

# An Overview of Arista Ethernet Capture Timestamps

High performance data analysis is an ever increasing requirement in modern networks with a vast array of use cases including; network troubleshooting, application performance tracking, security or threat analysis and legal compliance (such as RTS-25/MiFID II). However, such analysis is only as good as the accuracy of the data it's based on. One of the key requirements for accuracy when performing any kind of analytics is understanding precisely when a packet was captured. In modern network infrastructures this is not always a simple matter as there may be multiple links, network tiers, or data centers separating the data capture point and device performing the analysis.

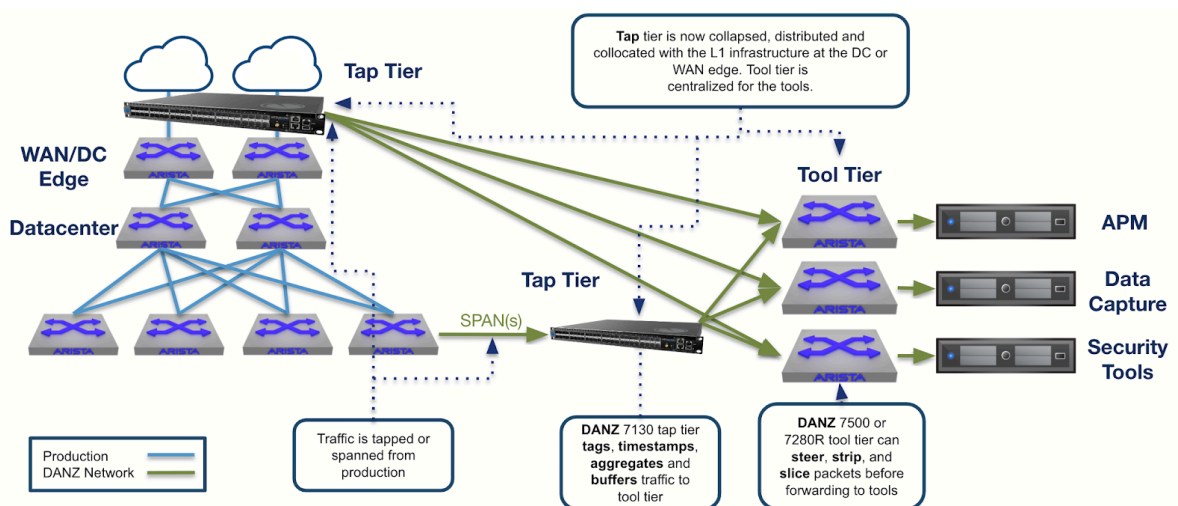
A common mechanism for preserving capture time accuracy is the insertion of timestamps in the data capture. While this method is somewhat ubiquitous within the industry, the specifics of the implementation differ widely. Variances include not only the accuracy of the timestamps, but the format itself (i.e length, location, self-contained vs overwritten, need for post processing etc).

## Timestamp Aware Devices

Broadly speaking there are two classes of timestamp-aware devices:

- 1. Consumers** - These devices such as analytics engines and packet capture systems ingest packets and pass the received data to local applications or write it to persistent storage. Typically these consumers will have their own timestamp logic that processes the packet arrival time when it reaches the systems NIC or OS. Higher precision consumers may use hardware based timestamping for higher precision and typically have a PPS or PTP input for synchronization across devices. In large distributed analysis networks this class of devices may ingest packets that have already been hardware timestamped at or near the wire capture source, enabling both local device and network latency to be accounted for and offset.
- 2. Transit Devices** - Transit devices may be active taps, tap aggregation switches, packet brokers or matrix switches. These devices do not perform any local analysis on ingested packets but rather qualify, process, and forward to consumers. Transit devices typically timestamp on ingress in hardware and may append metadata in addition to the timestamp. Higher precision systems have PPS or PTP input for synchronization across devices. Different functions of this class of device are offered on the Arista 7130, 7150, 7280R and 7500R Series platforms.

