Christian Hopps
LabN Consulting, LLC

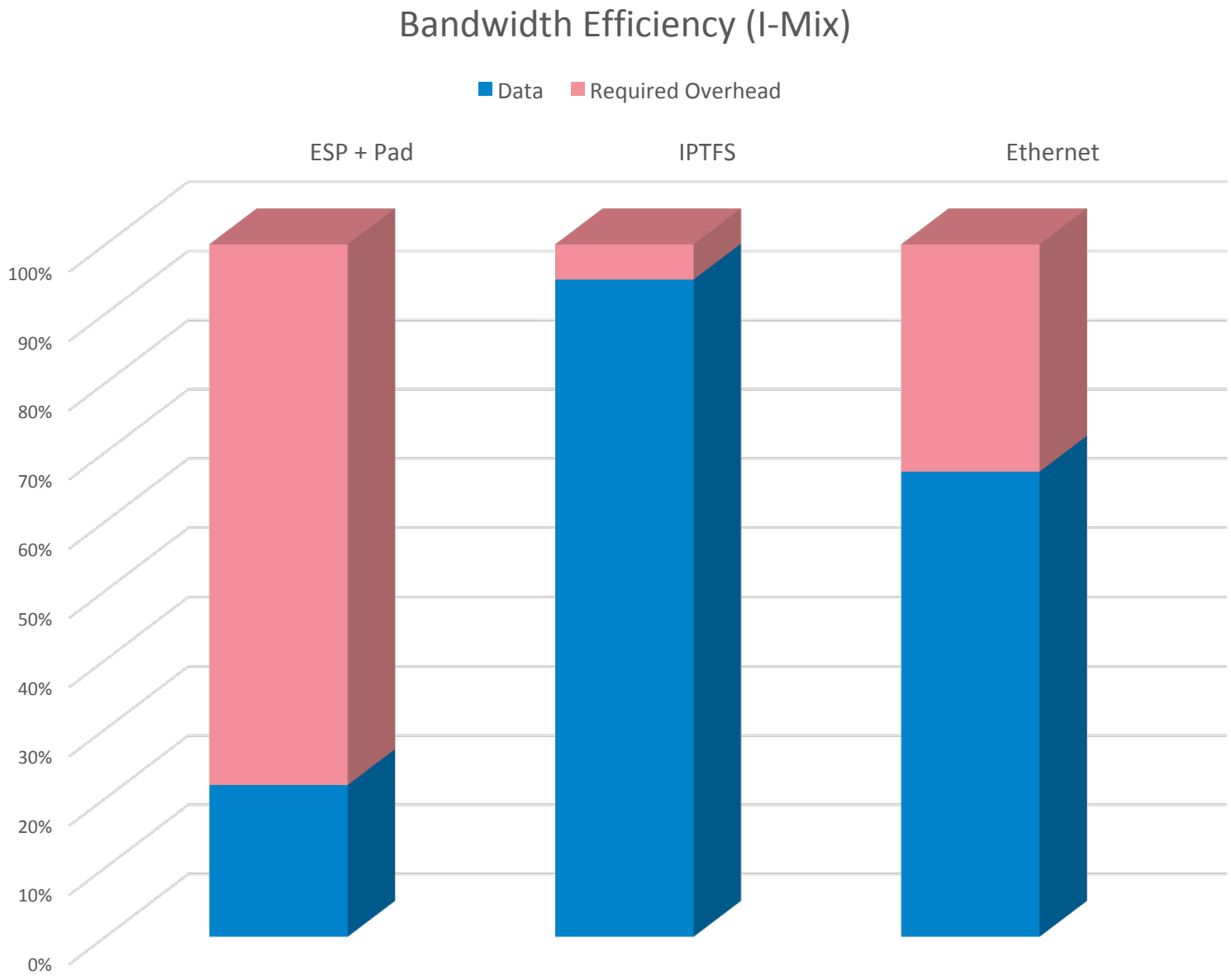
IP Traffic Flow Security

Improving IPsec Traffic Flow Confidentiality

IETF 105 – draft-hopps-ipsecme-iptfs-01 Update

Why is this Needed?

