cPacket Networks, Inc.

Microbursts in Data Centers

Impact, Detection and Measurement

White Paper

Table of Contents

Introduction *2*

Defining a Burst *2*

Bursty vs. Constant Data Traffic Patterns **2**

Defining a Microburst **3**

What Are the Causes of Microbursts? *4*

Application Data Patterns **5**

Traffic Patterns of Network Traffic Generated by Distributed Training of DNNs 5

Fan-In Traffic and the Impact of Middleboxes **5**

Impact of Microbursts and Middlebox Buffers on Network Performance 7

TCP Methods for Congestion Control and Reliable Data Transfer **9**

TCP Reno Congestion Control 10

ACK Compression 11

Batching Schemes **12**

Understanding Microbursts *12*

Understanding Microburst Effects on Networks *15*

Identifying and Measuring Microbursts *15*

Detection and Network Monitoring Limitations **16**

cPacket's Real-Time Microburst Detection Solution *18*

cPacket's High Resolution Counters **19**

cPacket's cBurst **20**

Conclusion *22*

References *23*

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

Introduction

Key sectors such as finance, healthcare, defense, and research are dependent on reliable and high-performing networks characterized by the need for zero downtime, continuous adaptation to change, and ongoing risk mitigation. Network performance may be impacted due to various issues, including hardware failures, configuration errors, and specific patterns of data delivery. One of these issues is known as a data *burst* - a pattern that can adversely affect the performance of a network by overwhelming packet buffers and causing packet loss, spurious retransmissions, and increased latency.

Microbursts are a type of burst defined by their especially brief duration. Despite their short duration, microbursts can cause all the damage of longer bursts while being significantly harder to detect. This combination makes network debugging and root cause analysis difficult, extends mean time to resolution (MTTR), and can result in significant downstream business impact.

Defining a Burst

A burst is “a sudden increase in something, especially for a short period” [1] (Cambridge Dictionary, n.d., https://dictionary.cambridge.org/dictionary/english/burst). Bursts are phenomena that can be found all around us and across different disciplines. Some examples of burst phenomena are earthquakes, solar flares, volcanic eruptions, flash floods, firing neurons, traffic jams, cyberattacks and data network traffic. In the specific case of data network traffic, a traffic burst can be defined as a “condition in which network activity rises suddenly and for a short period of time” [2] (The Network Encyclopedia, n.d., http://www.thenetworkencyclopedia.com/entry/burst/).

Bursty vs. Constant Data Traffic Patterns

Bursty and constant traffic patterns exhibit distinct characteristics, as illustrated in Figure 1. The two upper graphs compare the number of packets over time for bursty and constant traffic. Bursty traffic (left) shows intermittent spikes in packet count, while constant traffic (right) maintains a steady packet flow. The lower graphs compare throughput over time for bursty and constant traffic. Bursty traffic (left) results in fluctuating throughput levels, whereas constant traffic (right) maintains a more stable throughput. Understanding these patterns is important for network designers and managers to optimize network performance and resource allocation effectively.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

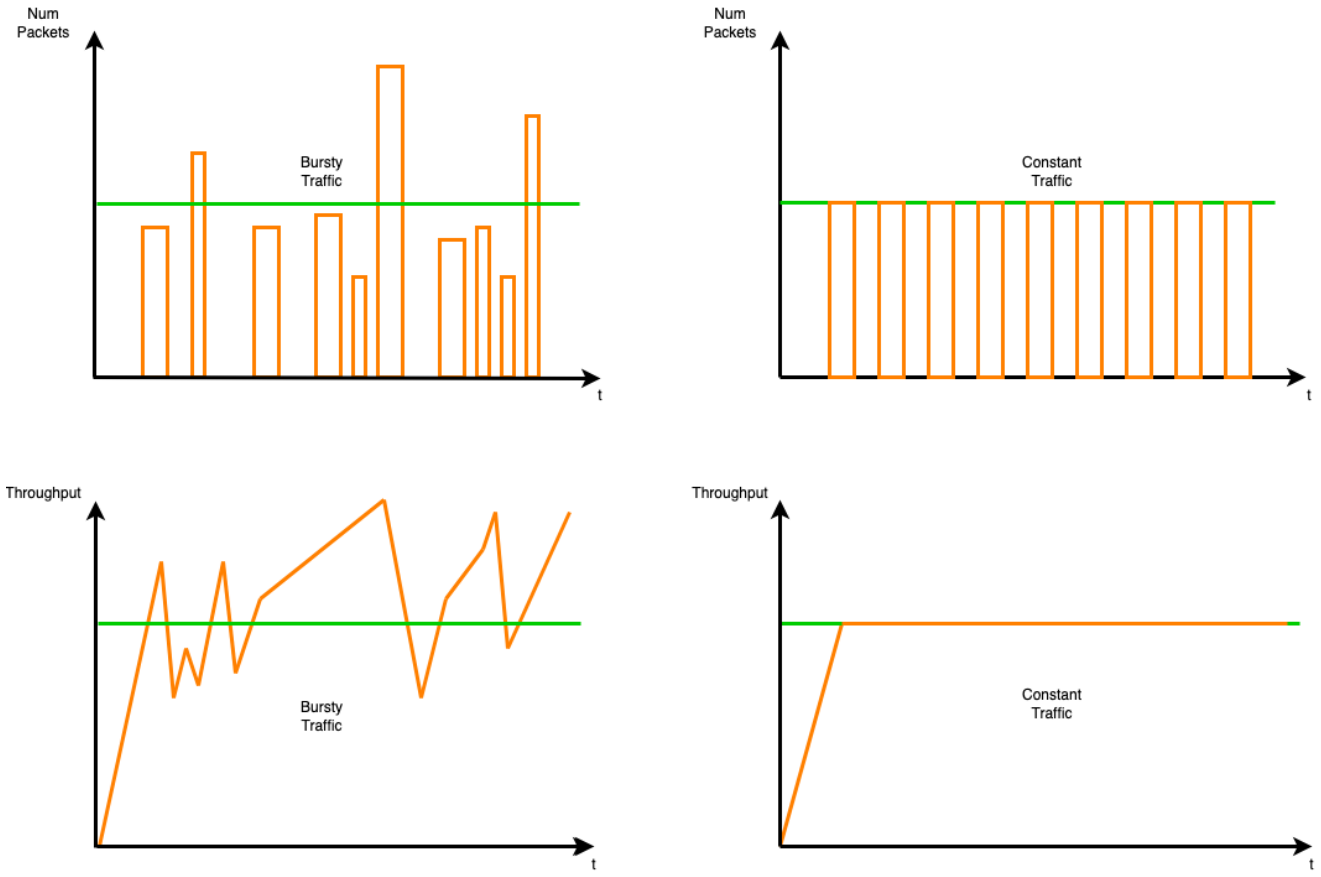


Figure 1. Bursty vs. Constant Data Traffic Patterns

Defining a Microburst

The definition of a *microburst* is like the definition of a burst above except the period when the burst occurs is very short, in the order of milliseconds (e.g., 1 to few hundreds of milliseconds). Despite being only brief spikes in network traffic, microbursts are capable of severely impacting network performance, particularly in cases like financial markets where steady data flow is vital. Figure 2 below shows the significant data rate difference between the average and microburst data rates.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

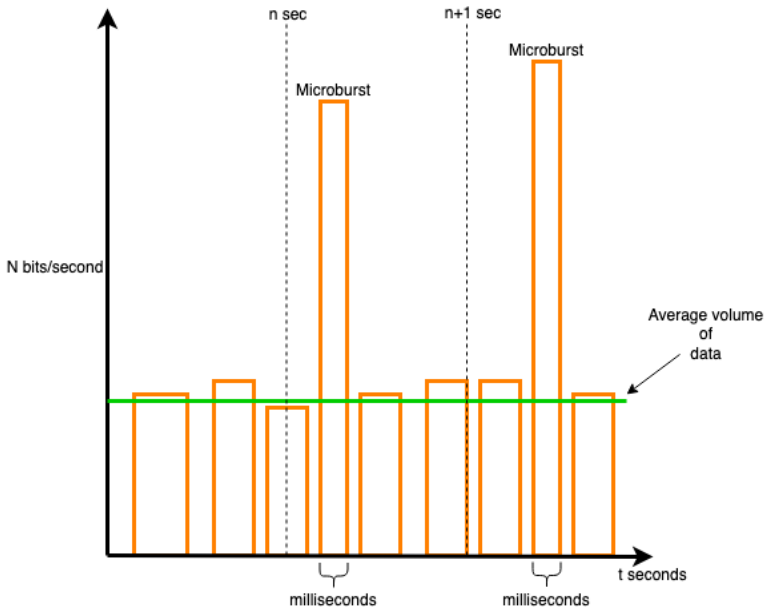


Figure 2. Microbursts

What Are the Causes of Microbursts?