In real world environments both transit devices and consumers are deployed. Optimally the transit device performs the packet replication, timestamping, and forwarding to Consumers removing the need for different devices to perform these discrete functions. Additional tiers of transit devices may exist between the device applying the timestamp and the Consumers. These tiers are usually included to provide aggregation and intelligent filtering or traffic-steering capabilities, which enables the capture of a much larger number of individual network links cost-effectively by reducing the number or distribution of Consumers required. Assuming that the traffic is timestamped on ingress by the transit device, the additional tiers of transit devices do not affect the precision of the capture. This is a key point for large distributed networks that require precision capture for analytics or regulations.



### Timestamp formats and locations on transit devices

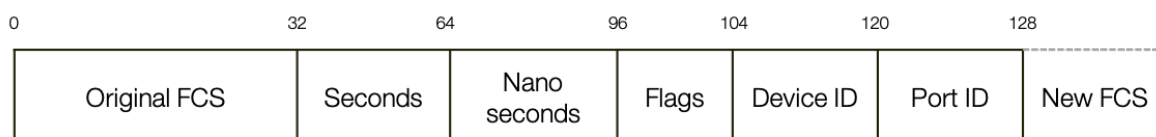
While specific implementations vary between vendors there are typically two primary methods to implement timestamps: overwriting fields in existing Ethernet or IP headers or the addition of a self-contained trailer.

1. **Field Overwriting** - Many implementations (including optional modes on both the Arista 7150 and 7280 series) take an existing field within a frame header or trailer and replace that value with the hardware timestamp. This approach is beneficial as it typically done in hardware at line rate and does not extend the overall length of the packet and therefore reduces the buffering burden on the aggregation switches, which tend to be oversubscribed. The negative aspect of this approach is that the size of the timestamp is dictated by the field it replaces (32 bits for the CRC field for example). If the field is less than 64 bits wide it requires some additional data (such as a keyframe) must be provided to the Consumer to correlate the timestamp back to UTC. A keyframe may also be required to correlate metadata such as input device and port.
2. **Self-contained frame trailer** - Another implementation to apply the timestamp is to expand the original packet with a set of custom fields. When such an implementation is used the FCS field must be updated to ensure the packet that leaves the device is a valid Ethernet frame that can be forwarded normally by subsequent transit devices. The self-contained frame trailer extends the size of the packet which can create some concerns around latency and throughput, i.e a line rate non-timestamped traffic flow can no longer egress out of a same-speed interface at line rate once timestamped. The benefits of self-contained timestamps revolve around flexibility and self-containment. The accuracy of the timestamp is not dictated by the size of an existing field and allows additional metadata to be built into the trailer, such as fields for the identification of the device and interface which applied the timestamp. This negates the needs for another mechanism to communicate this metadata to the Consumer.

### Timestamping on the Arista 7130 Series

Arista 7130L and K Series devices support inline tapping and replication, tap aggregation, and timestamping for 1G and 10G interfaces. These L1+ devices support sub-nanosecond timestamp precision via the MetaWatch application and are typically implemented as an inline active tap device performing tapping/replication of the production links, or as physically close to the passive tap performing the replication. The active tap function replicates traffic through the system in 5 ns while simultaneously replicating the packets to the MetaWatch application.

Timestamping is supported via the MetaWatch application on 7130L and K Series systems. The same timestamp format is used on all 7130 Series platforms.



7130 Series Timestamp

The 16 byte (128 bit) 7130 timestamp header is entirely self-contained and is inserted at the end of the Ethernet frame. A valid FCS is also computed and appended to the end of the trailer, which makes timestamped packets routable by subsequent devices. The combined seconds and nanoseconds fields of the timestamp are 64 bits in length, allowing advertisement of full UTC time without the need for additional sync mechanisms such as keyframes. An additional metadata tag is available to optionally instrument fractional nanoseconds with attosecond resolution, making the 7130 the highest precision platform in the Arista portfolio.