- Current Solution: ESP + Padding 1:1
- Not Deployable.



Solution Cost (I-Mix)

	ESP + Pad	IPTFS	Enet
Bandwidth Used	1Gb	1Gb	1Gb
I-Mix Throughput	219Mb	943Mb	672Mb

Update From version-00

- Updates based on comments received on mailing list and during IETF104 from ipsecme and TSV folks.
- IKEv2 TFS Type transform type introduced.
- Notification Status Message for indicating dont-fragment.
- Congestion Control information is now in-band, instead of using IKE.
- Congestion Control information and text changed to align with published TCP friendly congestion control algorithms.
- Appendix illustrating how to implement TCP friendly CC algorithm.

New IKEv2 Transform Type – TFS

- New Transform Type “TFS Type” (TBD - 6?).
  - 0 – None.
  - 1 – TFS_IPTFS_CC (congestion controlled).
  - 2 – TFS_IPTFS_NOCC (non-congestion controlled).
- Used during Child SA establishment (SA_INIT and CREATE_CHILD_SA)

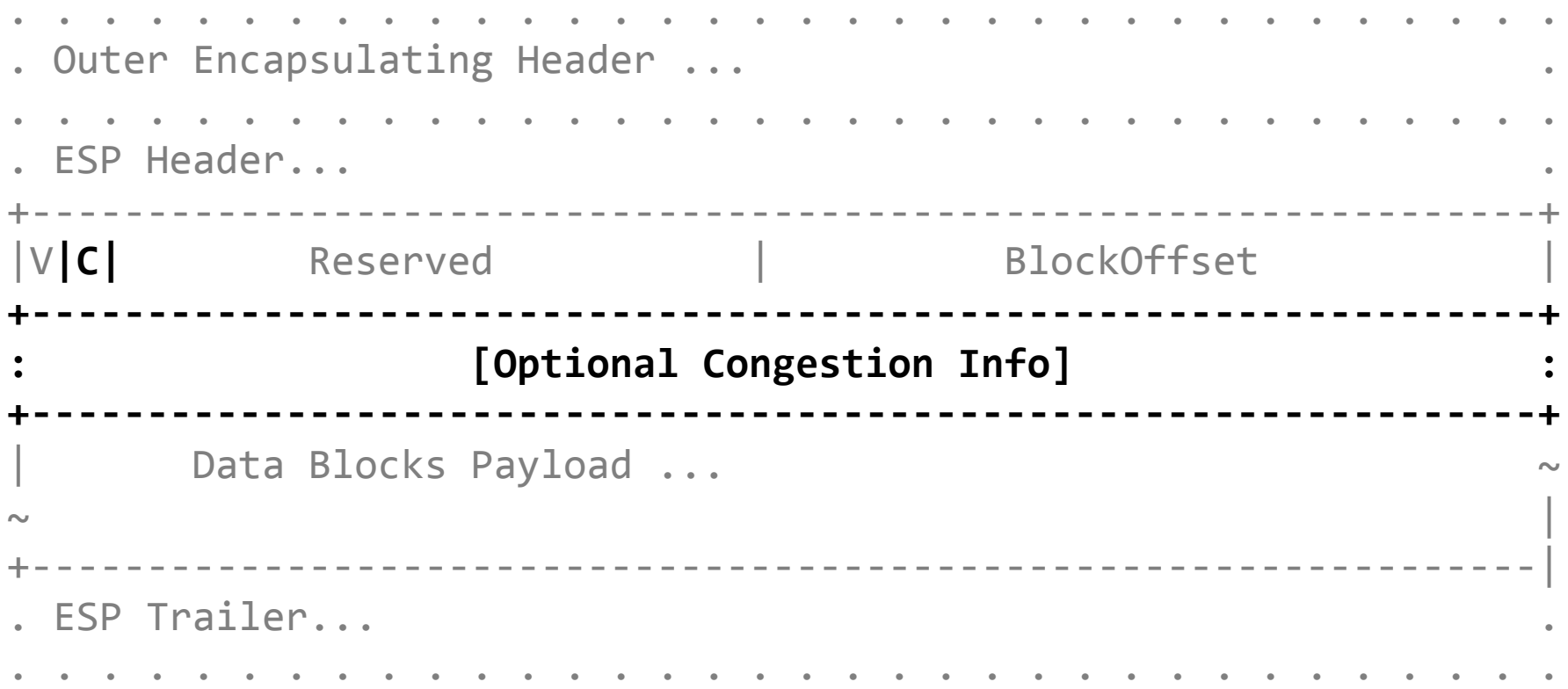
New IPTFS_REQUIREMENTS Notify Message

- Sent during SA_INIT (or CHILD_CREATE) when accepting TFS transform to indicate the sender should only aggregate and not fragment packets.
- 1 octet of flag data.

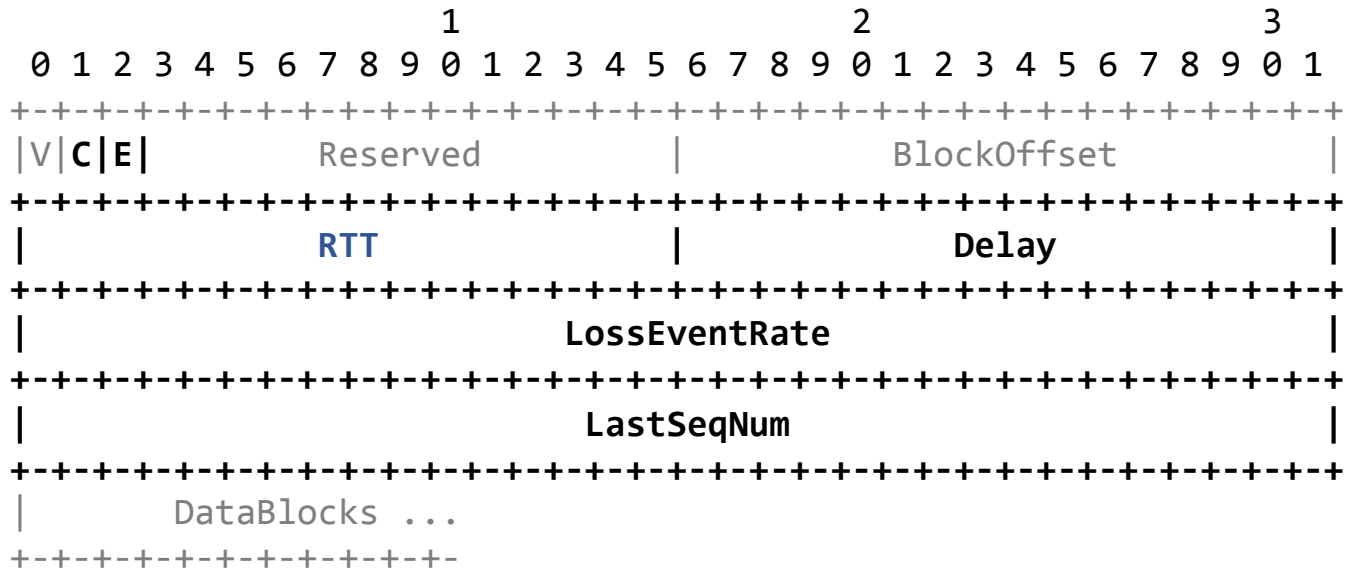
```
+-+-+-+-+-+-+-+-+
|0|0|0|0|0|0|0|D|
+-+-+-+-+-+-+-+-+
```

- 1 bit defined – “D bit” for don’t Fragment.

IP-TFS (updated) Packet Format



ESP Congestion Control Payload Format



- **C** :: Congestion Control set to 1 for this format, 0 for Non-CC.
- **E** :: ECN bit were used in calculating the **LossEventRate**.
- **RTT** :: Sender's round trip time estimate in milliseconds.
- **Delay** :: Millisecond estimate between receiver receiving **LastSeqNum**, and sending this info.
- **LossEventRate** :: 1/LossEventRate is the receivers calculation of the current loss event rate
- **LastSeqNum** :: The latest sequence number received by the receiver.

