Christian Hopps
LabN Consulting, LLC

IP Traffic Flow Security

Improving IPsec Traffic Flow Confidentiality

Why?

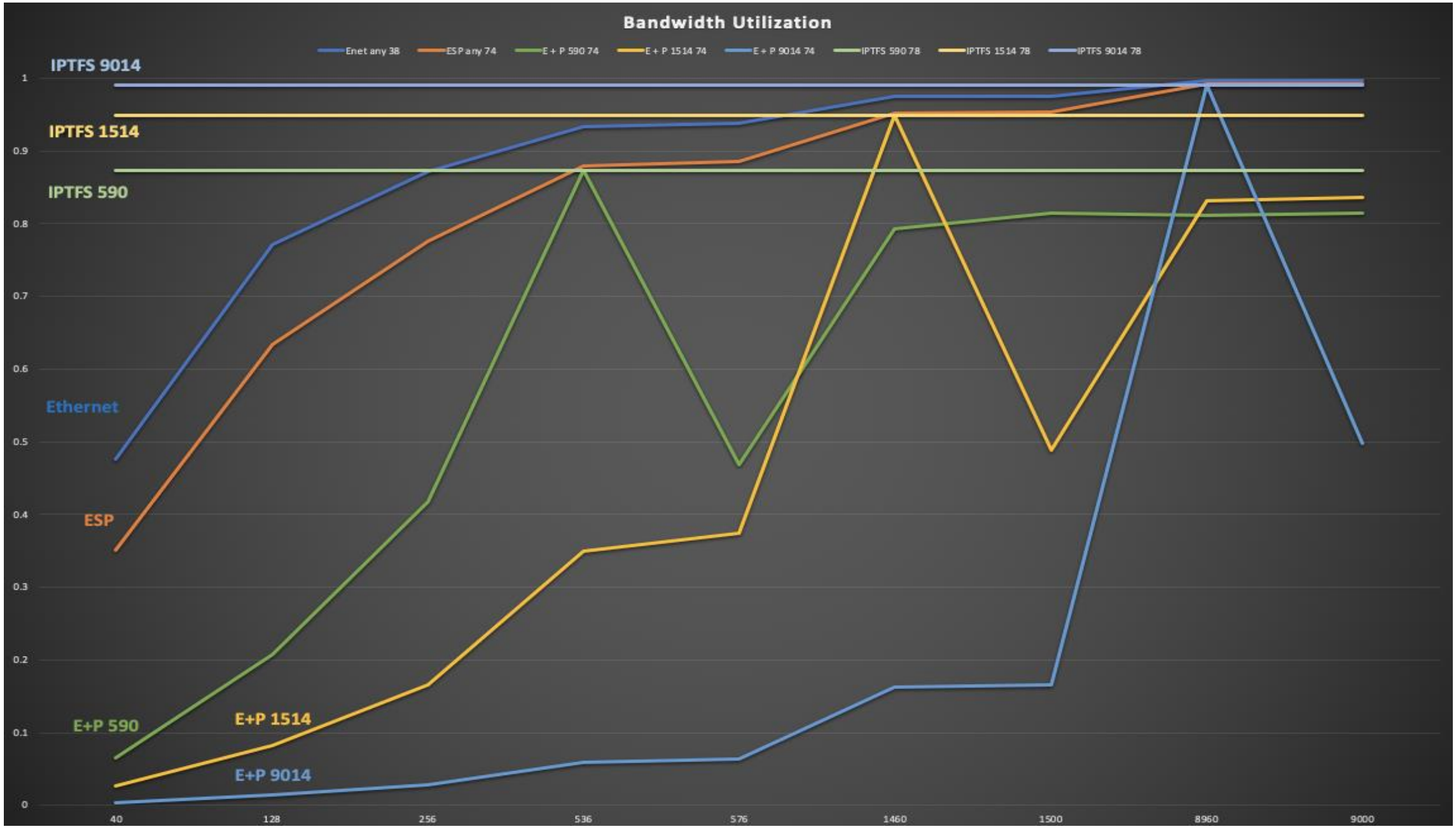
- Traffic Analysis is the act of extracting information about data being sent through a network (RFC4301, [AppCrypt]).
  - Need to protect against this.
- One may directly obscure data using encryption (IPsec/ESP).
- However, the traffic pattern itself exposes information due to variations in its shape and timing ([AppCrypt], [I-D.iab-wire-image], [USENIX]).
- Hiding the size and frequency of traffic is referred to as Traffic Flow Confidentiality (TFC) in RFC4303.

