Optimizing Edge Payloads for Cloud Costs: Payload Optimization with DeepIQ Edge or Kepware IoT Gateway

If you’re streaming real-time data from edge devices to cloud platforms like AWS IoT Core or Azure IoT Hub, the configurations at your edge can heavily influence your cloud expenses due to the billing structure of these platforms.

- What additional benefits can be derived from fine-tuning your edge configurations for better cloud cost management?
- How should you proceed with such optimizations?

This article addresses the topics mentioned above. It is detailed, but there’s a straightforward checkpoint for those who might already be optimized: Check the distribution of your message sizes using tools such as AWS CloudWatch or Azure Log Analytics.

If your message sizes are not significantly larger than the size that triggers an event then you might already be close to an optimal point

Cloud platforms calculate the number of events based on message size.

Currently:

- **AWS IoT Core** and **Kinesis Data Streams** use a 5 KB message size to define a single event.
- **Azure IoT Hub** use a 4 KB message size per event.
- **Azure Event Hub** uses a 64 KB message size per event

If your message sizes are significantly larger than the size that triggers an event, you might already be close to an optimal point, as this article suggests. However, if the message sizes are not consistently significantly larger (~3x) than this threshold, there could be substantial savings from optimization efforts.

Edge devices like Kepware IoT Gateway offer configurable parameters like message buffering time. These settings directly influence the frequency of transmissions and the size of individual messages. However, finding the right balance isn't straightforward. Data from edge devices often arrives unpredictably, leading to message size variability. Suboptimal configurations can increase billable events, unnecessarily driving up costs.

To tackle this, we introduce the concept of **Billing Payload Efficiency**—a metric that evaluates how well the transmitted payloads utilize the billable capacity for each event. Organizations can configure their edge systems by analyzing and optimizing this efficiency to achieve near-theoretical minimum costs while maintaining desired latency and reliability.

Introducing Billing Payload Efficiency

Billing Payload Efficiency (BPE) refers to how effectively the maximum capacity of a billable event is utilized. It can be calculated using the formula:

$$\boldsymbol{Billing\ Payload\ Efficiency} = \frac{\boldsymbol{100 * Payload\ Size\ (KB)}}{\boldsymbol{No\ of\ Events\ for\ Payload\ Size * Event\ Capacity\ (KB)}}$$

This concept is illustrated below.

Message Size Range	Payload Size	Event Capacity	# of Events/ Payload Size	BPE	Average BPE
	4.00	5	1	80.00	
	4.25	5	1	85.00	
	4.50	5	1	90.00	
	4.75	5	1	95.00	
4 to 6	5.00	5	1	100.00	75.00
	5.25	5	2	52.50	
	5.50	5	2	55.00	
	5.75	5	2	57.50	
	6.00	5	2	60.00	

Message Size Range	Payload Size	Event Capacity	# of Events/ Payload Size	BPE	Average BPE
	3.00	5	1	60.00	
	3.25	5	1	65.00	
	3.50	5	1	70.00	
	3.75	5	1	75.00	
3 to 5	4.00	5	1	80.00	80.00
	4.25	5	1	85.00	
	4.50	5	1	90.00	
	4.75	5	1	95.00	
	5.00	5	1	100.00	

This implies that when the messages are between 4 and 6, assuming a uniform distribution as shown, the billing efficiency would be around 75%.

The illustration below shows how optimization through better message sizing and frequency can lead to a 26% reduction in costs.

	Message Characteristics				Standard Model		Optimized Model	
AWS Cost/ 1 Million Events	Min	Max	Max Event Size	Sum Payload/ Day	# of Events	Cost	# of Events	Cost
$1.00	3	5	5	40,000,000	10,000,000	$10.00	8,000,000	$8.00
$1.00	4	6	5	50,000,000	15,000,000	$15.00	10,000,000	$10.00
$1.00	6	12	5	90,000,000	23,333,333	$23.33	18,000,000	$18.00
					Average	$16.11		$12.00
							Cost Reduction Achieved through Optimization	26%