TCP Friendly Congestion Control

- Update the Congestion Control section to align with RFCs
  - RFC5348 - TCP Friendly Rate Control (TFRC): Protocol Specification
  - RFC4342 - Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)
  - RFC3168 - The Addition of Explicit Congestion Notification (ECN) to IP
- Added Appendix with directions on how to implement CC.
  - Describes how to use IPTFS CC information in standard formula
    - $$pps = 1/R\sqrt{2p/3} + 12\sqrt{3p/8} \cdot p(1+32p^2)$$
  - Can be used with references RFCS to implement TCP friendly congestion control.

Summary

- Updated based on WG comments.
  - Congestion control
  - Don’t Fragment
- Update based on implementation experience.
  - IKEv2 Transform Type
  - CC Algorithm implementation guidance.
- Ready for WG Adoption?

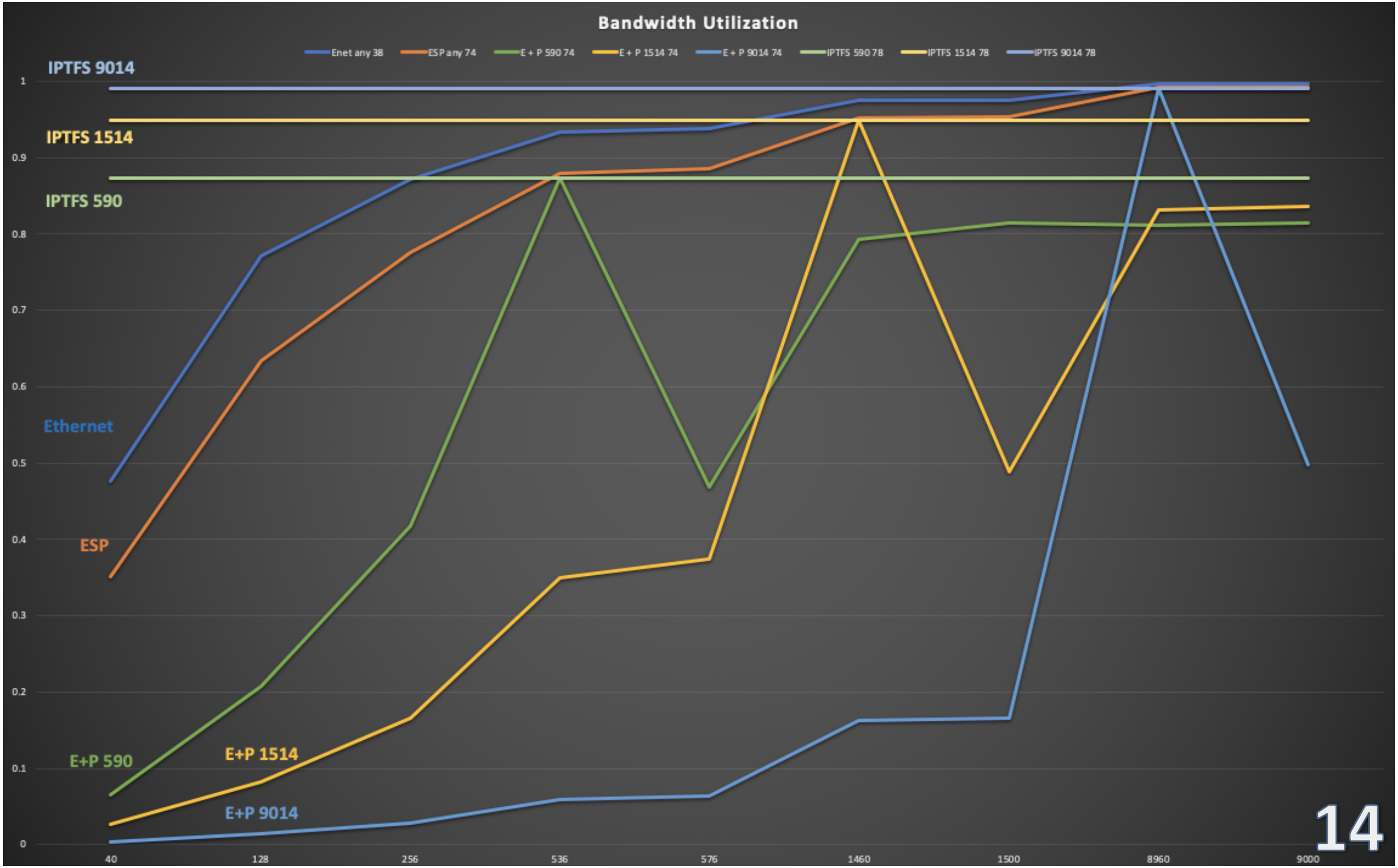
Questions and Comments

Backup Slides

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.

