International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3487-3494
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0512

# Comparative Performance Analysis of IDM and Traditional HTTP Flood Defence

*Sreeja Nair M P[1], Preetha Mathew K[2], Mathew Cherian[3]*
*[1]Research Scholar, Cochin university College of Engineering Kuttanad, Alappuzha, Kerala, India.*
*[2]Professor, Cochin university College of Engineering Kuttanad, Alappuzha, Kerala, India.*
*[3]Rtd.Professor, Cochin university College of Engineering Kuttanad, Alappuzha, Kerala, India.*
***Emails:*** *sreejanairmp@gmail.com[1], preethamk@cusat.ac.in[2], mathewch@cusat.ac.in[3]*

## Abstract
*The IDM focuses on both secondary, tertiary defense and includes primary and secondary monitoring within a five-phase process to identify defenses against HTTP GET flooding attacks. In the first phase, primary monitoring, incoming traffic is filtered through IP addresses, ports, protocols and packet types, with a monitor queue to provide active supervision and an overflow function to mitigate the load on the system. In the second phase, secondary monitoring with tokenization, Random Forests facilitate feature extraction to improve the detection angle of precision and recall. Tokenization provides the facility to handle time behavior of the requested packets. The overall effectiveness of this multi-layered defense can be derived from the extent monitoring for behavior, along with any external intrusion detection systems (IDS) and firewalls, contribute to any anomaly detection phase. It continuously improves detection accuracy and response times by learning from historical traffic behaviour and adjusting its defence systems. By taking this proactive stance, IDM improves overall system security and performance while lessening the burden on server infrastructure.*
***Keywords:*** *Token, Impoundment, Defense, packet loss*

## 1. Introduction
Previous research has shown a large gap in handling abrupt traffic spikes on essential web services, underscoring the urgent need for an efficient defence against HTTP GET flooding. Impoundment Defence Mechanism, which is specially designed to lessen congestion on vital websites, was created in response to this difficulty. IDM stands out because it emphasizes preserving fluid data transfer, which provides a strong defence against flooding attacks and ensures quick access for authorized users, particularly in high-priority situations when continuous service is essential. To improve application resilience against ongoing DDoS attacks, our research offers a novel defence method against HTTP GET flooding. Before potential attack requests reach the vulnerable web-server layer, our model Impoundment Defence Mechanism (IDM) is seamlessly incorporated into the reverse proxy architecture to provide intelligent management. Comparing IDM to a reservoir next to a dam, it uses an adaptive reservoir-style approach to control request volumes, much like a water flow controller. This analogy highlights the usefulness of IDM in dynamically controlling and stabilizing application performance. By comparing IDM's strategic management to reservoir water level control, the comparison demonstrates how well its intelligent control mechanisms protect applications against disruptions. In this paper, we present a new defence tactic against HTTP GET flooding that improves the resilience of applications against persistent DDoS attacks. When implemented within a reverse proxy framework, the Impoundment Defence Mechanism (IDM) intelligently screens and manages potentially harmful requests before they reach the vulnerable web server layer. In the same way that a reservoir regulates water flow near a dam, IDM uses an adaptive technique to monitor and control the quantity of incoming traffic. This illustration highlights the practical technique that IDM uses to dynamically adjust network traffic to stabilize and improve application performance. The comparison demonstrates how IDM, like water levels, can be used to proactively manage traffic to strengthen

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3487-3494
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0512

applications against potential disruptions. Aligned with the hydraulic analogy, our approach parallels the engineering methods used in building reservoirs to manage fluctuations in river water levels during extreme conditions. Impoundment play a crucial role in flood control, using sophisticated techniques to effectively regulate downstream water flow. Their design demands careful planning, often involving the construction of dams in valleys or leveraging natural and artificial basins. This precision ensures that impoundments can strategically control water levels, embodying a balance between engineering efficiency and environmental stewardship. Similarly, IDM mirrors these principles, adeptly managing the influx of requests by controlling overflow and ensuring a balanced outflow, thus maintaining system stability and preventing overload. In IDM, tokenization happens in real-time, awarding tokens to valid requests that reach within the allotted window for impoundment access. In the event of flooding, requests containing valid tokens are permitted to move forward to the server, establishing a haven inside the impoundment. This clever approach prioritizes authorized and time-sensitive requests, strengthening the system's resiliency while simultaneously optimizing server resource allocation. Consequently, IDM greatly improves the infrastructure's efficiency and security while making sure that vital resources are used efficiently in situations involving high traffic or attacks. The system uses a comprehensive packet monitoring module to properly analyse all incoming requests before IDM takes over its function in the defence strategy. Important information is gathered by this module, such as request rates, timestamps, packet sizes, and source and destination IP addresses. The successful operation of IDM depends on this thorough pre-processing stage, which makes it possible to identify and mitigate possible hazards early on. Our method provides a flexible and adaptive defence system against DDoS attacks, guaranteeing uninterrupted service availability for authorized users even in the face of sophisticated cyber threats. It is based on the concepts of reservoir dynamics and is bolstered by thorough packet analysis. [16-20]