A wide variety of elements can trigger data traffic microbursts. The works of Kapoor et al. [3] and Shan et al. [4] identify several factors, which may contribute to microburst generation, including:

- Application data patterns - sending large chunks of data, sending data in smaller increments, etc. For example:
    - Traffic from distributed machine learning (e.g., training of distributed deep neural networks).
- Fan-in traffic - multiple concurrent sources communicate with a single destination.
- TCP methods for congestion control and reliable data transfer, for example:
    - Slow-start, congestion avoidance, ACKs.
- Fine-tuning of OS-level TCP stack parameters to optimize network performance.
- Operating System (OS)/Network Interface Controller (NIC) performance optimization techniques to reduce CPU overhead, for example:
    - Batching schemes.

The presence of microbursts can therefore be a result of one or various factors. A brief description of a subset of the above potential sources of microbursts is provided next.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

Application Data Patterns

Application data patterns play a significant role in the generation of microbursts. The characteristics of the data transmitted, such as size, frequency and burstiness, directly influence the occurrence and severity of microbursts. Applications sending large amounts of data in short bursts, rather than at a steady state, are more likely to contribute to microburst formation. This bursty traffic pattern can be attributed to various factors, such as the nature of the application. For example, deep neural networks (DNNs) are a type of application, which has become increasingly widespread across different industries' data centers and exhibit a large degree of burstiness. The next section briefly describes the characteristics of network traffic generated by distributed training of DNNs.

Traffic Patterns of Network Traffic Generated by Distributed Training of DNNs

The research conducted by Luangsomboon et al. [5] studies the highly bursty nature of traffic generated by distributed ML training workloads. This research focuses on the network traffic generated by DNNs during the training phase. In this phase, network traffic is characterized by sudden and significant traffic spikes (i.e., microbursts). These microbursts lead to increased latency and packet loss and, in general, adversely affect network performance. Some of the key factors identified by this research contributing to generating microbursts are:

- Distributed DNN training generates network traffic with high burstiness over short intervals (sub-5ms).
- DNN training follows an on-off pattern, with lengthy compute periods at worker nodes with few transmissions alternating with periods of bursty transmissions when the workers exchange the gradients computed from the backpropagation phase.
    - Distributed training frameworks can orchestrate transmissions from worker nodes in a single DNN application to avoid multiple workers sending larger bursts simultaneously. However, highly bursty traffic still poses a challenge when multiple DNN applications run concurrently.

Fan-In Traffic and the Impact of Middleboxes

Fan-in traffic, which happens when multiple sources send data to a single destination, can have a significant impact on network performance, particularly in latency applications. The additive nature of fan-in traffic can lead to the generation of microbursts: the aggregation of data from multiple sources at a single destination, namely a middlebox, can result in brief periods of high-volume traffic. RFC 3234, titled "Middleboxes: Taxonomy and Issues" [6] defines a middlebox as follows:

> "A middlebox is defined as any intermediary device performing functions other than the normal, standard functions of an IP router on the datagram path between a source host and destination host."

A wide variety of middleboxes can be found in a network, such as: NATs, firewalls, routers, switches, load balancers, IDSs, proxies, application accelerators, etc.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

Fan-in traffic creates a challenging trade-off between burst tolerance and latency. To manage the bursty nature of fan-in traffic, larger buffer depths are needed at the middlebox. However, increasing the buffer depth also increases latency, since packets spend more time in the buffers before they are processed. This trade-off is particularly critical for latency sensitive applications, such as electronic trading systems, high-performance computing and DNNs. In this case, smaller buffer depths would be preferred to keep latency low; however, it comes at a cost of reduced burst tolerance and leads to packet loss when buffers fill up during the presence of microbursts. Figure 3 describes the fan-in ratio for a middlebox with N sources.

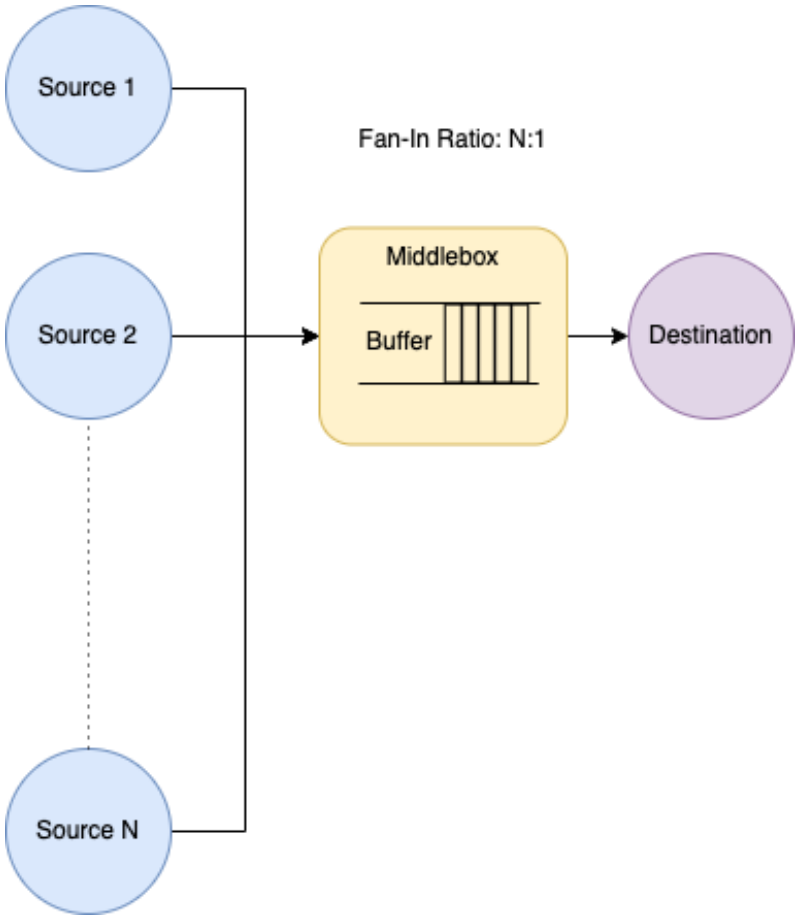


Figure 3. Fan-In Ratio of N:1

We can use the following generic definition of a data center to further describe fan-in as a source of microbursts:

> A data center is a network of clustered computing, application, and storage resources. These resources are interconnected via a complex cabling and middlebox network infrastructure.

Figure 4 illustrates a simplified data center layout, featuring two clusters of server racks. These clusters are interconnected through Top-of-Rack (ToR) switches, which are part of a broader network interconnect infrastructure. To illustrate this, consider three servers—Server1, Server2, and Server3—located in rack N. These servers simultaneously send data through Top-

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

of-Rack (ToR) Switches N and M to Client1, located on rack M. This example results in a fan-in ratio of 3:1.

