Understanding ILDA Digital Network (IDN)

Looking at Wireshark Traces

Version of May 26, 2021

Contributor(s) Matthias Frank, Uni Bonn

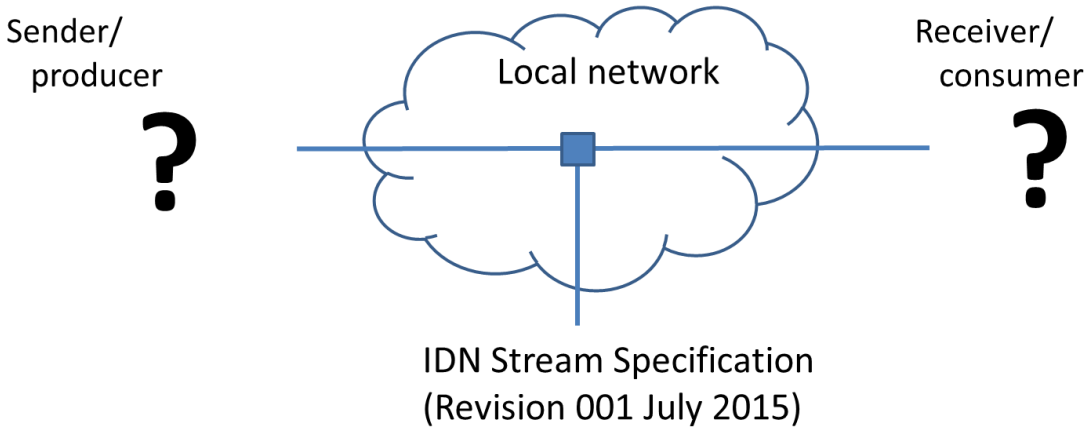
Contents

1. Abstract

What was the goal of preparing this document?

The main idea was to provide a single document with a comprehensive overview of which information is needed to get actively involved into IDN activities like IDN software and/or hardware development. This document gives references to the official IDN specifications (finalized and/or drafts) and to the relevant parts of these.

Wireshark traces are used to illustrate the payload content of network packets belonging to IDN streams and the procedures of IDN service discovery. The interested reader can install the Wireshark software [Wireshark-SW], [Wireshark-WiKi] and in detail look at the Wireshark traces that are also discussed in the remaining sections of this document. Those Wireshark traces are available for download at [IDN-Wireshark-Traces].



*"The IDN-Stream standard **describes the encoding of laser show artwork into digital data streams.** This can be from single laser projector data sent across a network connection up to entire laser shows including multimedia content, stored in data files for playing back in a target environment."*

Important terms and notions:

- **IDN Producer**: The so called IDN producer is the source (or sender) of IDN stream data. This could be a pure software (a laser show system) or also embedded software (e.g. a converter sampling the ILDA ISP-DB25 analogue output and generating a continuous IDN stream).
- **IDN Consumer**: The so called IDN consumer is the receiver of IDN stream data. Again this could be pure software (e.g. for visualization of IDN laser data), an embedded system (e.g. built into a laser projector to control scanners and laser diode drivers), or others.
- **IDN-Hello Server**: The term server (or IDN-Hello Server) has been introduced in the context of the IDN Hello Protocol: It is an IDN consumer that is capable to use some procedures for IDN device and service discovery, perform management tasks, and others.

- IDN entities may also have the double role of being IDN consumer and producer. E.g. an IDN-Recorder being able to receive (and record) IDN streams and later on playback the recorded streams to other IDN consumers. Or an IDN-Switcher (the digital counterpart of the famous 19-inch analogue Horaciotronic 2.0) maybe used to receive IDN streams from different IDN consumers and forward (or block) these streams to a set of several laser projectors.

2. IDN Stream Data (Laser Graphics)

To start with an implementation of IDN for laser streams, it is sufficient to look at the IDN Stream Specification [IDN-Stream, stable version of 2015] and just the packet header of the IDN Hello Protocol Specification [IDN-Hello, section 2.1]. This overview gives examples of IDN stream data that has been created by an IDN producer and could be processed by an IDN consumer.

Current implementations of IDN are using the User Datagram Protocol (UDP) of the Internet Protocol family and IDN consumers are receiving IDN UDP datagrams on the well-known UDP port 7255 (decimal). Cf. section 7 of [IDN-Hello]. The payload of these UDP datagrams contains the IDN data.