Current Available Option: ESP + Padding

- RFC4303.
- Send fixed-sized ESP packets with padding.
  - Each ESP packet can only carry 0 or 1 IP packets padded to fixed size.
- Sub-optimal performance.
  - Increased latency.
  - Low bandwidth.
    - For many inner packet sizes, the reduction is drastic.

Proposed IPsec Improvement - IP-TFS

- Continue to use IPsec/ESP
- Fragment and aggregate IP packets into new IPsec/ESP payload.
- Minimal latency increase.
- Constant High Bandwidth.
  - Higher than raw Ethernet for small to medium inner packet sizes.


Bandwidth Utilization
Enet any 38
ESP any 74
E + P 590 74
E + P 1514 74
E + P 9014 74
IPTFS 590 78
IPTFS 1514 78
IPTFS 9014 78
IPTFS 9014
IPTFS 1514
IPTFS 590
Ethernet
ESP
E+P 590
E+P 1514
E+P 9014
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
40
128
256
536
576
1460
1500
8960
9000

Key Design Points

- Improve on existing IPsec (ESP + Padding) option.
  - Fragment and Aggregate inner packets.
- Fixed-size encapsulating packets.
- Constant send rate.
- Unidirectional.
- Congestion Controlled and Non-CC operating modes.
- Uses IPsec/ESP.
- [Optional] IKEv2 Additions.
- Minimize configuration required.

IPsec Transport

- Use IPsec/ESP (encrypted encapsulation) as transport.
- Input packets are fragmented and aggregated into IPsec/ESP.
- New IP Protocol Number for new ESP payload (framing).

Fixed-Size Packets / Constant Send Rate

- Packet size never varies.
- Packet size manual or automatic configuration.
- Can use Path MTU Discovery for automatic optimal configuration.
- Constant send rate.
- Provides for transport flow confidentiality.

Unidirectional/Bidirectional

- Data path is unidirectional.
  - Sender to Receiver.
- Congestion-Control (CC) info is sent in reverse direction.
  - Receiver to Sender.
- Configure 2 paths for bidirectional operation.

Variation Fully Allowed

- Egress must accept packets at any rate.
- Egress must accept packets of any size.
- IPSec tunnels can start in normal "IP Mode", transition to IP-TFS.
  - SA reset required to leave IP-TFS mode.

Congestion Controlled (CC) Mode

- Packet send rate adjusted, as packet size fixed.
  - Congestion causes packet drops not byte drops.
- CC Info sent from Receiver to Sender using IKEv2.
- Sender uses CC algorithms to modify packet send rate.
- CC algorithm a local choice.
  - No need to standardize.
- Circuit breaking supported.
- ECN supported, but off by default.

