

A tool to design networks in entertainment venues

Andrew Barron
2024



University of
St Andrews

1 Abstract

Computer networks are increasingly used in entertainment venues such as theatres, concert halls, and arenas as the main way to distribute lighting control data, sound, and video around the space. Compared to traditional approaches, networks allow for greater redundancy, higher data throughput with smaller and lighter cables, and less equipment being toured around the country.

However, these networks are a drastically different way of thinking compared to what the industry has been used to for the last many decades where point-to-point links were the only option.

This project developed a design tool for planning IP networks to be used in this context. The tool includes data about specific hardware products and software protocols from the industry and allows the user to put these entities together to create a system. Using information from specification documents and real-world measurements, the tool validates the configuration giving the technician confidence that their plan will work when deployed. The target users for this tool are experienced event technicians and designers who are unfamiliar with computer networking concepts, so the tool presents information in a way which is accessible to these people, enabling them to learn while they work. The requirement specification and project evaluation included input from real users, who reported that the project made them feel more positive about using networks in their work.

2 Declaration

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated.

The main text of this project report is 18,729 words long.

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>. This work is published online at <https://awmb.uk/2025/04/entertainment-networks>, and the author can be contacted there.

3 Contents

1	Abstract.....	2
2	Declaration.....	3
3	Contents	4
4	Introduction	6
5	Context survey.....	8
5.1	Common protocols and hardware.....	8
5.1.1	Lighting control	8
5.1.2	Sound	10
5.1.3	Video	12
5.1.4	Other control data	12
5.2	Why change to IP?	12
5.3	The transition to IP.....	13
5.4	Protocols	14
5.4.1	Unreliable networks in a mission-critical application	15
5.5	Current tools	16
6	Ethics.....	17
7	Software engineering process.....	18
8	Requirements.....	20
8.1	Project objectives	20
8.2	Stakeholders	20
8.3	Functional and non-functional requirements.....	20
8.3.1	Functional requirements.....	21
8.3.2	Non-functional requirements	21
8.4	Requirements survey	21
8.5	Personas	26
9	Measurement and data collection.....	29
9.1	Methodology	29
9.2	Results and analysis.....	32
9.2.1	Art-Net	32
9.2.2	sACN.....	34
9.2.3	Dante	35
9.2.4	NDI and NDI-HX	38
9.3	Measurement conclusion	40
10	Design	41
10.1	Nielsen's usability heuristics	41
10.2	Health and safety consequences	45
10.3	Database.....	45
10.4	Data management	48
10.5	Homepage	49
10.6	Network design interface	51
10.7	User onboarding.....	55
10.8	Design wizard.....	56
10.9	File management.....	58

11	Implementation	59
11.1	Libraries and frameworks	59
11.2	Database: data layer	60
11.3	Frontend: application and presentation layers	61
12	Evaluation and critical appraisal	64
12.1	Meeting project requirements	64
12.1.1	Primary objectives.....	64
12.1.2	Secondary objectives.....	65
12.1.3	Functional requirements.....	66
12.1.4	Non-functional requirements	66
12.2	User evaluation.....	67
12.2.1	Respondent demographics	67
12.2.2	Sentiment and confidence	67
12.2.3	Qualitative responses	68
13	Future work	71
14	Conclusion	72
15	Bibliography	73
16	Appendix: Ethics approval.....	76
17	Appendix: Requirements survey	78
18	Appendix: Evaluation questionnaire.....	80
19	Appendix: Application tour	82
20	Appendix: Screenshots of application	86
21	Appendix: Database schema	96
22	Appendix: Testing summary	98
22.1	Map and reports.....	98
22.2	Design validation.....	102
23	Appendix: Conformance to ISO 9241-210:2019	105
23.1	Table B.10[31] — Checklist for assessing applicability and conformity with ISO 9241-210:2019	
	105	

4 Introduction

Across the live events industry – theatres, concert halls, night clubs, installations, and even corporate AV – a shift is happening. For years, lighting control data, audio feeds, and video content have moved around venues using basic analogue signals or point-to-point digital data transmission. Now, though, to support the ever-increasing scale of events and the need for greater flexibility in systems, IP networks are replacing these simple signals and becoming a critical part of venue infrastructure. This transition comes with a range of benefits: old analogue solutions tend to require one cable for every data channel and some existing point-to-point digital systems meet their limit around 48 channels, whereas computer networks are theoretically unlimited in their scale. Computer networks can be built atop widely available commodity hardware, offering flexible solutions and helping to reduce cost and increase accessibility. Because of the large bandwidth available, different departments can share infrastructure, again reducing cost and setup time.

To achieve these benefits though, you first need to set up a computer network – which for users unfamiliar with networking concepts, is easier said than done. Ask on a discussion forum about technicians’ experience with networks, and while you will find a fair number of evangelists, you will also hear horror stories where incorrect network configuration caused big problems for someone’s event and they have felt uneasy about networking ever since. As computer science concepts become more and more integrated into everyday life, computer scientists are faced with new user experience design challenges in order to make these technologies accessible to a broader audience, and entertainment network design is no different. This project aims to address this challenge.