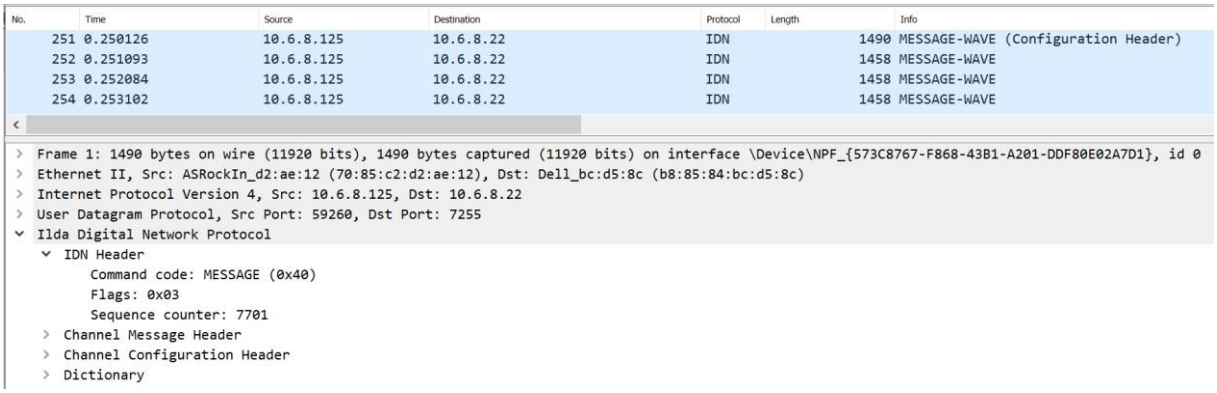
The IDN Hello Specification [IDN-Hello] and the IDN Stream Specification [IDN-Stream] define in detail how the UDP payload of such IDN network packets look like.

Laser Continuous Graphic Mode

In the IDN Continuous Graphic Mode (of Laser Graphic), often also called “wave mode”, “*…an uninterrupted sequence of samples representing a waveform …*“ [IDN-Stream] is transmitted in IDN network packets, the so called IDN channel messages.

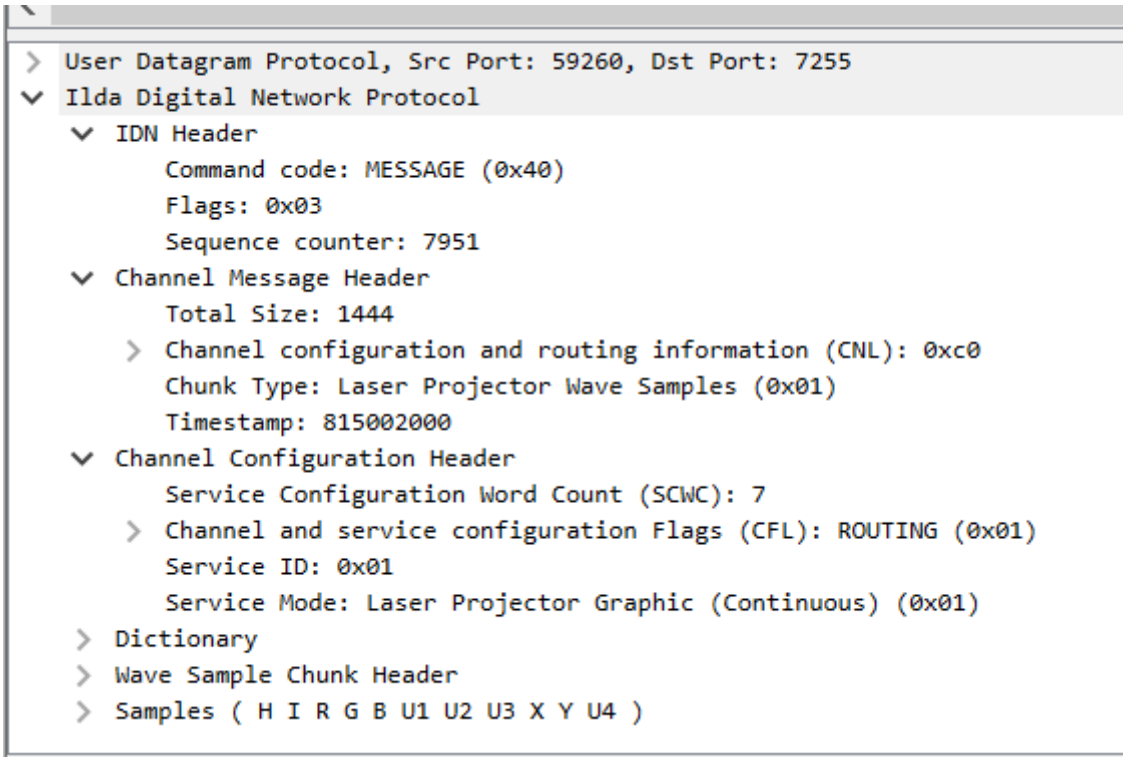
The following Wireshark trace file is used to illustrate IDN wave mode packets:

File: IDN-laser-continuous-mode.pcapng

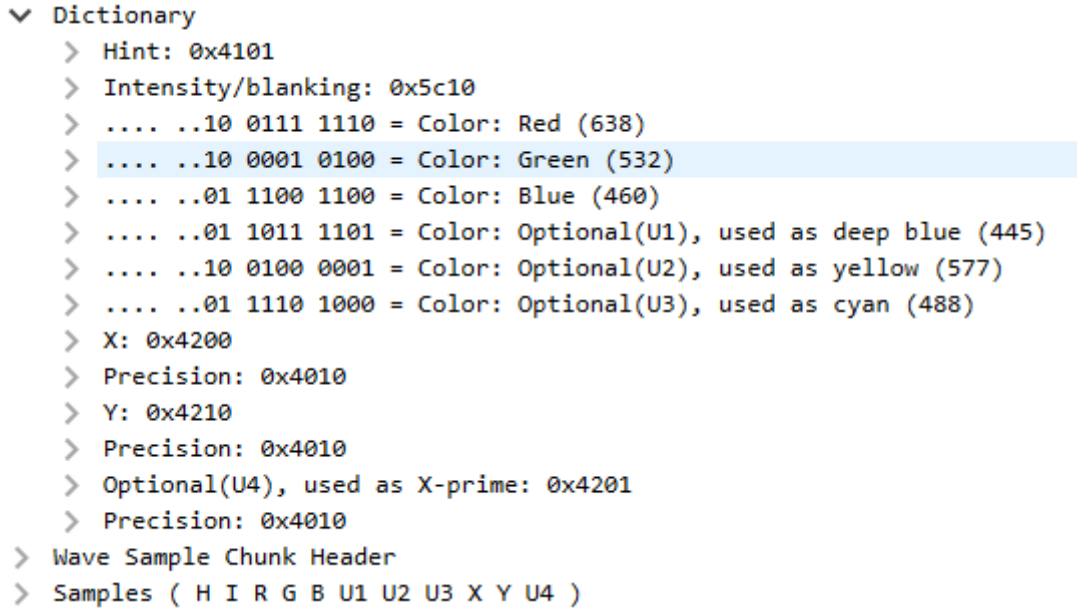


The payload of each IDN datagram begins with the IDN-Hello packet header of 4 byte length. All IDN stream datagrams presented in this section have the command code 0x40 (Realtime stream: channel message). Cf. section 2 of [IDN-Hello].

The first packet of an IDN stream needs to have channel configuration information. The presence of a configuration header is indicated by a bit flag in the channel message header. The first three headers contained in the Wireshark trace message #251 are shown below:



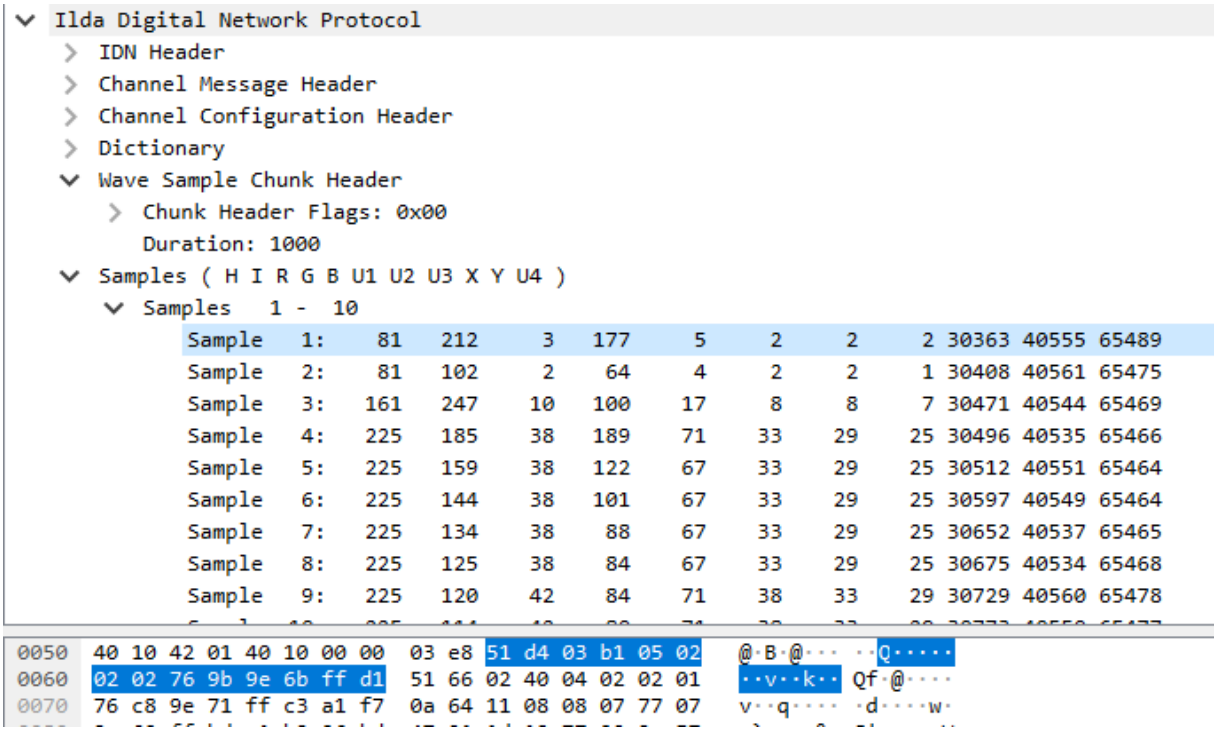
The channel configuration header is followed by a so called dictionary which defines which elements are present in the sample data structure (e.g. coordinates X, Y, optional Z; 3 colors RGB, or 6 colors) and at which resolution (8 bit, 16 bit, …). The Wireshark wave trace uses the following dictionary:



This dictionary is identical to the one of [IDN-Stream] section 3.4.10 as being recommended for ISP-DB25 Backwards Compatibility.

