Christian Hopps
LabN Consulting, LLC

IP Traffic Flow Security

Improving IPsec Traffic Flow Confidentiality

IETF 109 – “draft-ietf-ipsecme-iptfs-03”

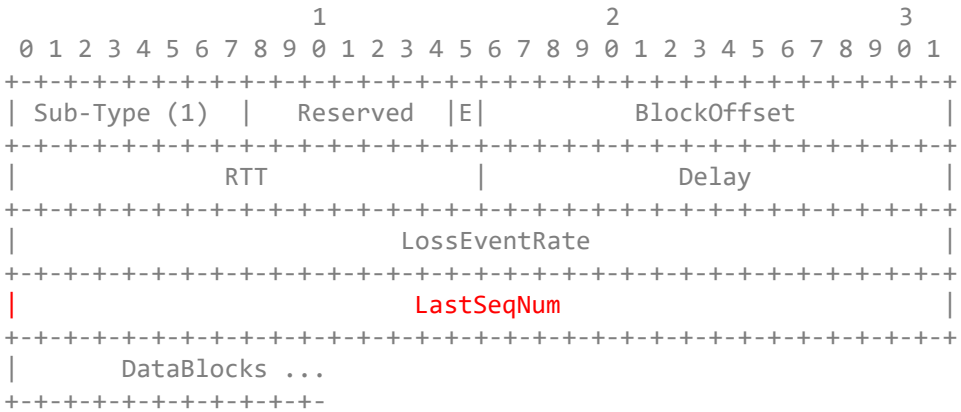
Update Since IETF 108

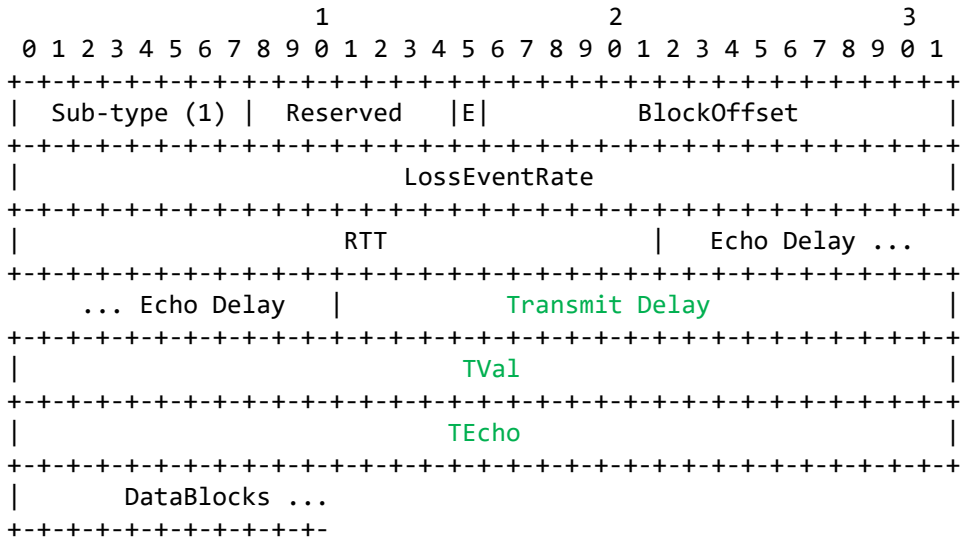
- Changes largely based on list discussions
- draft-ietf-ipsecme-iptfs-02 published Sept 30, 2020
  - Clarified fragment in following sequence number per WG feedback.
  - Add text highlighting ability to support zero-conf on receive.
    - Some WG discussion, should not be a MUST support (isn't).
- draft-ietf-ipsecme-iptfs-03 published Nov 15, 2020 (IETF109)
  - Removed Zero-Conf functionality text
  - Removed IP protocol number assignment
  - Retain ESP Payload Type, assign value of 0x5
  - Congestion Control Updated
    - Change from "last received sequence number" to more standard "timestamp and echo"
    - Added "Transmission Delay" in addition to "Echo Delay"

Mailing List Discussions

- IP Number – Early Allocation Request
  - IETF 108 Benjamin (AD) indicated do request
  - Chair (Tero) still objects as too much trouble justifying
  - State based negotiation (e.g., IKE) is not the only use case of IPsec
  - Move to backup plan, just assign an ESP payload type of 0x5
- Zero-conf receive support
  - Simple to implement
  - Useful in non-IKE scenarios to simplify configuration (good Ops)
  - Controversial for some reason
  - Removed

