Entropy Based Worm and Anomaly Detection in Fast IP Networks

Arno Wagner

wagner@tik.ee.ethz.ch

Communication Systems Laboratory

Swiss Federal Institute of Technology Zurich (ETH Zurich)

Outline

- Dataset
- Entropy and Compression
- Observable Effects During Worm Outbreak
- Sampling, Compression Algorithm, Performance
- Summary

The DDoSVax Dataset

Project URL:
`http://www.tik.ee.ethz.ch/~ddosvax/`

- NetFlow v5
- From all SWITCH border routers
- About 60.000.000 flows/hour
- ∼200k internal, ∼800k external IPs/hour
- Unsampled
- Stored in full since March 2003

Entropy, Kolmogorov Complexity

- **Entropy:**
  Expected information in an object from a set with a specific selection probability for each element.
- **Kolmogorov Complexity:**
  Information in a specific (binary) object.

Kolmogorov Complexity cannot really be measured.
Entropy can be estimated by compression.

Entropy Estimation by Compression

1. Represent data object in binary form
2. Compress
3. Entropy estimation [bit/bit] is

$$min\left(1, \frac{\text{compressed size}}{\text{original size}}\right) \in]0, 1]$$

- Relatively bad accuracy (worst case: encrypted data), but not that far off
- Usable for relative comparisons

Effects of Worm Outbreak

Normal traffic:

- Many contact few (servers)
  many contact many (P2P)
- Connections are mostly successful (bidirectional)
  $\Rightarrow$ Flow set is mostly symmetric

Worm outbreak traffic:

- Few hosts contact many
- Most connections fail
  $\Rightarrow$ Flow set is asymmetric

Generic properties of *any* scanning worm!

Compression statistics

Most promising (determined experimentally)

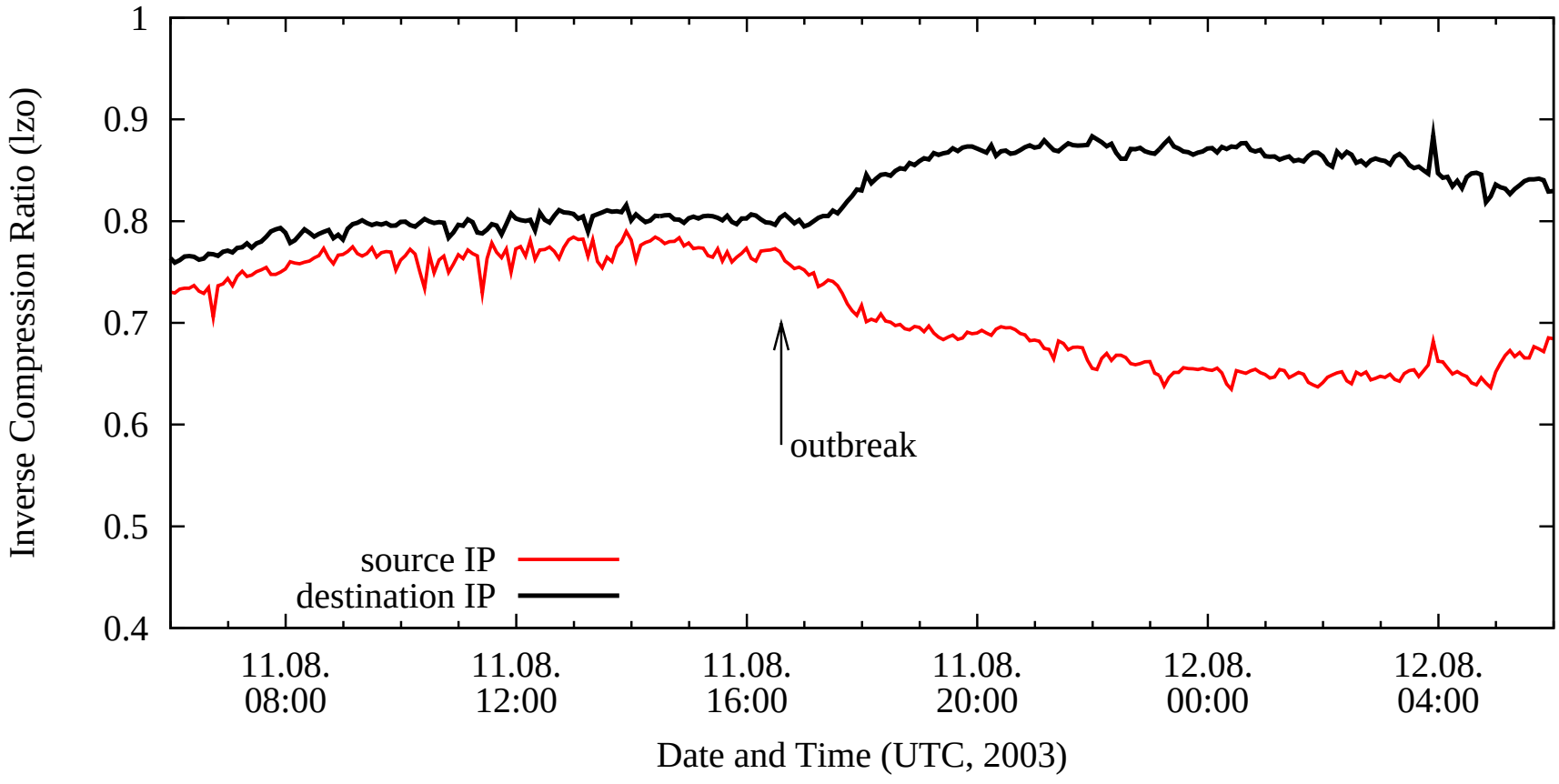
- Source IP
- Target IP
- Source port
- Target port