From [IDN-Stream, 3.4.10]: "*For devices sampling the waveforms on the ISP-DB25 connector, the default setting SHALL be a sampling frequency of 100kHz with 100 samples per IDN channel message with the above tags and signal assignments resulting in one message per millisecond of content.*"

The wave sample chunk header and the first sample values are shown below. The IDN channel message has a total of 100 samples and the duration of 1000 (microseconds) refers to the sampling frequency of 100 kHz.



As recommended by [IDN-Stream, section 2.2] a configuration header is repeatedly contained in the IDN channel message and thus visible in the Wireshark trace again after about 250 ms, i.e. every 250th message in the trace carries the configuration header with the dictionary.

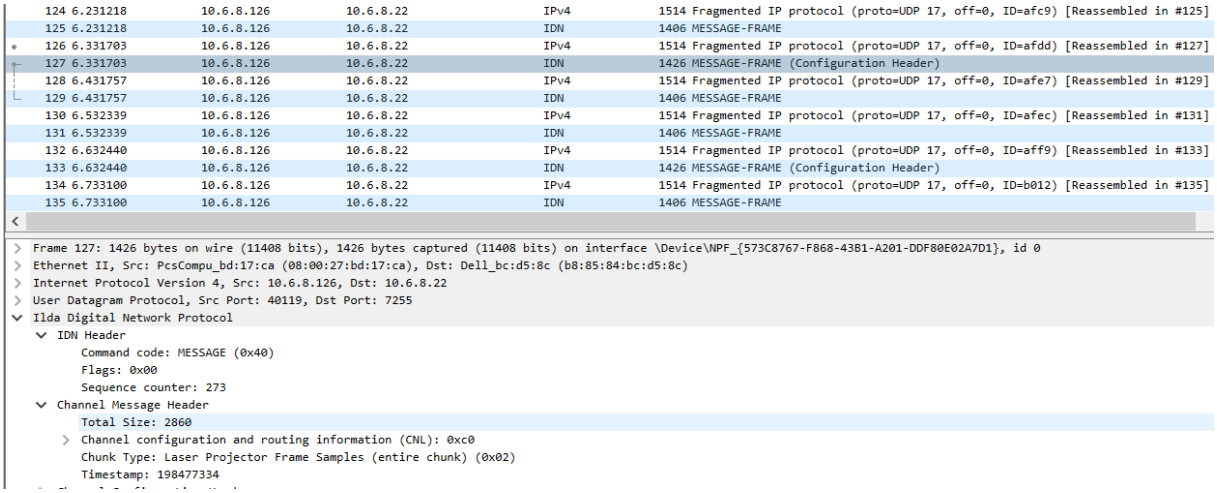
Laser Discrete Graphic Mode

In the IDN Discrete Graphic Mode (of Laser Graphic), often also known as “frame mode”, the IDN channel messages each contain sample data of a full laser frame. [IDN-Stream, section 3.3]: “*These frames are drawn one after another as they are scheduled. Frames that started to draw must finish before the next frame is processed. Newer frames replace older ones and in case of an overrun (more frames received than processed), older frames are dropped.*”

From a software point of view of an IDN producer, the IDN frame mode is much easier to be implemented (compared to the IDN wave mode), in particular if the laser show software system already has a frame based API (application programming interface) with access to the laser DAC software drivers (such as the old EasyLase API, the Medialas MLD API, and others).

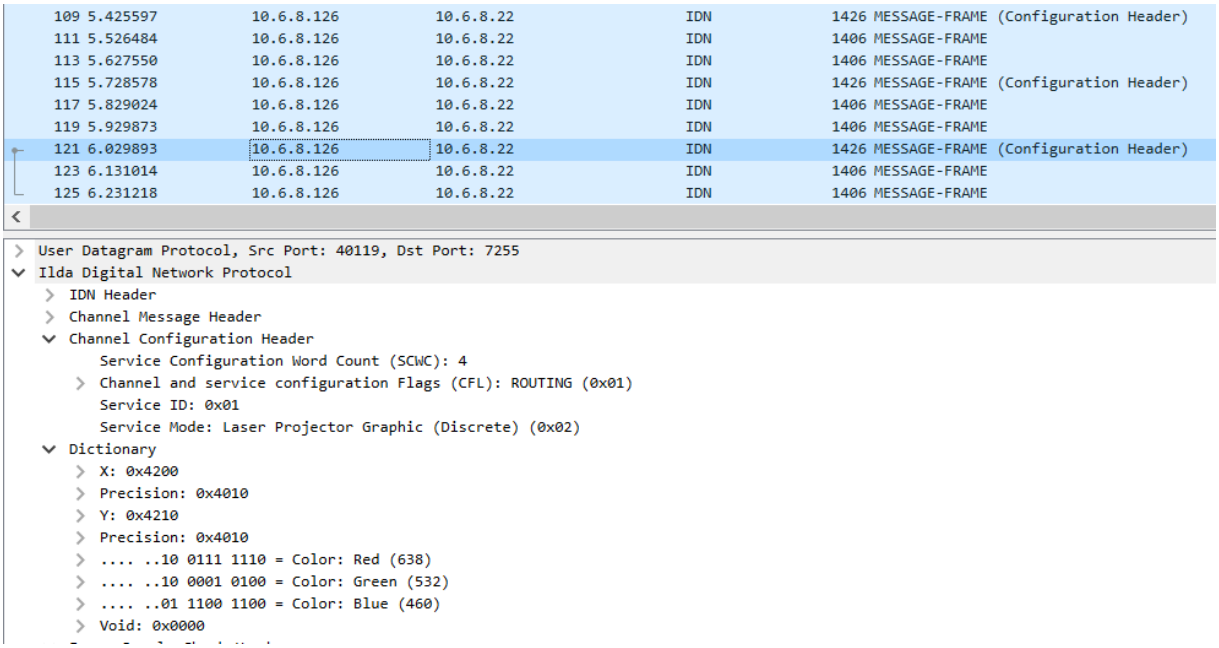
The following Wireshark trace file is used to illustrate IDN frame mode packets:

File: IDN-laser-discrete-mode.pcapng

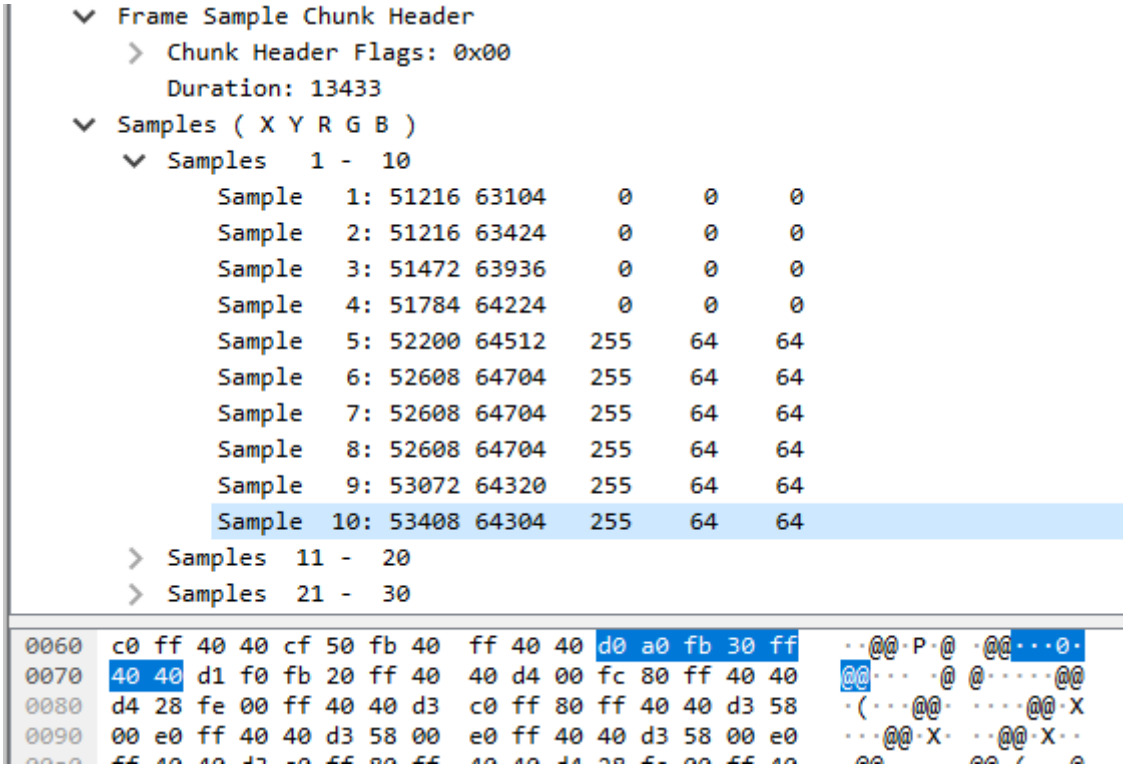


Again, the payload of each IDN datagram begins with the IDN-Hello packet header of 4 byte length (command code 0x40). The Wireshark trace shows that the size of the IDN frame messages is larger and IP fragmentation is applied. The screenshot example shows the IDN channel message header indicating a total size of the IDN message of 2860 bytes. The Internet Protocol (here IPv4) typically splits up larger UDP datagrams into fragments of about 1500 bytes.

As already seen with the Wireshark trace of the IDN wave stream, from time to time a configuration header is included in the IDN channel messages. The next screenshot also shows the dictionary, which is much more simple in this case: The sample structure only contains coordinates X and Y (with the so called precision tag of the dictionary the resolution is extended from 8 bit to 16 bit) and the colors red, green and blue (each 8 bit).



The frame sample chunk header and the first ten samples are shown below:



The amount of samples contained in such a full frame (in an IDN channel message) and the duration (in microseconds, in the frame sample chunk header) refer to the scanrate of the source of the IDN stream.

In the example above, a total of 404 samples is coming together with a duration of 13433 microseconds. This refers to a scanrate of about 30000 samples/second (which actually was selected in the source laser system when creating the Wireshark trace).

3. IDN Unit and Service Discovery

The IDN-Hello specification (draft) became available later than the IDN-Stream specification (revision 001 of July 2015). First prototype implementations used manual configuration of IP addresses of IDN consumers on the sending side of an IDN producer.