The objectives of this project are to compile an extendible data set of entertainment equipment which relies on computer networks by measuring device performance in representative configurations. This data will be used to produce a tool aimed at technicians who are experienced in their field but new to computer networking. The tool will support them in planning networks through helping them identify requirements and start their design using a wizard, and validate their configuration is functional through a set of automated checks. The tool will also allow them to save and share plans with others. Since the target audience is networking novices, careful design is required to ensure the tool helps these users and inspires confidence.

To meet these objectives, a human-centred design approach was used. The author of this project works and volunteers in the entertainment industry, so using this experience along with formal consultation with colleagues within the University, a requirement specification was drawn up to inform the work. From here, a software tool was designed and developed, then these same technicians were asked to evaluate the artefact. The evaluation showed the project objectives were met and forms a strong foundation for future work and interaction with users.

This report is quite extensive due to the broad range of activities which made up the project, including measurement, interaction with real users, and software development. It is written with enough detail to be understandable by a computer science undergraduate without entertainment industry experience, and for another future project to carry the work on based on the evaluation and feedback received. We first explore the history of lighting control, sound transmission, and video transmission in venues, assessing protocol specification documents and the (somewhat limited) existing literature in the field about the transition to network-based approaches (section 5). After explaining the ethical implications of the project (section 6), we more formally define the software engineering process used and explain how user feedback informs the system design by complying with ISO 9241-210:2019 (section 7). Following the guidance in the standard, the project requirements are defined (section 8) including analysing the results of the requirements elicitation questionnaire completed by real users. Next, a measurement procedure is carried out where the network usage of different entertainment equipment is characterised (section 9) before the tool is designed (section 10) and implemented (section 11). Through another questionnaire shared with real users, the tool is evaluated against the project objectives (section 12), before suggestions of future work are made based on the findings of this project and the results of the evaluation (section 13).

5 Context survey

The traditional approach to moving data around venues usually involves digital serial connections or plain analogue signals. This context survey briefly summarises the kinds of data which needs to be moved around a venue, highlighting the traditional non-networked approach and an IP-based replacement. We then assess why it is worth making the change to an IP-based approach, and explore some of the challenges of the transition. Next, some of the IP protocols discussed in this project are explored in more detail, followed by a look at some current tools in the industry similar to the one we aim to develop.

5.1 Common protocols and hardware

5.1.1 Lighting control

DMX-512 (usually abbreviated to DMX) is a standard for sending digital data and is primarily used for controlling lights[18]. It is an eight-bit serial protocol where each packet contains 513 slots. The first slot is a start code. The other 512 slots are data: each slot is known as an address to which some value is sent. Fixtures are ‘daisy-chained’ together using 5-pin XLR cable (known as DMX cable) to form a line, and each fixture responds to instructions at one or more addresses in the packet. Each group of 512 addresses is known as a universe; one universe is transmitted along each DMX cable.

In Figure 1, we see an example configuration of traditional lighting control. Three cables run from the controller, each carrying one universe of data (U1, U2, and U3). Underneath each fixture, the device’s start address is listed followed by its footprint in brackets. The top left fixture has start address 1 and a footprint of 3, which means it is controlled by three sequential addresses starting from 1. For example, DMX address 1 could represent red light intensity, address 2 as the green light intensity, and address 3 as the blue light intensity, allowing colour mixing. The next fixture in the line starts reading from address 4 and has a footprint of 10 (perhaps using additional parameters for motor movements or other effects), and so on down the line.

The second row of fixtures have much larger footprints, so we fill the 512 available addresses with only four fixtures (compared to universe 1, where after 5 fixtures there were still a further 475 addresses available). With modern fixtures which have a wide variety of effects, it is not uncommon for a single fixture to require hundreds of addresses.

Start addresses do not have to be sequential nor unique: we see this in universe 3 where each start address starts at a round number. This leaves gaps in the address space, but makes remembering addresses easier. We also see the two fixtures in the bottom right of the figure share an address, so they cannot be controlled individually.