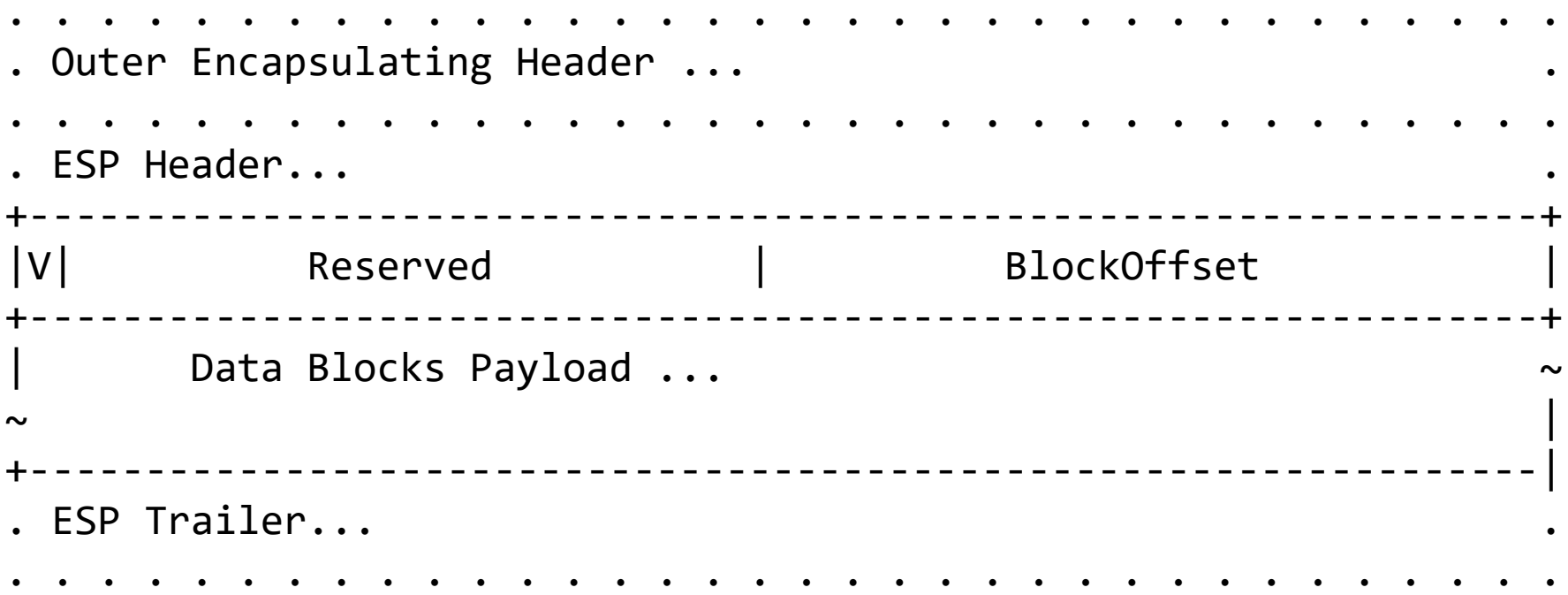
Non-Congestion-Controlled Mode

- For use when IP path bandwidth can be guaranteed.
- Packet loss reported by receiver to admin/operations.
- Optional CC info can be used to report packet loss from sender.
- Optional CC info can be used for circuit breaker.

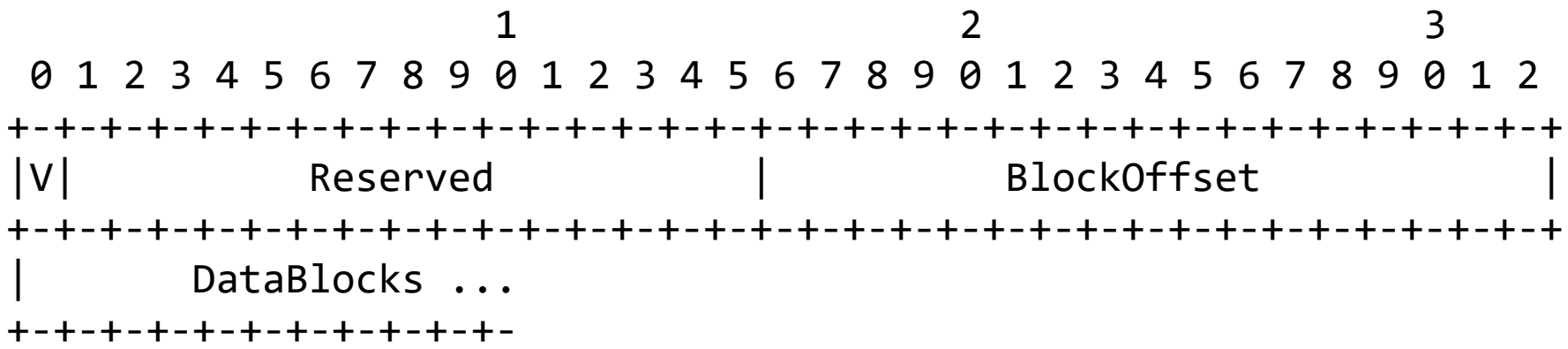
IKEv2 (CC Info)