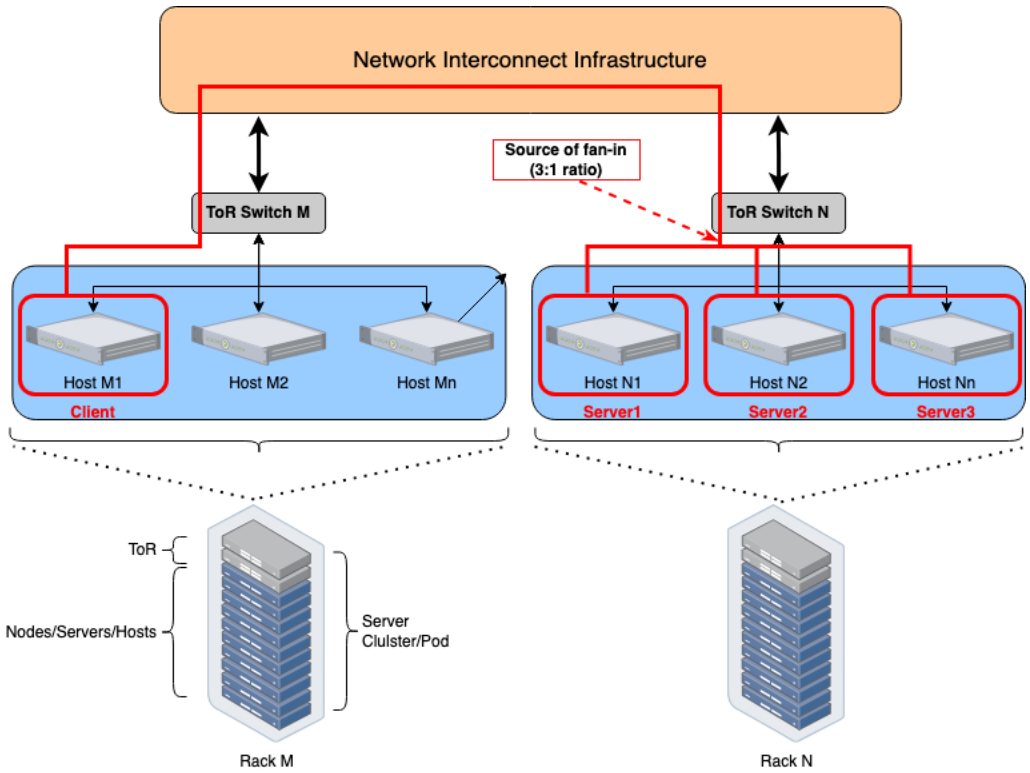


Figure 4. 3:1 Fan-In Ratio Between Two Interconnected Racks on Minimal Spine-Leaf Network

Impact of Microbursts and Middlebox Buffers on Network Performance

When packets from multiple sources arrive simultaneously at a middlebox, such as a switch or router, they are typically placed into a buffer queue. The middlebox processes these packets on a first-in, first-out basis, or according to other policies, such as QoS, traffic shaping, and fair queuing mechanisms, among others. If the rate of incoming packets exceeds the processing capacity of the middlebox, the buffer queue starts to fill up. As the queue decreases, packets experience longer queueing delays, resulting in higher latency. If the buffer becomes full and is unable to take additional packets, the middlebox may start dropping packets, leading to packet loss. This situation is more likely to occur during microburst events, where many packets from multiple sources all arrive at the middlebox in a short time.

Let's define the following conceptual high-level queuing model to describe the relationship between incoming packets and the capacity of the middlebox to process those packets:

Definition of Terms:

- $\lambda(t)$: incoming packet rate (packets/second)
- $\bar{\lambda}$: average rate of incoming packets (packet/second)

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

- μ_b: outgoing packet rate (packets/second)
- $Q(t)$: queue depth at time t (packets)
- $L(t)$: latency experienced by packets at time t (seconds)
- B: maximum buffer size (packets)
- t_s: start of microburst event
- t_e: end of microburst event

When $\bar{\lambda} < \mu_b$ the buffering system is **stable**

When $\bar{\lambda} > \mu_b$ the buffering system is **unstable**

- And $\bar{\lambda} - \mu_b$ is the long term average rate (packets/second)

During a microburst event, occurring within the timeframe defined by: $t_s \leq t \leq t_e$

- When $\lambda(t) > \mu_b$, then
  - $Q(t)$ increases
  - $L(t)$ increases, where $L(t) \approx f\big(Q(t)\big) = \frac{Q(t)}{\mu_b}$ is an increasing function in size of Q
  - **Probability of packet loss increases**
  - $if\ Q(t) = B$ (queue depth reaches buffer size), then
    - **Any incoming packets are dropped**.

Noteworthy Points:

- Packets can still be dropped when $\bar{\lambda} < \mu_b$
- The higher the variability in incoming packet rate $\lambda(t)$, the higher the probability $\lambda(t) > \mu_b$ at some point, resulting in higher probability of packet loss.

The high-level model above provides a useful explanation to better understand the relationship between microbursts, incoming packet rate, buffer processing, latency, and packet loss. The model can be summarized as follows:

- If the rate of incoming packets exceeds the buffer processing capacity, the queue depth increases over time.
- As queue depth increases, packet latency also increases.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

- If the queue depth reaches the maximum buffer size, the buffer will start dropping packets, leading to packet loss.

Figure 5 illustrates what may happen when packets from multiple sources in a hypothetical 10Gbps network are received simultaneously by a middlebox:

- Received packets from all ingress ports are queued in middlebox buffer.
- Once the middlebox buffer is full, further incoming packets are dropped.

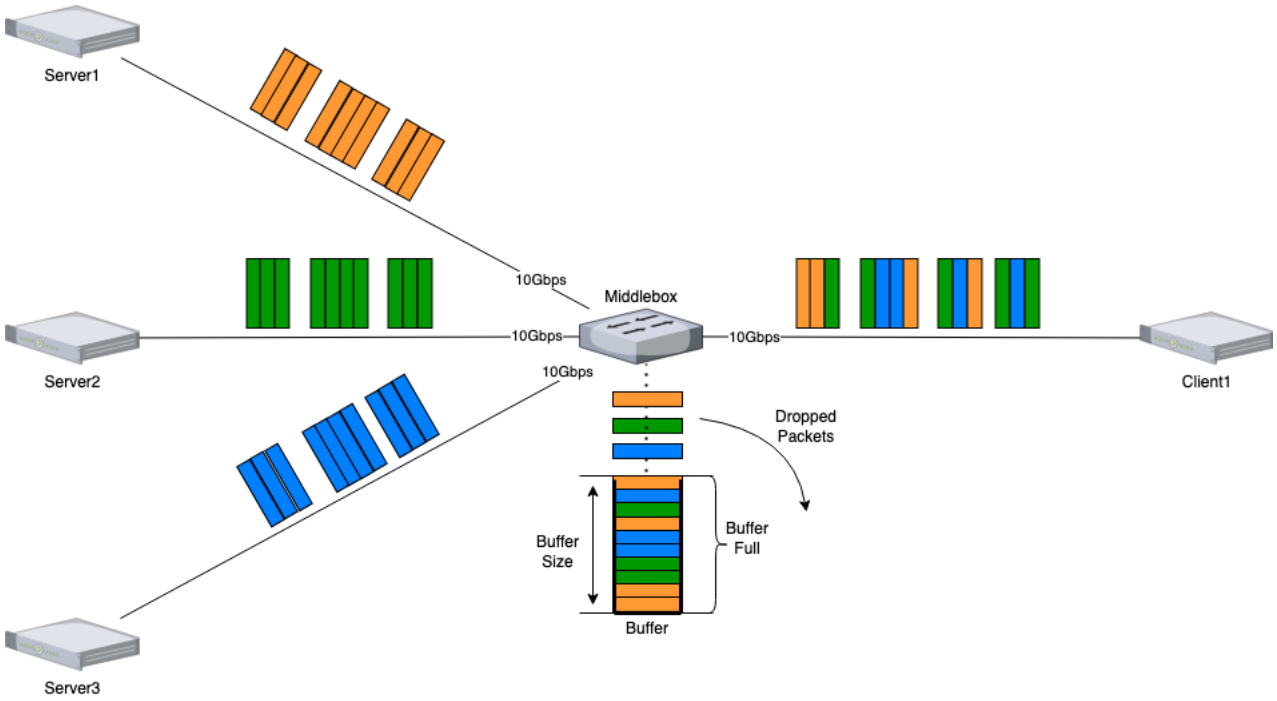


Figure 5. Middlebox Packet Buffer Queuing

Microbursts effectively increase the rate of incoming packets, which impacts buffer processing in middleboxes. When microbursts occur, the rate of incoming packets may exceed the buffer processing capacity and cause a rapid increase in the buffer queue depth. Packet latency increases as the buffer queue grows and packet loss happens when the queue length reaches the maximum buffer size (e.g., buffer full).