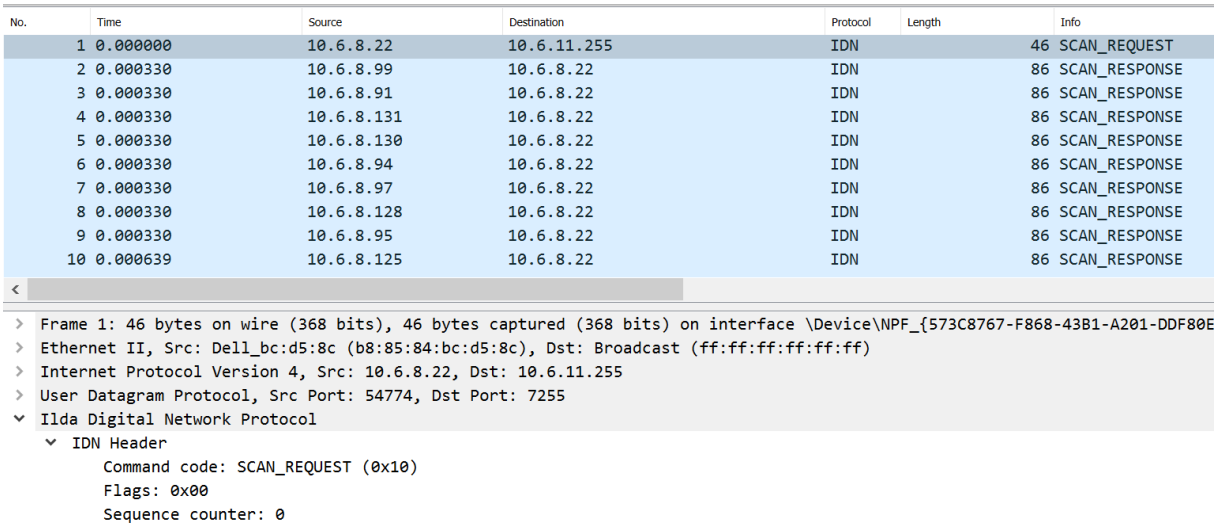
The service discovery procedures defined in [IDN-Hello] significantly help to support automatic configuration of IDN network elements and services independent from specific IP addresses.

Scan Network for IDN Units

The following Wireshark trace file is used to illustrate the network scan for IDN units:

File: IDN-scan-single-service-units.pcapng

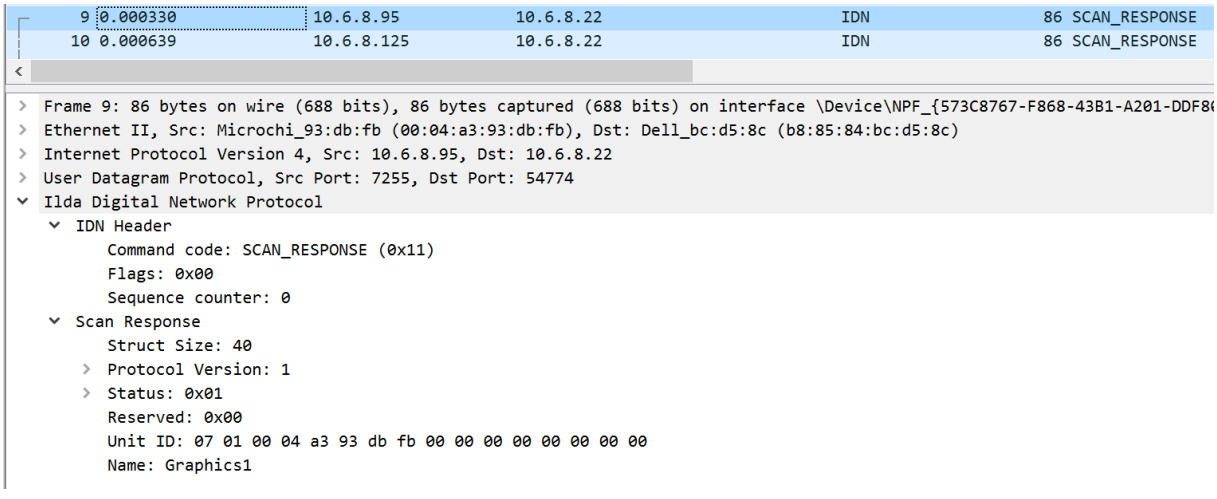
The screenshot below shows a broadcast message (IDN scan request) to the local network (IP destination address 10.6.11.255). The command code of the IDN-Hello header is 0x10. The requesting IDN producer gets replies from 9 IDN units (IDN scan response).



