# HDLC vs 64b/66b

## Analysis of transmission overhead of HDLC and 64b/66b for different types of Ethernet traffic

Vladimir Oksman
Broadcom Corporation
March 2002

# Background

- **The HDLC encapsulation (recommended by ITU-T G.993.1) uses octet stuffing for proper identification of the start and the end of the transmit packets. Due to the octet stuffing, the transmission overhead changes depending on the statistics of the transmit data. A concern was expressed that these changes may seriously impact system performance**
- **As a possible alternative, a 64b/66b coding was suggested**
- **This presentation analyses overhead characteristics of both encapsulation techniques**

# Basic approach

- For both encapsulation techniques the overhead is analyzed as a sum of two components:
  - fixed component (O_f), determined for the best possible transmit data content
  - statistical component (O_s), determined by the transmit data contents

- It is assumed that prior to encapsulation the Preamble and SDF octets of the Ethernet frame are discarded. The fixed overhead of the Ethernet frame includes 18 octets:
  - 12 octets of the Address field
  - 2 octets of the Type/Length field
  - 4 octets of the FSC field

- The minimum and maximum length of the Ethernet frame is, respectively, 64 and 1518 octets

# HDLC: fixed overhead

- **The fixed components of the HDLC overhead contains:**
  **- opening flag - 1 octet**
  **- Address and Control fields - 2 octets**
  **- FSC field - 2 octets**
  **- Closing flag - 1 octet**

  **Total: 6 octets**

- **If the transmit Ethernet frame contains N octets, the fixed overhead is estimated as:**

$$O\_f = \frac{6+18}{N+6} = \frac{24}{N+6}$$

# HDLC: statistical overhead

- **With HDLC encoding, transparent octets 7E and 7D, if occur in the data stream, are replaced by a pair of octets (7D 5E, 7D 5D, respectively). Accordingly, occurrence of *m* transparent octets in the frame will result in additional overhead of**

$$O\_s = \frac{m}{N+6}$$

- **Assuming that the probability that any octet in the Ethernet frame will get a value of a transparent octet (either 7D or 7E) equals *p,* the probability that the transparent octet will appear from 1 to *m* times, and will not appear at all, respectively, will be:**

$$P_{1-m} = \sum_{1}^{m} \binom{m}{N} \cdot p^{m} \cdot (1-p)^{N-m}$$

$$P_{m=0} = 1 - P_{1-N} = 1 - \sum_{1}^{N} \binom{m}{N} \cdot p^{m} \cdot (1-p)^{N-m}$$

# HDLC: statistical overhead

- **The average value of the statistical overhead may be estimated using the average frequency of transparent octets occurrence:**

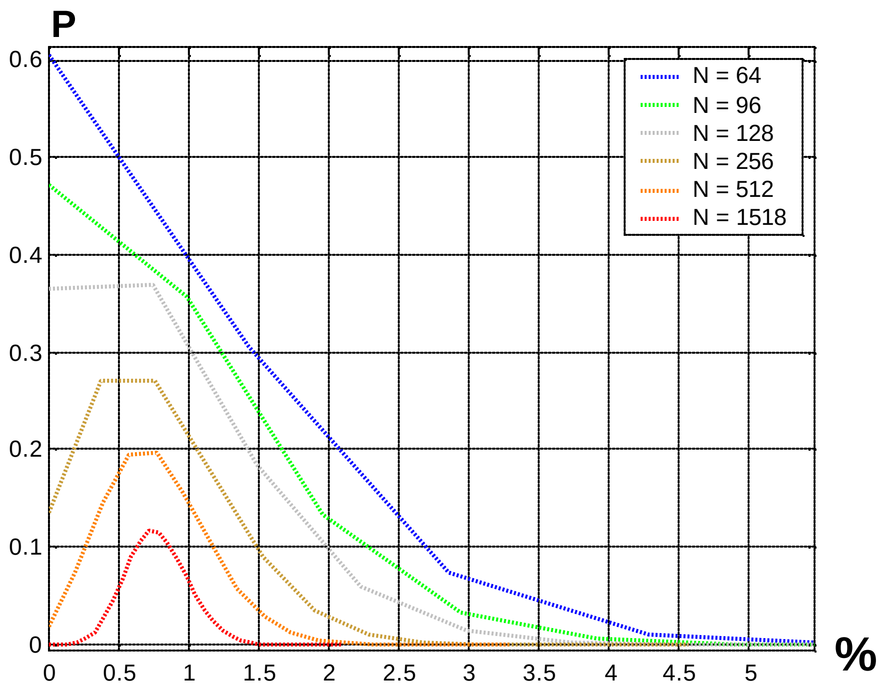
$$O\_sa \approx round\left(\frac{N \cdot p}{N+6}\right)$$