- Use IKEv2 for CC info advertisement.
- Use INFORMATION "exchange" Notification Data.
- Periodic send interval (e.g., 1 per second).
- CFG_REQUEST/CFG_REPSPONSE used to configure interval.
- 0 interval allowed for no send.
- * Non-reliable transport (*may need to change).

IP-TFS Packet Format

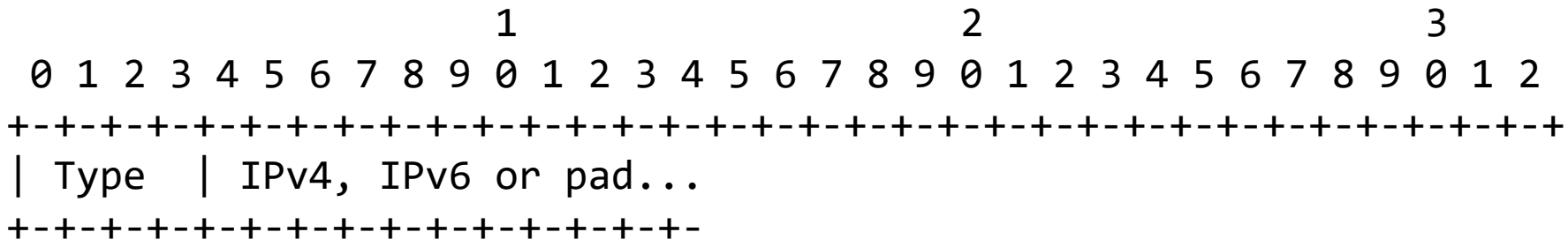


ESP Payload Format



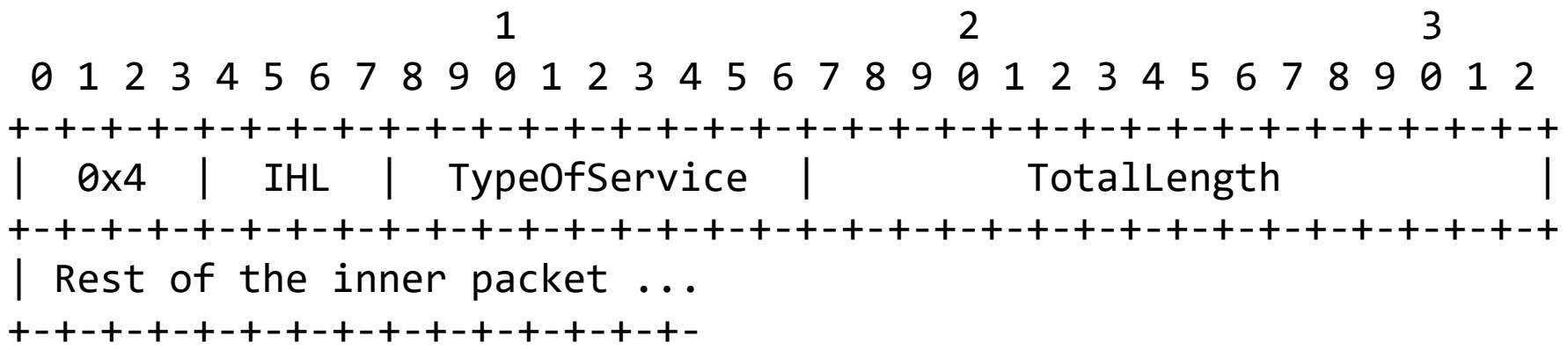
- **V** :: Version, must be set to zero and dropped if set to 1.
- **Reserved** :: set to 0 ignored on receipt.
- **Block Offset** :: This is the number of bytes before the next IP/IPv6 data block. It can point past the end of the containing packet in which case this packet is the continuation of a preivous one and possibly padding. NOTE: This can point into the next packet and yet the current packet can end with padding. This will happen if there's not enough bytes to start a new inner packet in the current outer packet.
- **Data Blocks** :: variable number of bytes that constitute the start or continuation of a previous data block.