Stochastic Payload Distributions and Expected Billing Payload Efficiency

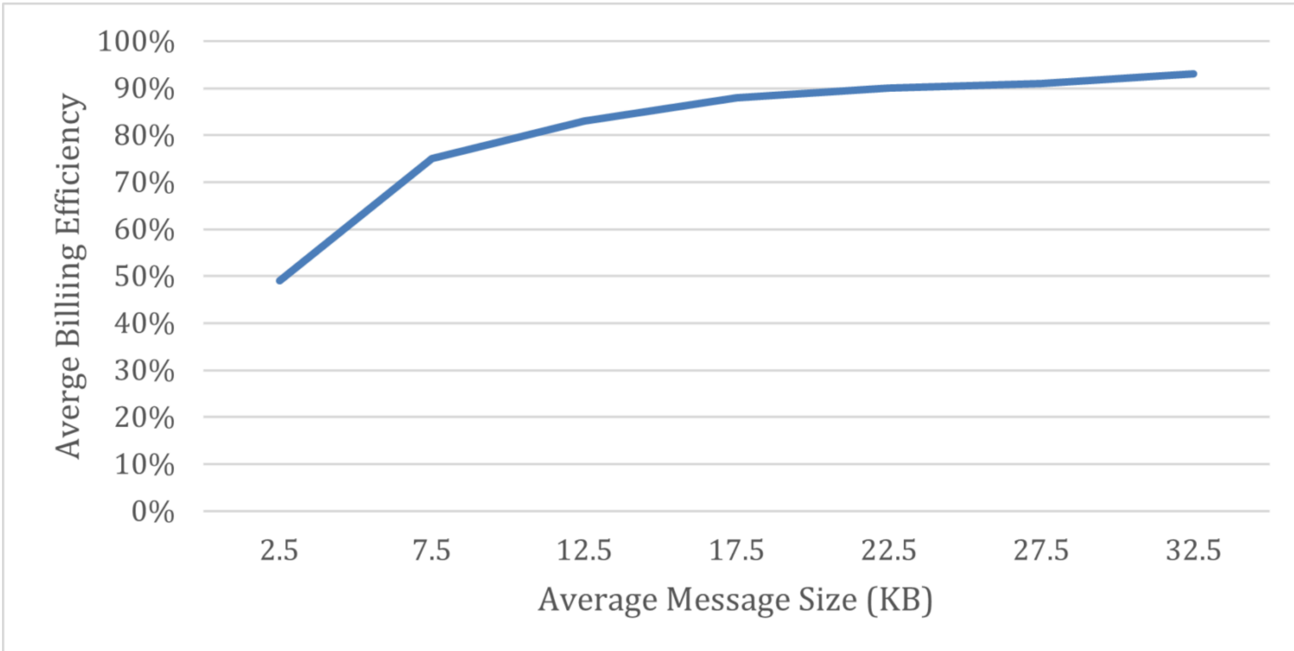
When working with edge devices like the **Kepware IoT Gateway**, the stochastic nature of data generation at the source introduces variability in payload sizes. For example, Kepware is often configured to publish data only when changes occur. Consider a sample scenario where a buffering interval of 1-second results in payload sizes varying significantly, ranging between 5 KB and 15 KB. Such randomness in payload sizes directly impacts billing efficiency and overall costs, making it crucial to optimize buffering strategies.

Introducing Average BPE

To account for this variability, we introduce the concept of Average BPE, which represents the expected value of BPE for a given payload size distribution. This metric provides a quantitative framework to evaluate and optimize buffering strategies under varying conditions.

