental feature of named data networking (NDN). This feature aims to improve consumers' quality of experience and reduce network traffic by reducing content retrieval latency and eliminating redundant communication, respectively. However, the negative aspects of request aggregation have not been studied. In this paper, we inspect different facets of request aggregation and introduce one of its harmful behavior, which can create an implicit Denial of Service (iDoS) vulnerability.

CCS CONCEPTS

• **Security and privacy** → *Denial-of-service attacks*; • **Networks** → *Network simulations*; *Wireless access networks*;

KEYWORDS

Information-centric networks, DoS, request aggregation

ACM Reference format:
Gaurav Panwar, Reza Tourani, Satyajayant Misra, and Abderrahmen Mtibaa[†]. 2017. Request Aggregation: The Good, The Bad, and The Ugly. In *Proceedings of ICN '17, Berlin, Germany, September 26–28, 2017,* 2 pages.
https://doi.org/10.1145/3125719.3132110

1 INTRODUCTION

NDN utilizes features such as in-network content caching and request aggregation with the objective of reducing the core network traffic and improving packet delivery characteristics. These features are pertinent in the context of multimedia content being the majority of the Internet traffic and the traffic following a heavy-tailed Zipf popularity distribution [1, 2]. That is, a small proportion of contents make up the majority of the consumer requests. There have been extensive efforts on improving the efficiency and effectiveness of in-network caching [3]. However, request aggregation has received less research attention.

In this paper, we investigate different aspects of request aggregation and briefly discuss its advantages and drawbacks. Further, we quantify the gain (*good*) and loss (*bad*) due to request aggregation at various network entities (e.g., content providers, consumers, and routers). Our main contribution is a discussion of a little investigated behavior of request aggregation, which can create an implicit DoS (*iDoS*) attack on consumers' requests (the *ugly*). To evaluate request aggregation impact, we studied it in three scale-free network topologies (Table 1). The links have *500 Mbps* bandwidth and *1 ms* delay to prevent congestion and packet drops. All consumers run constant bit rate applications with 10 packets-per-second request rate. In this paper, we disable caching, to focus on the request aggregation behavior, and use ndnSIM to run simulations for 300 seconds under two scenarios: enabled request aggregation and disabled request aggregation.

Table 1: Network Topologies Specifications.

	Topo. 1	Topo. 2	Topo. 3
Routers	200	400	600
Content Providers	1	1	1
Consumers	40	80	120

2 THE GOOD

On the bright side, request aggregation prevents redundant content transmissions at the network core. This is especially useful in streaming applications in which several consumers tend to request a content roughly concurrently. In such a scenario, a router forwards the first arriving request towards the content source and aggregates the subsequent requests, thus preventing redundant transfer of copies of a data across the network. Consumers' content retrieval latencies also reduce due to aggregation.

Figure 1 compares the reduction of traffic load on the network with and without request aggregation. With request aggregation, the load was as low as 3000 interests, seen across all utilized network links. This is because each link only sees unique interests passing through while other interests for same content are aggregated. Without aggregation, the load on the network links increase with more requests and as the number of consumers increase. This behavior is consistent across the studied network topologies.

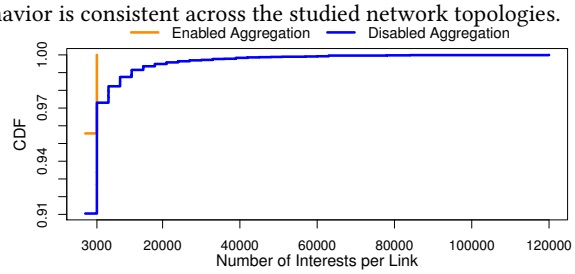


Figure 1: CDF of network traffic (Topology 1).

[†] This work was supported in part by the US National Science Foundation Grants 1345232, 1719342, and 1248109. The information reported here does not reflect the position or the policy of the federal government.

ICN '17, September 26–28, 2017, Berlin, Germany

ACM ISBN 978-1-4503-5122-5/17/09…$15.00
https://doi.org/10.1145/3125719.3132110

3 THE BAD

One downside of request aggregation is that it precludes obtaining all consumers' preferences. In today's IP network, the one-to-one connection between the consumer and the providers (e.g., Netflix, Amazon) allows the providers to collect per-user statistical information. In ICN, aggregation of requests will prevent providers from obtaining all users' statistics, which is important in today's recommendation era where consumers' preferences is important for better customer service and also business outcomes. In [4],

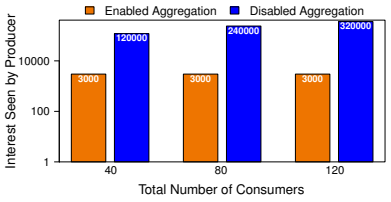


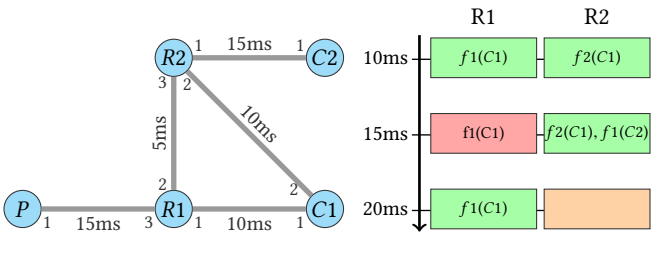
Figure 2: Provider perception of the requested content.