## 2. Traditional HTTP Defense Mechanisms

- **Rate Limiting**: Rate-limiting ensures equitable resource distribution and system stability by limiting client requests within a predetermined duration, preventing server overload. By reducing disruptions from both planned and accidental surges in requests, these measures aid in the management of large traffic volumes and improve the system's resilience against DDoS attacks.
- **Traffic Filtering:** To identify and stop requests from known malicious IP addresses or patterns connected to DDoS assaults, use firewalls or intrusion prevention systems to filter malicious traffic. By taking these precautions, servers can lessen the effects of security breaches and protect themselves from unwanted access. [1-5]
- **Load Balancing:** Use load balancing strategies to split up incoming traffic among several servers in an even manner. This keeps any one server from getting overloaded and experiencing performance snags. This strategy optimizes resource usage, enhances system reliability, and ensures consistent performance levels, even during periods of high demand or traffic spikes.
- **Content Delivery Network (CDN):** Deploying a CDN caches and serves static content from distributed servers, reducing the strain on the origin server and dispersing traffic to mitigate DDoS attacks.
- **Web Application Firewall (WAF):** Implementing a Web Application Firewall (WAF) is crucial for filtering and monitoring HTTP traffic between a web application and the internet. [11-15]
- **Anycast Routing:** Deploying Anycast routing is essential for distributing incoming traffic across multiple data centers, enhancing resilience against DDoS attacks by dispersing the load across various network locations.

## 3. IDM Experimental Results

The test environments are subjected to flood-induced events as well as regular HTTP traffic during testing.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3487-3494
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0512

Normal conditions result in traffic being generated within predetermined timeframes with moderate packet structures. After that, a series of high-size, time-exceeded packets is added to mimic flood circumstances. When the new algorithm is used, network throughput noticeably increases; this is especially beneficial for reducing HTTP flooding rates. A summary of the data flow, *df*, across the impoundment is as follows:

$$df = R.S \quad (1)$$

Here, R denotes the response rate within the timeframe of T1, and S represents the maximum storage capacity of the server. The calculation for determining the maximum bandwidth is as follows:

$$throughput \leq \frac{R(Wn)}{rrt} \quad (2)$$

The mean frequency of requests made within a certain period is represented by the average request rate or ***rtt.*** By employing a impoundment method, packets are guaranteed to persist until the client releases end, which eliminates server idle times. As the following formula shows, this leads to a twofold improvement in throughput over the conventional case:

$$throughput = df + (\Delta S/\Delta T)rtt \quad (3)$$

In this case, the packet loss ratio is very low. The packet loss ratio ($Plr$), which measures packet loss, and expresses the percentage of lost packets over transmitted packets. When flooding occurs and packet counts are above certain thresholds, the amount of overflow is directly proportional to the packet loss ratio. The following formula is used to determine the packet loss ratio (***plr***):

$$\text{Packet loss ratio,} Plr = K\frac{\sum_0^n O(n)}{t} \quad (4)$$

K is a constant. Notably, in the proposed system, there is virtually no chance of losing packets, as all normal requests are processed efficiently.