The timestamp header includes both Device ID and Port ID, supporting user defined values. This enables a Consumer to identify where in the network the packet was captured even if there are multiple tiers of aggregation between the tap and the Consumer. This is particularly useful for application performance tracking, where the Consumer may receive multiple copies of the same packet at distinct locations throughout the network. Additionally, a single bit within the Flags field confirms if the FCS of the original packet was valid or not.

The 7130L and K Series use an OCXO or Rubidium clock for local reference. The optional optional Rubidium clock provides superior accuracy during holdover. Time can be synced via NTP, PTP or PPS. When synchronised via PPS the 7130 timestamps are capable of +/- 75 picosecond accuracy.

The 7130 timestamps are captured on the first byte of a packet on the ingress port. The propagation delay between the ingress front panel and the FPGA is automatically calculated by MetaWatch and offset before applying the timestamp. This avoids the need for a Consumer to adjust for any offsets or perform any other delay based computation on the received timestamp.

The 7130 timestamp format is also user expandable with support for custom TLVs. This allows additional metadata to be passed between the timestamping and analysis devices. This additional metadata includes sequence numbers or optional user defined tags.

### Timestamping on the Arista 7280R/7500R Series

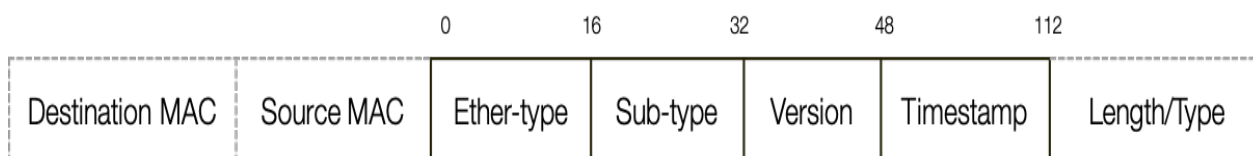
The Arista 7500R/7280R series support tap aggregation, timestamping and packet brokering for 1/10/25/40/50/100G interfaces. They provide the ultimate flexibility in packet qualification, filtering, manipulation, deep packet inspection, and flow based steering and are ideally suited as the aggregation and transit devices which steer the appropriate packets from taps or mirror interfaces to the various Consumers. The fixed configuration 7280R and modular 7500R systems provide identical functionality enabling deployment decisions to be made based on form-factor and environmental scale.

The R series discipline the time stamping clock using hardware assisted PTP and also support transparent or boundary clock operation for scaling a PTP infrastructure. Timestamping is supported on traffic replicated via a mirror session when the device is used as a production switch, or on ingress traffic when deployed in Tap Aggregation mode.

There are three discrete timestamp options available on the 7280R and 7500R systems:

1. 64-bit header timestamp; i.e., encapsulated in a L2 header (>=EOS-4.18.1F)
2. 48-bit header timestamp; i.e., encapsulated in a L2 header (>=EOS-4.20.0F)
3. 48-bit timestamp that replaces the Source MAC (>=EOS-4.20.0F)

**A) The 64 bit encapsulated timestamp** adds 14 bytes (112 bits) and is self-contained within the Ethernet header. Full UTC can be advertised without the need for additional keyframes.



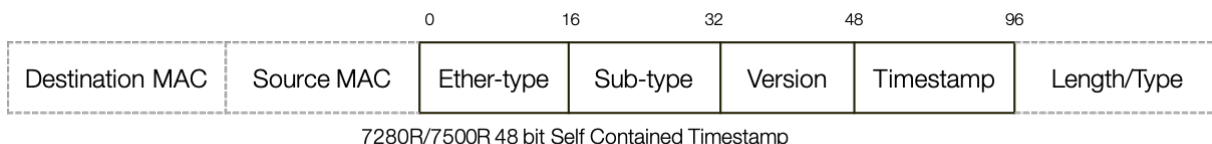
7280R/7500R 64 bit Self Contained Timestamp

The last 32 bits of the 64 bit timestamps are captured at the end of the ingress pipeline, the applied timestamp will therefore be the packet reception time plus the time taken to serialise the first 256 byte of a frame (or the entirety of the serialization time for sub-256 byte frames). The time taken for serialization is a function of the packet size (up to 256 byte) and the speed of the ingress interface.