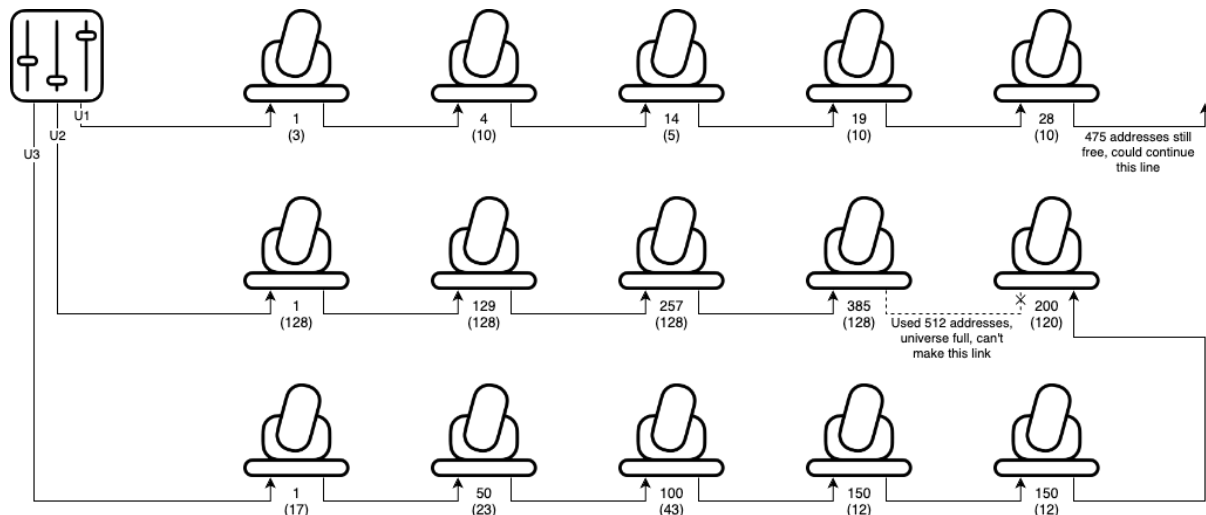


Figure 1: An example lighting control configuration using traditional DMX-512 only. The desk sends three universes of data on three cables. Each fixture's start address is written below the fixture and the footprint in brackets.

DMX was borne out of a time where each fixture used exactly 1 address — light intensity — so one universe would control up to 512 lights. As shown, this is no longer true with modern fixtures, so large events will need multiple universes.

It would be unwieldy to require a cable for every universe from the desk to each group of lights across the venue, and most control desks don't have enough 5-pin outputs to send data for every universe which needs to be controlled. IP offers a solution. Art-Net[3] and sACN[8] are the two leading standards in this area, and both serve to transmit DMX data over a network from controller to fixtures. Some very modern fixtures can take Art-Net and sACN signals in directly, but generally devices known as “nodes” act as breakout boxes to convert the IP data into traditional DMX-512.

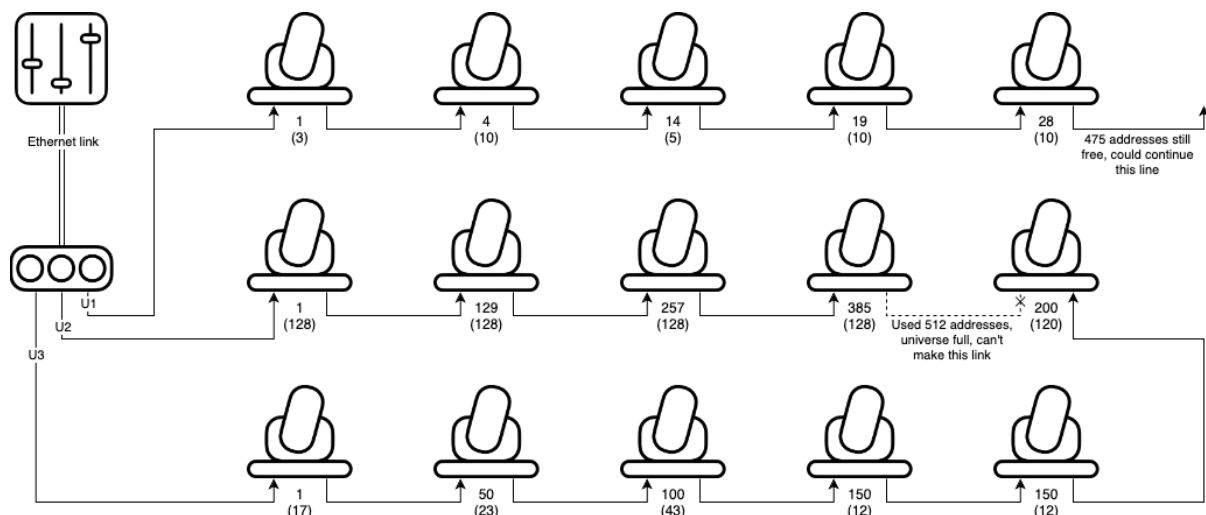


Figure 2: An example lighting control configuration where the control desk sends lighting data over IP to a node, which can be located closer to the fixtures and result in fewer long cables needing to be run across the room.

In Figure 2, we add a DMX node between the control desk and the first fixture in each universe. This node can be located physically near the fixtures — perhaps near the stage power

supplies – so the three DMX cables to the first fixture of each universe can be short, and a single long Ethernet cable can run from the control desk to the node. It also allows us more 5-pin DMX outputs than may be built into the control desk.

5.1.2 Sound

Analogue audio is usually transmitted as a balanced signal over a 3-pin XLR cable. The main limitation here is that each cable can only carry one audio signal. There can often be more than thirty microphones on stage and eight or more audio mixes being made at the control position at the back of the auditorium, meaning lots of cables needing to be laid between the stage and control desk.