Comparison Data


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

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

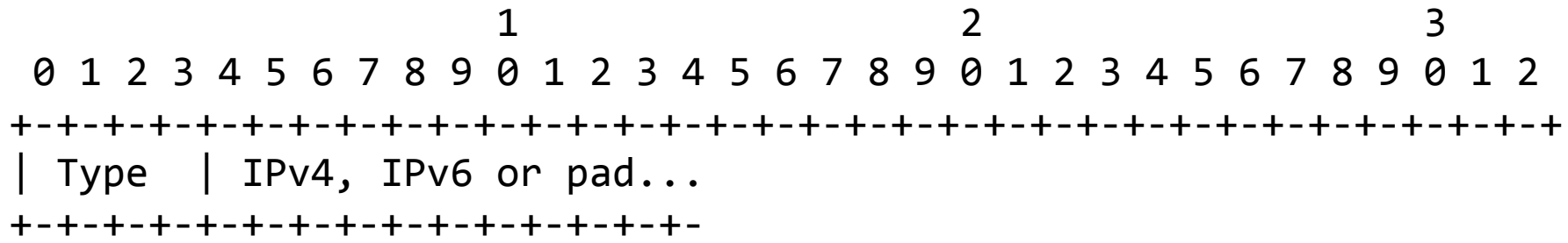
	Enet	ESP	E + P	E + P	E + P	IPTFS	IPTFS	IPTFS
	any	any	590	1514	9014	590	1514	9014
Size	38	74	74	74	74	78	78	78
40	47.6%	35.1%	6.5%	2.6%	0.4%	87.3%	94.9%	99.1%
128	77.1%	63.4%	20.8%	8.3%	1.4%	87.3%	94.9%	99.1%
256	87.1%	77.6%	41.7%	16.6%	2.8%	87.3%	94.9%	99.1%
536	93.4%	87.9%	87.3%	34.9%	5.9%	87.3%	94.9%	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.

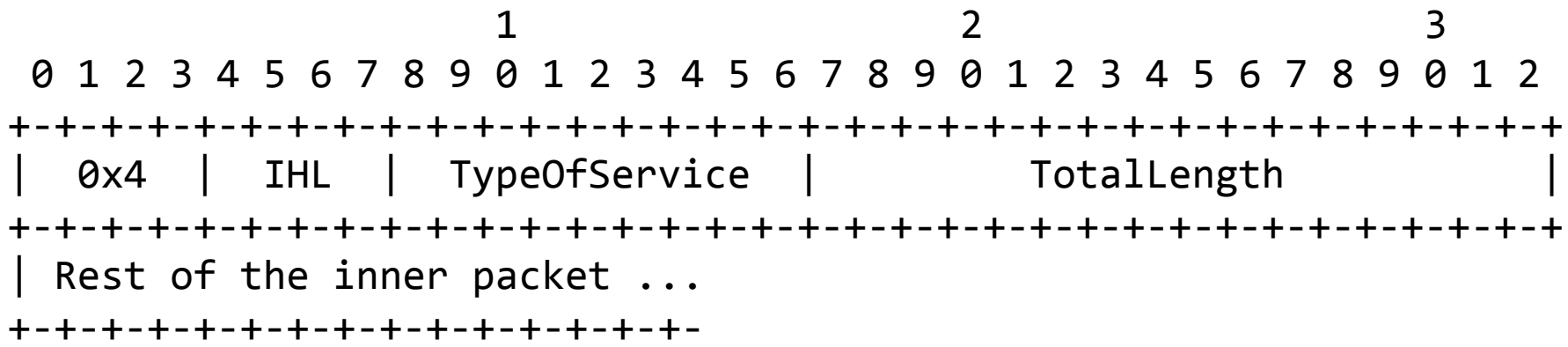
	ESP+Pad 1500	ESP+Pad 9000	IP-TFS 1500	IP-TFS 9000
40	1.14 us	7.14 us	1.17 us	7.17 us
128	1.07 us	7.07 us	1.10 us	7.10 us
256	0.97 us	6.97 us	1.00 us	7.00 us
536	0.74 us	6.74 us	0.77 us	6.77 us
576	0.71 us	6.71 us	0.74 us	6.74 us
1460	0.00 us	6.00 us	0.04 us	6.04 us
1500	1.20 us	5.97 us	0.00 us	6.00 us