Data Blocks



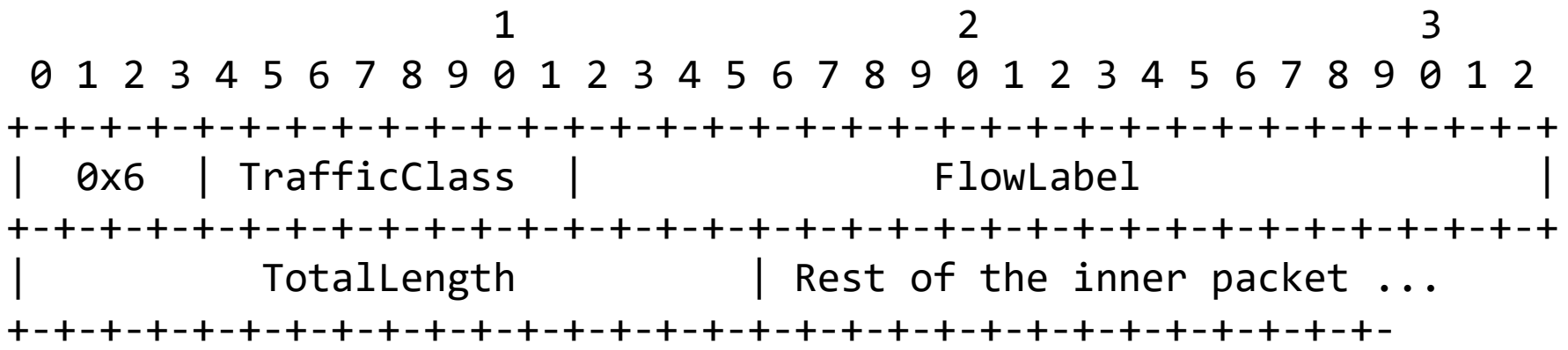
- **Version**
  - 0x0 for pad.
  - 0x4 for IPv4.
  - 0x6 for IPv6.

IPv4 Data Blocks



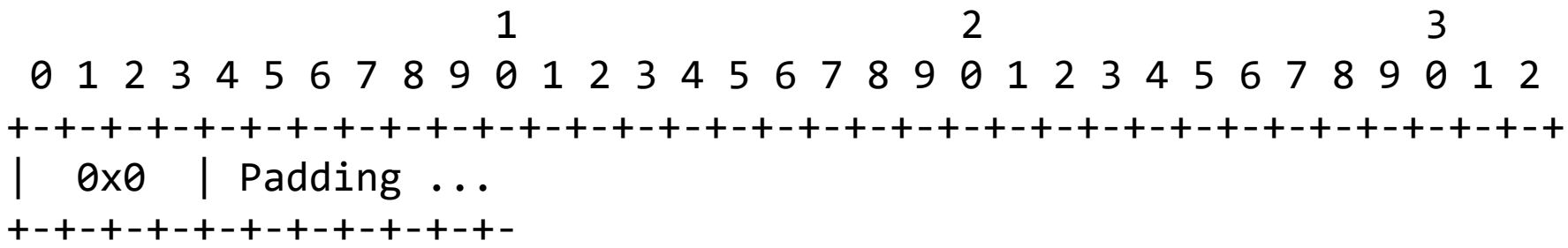
- **Version** :: 0x4 for IPv4.
- **Total Length** :: Length of the IPv4 inner packet.