Updated Congestion Control Payload Format





- Better matches RFC5348, identified as part of pre-TSV review and implementation
- TVal – Opaque timestamp from sender
- TEcho – Returned TVal to sender with Echo Delay indicating held time
- Echo Delay (21 bits) microseconds – Delta time from receiving TVal to sending in TEcho
- Transmit Delay (21 bits) microseconds – The current sending rate (packet delay)
  - Combined with local transmission delay to determine minimum RTT based on logical tunnel rate.
  - Required for fast packet paths where the in network RTT is smaller

Open Issues/Last Meeting Comments

- Transport Review (congestion control)
  - Suggested by Chair (Yoav) during IETF 108
  - Latest update based on implementation experience
    - Previous version worked fine, but was overly clever and restricting
  - Had meeting with David Black, ready to move on this

Other Notes

- Open source implementation
  - Implemented in VPP and Strongswan
  - Congestion Control Supported
  - IKEv2 Supported
  - In publication process now – hoping to release next month
- Open to collaboration/interoperability testing.

Moving Forward

- All issues raised by WG addressed in current version
- Transport review seems the remaining action
- As part of WGLC?

Questions and Comments

Backup Slides

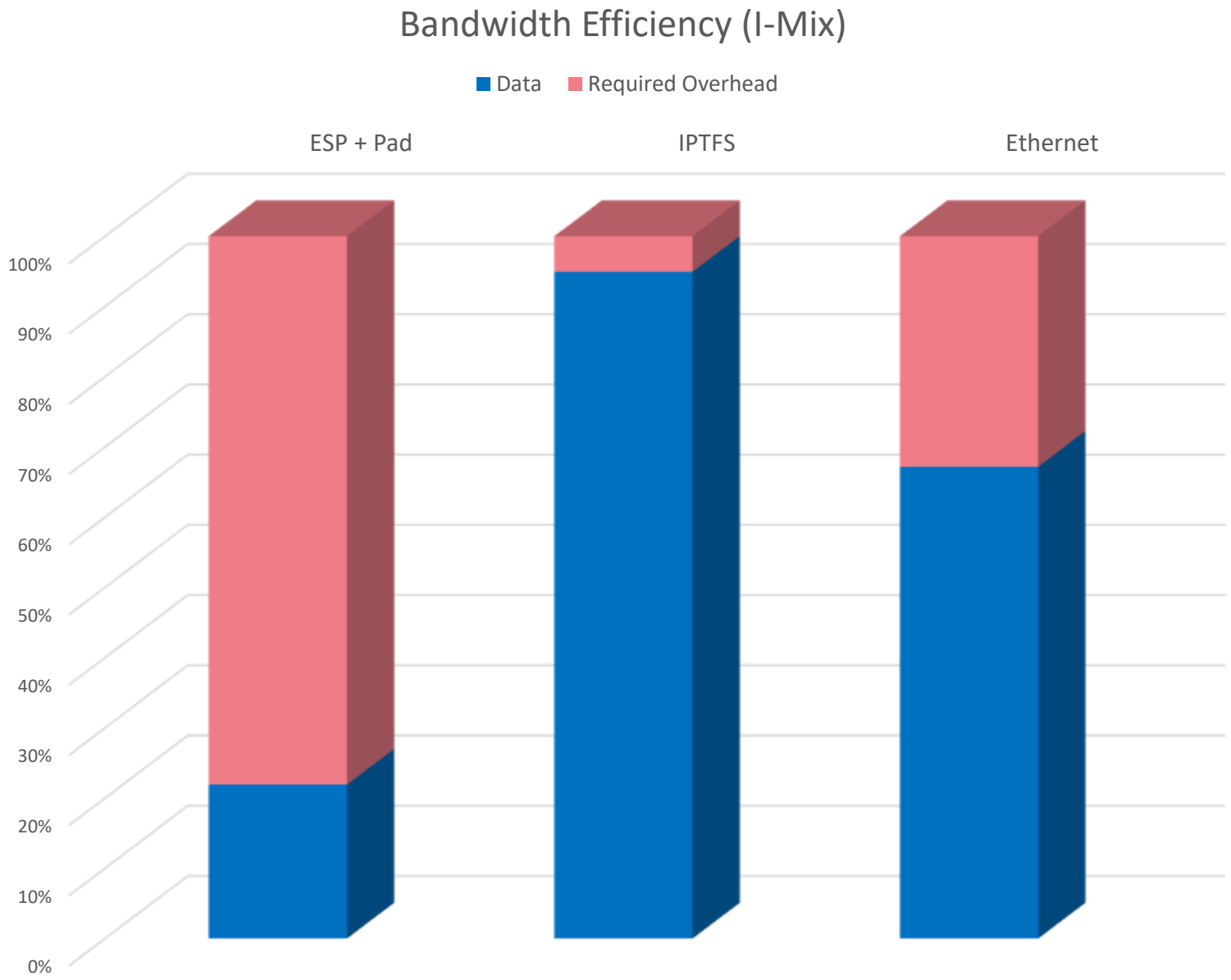
Comparison Data

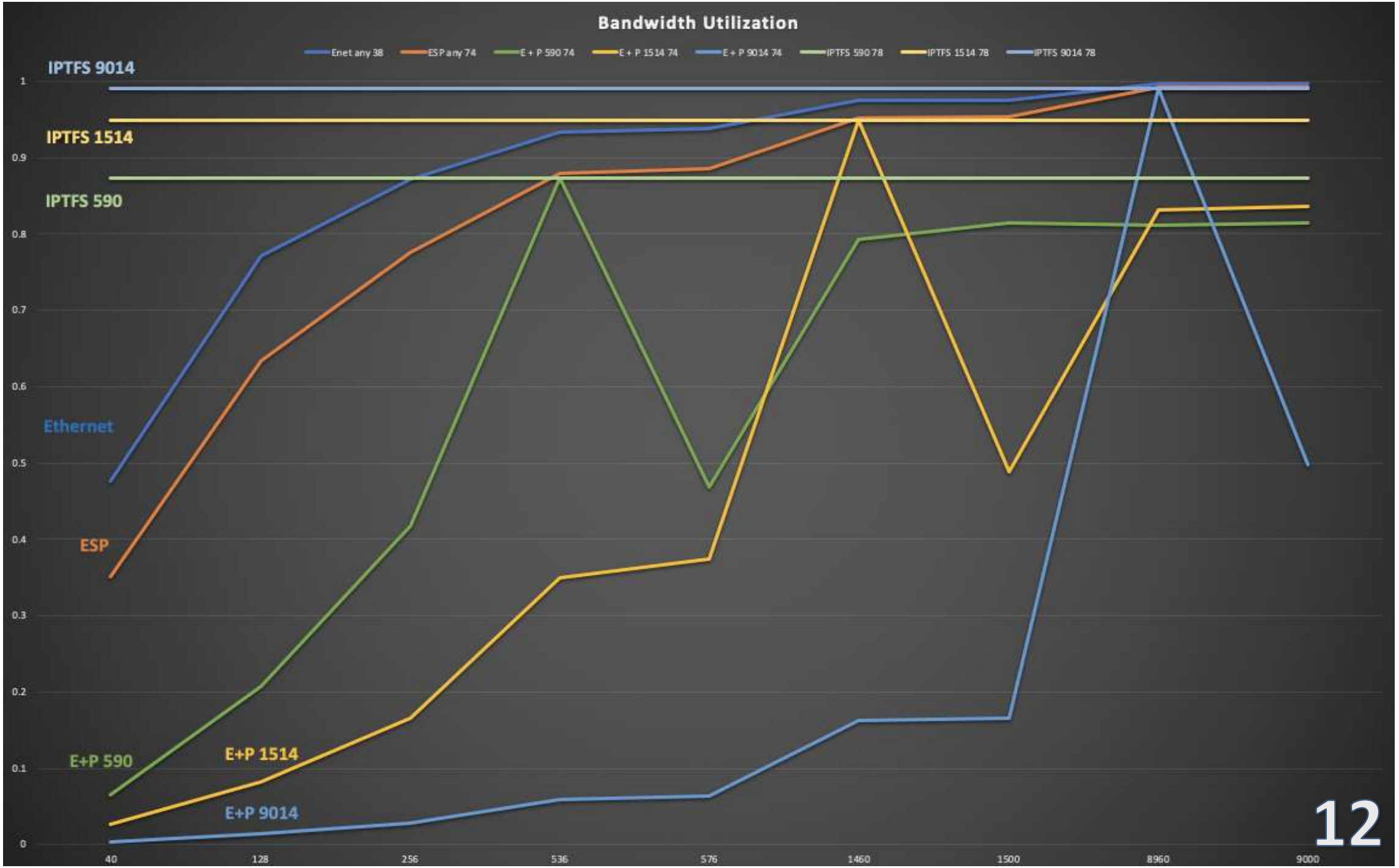
Why is this Needed?

- 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




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

Transport Mode

- Motivation is common GRE/IPsec-Transport Use
- Some interest in generic transport mode.
- What IP header fields to support
  - Simple
    - No fields – GRE Support
      - If the packet header is different then the last, pad current IPTFS out and start new one
      - If is inefficient due to frequent header differences, then use tunnel mode.
    - All Fields
      - IP header replicated inside payload for each packet
      - Similar to tunnel mode, but less efficient.
  - Complex
    - IP Header compression Ideas (deviations, etc)
      - Complex solution in need of a problem?
- Enough separable work to publish as a separate document.