Simulation and Analysis

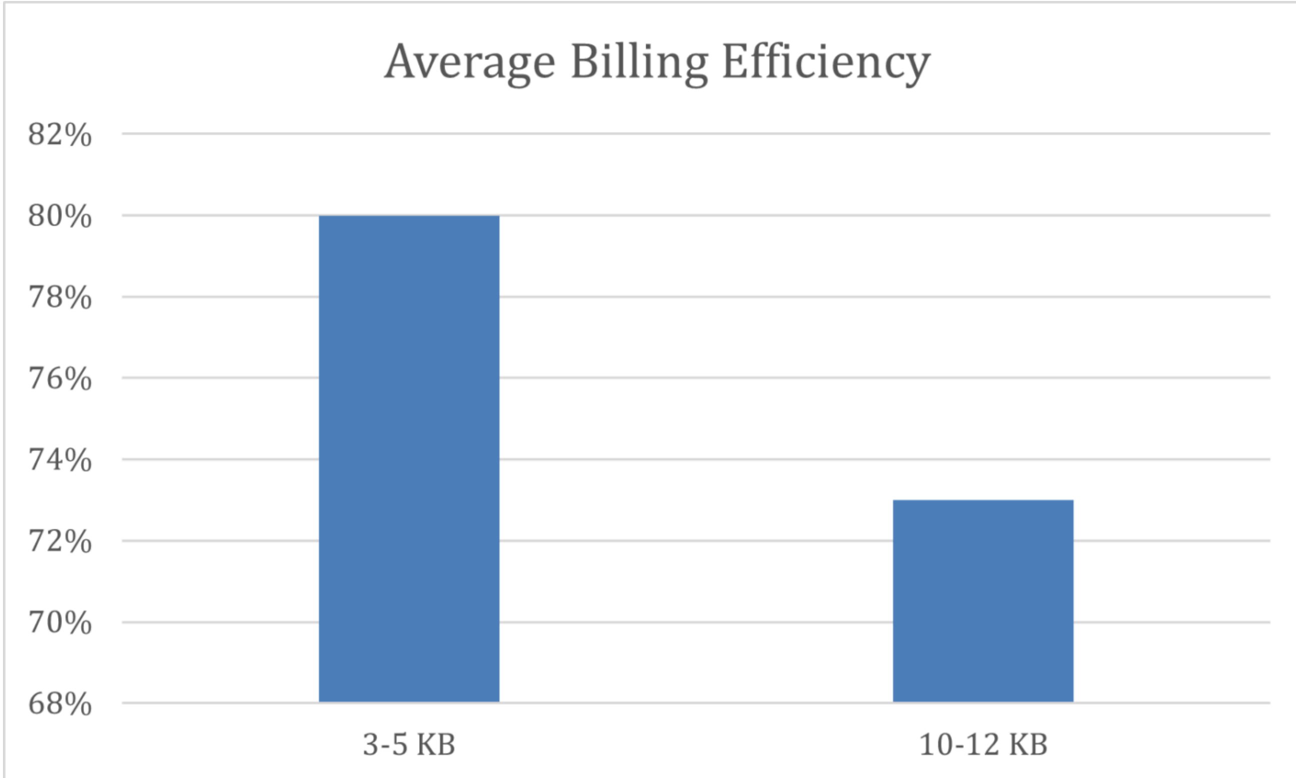
Average BPE can be estimated for various distributions of payload sizes. Assuming the payload is distributed uniformly with a range of 5KB, the average BPE versus the average message size is shown in the figure below.   Although specific results vary depending on the distribution, the key principles remain consistent.



Let's analyze the above figure to understand the overall trend. When your average message size is 2.5 KB, you waste approximately 50% of the available billable event capacity. In other words, adopting a better buffering strategy could reduce your IoT Core costs by 50% for the same data volume.

Two key trends emerge from this analysis. First, smaller messages tend to have lower billing efficiencies due to how billable events are calculated. Second, billing efficiencies gradually converge toward 100% as message sizes increase.

The relationship is not a straightforward, monotonically decreasing one. For instance, consider the Average Billing Efficiency under two scenarios: in the first scenario, the payload varies uniformly between 3 and 5, resulting in an Average Billing Efficiency of 80%. In contrast, in the second scenario, where the payload varies uniformly between 10 and 12, the Average Billing Efficiency drops to 73%. As a result, it is essential to analyze the factors influencing efficiency across different payload ranges carefully.



While buffering for larger message sizes can significantly improve efficiency, it may not always be feasible due to latency constraints. The optimal strategy should balance cost efficiency and latency requirements tailored to your use case's needs and priorities.

It is important to note that the results depend on the actual payload distribution. To optimize for your payload, it is crucial to simulate the distribution accurately and then use tools to calculate the Average Billing Efficiency.

Many edge sources produce data at high volumes. Even at the highest transmission frequency, the payload size may be significant. For these scenarios, explicit optimization of payload size may only have marginal benefits since the number of billable events is already close to the optimal number. However, for scenarios where edges produce data at low volume, the right strategy can have a significant impact.

Cloud Service Billing Overview

AWS IoT Core or Kinesis Data Streams

AWS IoT Core and Kinesis Data Streams are chargesdbased on **Message Size**: Messages up to 5 KB are considered a single billable event. Larger messages are split into multiple 5 KB chunks, each billed separately.

Azure Event Hubs

Charges are based on the event size. Each event up to 64 KB is considered one billable event. If an event exceeds 64 KB, it is split into multiple chunks of up to 64 KB each, with each chunk billed as a separate event.

Azure IoT Hub

Charges are based on the message size. Messages up to 4 KB are counted as a single billable event. Larger messages are divided into multiple 4 KB chunks, and each chunk is billed as one billable message.

Payload Optimization

Benefits of Larger Messages

1. **Cost Savings**: Larger, efficiently sized messages reduce the number of billable events.
2. **Improved Downstream Processing**: Consolidated messages allow for bulk processing, enhancing system performance.

Trade-offs and Challenges in Optimizing Billing Payload Efficiency

1. **Increased Latency**: Higher buffering times can delay data transmission, impacting real-time processing requirements.
2. **Risk of Lost Messages**: Without store-and-forward capabilities, longer buffering times increase the impact of connectivity issues.