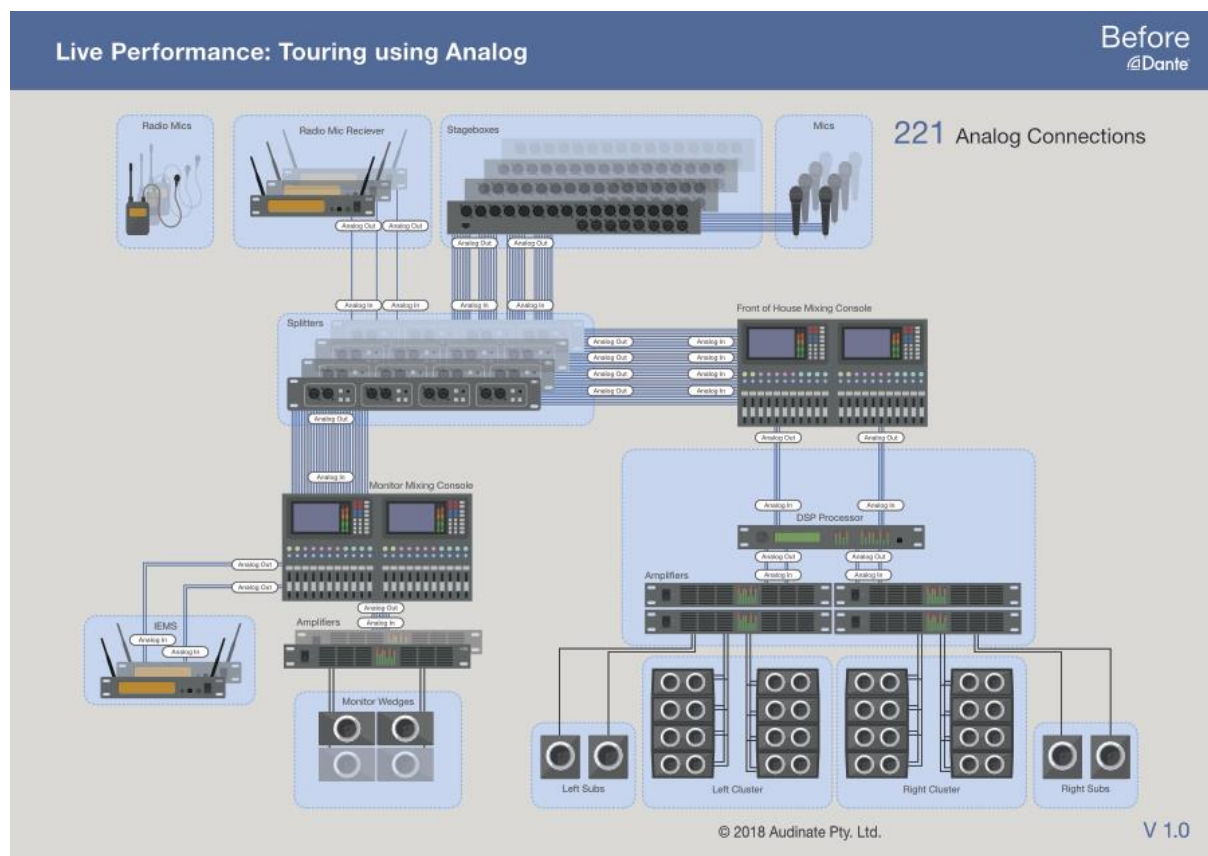


Figure 3: From [4], an example analogue audio configuration for a touring band including wireless and wired inputs, monitoring, and front of house sound.

Figure 3 shows an example sound configuration for a touring band¹. It shows a plan including a variety of microphones being split between a front of house mixing console (for the audience to hear) and a monitor mixing console (for the artists on stage to hear themselves)

¹ This document comes from the sales department of a commercial audio-over-IP solution vendor, but is not an unrealistic representation of a potential configuration.

plus the various speakers and outputs required, coming to a total of 221 analogue connections to be made.



Figure 4: A 32-channel (24 inputs, 8 outputs) multicore cable. The box would be placed on the stage, the 'tails' are plugged into the mixing desk.

It is unlikely this configuration would be realised using 221 XLR cables: instead, you would use multicore cables or “snakes” (Figure 4), which enclose multiple XLR cables in a single jacket. While this may reduce the number of long cable runs, multicore cable can weigh more than a kilo per metre so could be just as annoying and time-consuming to lay out.