- **The equal probability of any octet in the transmit data is assumed:**

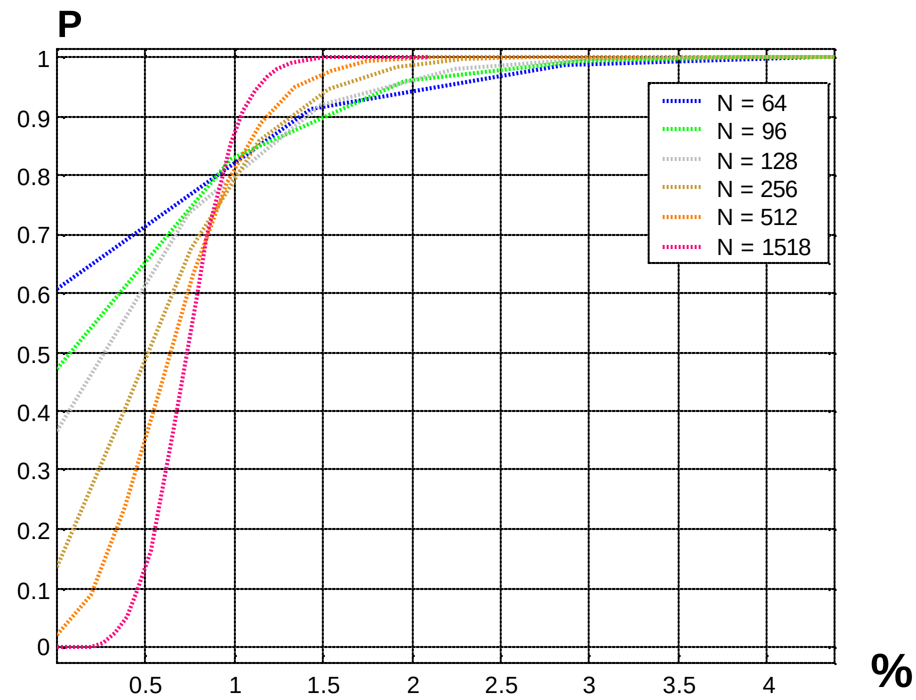
$p = 2/256 = 1/128$

# HDLC: statistical overhead

Probability distribution function



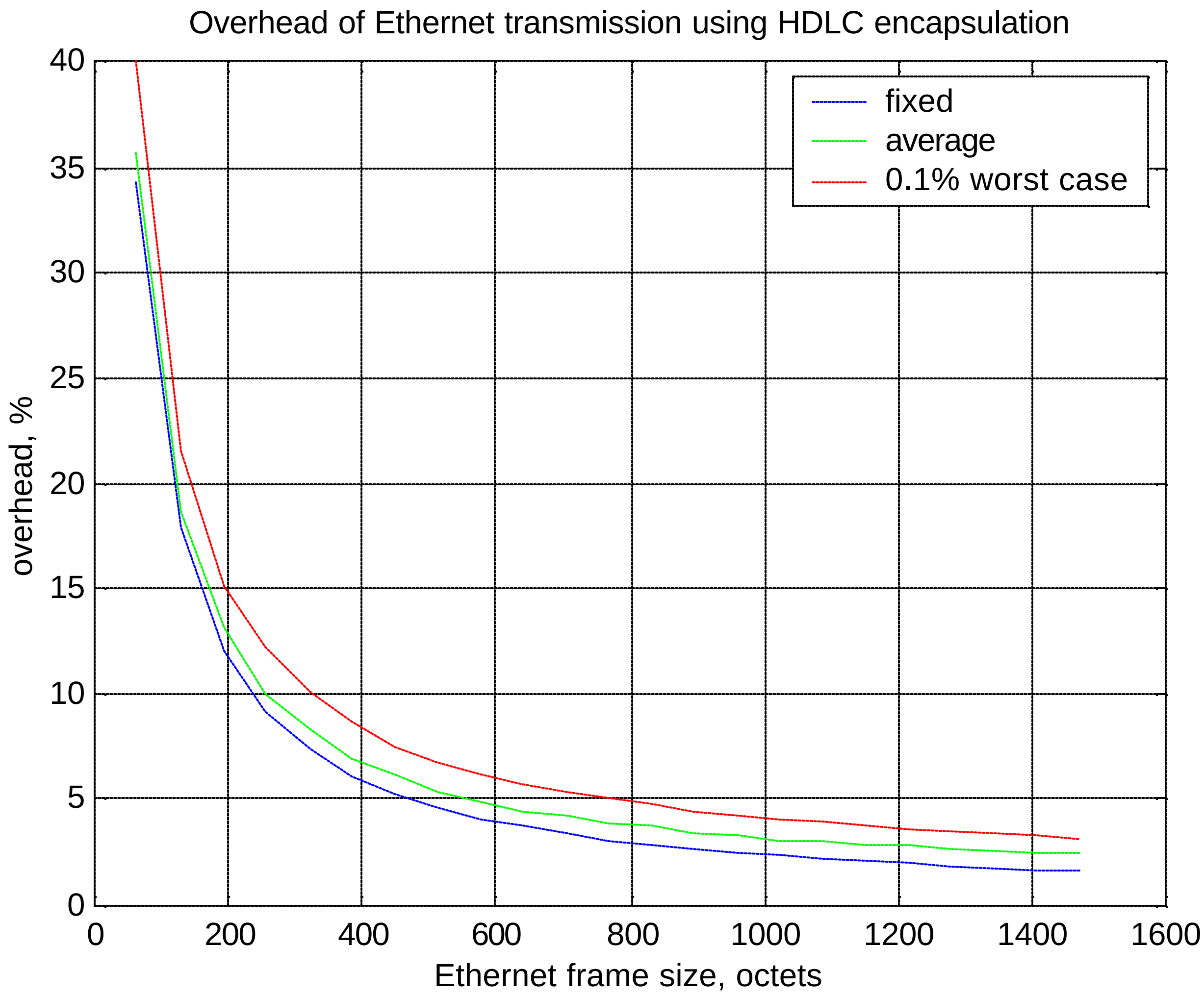

Cumulative probability function




**0.1% worst case statistical overhead**

| N | 64 | 128 | 256 | 512 | 768 | 1024 | 1518 |
|---|---|---|---|---|---|---|---|
| Overhead, % | 5.7 | 3.7 | 3.1 | 2.1 | 1.9 | 1.7 | 1.6 |

# HDLC: total overhead summary



# HDLC overhead: observations

- **Both fixed and a statistical components of the HDLC overhead depend on the length of the transported Ethernet frame, i.e. on the data traffic:**

  - the fixed overhead strongly depends on the traffic, and can vary from 1.57% for long messages to 34.3% for short messages

  - the statistical overhead is more stable and, with probability at least 0.999, doesn't exceed 1.6% for long frames and 5.7% for short frames

  - the total overhead may fluctuate almost twice due to it's statistical component (from 1.57% to 3.17%, for long messages), and almost 20 times (from 1.57% to 34.3%) due to it's fixed component

