Large-Scale Scanning of TCP’s Initial Window

Jan Rüth, Christian Bormann, Oliver Hohlfeld
Why look at Initial Windows?

Initial Window
unacknowledged bytes
"in flight" in first round
typically as a multiple of the MSS

Higher initial window
→ potential to transmit more data in fewer roundtrips

Initial Window
Why look at Initial Windows?

Amount of data bursted in an unprobed network

At the start, we don't know the bottleneck capacity
Why now?

IW 2-4
1998: RFC 2414
experimental

IWs measured
Padhye and Floyd
SIGCOMM '01

IWs in ISP
Qian et al.
IMC '09

IW 10
2013: RFC 6928
experimental

IW 1
Van Jacobsen
SIGCOMM '88

IW 1
1997: RFC 2001
standardized

IW 2-4
2002: RFC 3390
standardized

IW 10
Dukkipati et al.
SIGCOMM CCR 2010

IW 10
Linux Kernel
2011

IWs measured
Medina et al.
SIGCOMM CCR 2005
Why now?


IW 2-4
1998: RFC 2414
experimental

IWs measured
Padhye and Floyd
SIGCOMM '01

IWs in ISP
Qian et al.
IMC '09

IW 10
2013: RFC 6928
experimental

1997: RFC 2001

IW 1
Van Jacobsen
SIGCOMM '88

1998: RFC 2414

IW 1
1997: RFC 2001
standardized

2000: RFC 3390
standardized

Dukkipati et al.
SIGCOMM CCR 2010

2002: RFC 3390

Linux Kernel
2011

IWs measured
Medina et al.
SIGCOMM CCR 2005

IWs measured
Padhye and Floyd
SIGCOMM '01

IWs in ISP
Qian et al.
IMC '09

IW 10
2013: RFC 6928
experimental

IWs measured
Medina et al.
SIGCOMM CCR 2005

IW 10
2013: RFC 6928
experimental

IWs measured
Medina et al.
SIGCOMM CCR 2005

Why now?
Why now?

Paper Goal:
Actively monitor IWs in entire IPv4 space
Measuring IWs

Our Scanner

Probed Host

- SYN [MSS=..., WIN=...]
- SYN, ACK
- ACK, REQUEST
- ACK, SEG 1
- SEG n
- ACK n+1, WIN=2 \cdot MSS
- SEG n+1
- SEG n+2
- RST

Timeout

Retransmission

- Loss is a problem
  - Actually tail-loss
  - Do multiple scans
  - Scan early in the morning
  - Disable tail-loss probes
    - Do not enable SACK
- Trigger big response
  - HTTP and TLS

- Announce small MSS and large receive window
- Use ACK to test for more data
  - Was the host out of data or was the IW actually full?
Measuring IWs – Probe without prior knowledge

- **Send a client hello as the request**
- **Server hello contain certificate chains**
  - We further request options enlarging the reply (e.g., cert stapling)

- **Fails when SNI is enforced**

---

![CCDF Graph](image-url)

- Certificate Chain Length
- MSS 64, IW 1/2/4/10
- MSS 1336, IW 1/2/4

---

Jan Rüth, Christian Bormann, Oliver Hohlfeld
We want to probe all reachable IPv4 HTTP/TLS hosts

We implement the methodology in Zmap

- Bypasses the kernel stack
- Typically only used for enumeration
- We enable Zmap to send multiple packets
- We can manually craft connections and manipulate them

Modified Zmap, HTTP/TLS scanners available on Github

https://github.com/COMSYS/zmap
• TLS and HTTP do not agree
  ▶ Many TLS hosts still use IW 4

• HTTP scan triggers many abuse mails
  ▶ In contrast to TLS, this appears in access logs

• How much scanning is enough?
TLS and HTTP do not agree
- Many TLS hosts still use IW 4

HTTP scan triggers many abuse mails
- In contrast to TLS, this appears in access logs

How much scanning is enough?
Results – Who uses which IW?