Advantages with DeepIQ Edge

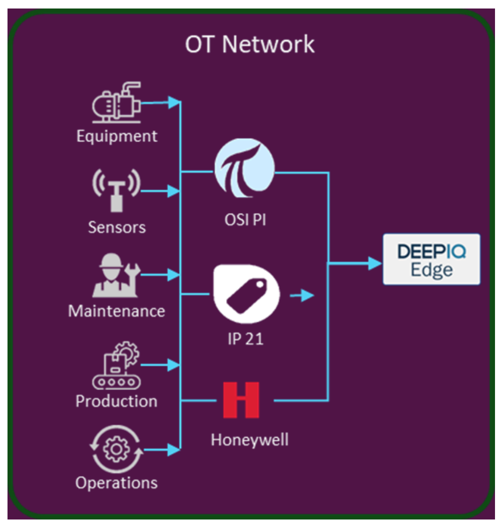


Figure1: DeepIQ Edge

DeepIQ Edge software is designed to enhance data transmission reliability and optimize cloud expenses for industrial applications. Here are its key advantages:

- Automated Payload Optimization: DeepIQ Edge employs advanced buffering techniques to ensure optimal transmission of messages. Adjusting buffer sizes and managing payloads efficiently maintains ideal sizes. For example, you might set up the message size to be 3.8 KB for Azure IoT hub or 4.8 KB for AWS IoT Core. This approach maximizes network efficiency and operational performance while minimizing cloud costs, especially in environments with variable data flow. Additionally, a timeout feature ensures timely data transmission, even if the buffer isn't full, which helps satisfy latency requirements.
- Support for Non-MQTT Payloads: Standard IoT workloads incur charges for both messaging and processing when forwarding data to other cloud services. DeepIQ Edge optimizes this by directly pushing data to AWS services like S3 or Kinesis Data Streams and Azure services like ADLS Gen 2 or Event Hub. This approach significantly reduces cloud costs by eliminating unnecessary intermediate processing layers.
- Comprehensive Historian Compatibility: DeepIQ Edge is engineered to integrate seamlessly with a wide range of historians, including OSI PI, IP 21, and Honeywell PhD. This compatibility ensures efficient ingestion of historical data, which enhances the analytical capabilities and data utilization of industrial operations. By supporting the ingestion of historical data, DeepIQ Edge offers unmatched flexibility and utility in managing and analyzing industrial data streams.

These features position DeepIQ Edge as a robust solution for industrial settings, reducing operational costs while maintaining high data integrity and transmission efficiency.

Optimization with Kepware

Kepware specializes in real-time connectivity to control systems and SCADA, enabling seamless communication with industrial devices for operational monitoring and control.

Configuring the ideal buffering time in Kepware IoT Gateway involves balancing latency, message size, and the potential impact of lost messages. Since Kepware IoT Gateway does not have store-and-forward capabilities, increasing the buffering time also increases the risk of losing more data during connectivity issues. If you have determined an ideal buffering time (e.g., 1500 ms), follow these steps to implement and monitor the configuration:

1. Access Kepware Settings:

- Log in to your Kepware server interface.
- Navigate to the IoT Gateway configuration.

2. Configure the Buffering Time:

- Select the specific IoT Gateway agent you want to configure.

- Locate the **Rate(ms)** setting.
- Set the Rate to your desired value (e.g., 1500 ms).
- Save the changes to apply the new buffering configuration.