All 4 fields are converted to host byte-order and compressed individually per measurement interval of, e.g., 5 minutes.

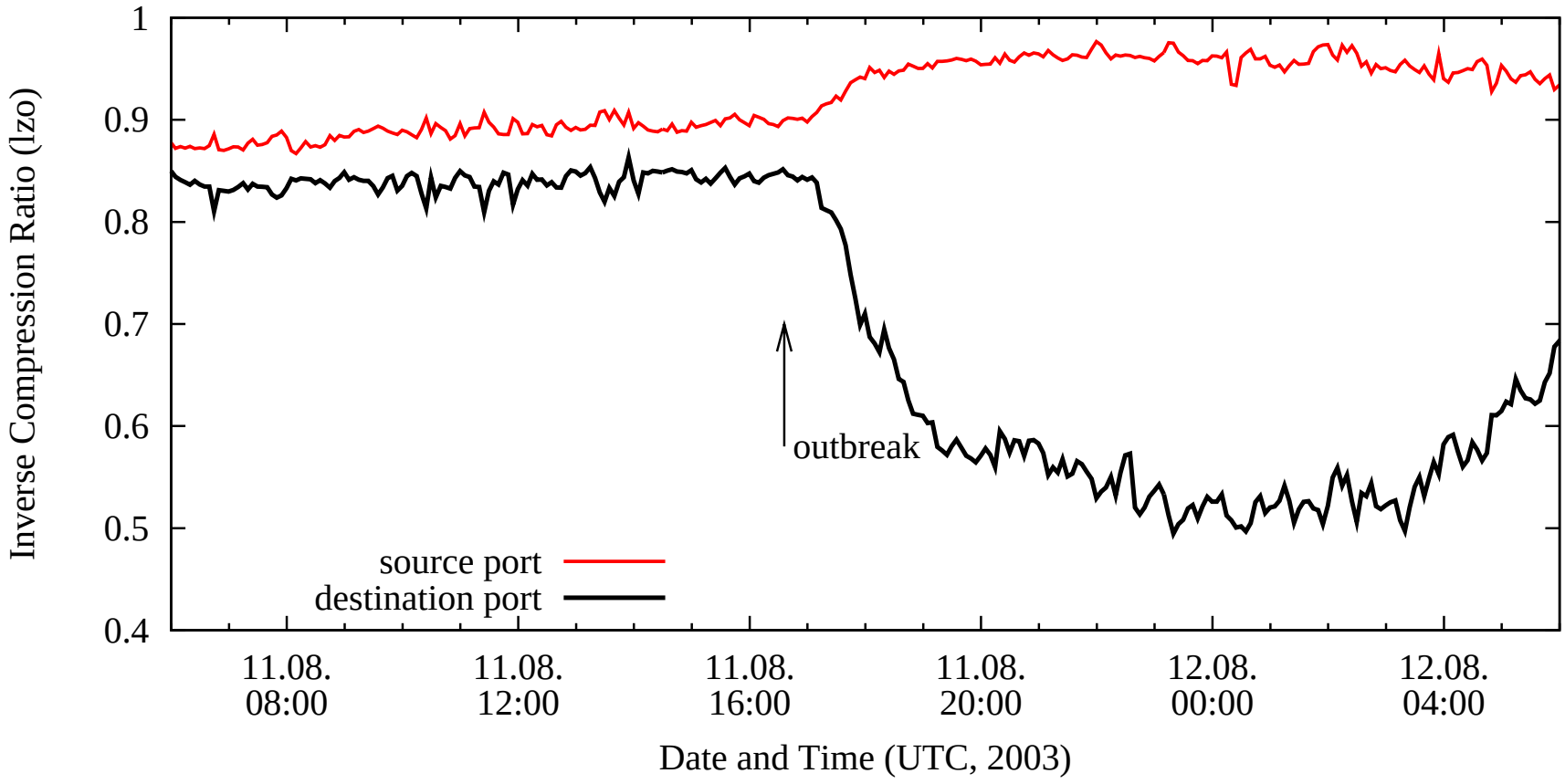
Example 1: Blaster Worm

- First observed August 11th, 2003
- Tries TCP connection to port 135
- Random target selection with local preference
- Initially infected 200’000. . .500’000 hosts in 8 hours