IPv6 Data Blocks



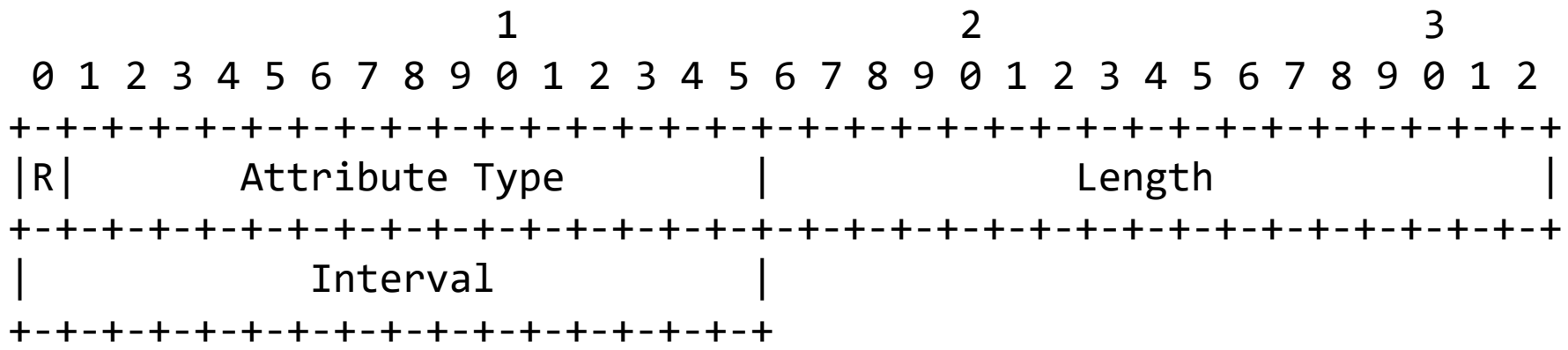
- **Version** :: 0x6 for IPv6.
- **Total Length** :: Length of the IPv4 inner packet.

Pad Data Blocks



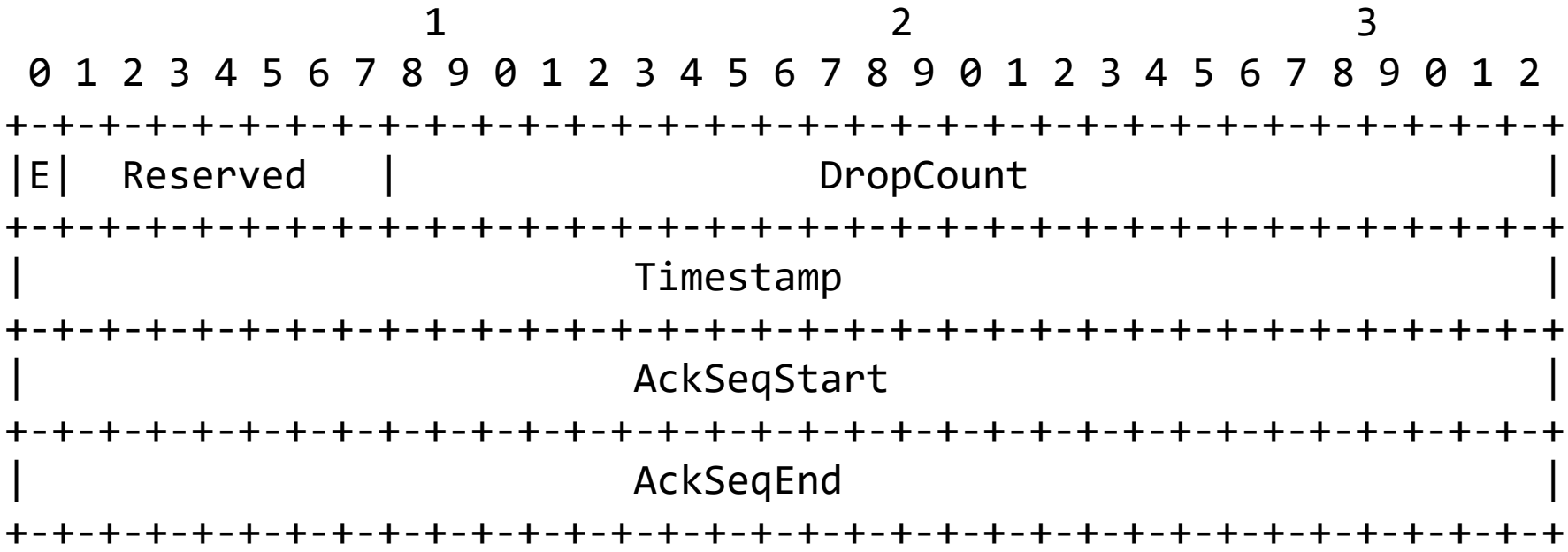
- **Version** :: 0x0 for Padding.
- **Padding** :: extends to end of the encapsulating packet.