Tourani *et al*. presented this concern and grouped the potential customer preference gathering approaches into manifest-free and manifest-based categories. They suggested a manifest-based mechanism in conjunction with ISPs' cooperation as the best solution. As illustrated in Figure 2, without request aggregation, the content providers receives all consumers' requests, while with aggregation only 3000 are received–a potential loss in business intelligence.

4 THE UGLY

In this section, we introduce the new *iDoS* attack. Other forms of DoS attacks against NDN have been extensively discussed [5]. An iDoS attack exploits the combination of NDN's request aggregation, loop prevention features, and multicast forwarding features. Figure 3(a) illustrates a network containing a provider *(P)*, two routers *(R1* and *R2)*, a multicast consumer *(C1)*, and a unicast consumer *(C2)*. As shown in Figure 3(b), at time 10 ms, *R1* and *R2* receive the same request from *C1* on faces *f1* and *f2* respectively. They insert the request in their PITs and forward it towards *P*. At 15 ms, *R2* receives a new request from *C2* for the same content on face *f1* and aggregates it with the existing PIT entry. At the same time, *R1* receives on face *f2* the same request it received at 10 ms (the request from *C1–R2*).

Due to redundant name and nonce values, *R1* detects a duplicate request (the red shaded box in Fig. 3(b)), drops this request and sends a "duplicate" negative acknowledgment (*NACK*) to its downstream node on face *f2*. At 20 ms, *R2* receives the *NACK* and removes the corresponding PIT entry (the orange shaded box in Fig. 3). Eventually, Consumer *C1* receives the requested content at 50 ms



(a) Network topology (b) Routers' PITs

Figure 3: The effect of request aggregation in the presence of a multicasting consumer.

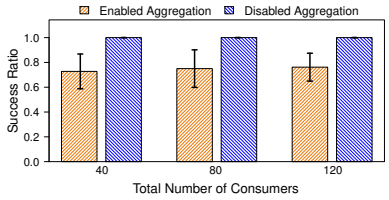


Figure 4: Content Retrieval success rate.

while consumer *C2* times-out on its request despite the existence of the content. This example illustrates an *iDoS* attack where a multicast consumer interrupts a unicast consumer's service.

In general, an *iDoS* attack can happen when: *(i)* there is at least one multicast consumer that shares an upstream router(s) with unicast consumers. *(ii)* Consumers request the same content object, which causes aggregation. *(iii)* Consumers' requests are aggregated on a PIT entry generated by a multicast consumer's request. *(iv)* The request paths converge in the network at an upstream router.

To quantify the negative impact of the *iDoS*, we compare (Figure 4), consumers' content retrieval success rates with and without aggregation; 50% of the consumers in each topology are multicasting consumers with three faces. When request aggregation is enabled, consumers successfully receive 72% of the requested content representing about 28% decrease of benign consumers' rates.

Potential Solutions: Here we propose some solutions to iDoS, which complement those discussed in [6]. *(i)* Consumers can use unique nonces on all faces when multicasting interests. This would eliminate duplicate *NACK*s in the network due to multicasted interests. The approach's downsides include redundant data delivery to multicasting clients and no control in case a client goes rogue and multicasts interests with same nonce. *(ii)* If a router receives a duplicate *NACK* on one of its faces and it has aggregated interests for the *NACK*ed interest, it can re-transmit another interest (with a different nonce) on the same face to avoid *iDoS*. Here multiple levels of aggregation might lead to multiple retries between routers. To handle this, the duplicate *NACK* packet can be modified to include all the nonces matching the *NACK*ed interest available at router initiating the *NACK*. With this modification, the router receiving the *NACK* can choose the nonce that will not generate a *NACK* thus reducing multiple retries.

5 CONCLUSION

In this paper, we discussed the benefits and shortcomings of request aggregation and quantified its impact. We have also introduced the (*iDoS*) vulnerability, discussed its impact, and proposed some solutions to mitigate it.

REFERENCES

[1] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of IMC*, pages 1–14, 2007.

[2] X. Cheng, J. Liu, and C. Dale. Understanding the characteristics of internet short video sharing: A youtube-based measurement study. *IEEE Transactions on Multimedia*, 15(5):1184–1194, 2013.

[3] G. Zhang, Y. Li, and T. Lin. Caching in information centric networking: A survey. *Computer Networks*, 57(16):3128–3141, 2013.

[4] R. Tourani, S. Misra, and T. Mick. Application-specific secure gathering of consumer preferences and feedback in ICNs. In *Proceedings of the ACM Information-Centric Networking Conference (ICN)*, pages 65–70. ACM, 2016.

[5] R. Tourani, T. Mick, S. Misra, and G. Panwar. Security, privacy, and access control in information-centric networking: A survey. *IEEE COMST (accepted)*, 2017.

[6] J. Shi. Ndn bug report #1966, update 20161027, 2016. https://redmine.named-data.net/issues/1966.