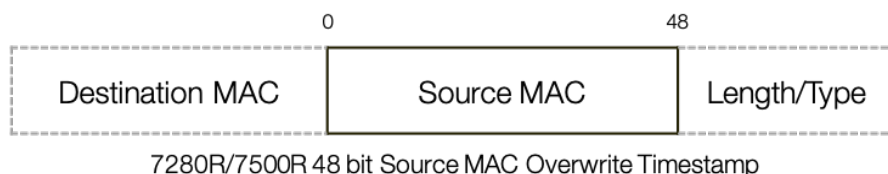
In order to recreate the real time of the frame's arrival the analytics engine/consumer must calculate the serialization time and apply a negative offset to the timestamped value. For example, -61ns for a 64 byte packet received on a 10GbE interface.

While the lower order 32 bits of the timestamp are captured on the ingress pipeline, the higher order 32 bits are captured on the egress pipeline. This creates the potential for a small rollover window as the packet moves between ingress and egress pipelines (especially in situations of heavy queueing/congestion), in such cases the timestamp will be 4 seconds into the future. It is recommended that the analytics engine/consumer be developed to disregard timestamps which vastly differ from those of the following/preceding frames.

**B) The 48 bit encapsulated timestamp** adds 12 bytes (96 bits) and is self-contained within the Ethernet header. The timestamp includes the 48 low-order bits of the UTC time. Providing the consumer is also running PTP (and therefore synchronised within several seconds) the remaining 16 higher order bits can be calculated and appended by the consumer, removing the need for Keyframes.



**C) The 48 bit overwritten timestamp** does not add any size to the packet, instead rewriting the contents of the source MAC field. The timestamp includes the 48 low-order bits of the UTC time. Providing the consumer is also running PTP (and therefore synchronised within several seconds) the remaining 16 higher order bits can be calculated and appended by the consumer.



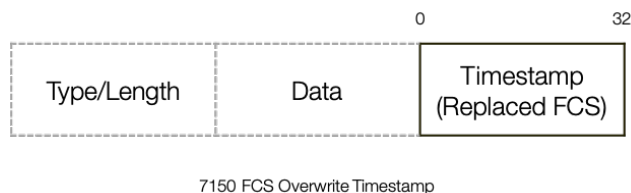
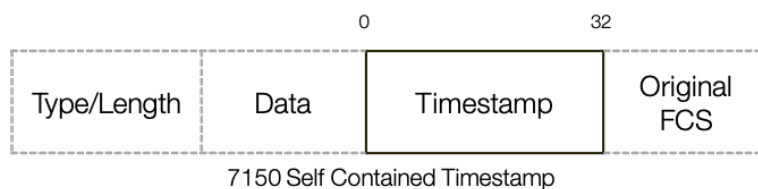
Identical to the 64 bit timestamp, both 48 bit timestamp variants are captured at the end of the ingress pipeline, the applied timestamp will therefore have the same considerations as the one mentioned in the previous 64 bit section. In order to recreate the timestamp at the point of ingress, the consumer will have to calculate and offset the serialization time for the first 256 bytes from the received timestamp.

## Timestamping on the 7150 Series

The Arista 7150 Series supports tap aggregation, timestamping and packet brokering for 1/10/40G interfaces. It provides significant flexibility around packet manipulation, filtering and flow based steering.