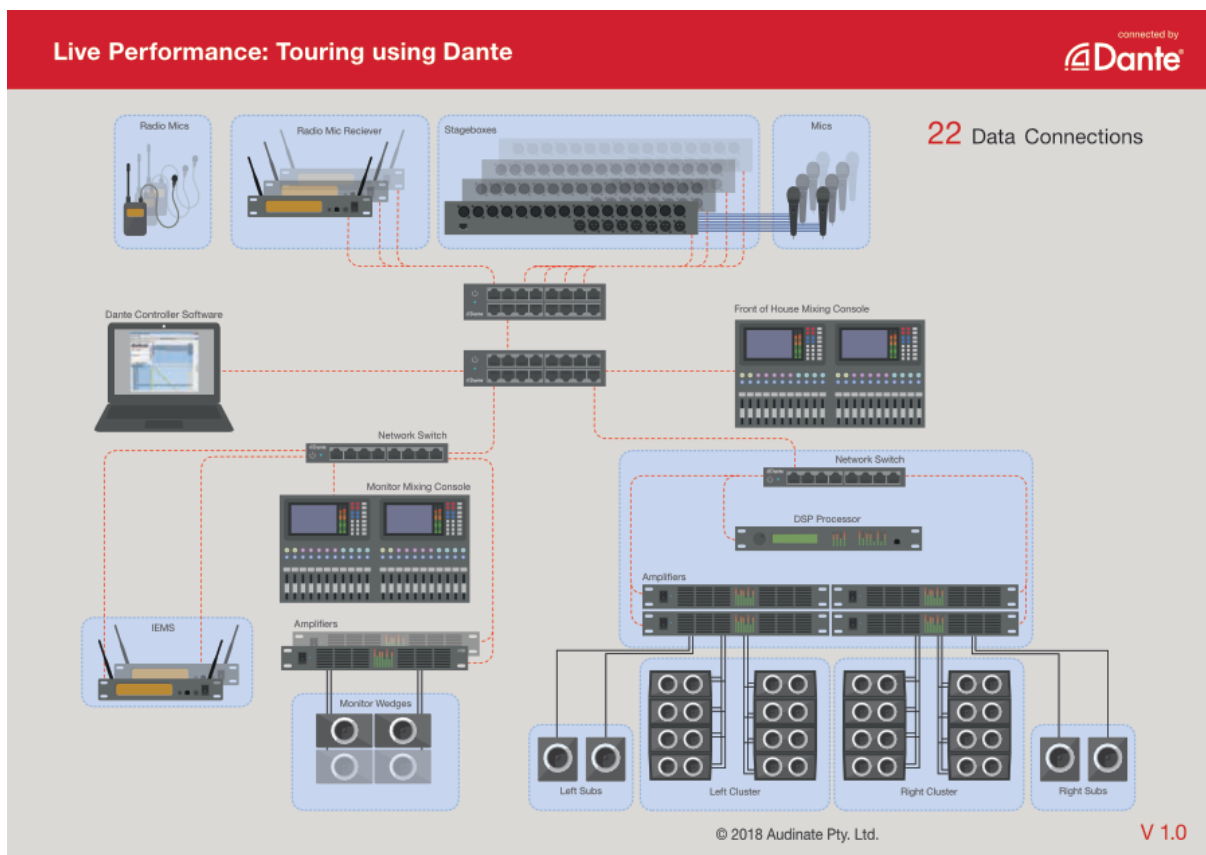


Figure 5: Using Dante, the number of analogue link cables is reduced compared to Figure 3. From [4].

Figure 5 goes on to show an IP-based configuration using Dante[24]. Dante is a proprietary audio-over-IP solution by Audinate, which has expanded to also support video. Dante is installed in devices by manufacturers who licence the appropriate hardware from Audinate. Offering up to 512 channels of audio in each direction on a single gigabit Ethernet link[20], the example tour would require only 22 data connections compared to 221. When you also consider how much more lightweight a Category 6 cable is compared to a multicore, the difference becomes clear.

5.1.3 Video

Like audio, the traditional approaches for production video use one cable for one feed. SDI (serial digital interface, using BNC cables) or HDMI are two common options.

NDI is an IP-based solution currently on the market[23]. Like Dante, it is a proprietary solution which can be licensed by manufacturers to add NDI functionality to their hardware. The system offers one-to-many video distribution with latency of less than 100ms, making it suitable for many broadcast and live scenarios.

5.1.4 Other control data

Many control consoles offer a remote control functionality through the use of an app[1] or programmable API[19]. These are popular as a convenience: where your control desk may need to be in a poor location for practical reasons, you can use the remote control from a tablet computer to move around and fine-tune designs and programming from other vantage points. Some ultra-compact equipment takes this further, omitting the screen and control functionality from the hardware and requiring the use of a tablet to remote control the unit[13]. APIs allow consoles to be included as part of automated procedures – for example, a sequence of events could run across all lighting, sound, and video systems in the event of an evacuation.

The two leading standards for this are MIDI Show Control[25] and OSC[21]. MIDI (Musical Instrument Digital Interface) was initially designed for things like electronic keyboards to communicate with computers for recording and playback. Now, it has been adapted for use in a variety of remote control interfaces over the original MIDI cable, or more commonly USB or the network. OSC (Open Sound Control) uses URI-style commands to send instructions to a receiver, making it more flexible and easier for humans to understand compared to MIDI notes, but requiring more processing on embedded hardware.