IKEv2 Config CC Info Sending Interval Attribute



- **R**:: 1 bit set to 0.
- **Attribute Type**:: 15 bit value set to TFS_INFO_INTERVAL (TBD).
- **Length**:: 2 octet length set to 2.
- **Interval** :: 2 octet unsigned integer. The sending interval in milliseconds.

CC Info Notification Data



- **E** :: A 1 bit value that if set indicates that packet[s] with Congestion Experienced (CE) ECN bits set were received and used in calculating the DropCount value.
- **Reserved** :: set to 0 ignored on receipt.
- **DropCount**:: For ack data block this is the drop count between AckSeqStart and AckSeqEnd, If the drops exceed the resolution of the counter then set to the max value.
- **Timestamp** :: The time when this notification was created and sent.
- **AckESPSeqStart** :: The first ESP Seq. Num. of the range that this information relates to.
- **AckESPSeqEnd** :: The last ESP Seq. Num. of the range that this information relates to.

Comparison Data

Overhead Comparison in Octets

Type	ESP+Pad	ESP+Pad	ESP+Pad	IP-TFS	IP-TFS	IP-TFS
L3 MTU	576	1500	9000	576	1500	9000
PSize	540	1464	8964	536	1460	8960
40	500	1424	8924	3.0	1.1	0.2
128	412	1336	8836	9.6	3.5	0.6
256	284	1208	8708	19.1	7.0	1.1
536	4	928	8428	40.0	14.7	2.4
576	576	888	8388	43.0	15.8	2.6
1460	268	4	7504	109.0	40.0	6.5
1500	228	1500	7464	111.9	41.1	6.7
8960	1408	1540	4	668.7	245.5	40.0
9000	1368	1500	9000	671.6	246.6	40.2

Overhead as Percentage of Inner Packet

Type	ESP+Pad	ESP+Pad	ESP+Pad	IP-TFS	IP-TFS	IP-TFS
MTU	576	1500	9000	576	1500	9000
PSize	540	1464	8964	536	1460	8960
40	1250.0%	3560.0%	22310.0%	7.46%	2.74%	0.45%
128	321.9%	1043.8%	6903.1%	7.46%	2.74%	0.45%
256	110.9%	471.9%	3401.6%	7.46%	2.74%	0.45%
536	0.7%	173.1%	1572.4%	7.46%	2.74%	0.45%
576	100.0%	154.2%	1456.2%	7.46%	2.74%	0.45%
1460	18.4%	0.3%	514.0%	7.46%	2.74%	0.45%
1500	15.2%	100.0%	497.6%	7.46%	2.74%	0.45%
8960	15.7%	17.2%	0.0%	7.46%	2.74%	0.45%
9000	15.2%	16.7%	100.0%	7.46%	2.74%	0.45%