TCP Methods for Congestion Control and Reliable Data Transfer

TCP mechanisms were designed to ensure reliable data transfer and ensure fair and efficient use of network resources. These same mechanisms can also inadvertently contribute to generating bursty traffic under certain conditions.

ACKs and retransmissions are examples of mechanisms that provide a reliable data transfer service, while congestion control aims to prevent a sender from overloading the network and ensures the fair and efficient use of network resources.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

The following two sections describe two examples, TCP Reno congestion control and ACK compression, both of which can cause bursty traffic.

TCP Reno Congestion Control

As previously stated, TCP's congestion control mechanisms (e.g., slow start, congestion avoidance) were designed to prevent the sender from overloading the network and adapting to network conditions. When congestion is detected, the congestion window ($cwnd$), which limits the amount of data TCP can send before receiving an ACK, decreases and data is sent at a slower rate. When no congestion is detected, the congestion window increases and data is sent at a faster rate. This rapid rate increase mechanism can inadvertently generate traffic bursts. Note that TCP also uses an advertised receiver window ($rwnd$), which indicates the maximum amount of data the receiver can accept, to limit amount of data the sender can transmit to the receiver before receiving an ACK (i.e. sending window). The sending window is limited by $\min(cwnd, rwnd)$.

Figure 6 below depicts the behavior of a typical TCP slow-start and congestion avoidance pattern. Note that this pattern will vary depending on the TCP variant (e.g., TCP Reno, TCP New Reno, TCP Cubic). The slow start and congestion avoidance pattern depicted on the graph below is representative of TCP Reno and illustrates the dynamics of TCP's congestion control algorithm, showing $cwnd$ (MSS bytes) or throughput and time in terms of RTT. An exponential growth pattern occurs during slow start, transitioning into a more linear or stabilized growth pattern during congestion avoidance. As TCP increases its sending rate during slow start, $cwnd$ grows exponentially until it reaches $ssthresh$. At this point, TCP transitions from exponential to linear increase send rate. In response to congestion indications, such as packet loss or increased RTT (latency), TCP adjusts $ssthresh$ down, prompting a return to slow-start or a more conservative congestion avoidance phase. TCP transmission rate is also influenced by the timeout mechanism known as RTO. If a TCP segment is not acknowledged within the RTO period, TCP retransmits the segment, possibly inferring congestion and, as observed on the graph, adjusting $cwnd$ and $ssthresh$.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

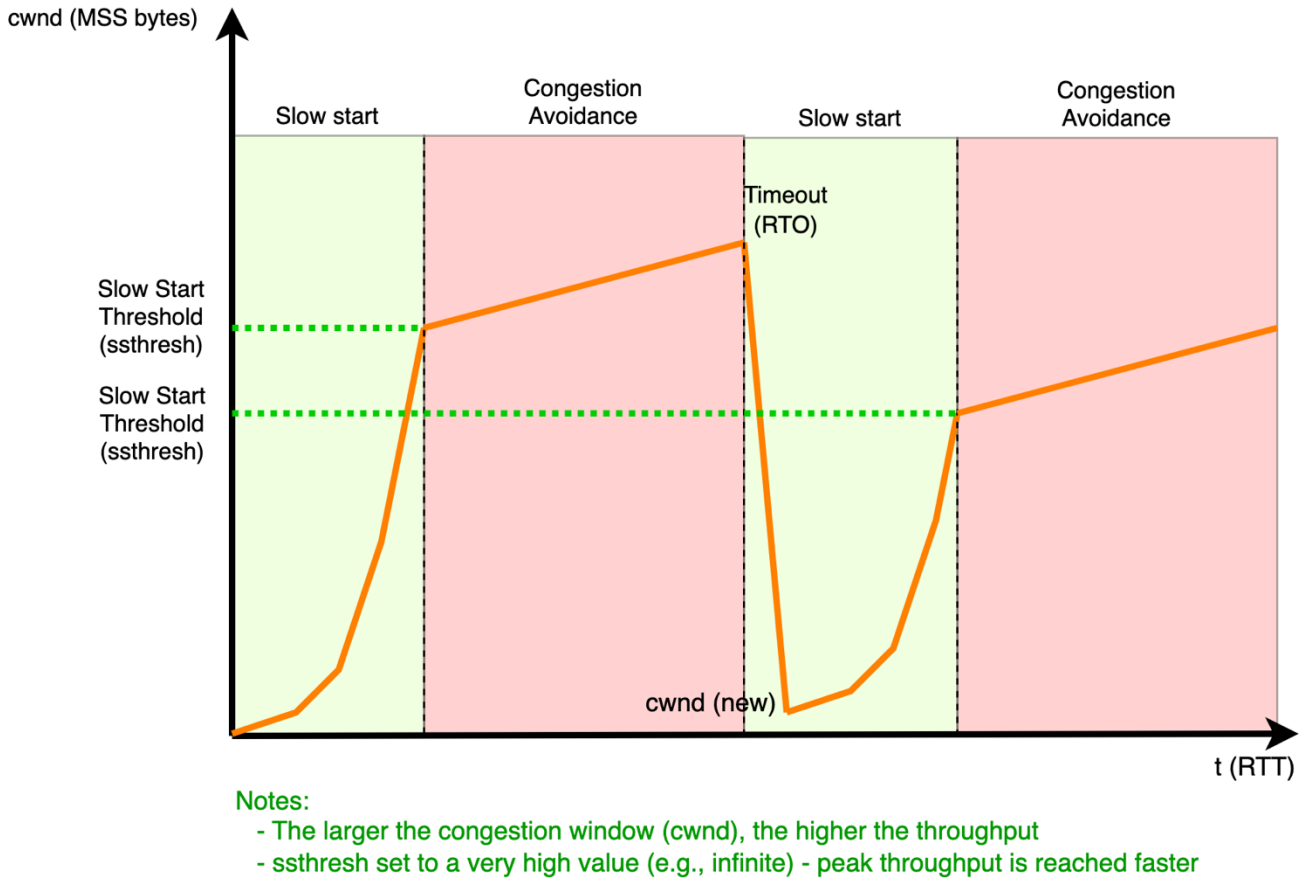


Figure 6. TCP Slow Start and Congestion Avoidance for TCP Reno

ACK Compression

ACK compression is an unintended consequence of using ACKs as one of the mechanisms to ensure reliable data delivery.

ACK compression, as described by RFC 3449, titled "TCP Performance Implications of Network Path Asymmetry" [7], can also contribute to generation of microbursts:

> "*When ACKs are queued behind other traffic for appreciable periods of time, the **burst nature of TCP traffic and self-synchronizing effects can result in an effect known as ACK Compression**, which **reduces the throughput of TCP**. It occurs when a series of ACKs, in one direction are queued behind a burst of other packets (e.g., data packets traveling in the same direction) and become compressed in time. This results in an **intense burst of data packets in the other direction, in response to the burst of compressed ACKs arriving at the server**.*"

The described TCP mechanisms designed to ensure reliability, fairness and efficiency can unintentionally generate microbursts under various network conditions, including congestion, link failures, high latency, and rapidly changing network topologies (e.g., due to dynamic resource allocation, load balancing or failover/redundancy strategies).

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

Batching Schemes

Batching schemes are useful tools in data centers, improving efficiency by combining multiple data packets into single tasks. While the use of batching schemes may reduce CPU load and improve performance, they can also contribute to the generation of microbursts.