5.2 Why change to IP?

“Converged networking” is the popular term for all technical departments in the theatre sharing a single network for their infrastructure, but other than sales material, there is not a single list of benefits to this approach. Based on industry experience and the work done during this project, the key benefits of switching from traditional approaches to converged networking include:

- **Fewer, smaller, easier-to-manage cable runs.** Enormous multicores or tens of cables running to the same location can be replaced by a single Ethernet cable. In many cases, even running two Ethernet cables to form a redundant link is still significantly lighter than the alternative. These cable runs also have a much larger bandwidth than analogue alternatives, allowing (for example) multiple high-definition video feeds to be sent over a single small connection.
- **Shared infrastructure reduces setup time and project cost.** Traditionally, most departments would need dedicated cabling between the control area at the back of the auditorium and key locations around the venue (the stage, flying areas above the stage, etc). With IP, departments can share a single Ethernet cable between switches at each end, and each department can still have some level of isolation from the others using VLANs.
- **More flexible topology.** Devices can be added to the network anywhere you deploy a switch, instead of always requiring another direct point-to-point cable run, and the flow of data is controlled in software. It becomes practical to move control desks into the centre of the auditorium during rehearsals to improve visibility to the stage and then move them to the back of the room for the shows, since you only need a single Ethernet cable to connect to everything in the venue. There is also the opportunity to send data in a one-to-many fashion without dedicated splitter hardware.
- **Wireless connectivity.** While usually not recommended for show-critical applications, having wireless access to video feeds or control systems can help with troubleshooting and system tuning.

5.3 The transition to IP

Event technicians tend not to be computer scientists, and thus do not usually have a strong background in designing, deploying, and managing computer networks.

Networks are often built piecemeal: an Ethernet cable is run between the lighting console and an sACN node, because more DMX addresses were needed. A second-hand broadband hub is connected to the sound console when someone wants to remote control it over Wi-Fi, and later a laptop is plugged in to record a show using Dante. A bigger switch is found in a cupboard and added to the setup, to make it easier to connect a stage box on stage. This approach is fine, but results in multiple isolated networks, issues with allocating addresses correctly, use of equipment which isn't an appropriate specification for the purpose, and misses out on a lot of the convenience and benefits of true converged networking.

There is limited existing academic research on this topic since it is somewhat niche to the industry and often revolves around proprietary technologies. There is also a lack of comprehensive training to help technicians gain a better understanding of core networking concepts. Some courses – such as the Dante certification programme – include modules on IP addressing, VLANs, TCP, and similar concepts[26], but really are an introduction to the company's own proprietary tooling and product-specific terminology. Some books have been written on the topic[10, 17], but may still be too complex for technicians to understand. Attending dedicated training takes time, which is often in short supply: 90% of workers in the

industry are freelancers[22], so if these people are not working, they're not making money. This can make it hard to dedicate time to training, so on-the-job education can be preferable.

5.4 Protocols

Particularly in the sound department, there are a variety of data transfer systems which happen to use Ethernet cabling, but not IP. AES50[6] sits at layer 1 in the OSI model, using Ethernet cabling due to its ubiquity, but not the Ethernet frame structure or media access control features. Audio Video Bridging (AVB)[16] comprises a collection of IEEE standards and sits at layer 2, transporting data in standard Ethernet packets and therefore can use standard Ethernet hubs and switches.

We, however, are primarily interested in standards which sit at layer 3 or higher. Just like on the Internet, most layer 3 traffic uses IP combined with layer 4 protocols like UDP or RTP. AVB can be configured to work in this way instead of at layer 2. Other layer 3 standards include AES67[7], Dante[24], Art-Net[3], sACN[8], and NDI[23].

The diversity of protocols at layer 3 – encompassing the transport of lighting control data, sound, and video – is what enables converged networking and is why we focus on layer 3 and higher in this project. Because these systems share many core technologies with standard computer networks, we can create entertainment networks with off-the-shelf switches and cabling: the hardware is readily available so easy to get started with. However, this introduces one of the challenges of venues' transition to IP: commodity hardware varies widely in its feature set, quality, age, and reliability, and usually in ways subtle enough that it is not obvious why a problem is happening. One of the aims of the tool we build in this project is to indicate these subtle issues early.

Some protocols, like Art-Net, perform much of their core functionality by acting as a wrapper around the original serial protocol (Figure 6). Others, like those focussed on audio and video, are technically much different to their traditional analogue counterparts: for example, Dante requires an analogue signal from a microphone to be digitised, then it packages up four audio channels into one packet for transmission and uses PTP to synchronise audio playback across the many potential devices in a system. Dante also uses the differentiated services code point in its packets to ensure that real-time audio and clocking data are appropriately prioritised across the network.