**Table 1 Performance Measures of The Proposed System**

| Performance measures | Result |
|---|---|
| The average number of requests waiting in the impoundment | 0.00666 |
| Average number of requests in the system | 66.6666 |
| The average time a request spends waiting in the impoundment | 0.00666 |
| The average time a customer spends waiting in the system | 0.06666 |
| Packet Loss Ratio | 0.00016 |
| Throughput | 0.8421 |

Packet loss in HTTP floods is very bad, especially in last-minute rushes. For websites that are focused on jobs or universities, where prompt request processing is essential, this presents a serious problem. To prevent packet loss, the impoundment concept makes sure that all requests are handled within the allotted period. Table 1 shows Performance Measures of The Proposed System [6-10]

## 4. Comparison to Traditional Systems

The comparison of the IDM method with conventional methods is explained in this section. Using only conventional HTTP defence techniques can expose systems to exploitation as cyber threats continue to grow in complexity and diversity. Attackers continuously create new methods to get beyond defences that are in place by taking advantage of flaws in apps and systems. Furthermore, new vulnerabilities are introduced by the quick speed of technology innovation, which may make it difficult for established ways to adequately address them. To effectively safeguard their digital assets and keep ahead of emerging threats, organizations must take a proactive approach to security, regularly evaluating and improving their defence strategies with cutting-edge technologies like threat hunting, machine learning, and behavioral analytics. Table 2, Table 3, Table 4, Table 5, Table 6, Table 7

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3487-3494
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0512

### 4.1 IDM Vs Rate Limiting

**Table 2 Comparison with Rate Limiting**

| Parameters | Rate Limiting | IDM |
|---|---|---|
| User Experience | Too restrictive rate limiting can negatively impact legitimate users | All legitimate users get a chance to use the server |
| Scalability | Not sure to maintain scalability | Ensure scalability due to impoundment pack formation and grow or shrink property |
| False positives | Minimizing false positives, legitimate users are mistakenly restricted | Minimizing the restriction to the legitimate users |

### 4.2 IDM Vs Traffic Filtering

**Table 3 Comparison with Traffic Filtering**

| Parameters | Traffic filtering | IDM |
|---|---|---|
| False Positives | Incorrectly identify legitimate traffic as malicious, leading to the blocking of valid requests. | Second-level filtering by random forest identifies exact requests |
| False Negatives | Sometimes system may fail to identify malicious traffic | Second-level filtering by random forest identifies malicious requests |
| Hardware cost | High | Low |

### 4.3 IDM vs. Load Balancing

**Table 4 Comparison with Load Balancing**

| Parameters | Load balancing | IDM |
|---|---|---|
| Complexity and Cost: | More hardware required | Not much required |
| Session Management Challenges: | Managing and maintaining session state across multiple servers can be challenging. | Only one server |
| Single point failure: | If the load balancer goes down, it can disrupt the entire system | Sometimes disrupts the entire system |
| Learning and Configuration Overhead: | Require a good understanding of the application and traffic patterns. Improper configurations may result in suboptimal performance. | No need to understand traffic patterns. Identify malicious or flooding |

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3487-3494
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0512

| | | |
|---|---|---|
| Difficulty Handling Long-Lived Connections: | Those used in some streaming applications can be challenging | No idea about load balancing |
| Security Challenges: | Load balancing may introduce security challenges, especially if not properly configured. | Security maintains |

### 4.4 IDM Vs CDN

**Table 5 Comparison with Load Balancing**

| Parameters | CDN | IDM |
|---|---|---|
| Cost | High | Low |
| Complexity And Configuration | More complex configuration needed | Not much complex |
| Caching Issues | Cached content may become outdated if not managed properly. Cache-control | Caching not needed |
| Dynamic Content Handling | Not effective | Effective |

### 4.5 IDM Vs WAF

**Table 6 Comparison with WAF**

| Parameters | WAF | IDM |
|---|---|---|
| Cost: | high | low |
| False Positives: | blocking legitimate traffic that is mistakenly identified as malicious | Second-level monitoring identifies malicious packets |
| False Negatives: | Allowing malicious traffic to pass through undetected. | Second-level monitoring identifies malicious packets |
| Complexity and Configuration: | high | less |
| Resource Requirements: | Organizations need to ensure that their infrastructure can handle the additional load imposed by the WAF without affecting application performance. | Impoundment mechanism can handle overload |
| Scalability Challenges | Difficult | Easy |
| Costs | High | Low |
| Potential for Over blocking | In an attempt to block malicious traffic, WAFs may over block, preventing legitimate users from accessing certain features or content. | Legitimate users are accessible |