- **Traffic, but not statistical overhead, is the main factor leading to variations in the bit rate of data transport**

# 64b/66b: fixed overhead

- **Encoding rules for 64b/66b lead to the following component of the the fixed overhead:**
  **- start of packet - 1 octet**
  **- Sync preamble - 2 bits per 8 octets**
  **- End of packet - 1 octet**
  **Total: 2 octets and 1/4 octet per 8 octets**

- **If the transmit Ethernet frame contains N octets, the fixed overhead may be estimated as:**

$$O\_f = \frac{2 + 0.25 CEIL\left(\frac{N+2}{8}\right) + 18}{N + 2 + 0.25 CEIL\left(\frac{N+2}{8}\right)}$$

# 64b/66b: statistical overhead

- **The statistical component of 64b/66b overhead is due to:**
  **- Starting pad of 0-3 octets: s**ince the number of possible frame start positions is only 4, the maximum value of 3 octets should be applied for the worst case overhead estimation

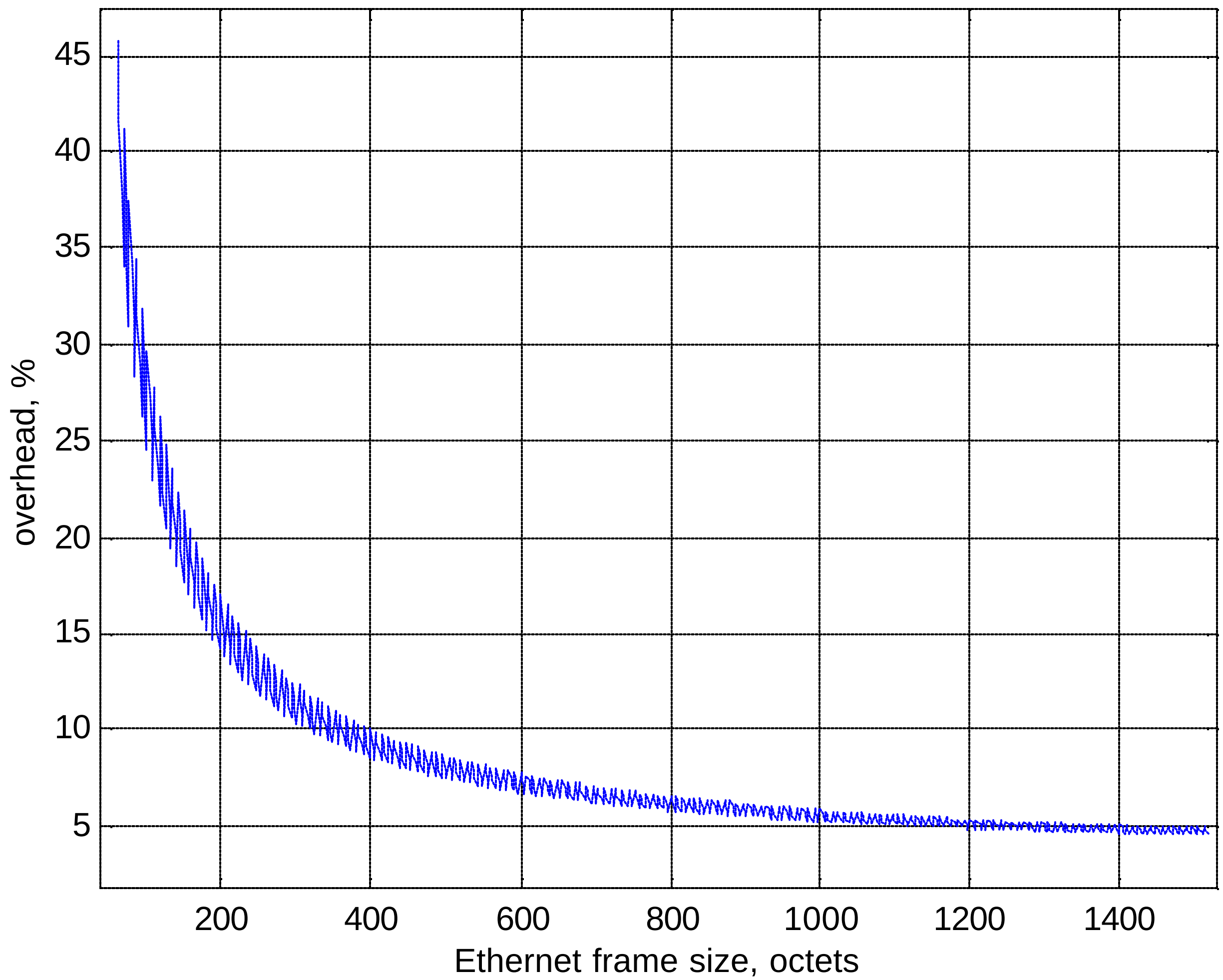
  **- Ending pad of 0-7 octets:** $E\_pad = 8 \cdot CEIL\left(\frac{N+2}{8}\right) - (N+2)$