Data Blocks



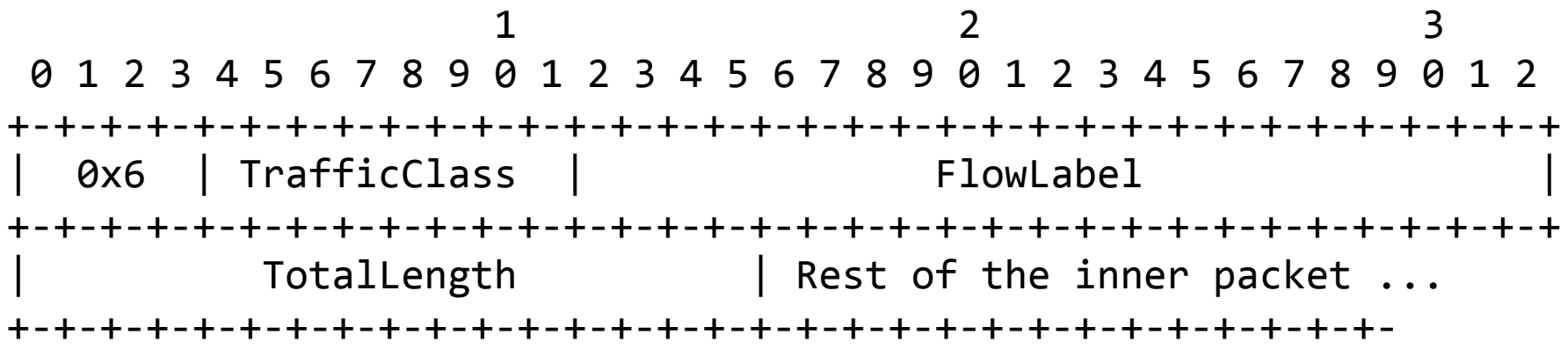
- **Version**
  - 0x0 for pad.
  - 0x4 for IPv4.
  - 0x6 for IPv6.

IPv4 Data Blocks



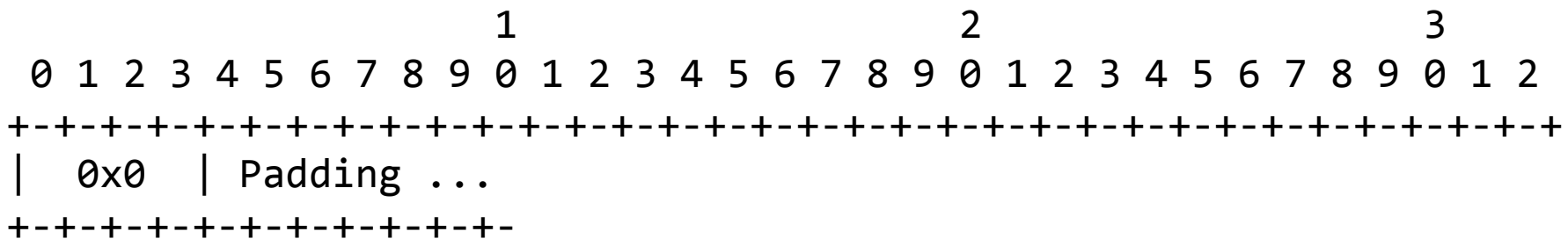
- **Version** :: 0x4 for IPv4.
- **Total Length** :: Length of the IPv4 inner packet.

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.

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).

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