According to Shan et al. [8], system batching schemes used in high-speed data centers to reduce CPU utilization can also contribute to generating microbursts. Some of the system batching schemes identified include:

- *Interrupt Coalescing (IC)* - Whereby the sender introduces traffic bursts into the network by delaying when an interrupt is raised by a few microseconds. This scheme allows for processing of multiple packets and ACKs together, thereby introducing microbursts in the network. In GNU/Linux, for example, a burst length can be ~ 3 packets for a 10Gbps network.
- *Large Segment Offload (LSO)* - LSO is a technique on the sender side, which enables the transport layer to send large virtual packets (up to 64MB in GNU/Linux). These packets are segmented into MTU-sized packets before entering the network. LSO effectively enables packet transmission to be deferred until some condition is met instead of when an ACK is received.
- *Receive Side Offload (RSO)* - RSO, as the name indicates, is a scheme on the receiver side which may also introduce microbursts. With this scheme, multiple received packets are merged into a single large virtual packet. Similar to ACK Compression, this results in multiple packets being acknowledged together, producing microbursts. Burst length introduced by this scheme is usually < 5 packets.
- *Jumbo Frames* - Using jumbo frames lets a network send data in larger chunks. This can lead to more network buffer use, longer wait times in queues, and a temporary reduction in available network bandwidth. These factors together can cause sudden bursts of network traffic.

Understanding Microbursts

Mathis, et al. [9] defines an equation, which, although simplified, can be used as a tool to understand the impact of microbursts on data networks. By analyzing how brief spikes in network traffic can cause increased packet loss and latency, we can better understand why TCP throughput decreases.

This equation is defined as a function $f\left(MSS, \frac{1}{RTT}, \frac{1}{\sqrt{p}}, \text{ where p} \sim \text{packet loss}\right)$:

Definition of Terms:

- ***MSS*** (Maximum Segment Size) -- largest amount of data that can be carried in a payload of a single TCP segment. It is the maximum size of payload (data) in a TCP segment, excluding the TP header and any IP or lower-layer headers.
- ***RTT*** (Round Trip Time) - analogous to network latency, and it can be defined as the time it takes for a signal or data packet to travel from source to destination, plus the time it takes for an acknowledgement of receipt from the destination to the source.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

- ***p*** - probability of packet loss in the network.

Where TCP throughput is calculated as:

- $TCPtput \approx \frac{MSS * C}{RTT \sqrt{p}}$, where $C = \sqrt{3/2} \cong 1$ and *p* = packet loss rate
- $TCPtput \approx \frac{MSS}{RTT \sqrt{p}}$

This equation simplifies TCP throughput calculation; unlike other models, it does not consider factors like bandwidth-delay product and network congestion nor the specific impacts of TCP congestion control algorithms in varying network conditions. However, it is sufficient for our illustration purposes.

Let's use the following two examples to illustrate the calculation of $TCPtput$ of a TCP connection as *p* varies and *MSS/RTT* remain constant, which highlight how much $TCPtput$ varies as packet loss increase.

- *MSS* = 1460B
- *RTT* = 80ms
- *p* = $1x10^{-11}$ % → ($1x10^{-13}$)
    a. $TCPtput \approx \frac{1460\ B * 8\ bits}{0.08\ sec * \sqrt{\mathbf{1X10^{-13}}}} \approx 462\ Gbps$
- *p* = $1x10^{-7}$ % → ($1x10^{-9}$)
    a. $TCPtput \approx \frac{1460\ B * 8\ bits}{0.08\ sec * \sqrt{\mathbf{1X10^{-9}}}} \approx 2.3\ Gbps$

The equation shows that TCP throughput decreases as packet loss *p* increases, which in turn highlight how microbursts can directly affect TCP throughput by causing temporary but significant packet loss.

Congestion resulting from microbursts can increase *RTT* as packets queue up, waiting to be processed by overwhelmed middleboxes. The Mathis equation shows that TCP throughput decreases as *RTT* increases, highlighting how microbursts can degrade network performance by delaying packet delivery.

Microbursts highlight the quick and unpredictable shifts in network conditions. The Mathis equation, incorporating packet loss *p* and *RTT*, highlight the responsiveness of TCP throughput in fluctuating network environments.

Figure 7 below illustrates the variation of TCP throughput as a function of *RTT* and *p*, per the Mathis et al., equation.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

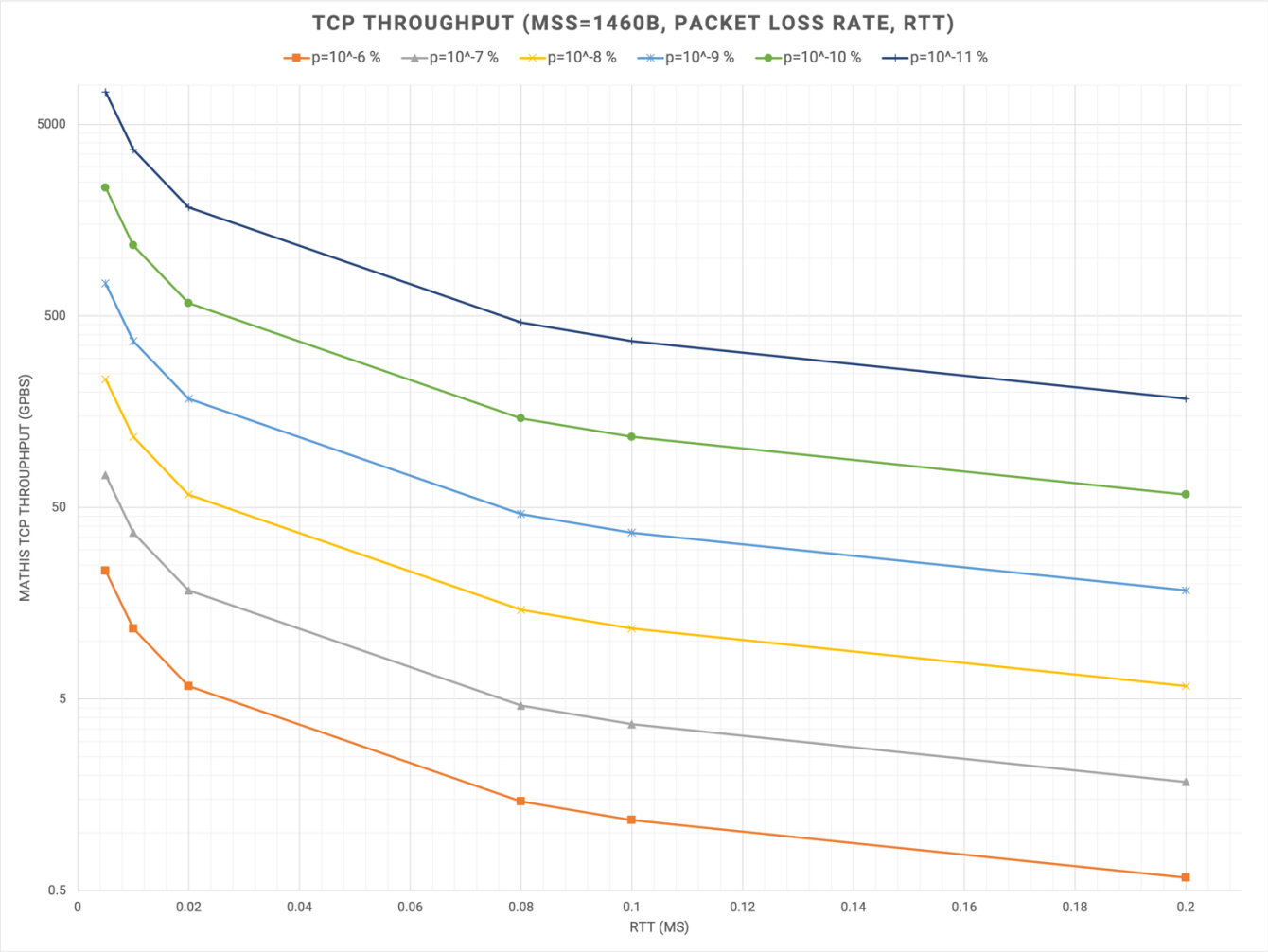


Figure 7. TCP Throughput vs. RTT and Packet Loss Rate (p)