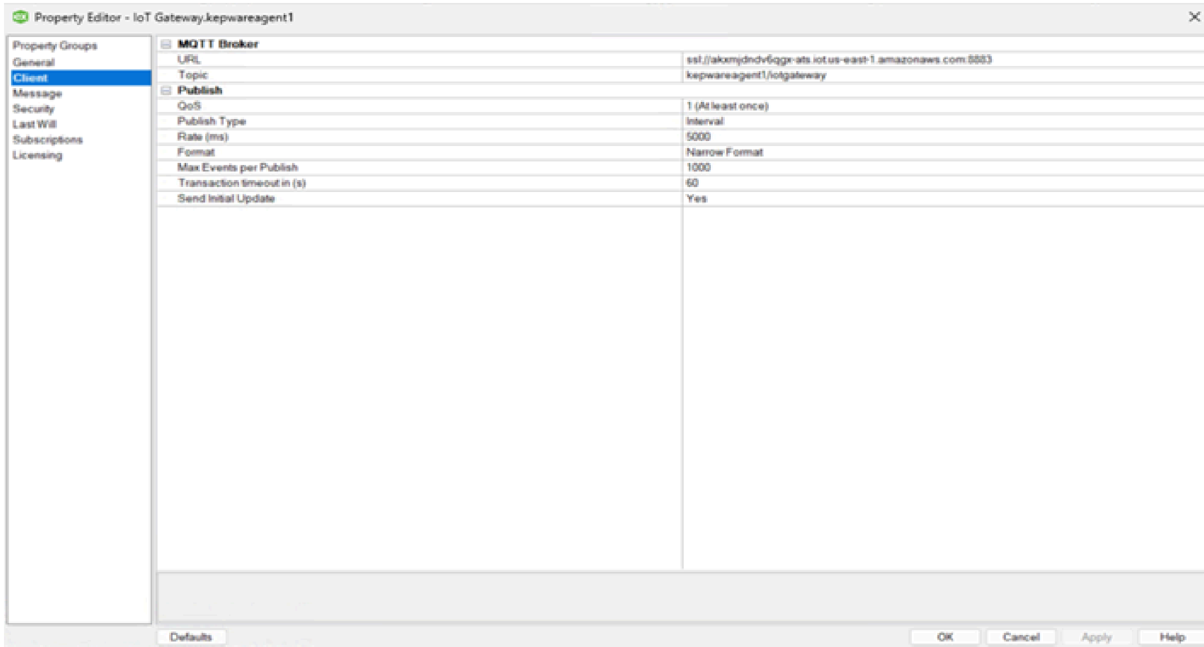


Figure 2: IoT Gateway

3. Monitor Logs in AWS CloudWatch or Azure Log Analytics

- Set up monitoring in AWS CloudWatch or Azure Log Analytics for logs to track metrics such as the number of events, average message size, and latency.
- Validate that the implemented buffering strategy aligns with cost and reliability goals.
- Track the distribution of message sizes and calculate average billing efficiency.

4. Adjust as Needed:

- Based on the avbove metrics, fine-tune the buffering time in Kepware to optimize performance further.
- Ensure that any adjustments maintain a balance between cost efficiency and latency requirements.

Monitoring and Feedback Loop

- **Billing Reports**: Regularly review cloud billing reports to measure cost impacts.
- **Kepware Logs**: Analyze logs to track message sizes and adjust configurations as needed.
- **Payload Efficiency Metrics**: Calculate Billing Payload Efficiency to identify inefficiencies and opportunities for improvement.

Conclusion

Billing Payload Efficiency is a new metric for IoT cost optimization. By aligning edge software configurations like DeepIQ Edge or Kepware IoT Gateway with the billing models of Cloud platforms, organizations can significantly reduce costs while improving efficiency. The key is to buffer data intelligently, balancing payload size against transmission frequency and minimizing wasted bandwidth.

Optimizing edge payloads for cloud cost efficiency is a vital consideration in the broader context of IT-OT convergence. This strategy is just one piece of a complex puzzle that includes managing the lifecycle of edge software, contextualizing IT and OT data, developing robust cloud data models with stringent versioning and governance, and simplifying the implementation of AI and digital twin workflows. Each of these elements presents its own set of challenges and nuances that must be expertly navigated to unlock the full potential of digital transformation initiatives.

The DeepIQ platform is equipped with a wide range of capabilities and tools tailored to streamline complex IT-OT convergence tasks. It supports a comprehensive array of functions, from constructing edge asset hierarchies to advanced IT contextualization, and facilitates streaming AI and digital twin workflows. As a unified solution, DeepIQ simplifies integration and accelerates the deployment of digital strategies. Engineered to address the intricacies of these processes, our platform ensures meticulous management of every convergence aspect, perfectly aligning with your business objectives.

For organizations looking to explore the full spectrum of benefits that IT-OT convergence can offer, the DeepIQ platform is a proven leader. We have implemented our solutions across some of the world's largest companies, demonstrating our ability to deliver scalable and effective results. Our platform not only optimizes costs but also empowers organizations to leverage their data more effectively, improving decision-making and operational efficiencies.

We invite you to explore further how the DeepIQ platform can transform your operations. For a deeper insight into our capabilities and the successes our customers have achieved, visit our website and access our comprehensive library of whitepapers and customer success stories at https://deepiq.com.

DeepIQ is on a mission to transform industrial processes by digitizing industrial expertise. Our vision is to drive end-to-end automation, enabling systems such as self-running power plants or drilling rigs using generative AI as the higher order reasoning layer operating over existing industrial automation technology stack.

DataStudio Deployment Guide

Career

Contact Us

+1 (832) 391-6834 info@deepiq.com