Blaster: IP Compressibility



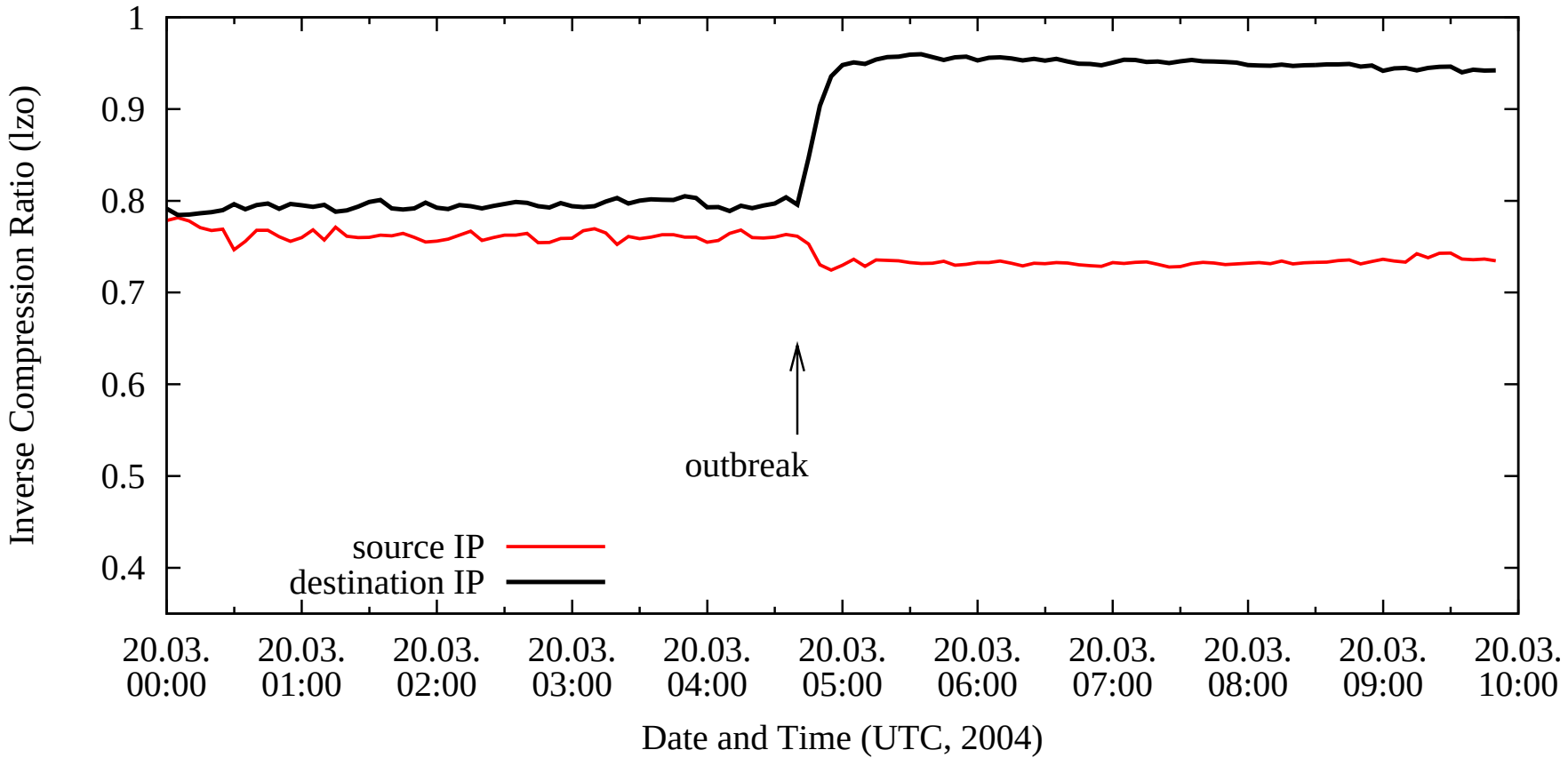
Blaster: Port Compressibility



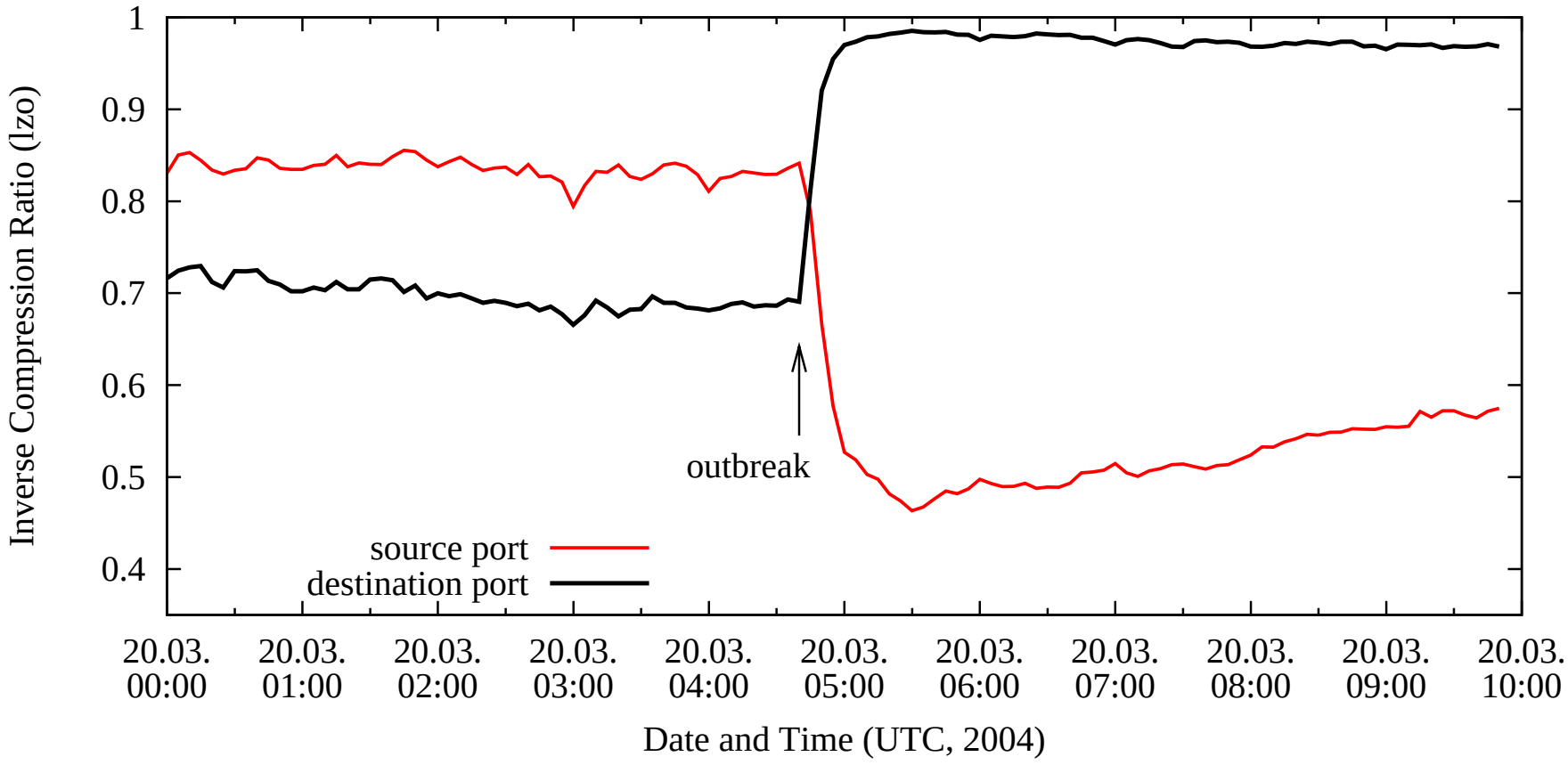
Example 2: Witty Worm

- First observed March 20th, 2004
- Infects a firewall product
- Random target selection
- Sends UDP packet with random target and fixed source port
- Initially infected $\sim$12'000 computers in 75 minutes