In Figure 7, which plots TCP throughput vs. *RTT*, each curve corresponds to a different packet loss rate, demonstrating the degradation in TCP performance with increasing latency and packet loss. The graphs help in understanding the consequences of microbursts in network performance, summarized as follows:

- Microbursts contribute to increased packet loss and RTT.
- During microburst incidents, network performance is consistent with higher packet loss rate and RTT curves.
- This results in a significant, albeit temporary, reduction in TCP throughput

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

Understanding Microburst Effects on Networks

Microbursts can degrade network performance significantly. Some notable effects of microbursts on a network include:

- Increased Packet Loss - Microbursts can lead to sudden, temporary overflows in network buffers, causing packet drops translating to degradation of quality or service interruptions.
- Increased Latency - Microbursts can cause packets to be queued in middlebox buffers for longer periods, resulting in higher latency and delayed packet delivery. This impacts application response times and reduces overall network performance.
- Impact to Applications - Microbursts can considerably affect the performance of latency-sensitive applications across various industries, leading to consequences such as:
    - DNN Training and Inference Workloads - Disruptions in communication between DNN nodes due to microbursts can result in slower training and inference times, potentially affecting development and deployment times of AI-driven applications.
    - Electronic Trading Systems - Latency is critical in electronic trading systems, where even millisecond delays can lead to delayed market data and slower execution times, resulting in missed opportunities or financial losses.
    - HPC Workloads - Delays caused by microbursts in data transmission between HPC nodes may lead to slower computation times and decreased performance.
- Misleading Network Capacity - By averaging conventional metrics (e.g., bandwidth utilization, packet loss rate, average throughput, latency, error rates, jitter) over longer periods, brief periods of high utilization caused by microburst may be masked, leading to a misleading perception of underutilization. This can result in inadequate network planning and unexpected performance issues during peak demand. Conversely, when packet drops are observed without identifying microbursts as the cause, the typical response is to add bandwidth — an expenditure that fails to address the root cause.
- Quality of Service (QoS) Enforcement Difficulties - Microbursts, with their quick and brief occurrences, may evade QoS controls, resulting in QoS policy violations and degraded service quality for priority traffic.

Identifying and Measuring Microbursts

Standard network monitoring tools, referred to here as low-resolution systems, such as those based on SNMP, NetFlow or sFlow, typically aggregate data over long time intervals (e.g., seconds, minutes). This aggregation can mask microbursts, limiting the ability of these systems to identify and diagnose them. Conversely, high-resolution systems can capture and analyze network traffic in real-time, enabling them to detect microbursts that are often missed by low-resolution systems.

Figure 8 illustrates the difference between low-resolution monitoring systems (e.g., averaging over short period: ~ seconds) vs. high-resolution monitoring systems (e.g., averaging over short period: ~ milliseconds). A high-resolution system can measure and detect the presence of microbursts.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

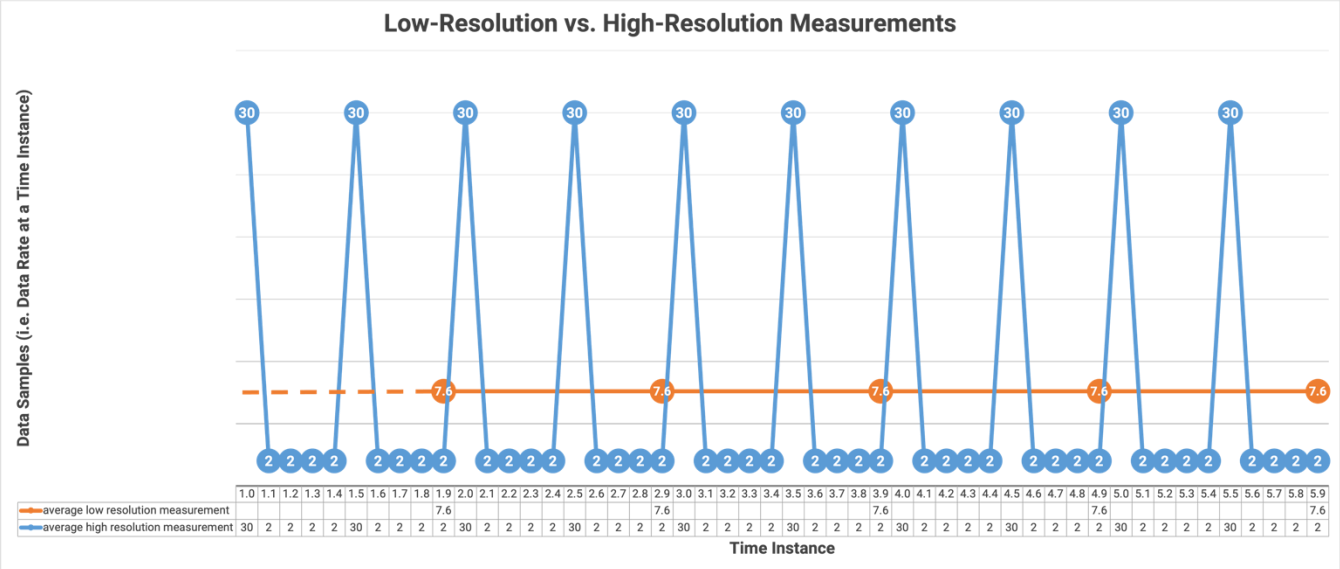


Figure 8. Low vs. High Resolution Measurements

Low-resolution measurements, with aggregate metrics over seconds, would be unable to detect the presence of large peak data samples, resulting instead in a misleading and likely much smaller value. High-resolution measurements, however, can effectively detect the presence of large peak data samples. Similarly, in the context of a system monitoring network performance, and specifically for the microburst measurement use case, a low-resolution system would be unable to measure and detect microbursts. If the data shown in Figure 8 above corresponded to throughput measurements collected from a 10 Gbps network, the following observations could be made:

- Real average data throughput: 2Gbps
- 30 Gbps microbursts every five time-instances
- Maximum low-resolution throughput measured: 7.6 Gbps
- Low-resolution measurements provide a maximum throughput value of 7.6 Gbps and miss the presence of the 30 Gbps microbursts, which are detected by high-resolution measurements.

Detection and Network Monitoring Limitations

Let us compare the difference in the amount of data transferred by a 1Gbps link and a 100Gbps link to illustrate the impact a microburst event can have in a short amount of time. A 1Gbps link can transmit 125MB of data per second, while a 100Gbps link can transmit 12.5 GB in the same time frame. During a microburst event, which can last from a few microseconds to a few milliseconds, a significant amount of data can be transferred. For instance, a microburst lasting 5 ms can transfer up to 625KB on a 1Gbps link and 62.5MB on a 100Gbps.

Figure 9 illustrates how the aggregation of data received from three servers can create microbursts. Servers 1, 2, and 3 each transmit at an average rate of 2 Gbps, resulting in a 6

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

Gbps aggregate average data rate. Short (10 ms) bursts from all three servers occur simultaneously, increasing the instantaneous aggregate data rate to 30 Gbps. Network monitoring with per-second averaging resolution cannot detect these brief spikes because the data is averaged over a longer interval. Monitoring with millisecond-level resolution, however, can detect the presence of microbursts by capturing traffic at or near real time.