- Most people in the Alexa list follow current RFCs
  - Here: similar distribution for HTTP and TLS

- Generally, we see older IWs in Access Networks
- CDNs and Cloud seem to be more up to date

<table>
<thead>
<tr>
<th>Service</th>
<th>HTTP IW1</th>
<th>HTTP IW2</th>
<th>HTTP IW4</th>
<th>HTTP IW10</th>
<th>TLS IW1</th>
<th>TLS IW2</th>
<th>TLS IW4</th>
<th>TLS IW10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akamai</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>EC2</td>
<td>0.0</td>
<td>1.8</td>
<td>3.4</td>
<td>94.7</td>
<td>0.2</td>
<td>1.3</td>
<td>2.6</td>
<td>95.8</td>
</tr>
<tr>
<td>Cloudflare</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>Azure</td>
<td>0.0</td>
<td>7.8</td>
<td>54.9</td>
<td>37.1</td>
<td>0.1</td>
<td>4.1</td>
<td>73.3</td>
<td>21.9</td>
</tr>
<tr>
<td>Access NW</td>
<td>3.5</td>
<td>50.2</td>
<td>20.8</td>
<td>21.7</td>
<td>4.5</td>
<td>17.6</td>
<td>67.1</td>
<td>10.4</td>
</tr>
</tbody>
</table>

Initial Window Size

- [Bar chart showing distribution of IW sizes for HTTP and TLS]

- [Table showing IW usage percentages for different services]

- Figure 5: Distribution of IWs per AS. Left, 3 HTTP and 3 TLS clusters or ASs that do not use IW 4. In the case of GoDaddy, 19.8% (32.7%) of the 137 k HTTP (193 k TLS) hosts use IW 4. In the case of Akamai, 38.6% (62.5%) of all HTTP (TLS) IPs use IW 10. Generally, we see older IWs in Access Networks. CDNs and Cloud seem to be more up to date with respect to IW configuration. Since especially service providers can benefit from such configurations, this suggests a trend towards more IW customization.

- This paper presents the first large-scale measurement of TCP's Initial Window Size (IW) configuration. The IW 10 peak is not clearly visible in Figure 3. Unlike our previous work, we observe 4 kB IW hosts, which use a static configuration, irrespective of the announced MSS. We found no obvious reason for these comparably large IWs.

- We differentiate between network types, content networks, and access networks. As shown in Table 3, content networks are classified based on their IP space, while access networks are categorized based on their reverse DNS record. This way, we classify 16% (18.1%) of all HTTP (TLS) IPs as belonging to access networks. To exclude server networks (e.g., Amazon and Akamai), we further match their reverse DNS record against a manually created ISP domain list and against a keyword list (e.g., "customer", "datacenter").

- Since our method does not require prior knowledge of the target, we used our scanner to manually probe few Akamai hosts. Access networks are classified based on their IP space. We observed non-RFC configurations such as much larger IWs.

- Initially, we extract hosts which encode their IW distribution in Table 3. Content networks are classified based on their IP space, while access networks are categorized based on their reverse DNS record. This way, we classify 16% (18.1%) of all HTTP (TLS) IPs as belonging to access networks. To exclude server networks (e.g., Amazon and Akamai), we further match their reverse DNS record against a manually created ISP domain list and against a keyword list (e.g., "customer", "datacenter").

- Generally, we see older IWs in Access Networks. CDNs and Cloud seem to be more up to date with respect to IW configuration. Since especially service providers can benefit from such configurations, this suggests a trend towards more IW customization.

- This paper presents the first large-scale measurement of TCP's Initial Window Size (IW) configuration. The IW 10 peak is not clearly visible in Figure 3. Unlike our previous work, we observe 4 kB IW hosts, which use a static configuration, irrespective of the announced MSS. We found no obvious reason for these comparably large IWs.

- We differentiate between network types, content networks, and access networks. As shown in Table 3, content networks are classified based on their IP space, while access networks are categorized based on their reverse DNS record. This way, we classify 16% (18.1%) of all HTTP (TLS) IPs as belonging to access networks. To exclude server networks (e.g., Amazon and Akamai), we further match their reverse DNS record against a manually created ISP domain list and against a keyword list (e.g., "customer", "datacenter").

- Since our method does not require prior knowledge of the target, we used our scanner to manually probe few Akamai hosts. Access networks are classified based on their IP space. We observed non-RFC configurations such as much larger IWs.

- Interestingly, we found that the number of GoDaddy hosts is a recent proposal. Generally, we see older IWs in Access Networks. CDNs and Cloud seem to be more up to date with respect to IW configuration. Since especially service providers can benefit from such configurations, this suggests a trend towards more IW customization.
Conclusion

• Distributions dominated by RFC-recommended values
  ▶ Still a lot of IW 2 and IW 4
  ▶ Heavily used infrastructure and popular hosts seem to be on IW 10

• We also find some customization
  ▶ Some hosts have very large IWs

• Periodic 1% scans are available at https://iw.comsys.rwth-aachen.de
• Source code available at https://github.com/COMSYS/zmap