- **The total statistical overhead, respectively, may be estimated as**:

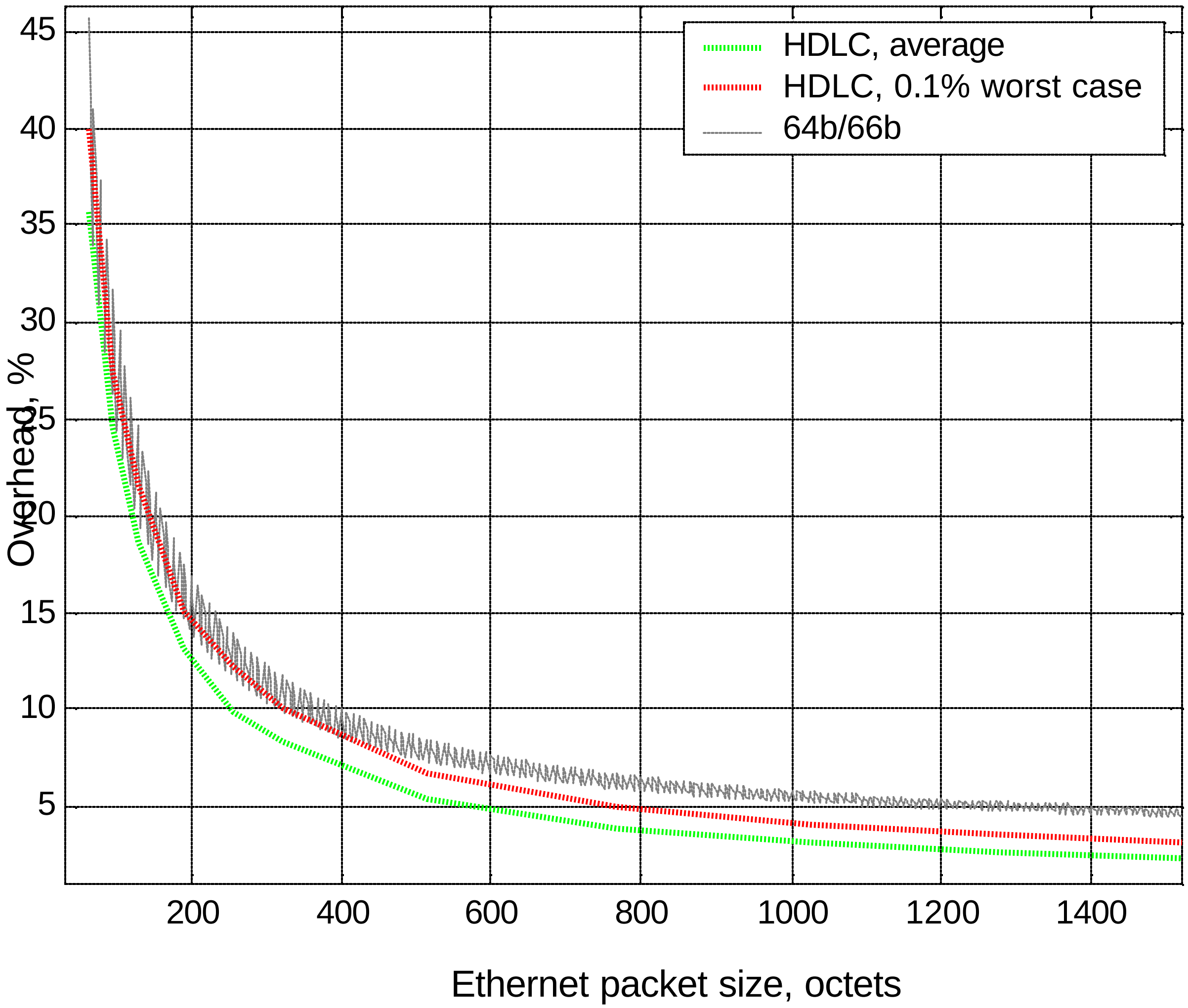
$$O\_s = \frac{3 + CEIL\left(\frac{N+2}{8}\right) - (N+2)}{N+2+0.25CEIL\left(\frac{N+2}{8}\right)}$$