Bandwidth Utilization over Ethernet

	Enet	ESP	E + P	E + P	E + P	IPTFS	IPTFS	IPTFS
	any	any	590	1514	9014	590	1514	9014
Size	38	74	74	74	74	78	78	78
40	47.6%	35.1%	6.5%	2.6%	0.4%	87.3%	94.9%	99.1%
128	77.1%	63.4%	20.8%	8.3%	1.4%	87.3%	94.9%	99.1%
256	87.1%	77.6%	41.7%	16.6%	2.8%	87.3%	94.9%	99.1%
536	93.4%	87.9%	87.3%	34.9%	5.9%	87.3%	94.9%	99.1%
576	93.8%	88.6%	46.9%	37.5%	6.4%	87.3%	94.9%	99.1%
1460	97.5%	95.2%	79.3%	94.9%	16.2%	87.3%	94.9%	99.1%
1500	97.5%	95.3%	81.4%	48.8%	16.6%	87.3%	94.9%	99.1%
8960	99.6%	99.2%	81.1%	83.2%	99.1%	87.3%	94.9%	99.1%
9000	99.6%	99.2%	81.4%	83.6%	49.8%	87.3%	94.9%	99.1%

Latency

- Latency values seem very similar
- IP-TFS values represent max latency
- IP-TFS provides for constant high bandwidth
- ESP + padding value represents min latency
- ESP + padding often greatly reduces available bandwidth.

	ESP+Pad 1500	ESP+Pad 9000	IP-TFS 1500	IP-TFS 9000
40	1.14 us	7.14 us	1.17 us	7.17 us
128	1.07 us	7.07 us	1.10 us	7.10 us
256	0.97 us	6.97 us	1.00 us	7.00 us
536	0.74 us	6.74 us	0.77 us	6.77 us
576	0.71 us	6.71 us	0.74 us	6.74 us
1460	0.00 us	6.00 us	0.04 us	6.04 us
1500	1.20 us	5.97 us	0.00 us	6.00 us

Related Work – IEEE

- An Ethernet TFS problem statement along with high level requirements were presented to the 802.1 Security Task Force at March 2019 meeting.
  - http://www.ieee802.org/1/files/public/docs2019/new-fedyk-traffic-flow-security-0219.pdf
- The group discussed complementary amendments to 802.1AE Media Access Control (MAC) Security (MACsec) to address the requirements and fit with existing MACsec.
- Progress on the above is anticipated in upcoming interim meetings.

Running Code

- https://github.com/LabNConsulting/iptfs [will be present by meeting]
- Proof-of-concept code.
- IP in UDP tunnel encapsulation.
  - UDP stands in for ESP
- Implements new IP-TFS payload.
  - Inner packet fragmentation and aggregation using Datablocks
- Implements Congestion Control Info Reports.
  - Sent in UDP rather than IKEv2.
- Auto-adjusts send rate correctly based on congestion.
- 2 implementations (Python and C).

Open Issues

- CC Information Report transmission.
  - Message IDs use.
  - Full INFO exchange (reliable not really needed)
    - The CC data is basically telemetry that doesn't need to be reliably delivered for TFS to function correctly.
    - If reverse direction is lossy could cause TFS tunnel teardown when there is no actual issue with the tunnel traffic.
  - Would it be useful to generalize/legitimize this "in-SA" unreliable notification in IKEv2?
    - Could do this separately, and use normal exchange method for now.

Questions and Comments

References

- [AppCrypt] - B. Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C", Nov, 2017.
- [I-D.iab-wire-image] - B. Trammell, M. Kuehlewind, "The Wire Image of a Network Protocol", Nov 05, 2018
  - https://datatracker.ietf.org/doc/draft-iab-wire-image
- [USENIX] - R. Schuster, V. Shmatikov, and E. Tromer, “Beauty and the Burst: Remote Identification of Encrypted Video Streams” 26th USENIX Security Symposium, August 16–18, 2017, Vancouver, BC, Canada
  - https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/schuster