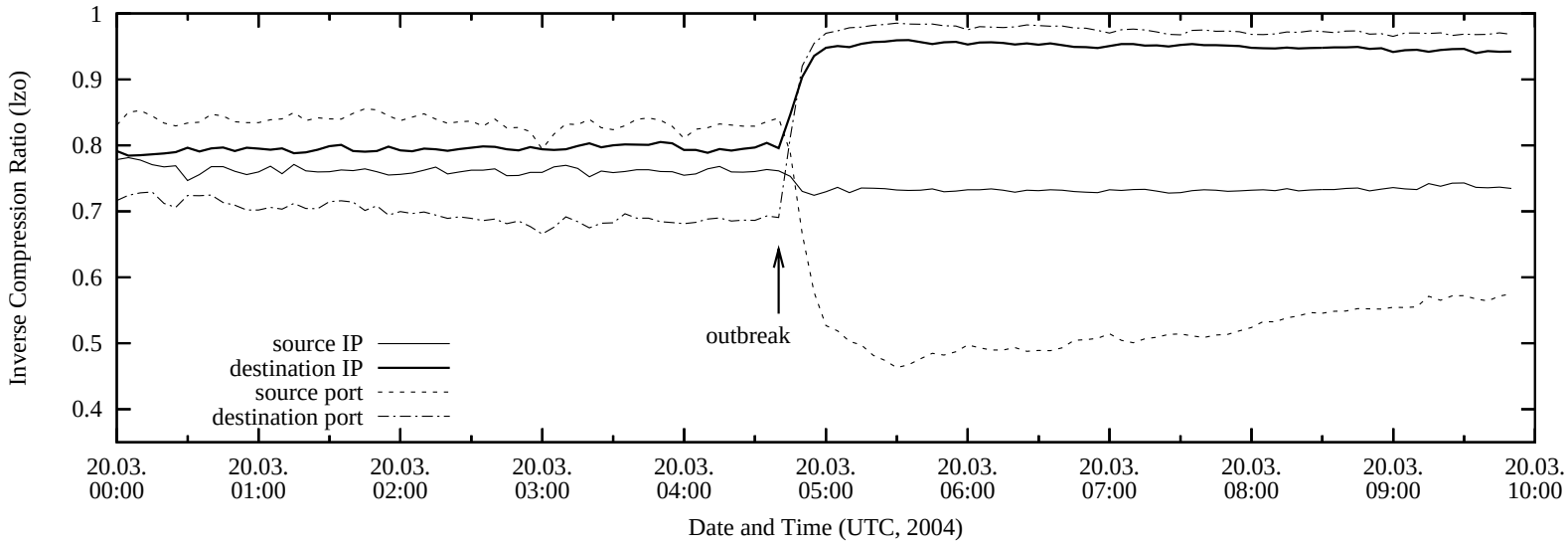
Witty: IP Compressibility

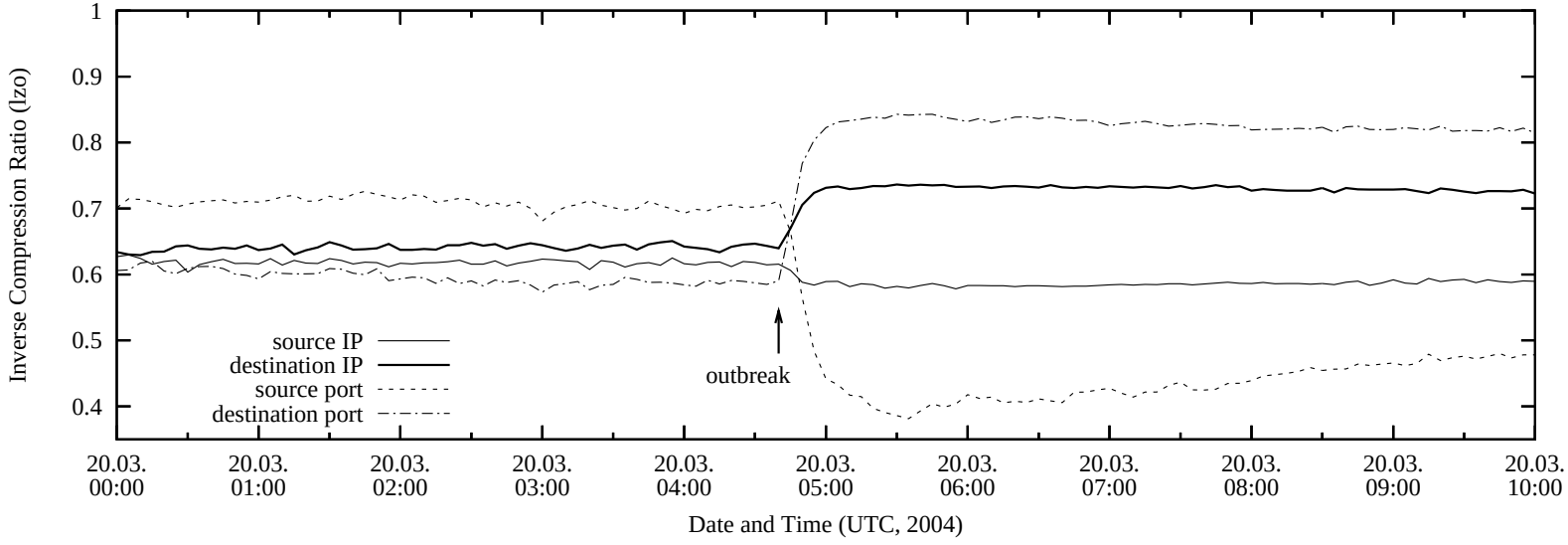


Witty: Port Compressibility



Sampling: Witty full vs. 5%

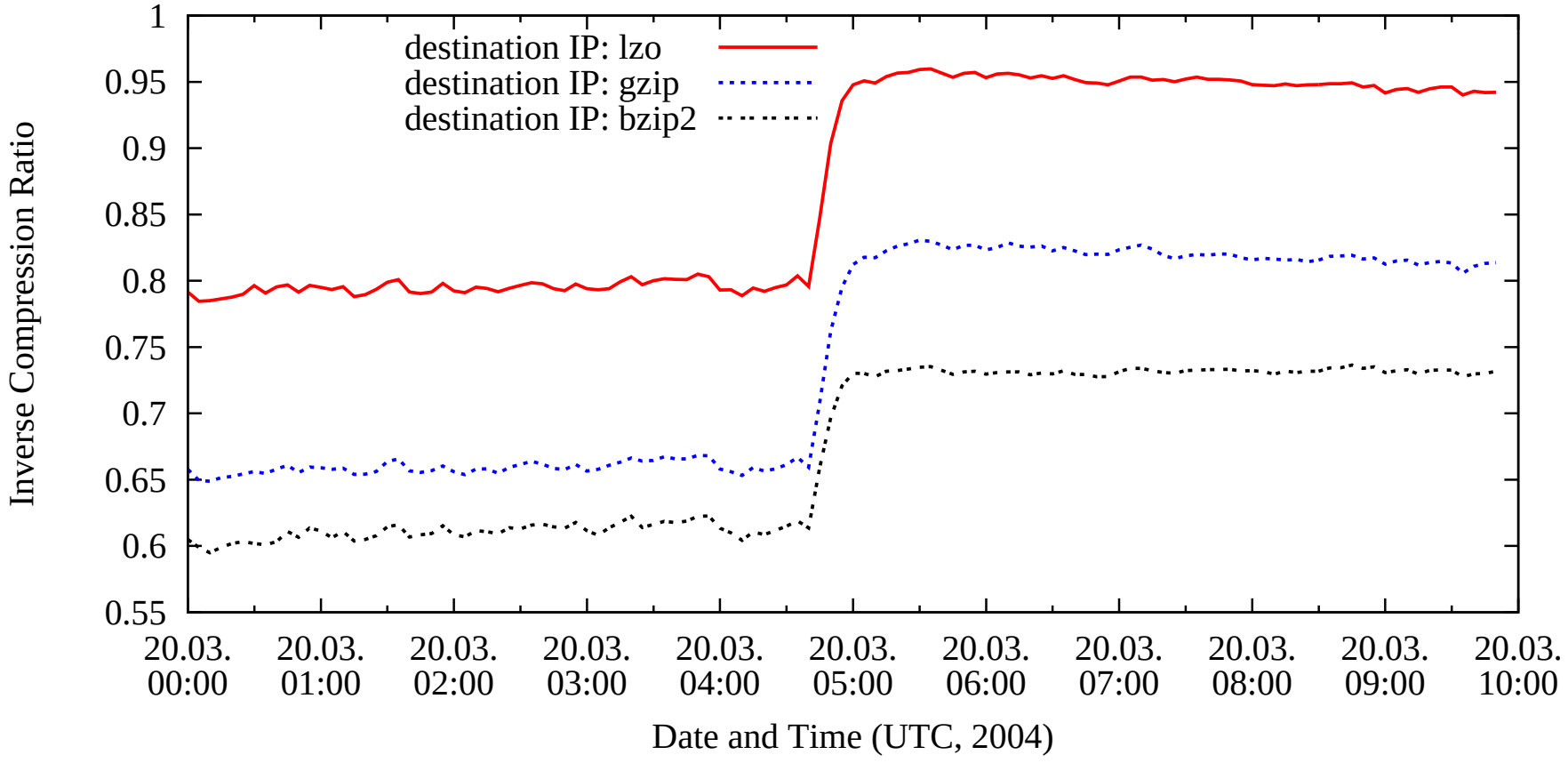




Compression Algorithms

- **lzo:** Lempel-Ziv variant
  very fast, bad compression
- **gzip:** Lempel-Ziv variant
  well-known GNU compressor, average in all respects
- **bzip2:** Burrows-Wheeler + Huffman coding
  slow, very good compression
- **Entropy estimation by value frequencies:**
  average speed, high memory needs

Compression Comparison



Resource Comparison

Method (Library)	CPU time / hour (Athlon 2800+) (60'000'000 flows/hour)
bzip2 (libbz2-1.0)	169 s
gzip (zlib1g 1.2.1.1-3)	52 s
lzo1x-1 (liblzo1 1.08-1)	7 s

Method	Memory per compressor instance
bzip2	7600 kB
gzip	256 kB
lzo1x-1	64 kB

Summary

- Generic approach
- Works for any fast random scanning worm
- Scales linear for CPU and I/O, constant for memory
- Not suitable for slower worms
- Only limited information about worm details

Thank You!