# 64b/66b: total overhead summary

Overhead of Ethernet transmission using 64b/66b encapsulation



# HDLC & 64b/66b overhead summary



# 64b/66b overhead: observations

- Same as HDLC, the 64b/66b overhead strongly depends on the Ethernet traffic
- As Ethernet traffic changes, the 64b/66b overhead may fluctuate more than 10 times (4.3% and 45.6% for long and short frames, respectively)
- The 64b/66b overhead always exceeds the average HDLC overhead, and in the most cases exceeds even the worst 0.1% HDLC overhead.

# HDLC overhead explosion

- **If a particular frame contains only transparent octets, the encapsulated frame could reach almost the double size relatively to the original frame.**

- **This so called “HDLC overhead explosion” raised a concern during some discussions**

# Impact on the system

- **System Complexity**: if a particular frame is exploded, more time is required for transmission of this frame. This additional time will be handled by the standard flow control mechanism at the $\gamma$-interface. No additional complexity
- **Multi-pair operation**: if a part of the frame intended for transmission over particular pair explodes, the additional transport time over this pair will slow down the frame transport. The efficiency may be improved if distribution of data between the pairs accounts transparent octets.

# Impact on the SLA

- **Can the frame explosion impact the SLA?**
  Consider the issue based on the assumption that the SLA objective is to provide a particular bit rate (10 Mb/s) during at least 95% of the time with measurement intervals of 1s

- **Example:**
  A 10Mb/s Ethernet traffic during 1s transports 1.25 Mbyte of data. If up to 2.5% of this amount of data will be transported with a twice lower speed, the SLA will be accomplished.

  The mentioned 2.5% is 31.25 Kbyte of data containing only transparent octets - at least 20 messages of maximum length during every second. Doesn't seems real, especially accounting that data compression is widely used for file transfer

# Possible improvement

- **Use HDLC bit stuffing instead octet stuffing**
  This reduces the maximum size of the overhead explosion to maximum 20% instead 100% of the encoded data. The demerit is, however, that the coded HDLC signal may include not integer number of octets (same demerit also observed in 64b/66b)

- **Use scrambling**
  Scrambler randomizes the transported Ethernet data signal prior to HDLC encoding and significantly reduces the probability of overhead explosion

- **Use both**

# Conclusions

- **Fluctuations of HDLC overhead are not critical, and cause a rather low reduction of the transmission bit rate in all practical case considering at least 0.1% worst case statistics**
- **Packet explosion due to HDLC overhead is not expected to make any influence on the SLA and other integrated system characteristics**
- **HDLC seems to be a rather convenient choice: it demonstrates highly efficient transmission in conjunction with low complexity of the implementation**