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.6.8.22	10.6.11.255	IDN	46	SCAN_REQUEST
2	0.000330	10.6.8.99	10.6.8.22	IDN	86	SCAN_RESPONSE
3	0.000330	10.6.8.91	10.6.8.22	IDN	86	SCAN_RESPONSE
4	0.000330	10.6.8.131	10.6.8.22	IDN	86	SCAN_RESPONSE
5	0.000330	10.6.8.130	10.6.8.22	IDN	86	SCAN_RESPONSE
6	0.000330	10.6.8.94	10.6.8.22	IDN	86	SCAN_RESPONSE
7	0.000330	10.6.8.97	10.6.8.22	IDN	86	SCAN_RESPONSE
8	0.000330	10.6.8.128	10.6.8.22	IDN	86	SCAN_RESPONSE
9	0.000330	10.6.8.95	10.6.8.22	IDN	86	SCAN_RESPONSE
10	0.000639	10.6.8.125	10.6.8.22	IDN	86	SCAN_RESPONSE

```
> Frame 1: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface \Device\NPF_{573C8767-F868-43B1-A201-DDF80E
> Ethernet II, Src: Dell_bc:d5:8c (b8:85:84:bc:d5:8c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 10.6.8.22, Dst: 10.6.11.255
> User Datagram Protocol, Src Port: 54774, Dst Port: 7255
v Ilda Digital Network Protocol
  v IDN Header
      Command code: SCAN_REQUEST (0x10)
      Flags: 0x00
      Sequence counter: 0
```

Below is the content of one of the IDN scan response packets (command code 0x11), returning the IDN Unit ID (here: 0x07010004a393dbfb) and a host name (here: Graphics1), among others.



9	0.000330	10.6.8.95	10.6.8.22	IDN	86	SCAN_RESPONSE
10	0.000639	10.6.8.125	10.6.8.22	IDN	86	SCAN_RESPONSE

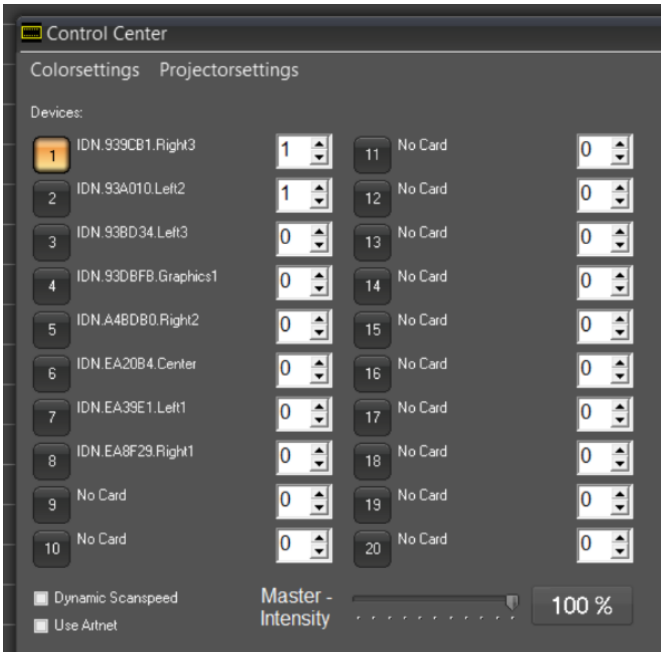
```
> Frame 9: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{573C8767-F868-43B1-A201-DDF8
> Ethernet II, Src: Microchi_93:db:fb (00:04:a3:93:db:fb), Dst: Dell_bc:d5:8c (b8:85:84:bc:d5:8c)
> Internet Protocol Version 4, Src: 10.6.8.95, Dst: 10.6.8.22
> User Datagram Protocol, Src Port: 7255, Dst Port: 54774
v Ilda Digital Network Protocol
  v IDN Header
      Command code: SCAN_RESPONSE (0x11)
      Flags: 0x00
      Sequence counter: 0
  v Scan Response
      Struct Size: 40
    > Protocol Version: 1
    > Status: 0x01
      Reserved: 0x00
      Unit ID: 07 01 00 04 a3 93 db fb 00 00 00 00 00 00 00 00
      Name: Graphics1
```

The IDN unit ID is like a globally unique serial number and can be used to unambiguously distinguish between different IDN units (devices) and, moreover, recognize devices in a local network even if the IP addresses have changed. The host name is a human user friendly name to identify the IDN unit. In the Wireshark trace, unit names like Left3, Left2, …, Center, Right1, Right2, …, Graphics1 are used.

Ask Unit for IDN Services

Once the available IDN units are known to an IDN producer after all replies to the (broadcasted) IDN scan request have been received, the IDN producer can ask each IDN unit for the IDN services provided by that unit (command code 0x12 IDN service map request). The following screenshot shows the service map response (command code 0x13) by one of the IDN units mentioned above (here: IP address 10.6.8.99 with name Center), that also offers a single IDN service with service ID 1 and service name Center.