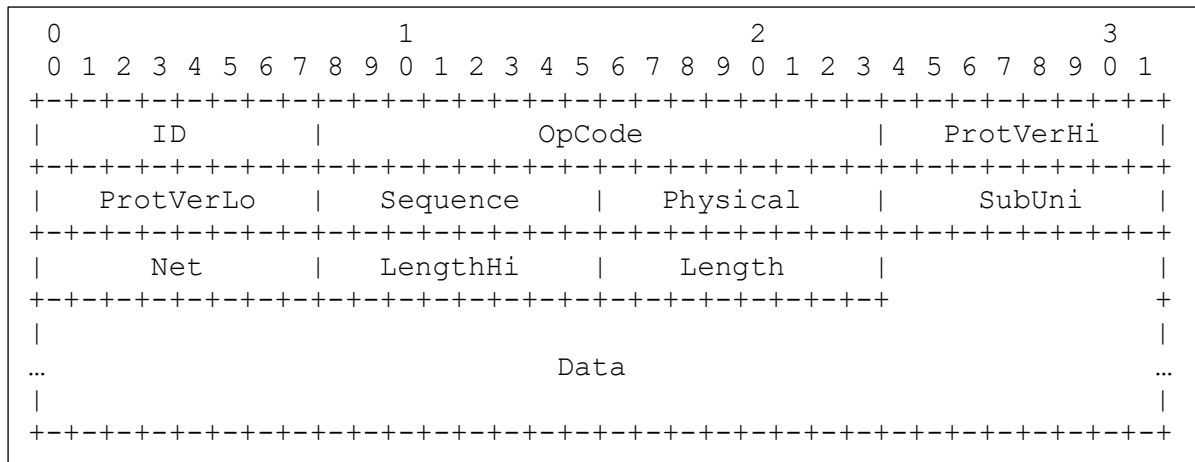


Figure 6: The packet structure for the ArtDmx packet in Art-Net is an array of serial DMX data preceded by some protocol management information and the universe number the DMX data refers to.

While there is a diversity of protocols available, there tends to only be one or two options for each type of data transfer. Those few options though tend to be well-supported – it is generally assumed that every digital lighting control console will support both sACN and Art-Net, and Dante’s website boasts its availability in “over 3,800 products from more than 550 manufacturers”[24]. This means there is a wide variety of hardware ready to be included in these converged networks, so finding suitable and compatible equipment for most applications should be possible.

5.4.1 Unreliable networks in a mission-critical application

Converged networking makes the network a single point of failure for a venue’s operation. It is therefore interesting that UDP is a common choice of transport layer protocol, given it (and IP below it) are unreliable technologies.

When choosing a transport layer protocol, deciding between TCP and UDP usually comes down to whether reliable transfer or fast transfer is more important.

In lighting, because of the serial origins of the DMX protocol, each DMX packet contains a complete universe of data and is rebroadcast regularly (at least once per second, but usually much more often than that). This would suggest that a dropped packet every so often does not matter so much, since the same data will be sent again very soon and probably arrive before a negotiated retransmitted packet would. In this case, the overhead of TCP is probably not worth it, so UDP is a natural choice.

In sound and video, it is difficult at first glance to prioritise reliable transfer or low latency. Dropped packets would result in broken-up playback, which would be incredibly disruptive to a live performance. High latency would result in out-of-sync playback of sound effects and video, or out-of-sync action on stage and amplification – possibly equally disruptive to the show. These protocols’ use of UDP and clocking systems like PTP suggests that they prioritise low latency, and that they assume the network they run on is reliable. Ensuring this

reliability (particularly when using heterogeneous commodity networking hardware as mentioned previously) is a challenge, and one which this tool aims to help with.

Audio-over-IP protocol Dante includes an interesting feature when exploring the use of unreliable networks. The Dante Controller software shows a histogram of packet delivery time for all transmitters, which highlights how it relies on PTP to synchronise packet delivery when the network can add such variability and unreliability to delivery time.

5.5 Current tools

Outside the entertainment industry, various major equipment manufacturers offer network design and modelling tools[27]. These are simply far too complicated for the average theatre technician without a computer science background to understand, and are not tailored to the industry — they are for enterprise system administrators.

Luminex manufacture a line of entertainment-specific network switches called GigaCore[28], and offer a design tool targeted at the industry named Araneo[29]. You can add switches and other equipment to a project, and the tool will perform some basic checks for configuration consistency and allow you to push your planned configuration to their devices. A primary limitation of this tool is the fact it is maintained by a manufacturer, so is not particularly compatible with heterogeneous environments comprising a variety of network switches and other equipment from lots of manufacturers. Additionally, the tool assumes a basic knowledge of networking concepts, which is not always the case.

CAD applications like Vectorworks have entertainment add ons[30] which add industry-specific features like load calculations, lighting fixture address management, and so on, but none have strong features for planning networks configurations or even IP addresses.

A design tool which truly supports the heterogeneity of many small-to-medium scale or piecemeal entertainment networks (such as the one we propose here) would best be built in an academic or community setting, because there is little commercial benefit to its production for any single equipment manufacturer.

6 Ethics

Ethical approval was initially sought for a single questionnaire, which comprised a user requirements survey followed by an evaluation of the developed tool. This structure was used to simplify the ethical approval process, due to the tight timescales of this project. This questionnaire was approved.