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3487-3494
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0512

### 4.6 IDM Vs Anycast Routing

**Table 7 Comparison with Anycast Routing**

| Parameters | Anycast Routing | IDM |
|---|---|---|
| Limited Control Over Routing Decisions | Organizations have limited control over which specific server a user is directed to | No distribution of routing data |
| Routing Convergence Time | Find next location of the router is a big task | No issue |
| Increased Network Complexity | High | Low |
| Difficulty in Debugging and Troubleshooting | Difficult to debug | Easy |

## 5. Results And Discussion

In this study, an experimental investigation is conducted to maximize packet forwarding and storage functionality by utilizing the impoundment concept in network architecture. By controlling the flow of incoming requests and efficiently handling surges above the server's processing capabilities, the impoundment acts as a tactical buffer. A Markov Model governs the dynamic interplay between impoundment inflow and outflow, providing a methodical explanation of the system's behaviour. Notably, excess requests simply move to the impoundment for storage when the cache memory threshold is reached. Discrete intervals are used for packet analysis and packet rejection results from failed verifications. One important performance indicator that shows how well requests are processed is the Success Ratio. All things considered, this experimental approach illuminates the subtle coordination of impoundment dynamics in network design, offering important new perspectives on packet flow and storage management optimization.

**Table 8 Token Requests, Accepted, Token**

| Trial# | TimeWin | Totel Rquest | Accepted | Token Issued | Rejected | Token Issue Ratio |
|---|---|---|---|---|---|---|
| 1 | 5 | 2345 | 2000 | 300 | 45 | 0.358851675 |
| 2 | 8 | 3420 | 3200 | 200 | 20 | 0.23923445 |
| 3 | 10 | 4495 | 4400 | 36 | 59 | 0.043062201 |
| 4 | 12 | 5570 | 5600 | 100 | 70 | 0.119617225 |
| 5 | 15 | 6845 | 6200 | 200 | 445 | 0.23923445 |

The table 7 depicts the process of tokenization. The request who are coming within the time frame are allotted tokens, then it will be processed when the server is free. Within the framework of the IDM system, server utilization (Fig.1) is always at its highest, setting it apart from other methods. This persistent increase in server activity highlights how resource-efficient the strategy is, which in turn increases hardware investment returns. The continuous high utilization rates indicate a well-managed workload, ensuring smooth operations free from bottlenecks and performance lapses. This performance resilience highlights how well the scheme distributes resources, enabling the server to skilfully meet user requests while maintaining optimal performance levels.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3487-3494
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0512

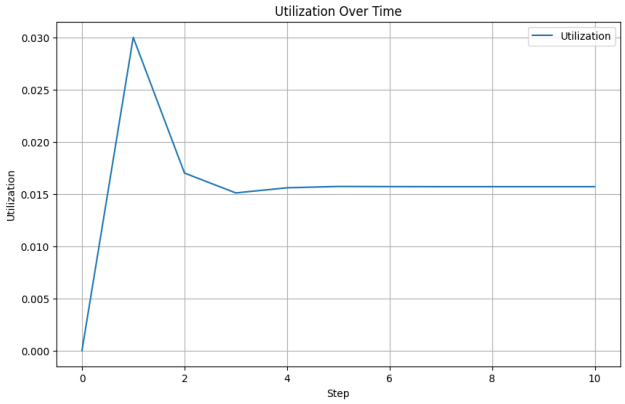


**Figure 1 Server Utilization**

## Conclusion

Cyber systems become exposed when application layer flooding attacks are unsuccessfully resisted by traditional DDoS defence techniques. These approaches rely on methods that can result in false positives and excessive resource use, such as rate restriction or IP blocking. Additionally, they can be unable to deal with asymmetric traffic patterns or adjust to changing attack strategies, making them vulnerable to advanced DDoS attacks. The absence of creative ideas in application layer DDoS research is impeding advancement and system security. This work presents the Tokenised Impoundment Based Defence (TIBD), which monitors high HTTP traffic without overloading resources. To overcome issues like request dropping, TIBD treats users as nodes making requests to time-based servers. Subsequent investigations may examine Data Mining techniques to enhance packet classification. A big step in fending off HTTP flooding attacks and adjusting to the ever-changing DDoS complexity is TIBD.