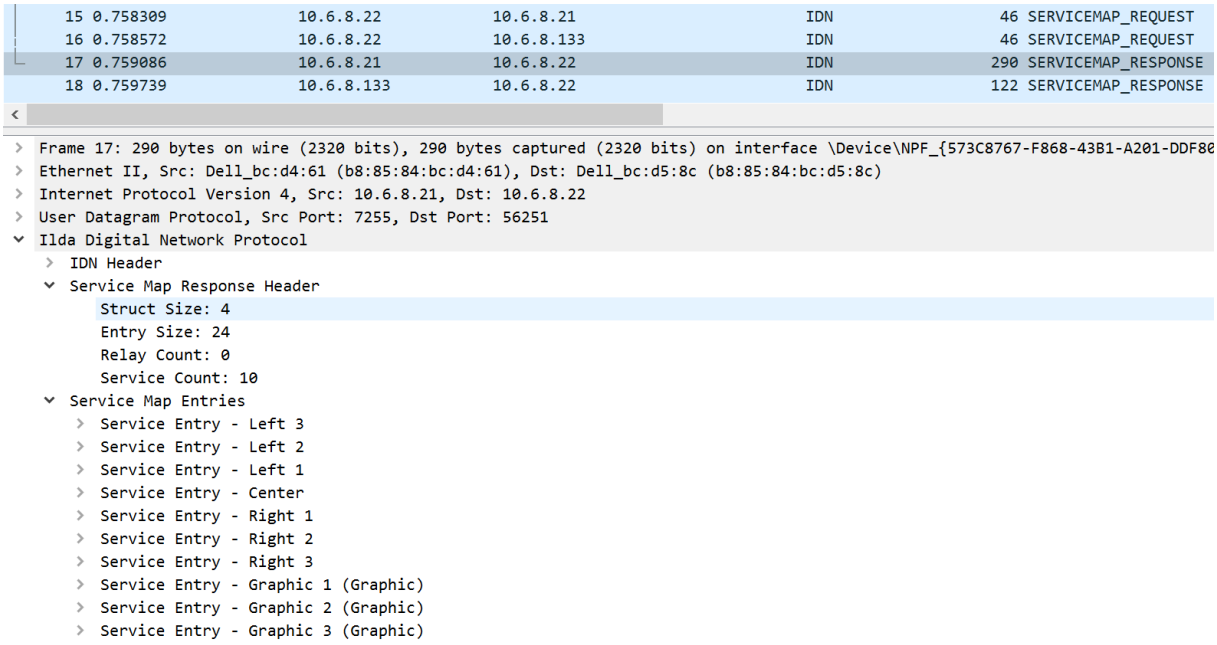
```
    53 0.750473    10.6.8.22     10.6.8.131    IDN    46 SERVICEMAP_REQUEST
    54 0.750513    10.6.8.99     10.6.8.22     IDN    74 SERVICEMAP_RESPONSE
    55 0.750567    10.6.8.22     10.6.8.130    IDN    46 SERVICEMAP_REQUEST
    56 0.750661    10.6.8.91     10.6.8.22     IDN    74 SERVICEMAP_RESPONSE

> Frame 54: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{573C8767-F868-43B1-A201-DDF80E02A7D
> Ethernet II, Src: Microchi_ea:20:b4 (d8:80:39:ea:20:b4), Dst: Dell_bc:d5:8c (b8:85:84:bc:d5:8c)
> Internet Protocol Version 4, Src: 10.6.8.99, Dst: 10.6.8.22
> User Datagram Protocol, Src Port: 7255, Dst Port: 54774
v Ilda Digital Network Protocol
  v IDN Header
      Command code: SERVICEMAP_RESPONSE (0x13)
      Flags: 0x00
      Sequence counter: 1
  v Service Map Response Header
      Struct Size: 4
      Entry Size: 24
      Relay Count: 0
      Service Count: 1
  v Service Map Entries
    v Service Entry - Center
        Service ID: 0x01
        Service Mode: Unknown (0x80)
        Flags: 0x00
        Relay Number: 0x00
        Name: Center
```



A software system can make use of the information returned from IDN scan response and service map response and can display this information to the user of the software, e.g. here in the menu to select color and projector settings. In this case, each IDN device is identified by e.g. “IDN.93DBFB.Graphics1” which contains part of the IDN unit ID and the IDN service name.

The following Wireshark trace file is used to illustrate the case of service map request/response where a single IDN unit offers several IDN services:

File: IDN-scan-multiple-service-units.pcapng



In this example the service count is 10 and thus the service map also has 10 entries (service names Left3, Left2, …, Graphic 3 (Graphic)).

4. References

List of IDN specification documents (both standard and draft) and other useful sources and documentation:

[IDN-Stream] IDN Stream Specification, Revision 001, July 2015

https://www.ilda.com/resources/StandardsDocs/ILDA_IDN-Stream_rev001.pdf

In development: Revsion 002 draft (NOT approved yet) March 17 2021

https://www.ilda.com/resources/Tech/IDN/IDN-Stream-v002---2021-03-17-draft.pdf

[IDN-Hello] IDN Hello Protocol Specification

In development: IDN-Hello draft Nov. 24 2020

https://www.ilda.com/resources/Tech/IDN/IDN-Hello---2020-11-24-draft.pdf

[Wireshark-SW] Wireshark network protocol analyzer, official project page, incl. free software download

https://www.wireshark.org/

[Wireshark-WiKi] Wireshark, general information on Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Wireshark

[IDN-Wireshark-Traces] Cloud share to provide the Wireshark traces used in this document, and also upcoming updates of this document

https://uni-bonn.sciebo.de/s/H7w83cHzxYfRyy1

5. Document History

When editing this document, please add the most recent version date, add contributors and change the filename accordingly.

Version of	May 26, 2021	- Wireshark traces/screenshots added
Version of	May 25, 2021	- Initial version
Contributor(s)	Matthias Frank, Uni Bonn	