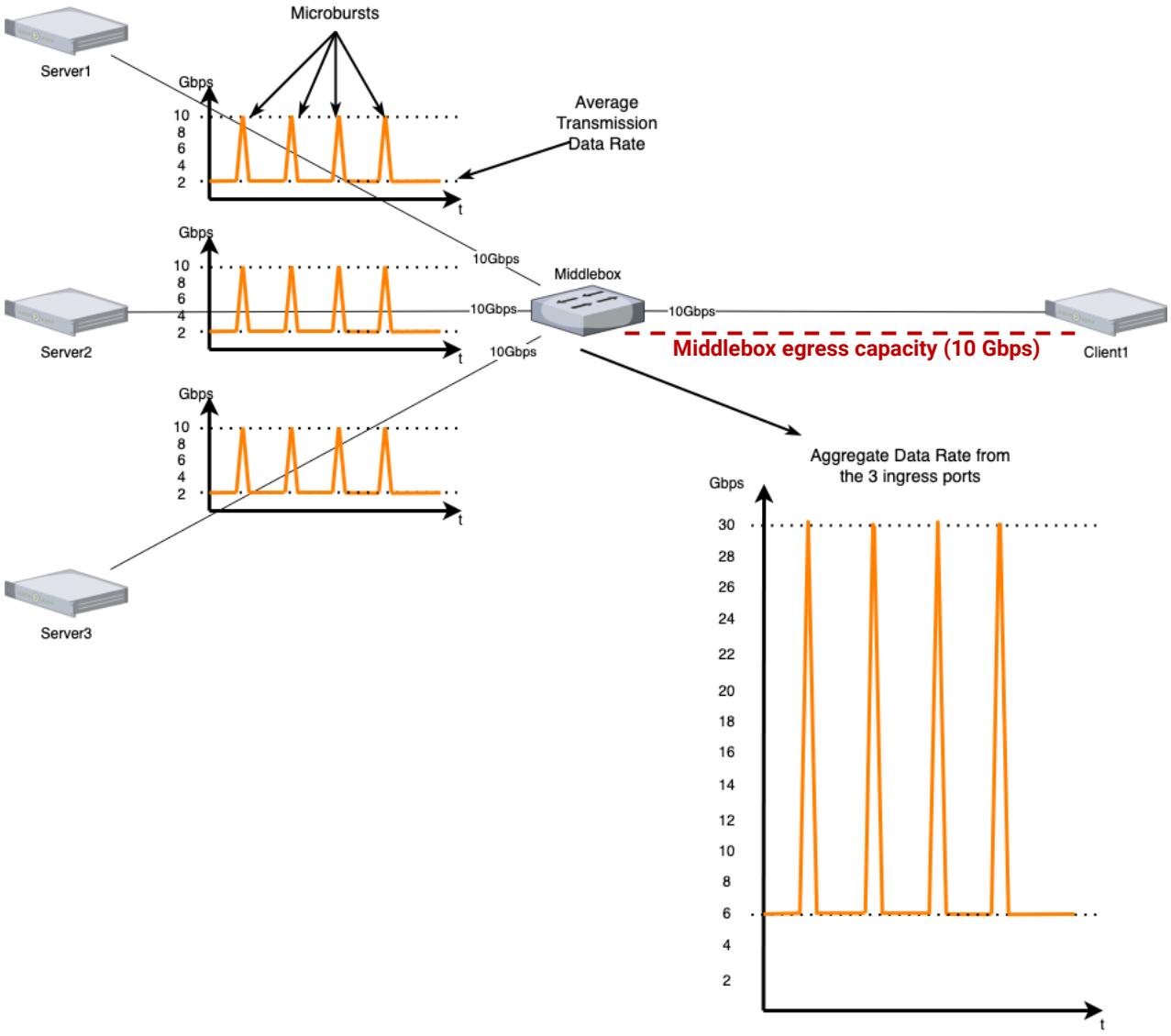


Figure 9. Aggregation of Data Received Simultaneously from Three Sources

While Figure 9 uses synchronized bursts to illustrate the concept, real-world traffic behavior is typically less predictable. Servers burst at different times and rates, and in most cases the aggregate traffic arriving at a middlebox remains below its forwarding capacity. Occasionally, however, bursts from multiple servers align, causing the instantaneous aggregate rate to exceed the middlebox's egress bandwidth. When this happens, excess traffic must be buffered, and if buffer space is exceeded, packets are dropped. This intermittent behavior

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

makes microbursts particularly difficult to diagnose: the same sources may appear benign most of the time, yet still produce brief congestion when timing happens to align.

Figure 10 shows this more realistic view. Bursty traffic from three independent sources typically overlaps only partially, producing aggregate rates below the middlebox's 10 Gbps egress capacity. Occasionally, bursts from two or three servers coincide, creating brief spikes that exceed the forwarding threshold. These short-lived microbursts can cause buffering and packet loss but are difficult to observe with per-second monitoring due to averaging effects.



Figure 10. Aggregate Traffic Relative to Middlebox Capacity

The brief but significant data transfer that occurs during microburst events can lead to network congestion and performance degradation. To effectively detect and mitigate them, network monitoring equipment with real-time capabilities is needed. Low-resolution systems that aggregate data over seconds may fail to identify these events entirely.

cPacket's Real-Time Microburst Detection Solution

cPacket, a leading provider of network observability systems with over 15 years of experience, recognizes the critical importance and challenges of detecting microburst events in real-time. To address this issue, the company has developed comprehensive hardware and software solutions that enable network operators to detect and measure microburst as they occur, even in today's increasingly demanding network environments. These solutions provide real-time detection, measurement, and visualization of microburst events using intuitive dashboards,

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

offering valuable insights into network behavior. cPacket's High Resolution Counters (HRCs) and cBurst provide a direct solution to the challenge of microburst detection in real-time.

cPacket's High Resolution Counters

cPacket's High Resolution Counters (HRCs) provide granular visibility to help detect and analyze microbursts effectively. With dedicated hardware based on ASICs and FPGAs behind every monitoring port, cPacket's solutions enable real-time, line-rate up to 100Gbps processing of network traffic without impacting performance or dropping packets. HRCs, implemented using the capabilities of these ASICs and FPGAs, provide microsecond-level measurement of network traffic, enabling the detection and analysis of microbursts.

HRCs enable drill-down capabilities into behaviors of any traffic profiles at millisecond resolution. HRCs allow users to drill down into the millisecond behavior of any of the counters reported by cPacket products and enable users with the capability to observe microbursts in real time.

Figure 11 compares average resolution (1-second) versus high-resolution (millisecond) throughput values for a 100Gbps port measurements using cPacket's high-resolution counters. Specifically, it highlights a scenario where the average throughput measurement is 1.17 Gbps, in contrast to the high-resolution measurement of 94 Gbps. This reveals a significant difference, with the high-resolution throughput measured being 80 times higher than the average.



Figure 11. Example of 100Gbps Port Throughput Measurement with Millisecond vs.1-Second Resolution

Figure 11 highlights the difference between millisecond resolution and 1-second resolution measurements and microburst detection. Relying on low-resolution measurements could leads to several issues, including:

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

- Network anomalies, such as packet drops, increased latency, and congestion, might be misdiagnosed if they stem from microbursts, which low-resolution measurements fail to capture.

- While it is possible to detect microbursts without high-resolution measurements, this typically involves indirect methods, such as experimental or heuristic approaches. These methods can be time consuming and resource intensive, potentially extending MTTR.

- Low-resolution network link measurements may indicate lower-than-actual utilization if they fail to detect frequent microbursts. This can result in underestimation of the necessary network bandwidth, leading to situations where host or middle-box ports may be unexpectedly overwhelmed by "unexpected" traffic.

cPacket's cBurst

cBurst was developed with the intent to provide the capability to characterize microbursts in networks. This feature provides microburst statistics (e.g., maximum, minimum, mean, and standard deviation) per second with millisecond resolution, tracked according to user specification. cBurst counts the number of milliseconds over a threshold, which is a more realistic representation of the load on a middlebox than the single worst case burst.

The difference between cPacket's cBurst and traditional technologies is that rather than simply detecting that a link is spiking or saturated, cBurst provides the profile or combination of profiles that make up the bursts. Ultimately, this identification allows for reallocation of network traffic to links with more capacity (again, as measured with hardware accuracy by cBurst) and eliminates the trial and error that ensues when only the aggregate traffic behavior is known. In summary, cBurst will identify the specific flows responsible for microbursts, eliminating any guesswork involved in the process.

With cBurst, network professionals can detect and understand the source of the microburst. By understanding the source of the microburst, the network administrator and the application owner can rebalance the load or reroute the offending data stream.