## Acknowledgment

Sincere gratitude to all who gave support and guidance in this work

## REFERENCES

[1]. J.Mertz and I. Nunes, “A qualitative study of application-level caching,” in JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015

[2]. J.B Pernabas, S.F Fidele, K.K Vaithinathan, “Enhancing greedy web proxy caching using weighted 0Random Indexing based Data Mining Classifier,” in Egyptian Informatics Journal 20 (2019) 117–130

[3]. S Kandula, D Katabi, M Jacob, A Berger, “Botz-4-sale: surviving organized DDoS attacks that mimic flash crowds”, Proc. Of Symposium on Networked Systems Design and Implementation (NSDI), Boston, May 2005.

[4]. Mudhakar Srivatsa, Aun Iyengar, and Jian Yin, “Mitigating Application-Level Denial of Service Attacks on Web Servers: A Client-Transparent Approach” ACM Transactions on the Web, Vol. 2, No. 3, Article 15, July 2008.

[5]. Jie Yu, Chengfang Fang†, Liming Lu, Zhoujun Li, “A Lightweight mechanism to Mitigate Application Layer DDoS Attacks”, Proc. Of Info scale 2009, LNICST 18, pp. 175191, 2009

[6]. Amit Praseed and P. Santhi Thilagam, “DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications”, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 21, NO. 1, FIRST QUARTER 2019.

[7]. Mori and J. Malik, “Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA,” in Proc. IEEE Computer. Soc. Conf. Computer. Vis. Pattern Recognition., vol. 1. Madison, WI, USA, 2003, pp. I–I.

[8]. J. Yan and A. S. E Ahmad, “Breaking visual captchas with naive pattern recognition algorithms,” in Proc. IEEE 23rd Annu. Computer. Security Appl. Conf. (ACSAC), Miami Beach, FL, USA, 2007, pp. 279–291.

[9]. M. Jensen, N. Gruschka, and R. Herkenhöner, “A survey of attacks on Web services,” Computer. Sci. Res. Develop., vol. 24, no. 4, pp. 185–197,2009.

[10]. L. Jimenez, M. Solera, M. Toril, C. Giljon, and P. Casas, “Content Matters: Clustering Web Pages for Analysis With web lust,” in IEEE ACCESS, Vol.9,2021

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3487-3494
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0512

[11]. M.Indana, Zulfa, H.Rudi and A.E. Permanasari., "Caching strategy for web application – a systematic literature review," in International Journal of Web Information Systems Vol. 16 No. 5, 2020
[12]. L.Yang, C. Chi, C. Pan and Y.Q. Nicole, "An intelligent caching and replacement strategy based on cache profit model for space-ground integrated network," Hindawi Mobile Information Systems Volume 2021, Article ID 7844929
[13]. J.C. Mogul, "Squeezing More Bits Out of HTTP Caches," IEEE Network May/June 2000
[14]. X. Hu, X. Wang, L. Zhou, Y. Luo, C. Ding and Z. Wang, "Kinetic Modeling of Data Eviction in Cache," 2016 USENIX Annual Technical Conference 351
[15]. H.A,Mirvaziri, "A New Method to Reduce the Effects of Http-Get Flood Attack,". Future Computing and Informatics Journal (2017); doi: 10.1016/j.fcij.2017.07.003
[16]. Wang et al. "Sky Shield: A Sketch-Based Defense System Against Application Layer DDoS Attacks", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 13, NO. 3, MARCH 2018
[17]. Tetsuya Hirakawa, Kanayo Ogura, Bhed Bahadur Bista, Toyoo Takata, "A Defense Method against Distributed Slow HTTP dos Attack",19th International Conference on Network-Based Information Systems,2016.
[18]. Yuri G Dantas, Vivek Nigam, Iguatemi E Fonseca, "A selective Defense for Application Layer Attack",2014 IEEE Joint Intelligence and Security Informatics Conference.
[19]. Karuna S. Bhosale, Maria Nenova, Georgi Iliev, "The Distributed Denial of Service Attacks (DDoS) Prevention Mechanisms on Application Layer", TELKIS 2017.
[20]. Supranamaya Ranjan, Mustafa Uysal, Edward Knightly, "DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 17, NO. 1, FEBRUARY 2009.