The 7150 series discipline the time stamping clock using hardware assisted PTP and also support transparent or boundary clock operation for scaling a PTP infrastructure. Timestamping is supported on traffic replicated via a mirror session when the device is used as a production switch, or on ingress traffic when deployed in Tap Aggregation mode.

The 31 bit 7150 timestamp header can be used in both self-contained and overwrite modes. Specifically, the timestamp can be inserted before the FCS in the ethernet trailer or can be configured to overwrite the FCS. The mode selected results in either 4 byte (insert) or 0 byte (overwrite) being added to the total length of the frame.



The 31 bit timestamp is based on a free-running 350Mhz counter which wraps every 6.135 seconds. In order to correlate the timestamp to full UTC time the 7150 will periodically send keyframes. Keyframes include the full UTC time and the current free-running counter value. Any Consumer looking to resolve timestamps to full UTC must have the appropriate logic to interpret and correlate both the 7150 timestamp and keyframe.

7150 series devices capture timestamps on the last byte of a frame on the ingress port, therefore in order to calculate the UTC value at the point of packet ingress the analytics engine must offset the serialization delay which is a pure function of packet size and the 7150 ingress interface speed.

## Arista Device Selection

Arista provides a rich product portfolio supporting the monitoring and telemetry requirements of a wide variety of deployments, with use cases from application performance monitoring to more demanding compliance requirements, such as RTS25. While a certain amount of overlap exists between these products, each has a specialization denoted in the following table.

Arista Platform to Use Case Mapping	
Use Case	Solution
Replacing SPAN or inline passive tap	7130L Series
Timestamping to nanosecond accuracy	7130L or 7150 Series
Timestamping to picosecond accuracy	7130L Series
Tap aggregation	7130L, 7150 or 7280R Series
Packet brokering/manipulation/filtering	7150, 7280R or 7500R Series

The latest addition to the Arista Portfolio, the 7130L series, provides high resolution timestamping and guaranteed 'on-the-wire' data replication while also enabling both the Tap and the Aggregation tier to be consolidated into a single device, reducing physical footprint as well as CapEx and OpEx.

The best of breed monitoring solution features Arista 7130L series paired with a 7280R or 7500R series acting as a packet broker. While the former provides the replication, timestamping and aggregation functions, the packet broker performs filtering, steering and 2nd tier aggregation functions. This creates a one stop solution which can reliably and accurately replicate data, deliver it to exactly (and only) the tools that specifically need to consume that data, while providing the exact same level of accuracy as would be achieved when connecting a discrete analytics engine to every single network tap in the network.

#### Santa Clara—Corporate Headquarters

5453 Great America Parkway,  
Santa Clara, CA 95054

Phone: +1-408-547-5500

Fax: +1-408-538-8920

Email: [info@arista.com](mailto:info@arista.com)

Ireland—International Headquarters  
3130 Atlantic Avenue  
Westpark Business Campus  
Shannon, Co. Clare  
Ireland

Vancouver—R&D Office  
9200 Glenlyon Pkwy, Unit 300  
Burnaby, British Columbia  
Canada V5J 5J8

San Francisco—R&D and Sales Office 1390  
Market Street, Suite 800  
San Francisco, CA 94102

India—R&D Office  
Global Tech Park, Tower A & B, 11th Floor  
Marathahalli Outer Ring Road  
Devarabeesanahalli Village, Varthur Hobli  
Bangalore, India 560103

Singapore—APAC Administrative Office  
9 Temasek Boulevard  
#29-01, Suntec Tower Two  
Singapore 038989

Sydney—R&D Office  
Level 5, 11 York Street  
Sydney, NSW 2000  
Australia



Copyright © 2019 Arista Networks, Inc. All rights reserved. CloudVision, and EOS are registered trademarks and Arista Networks is a trademark of Arista Networks, Inc. All other company names are trademarks of their respective holders. Information in this document is subject to change without notice. Certain features may not yet be available. Arista Networks, Inc. assumes no responsibility for any errors that may appear in this document. 03/19