For example, Figure 12 below demonstrates the case where two workloads are the dominant contributors to the microburst, driving 60.7Gbps and 30Gbps bursts where the rest of the workloads have bursts of less than 10Gbps. Rebalancing "Workload 1" or moving it to a separate link will let the network avoid congestion.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

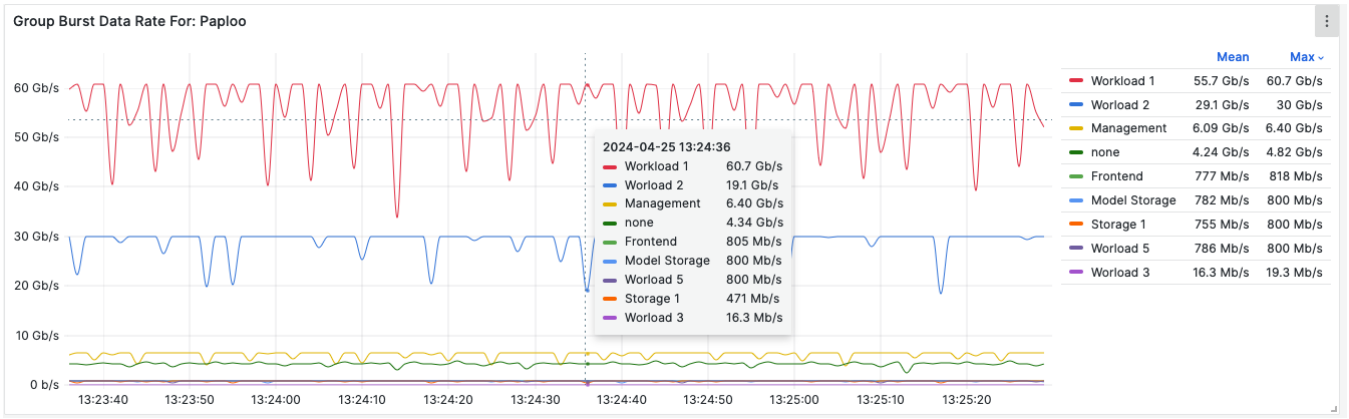


Figure 12. cBurst Detection of Dominant Microburst Contributors

cBurst provides even more insight beyond understanding the source of the microburst. With cBurst it is possible to measure the duration of the burst for each individual workload. Figure 13 below illustrates the capability to observe the length of each burst for every workload. Additionally, it exhibits the uneven distribution of the bursts across consecutive seconds.



Figure 13. cBurst Measurement of Microburst Duration and Distribution

Figure 14 below emphasizes the advantages of millisecond accuracy measurements. While 1-second resolution measurements miss the severity of microbursts, employing millisecond resolution enables their detection with precision.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

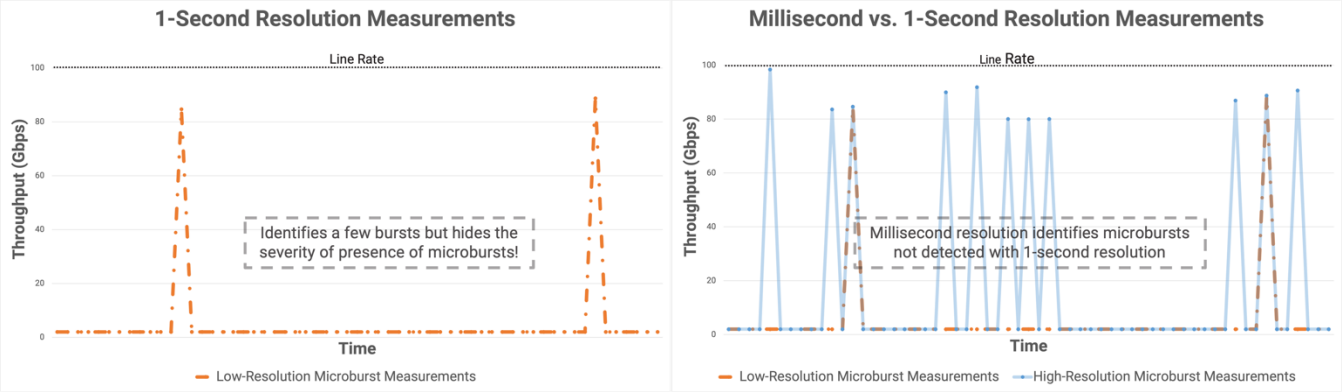


Figure 14. Microburst Detection with 1-Second Resolution vs. Millisecond Measurements

Conclusion

In today's rapidly evolving technological landscape, microbursts pose a significant challenge in operational production networks, particularly for applications such as electronic trading systems, HPC and increasingly ubiquitous AIML workloads. Microbursts can cause network congestion and performance degradation. This can significantly impact modern-day applications, such as increasing the Job-Completion-Time (JCT) for AIML workloads or missing trades in electronic trading systems. Without the ability to effectively detect and analyze microbursts, network managers are unable to resolve issues or proactively and cost-effectively mitigate their impact on network performance.

cPacket sets itself apart by offering a portfolio of high-fidelity products that measure network performance at an unmatched resolution. The company's High Resolution Counters and cBurst technology enable NetOps professionals to directly measure and analyze microbursts in real-time, rather than relying on secondary measurements or inferential analysis. This unique capability provides unparalleled visibility into network behavior at the packet level, allowing network operators to identify and characterize microburst behavior with precision.

cPacket, empowers network managers to identify and characterize microburst behavior with precision. By using the information from cPacket’s solution, organizations can:

- Reduce the MTTR to resolve packet drops or increased latency and jitter
- Proactively increase the bandwidth on cost sensitive links such as WAN connections
- Reallocate workloads or re-balance data streams to avoid microbursts

Track accurately the SLA for consumers of sensitive data such as market data feeds

cPacket’s microburst monitoring capabilities are built on its expertise and technology leadership in network observability, enabling network administrators to reduce MTTR and proactively address network issues. With advanced technical solutions and understanding of network performance and the microburst phenomenon, cPacket is ready to assist enterprises

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.

in monitoring, managing, and safeguarding their networks from the impact of sub-second traffic surges.

References

[1] "Cambridge Dictionary," n.d., . [Online]. Available: https://dictionary.cambridge.org/dictionary/english/burst. [Accessed 22 April 2024].

[2] "The Network Encyclopedia," n.d., . [Online]. Available: http://www.thenetworkencyclopedia.com/entry/burst/. [Accessed 22 April 2024].

[3] R. Kapoor, A. C. Snoeren, . G. M. Voelker and G. Porter, "Bullet trains: a study of NIC burst behavior at microsecond timescales," in *CoNEXT '13: Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, Santa Barbara California USA, pp. 133-138, 2013, https://doi.org/10.1145/2535372.2535407.

[4] D. Shan, F. Ren, P. Cheng and R. Shu, "Micro-burst in Data Centers: Observations, Implications, and Applications," *arXiV - CS - Networking and Internet Architecture,* 2016, arXiv:1604.07621v1 [cs.NI].

[5] N. Luangsomboon, F. Fazel, J. Liebeherr, A. Sobhani, S. Guan and X. Chu, "On the Burstiness of Distributed Machine Learning Traffic," *arXiv - CS - Networking and Internet Architecture,* arXiv:2401.00329v1 [cs.LG], 2023.

[6] S. W. Brim and B. E. Carpenter, *RFC 3234. "Middleboxes: Taxonomy and Issues",* IETF, 2002, [Online]. Available: https://doi.org/10.17487/RFC3234.

[7] H. Balakrishnan, V. Padmanabhan, G. Fairhurst and M. Sooriyabandara, *RFC 3449: "TCP Performance Implications of Network Path Asymmetry",* 2002, [Online]. Available: https://doi.org/10.17487/RFC3449.

[8] D. Shan and F. Ren, *Improving ECN marking scheme with micro-burst traffic in data center networks,* Atlanta, GA, USA: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp. 1-9, 2017, doi: 10.1109/INFOCOM.2017.8057181.

[9] M. Mathis, J. Semke, J. Mahdavi and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *ACM SIGCOMM Computer Communication Review,* pp. 67-82, 1997, https://doi.org/10.1145/263932.264023.

© cPacket Networks, Inc. | Confidential | Unauthorized reproduction or distribution, in whole or in part, is prohibited without explicit written consent.