As the project progressed, the focus shifted away from system design and more towards how people interact with the system. This led to an amendment being requested to split the single questionnaire into two: first asking about user requirements, then later asking for users to evaluate the project. This would provide some real user requirement information earlier in the development cycle so it could feed into the design and implementation process, instead of only the evaluation. This amendment was also approved.

The ethical approval document and amendment are included in section 16.

7 Software engineering process

A human-centred (or user-centred) approach as described in ISO 9241-210:2019[31] drove this project, because its benefits map closely our objectives – namely, improving accessibility of information and supporting technicians to use networks to their fullest potential. The standard suggests the following benefits can be found:

“Systems designed using human-centred methods improve overall quality, for example, by:

- a) increasing the productivity of users and the operational efficiency of organizations;*
- b) being easier to understand and use, thus reducing training and support costs;*
- c) increasing usability (effectiveness, efficiency and satisfaction)*
- d) increasing accessibility (for people from a population with the widest range of user needs, characteristics and capabilities);*
- e) improving user experience;*
- f) reducing discomfort and stress;*
- g) providing a competitive advantage, for example by improving brand image;*
- h) contributing towards sustainability objectives.”[31]*

Points (a), (e) and (f) form the motivation for starting this project, while points (b) and (c) are part of the project objectives and non-functional requirements outlined in section 8 below. The remaining points, while perhaps not explicitly in scope, are certainly not bad outcomes to pursue and can only help to improve the quality of software produced. Indeed, they would probably be included in the objectives of a larger project.

Poor usability risks missing a key project objective and mis-advising users, resulting in incorrect systems being installed. This would result in disruption to events and may discourage people from using computer networks at all in this application. A human-centred approach improves usability, and thus reduces these risks. That said, the structure described in the ISO document is not entirely applicable to this development environment: this is an individual project where time is limited. Recognising this, some of the standard requirements and recommendations are not applicable, as outlined in section 23.

Human-centred design is built atop four activities, which are outlined in Table 1, along with the methods used to carry out those activities in this project. These activities are integrated with system development by following an agile-inspired approach, tightly coupling design and implementation so the output of the planning activities remained front of mind and could feed into frequent iteration and improvement.

Activity	Methods
Understanding and specifying the context of use	<ul style="list-style-type: none"> • Section 5, Context survey • Section 8.2, Stakeholders • Section 8.4, Requirements survey • Section 8.5, Personas
Specifying the user requirements	<ul style="list-style-type: none"> • Section 8.1, Project objectives • Section 8.4, Requirements survey • Section 8.3, Functional and non-functional requirements
Producing design solutions	<ul style="list-style-type: none"> • Section 10, Design
Evaluating the design	<ul style="list-style-type: none"> • Section 12, Evaluation and critical appraisal

Table 1: Activities of human-centred design and the methods used to do those activities in this project.

Initially, a more traditional waterfall development approach was trialled, but this was quickly scrapped as when working individually and on a short timescale it became hard to follow the principles of human-centred design – namely, trying to refine designs based on experience using the application and understand how users would interact with the system as a whole.

The functional requirements in section 8.3.1 later are listed in a numerical order because they each build on top of each other. So, after creating a basic application framework and structure, one functional requirement was focussed on at a time in order, building on previous work and leaving clear expansion points which would be required for future development. This means a working version of the software was always available – even if some methods or buttons would result in a logged message rather than an action until a certain feature was implemented.

The application is sorted into components where appropriate, allowing for code re-use throughout the backend of the application, and consistency across its frontend UI elements.

A non-functional requirement of the system is that it is web-based. This resulted in TypeScript being the programming language of choice, with all dependencies supplying appropriate type signatures and the design tool adding its own. TypeScript encouraged the use of type-driven development, where the types act as a sort of test and ensure data passed around the application is valid. It also allows IDEs to offer intelligent suggestion of different object field names, which was very useful given the large and complex nature of the data structures used to represent devices, protocols, and network designs.

8 Requirements

8.1 Project objectives

The project objectives were specified at the project proposal stage as part of the DOER and are the foundations of the work being done. They are mostly described in terms of tasks a user may want to do within the system. The primary objectives of the project were defined as:

- **Database construction:** Create an extendible dataset showing how different hardware products and software protocols use the network and what they require to work efficiently.
- **Design tool:** Backed by the dataset, create a network design tool. The user will select their hardware and describe the kinds of data they are transporting, and the tool will validate their configuration.
- **Accessibility of information:** Through an evaluation process involving users in the target audience, ensure the tool is useful in practice and meets the goal of inspiring confidence in system design and enabling learning.

The secondary objectives of the project were defined as:

- **Extendibility:** Provide mechanisms to add to the dataset of hardware and software, preventing the tool from becoming outdated. Look at areas where other services could link into the core tool or plugins could be added.
- **Documentation generation:** Based on the user's design, create deployment paperwork which can be used in the field to install the system in a venue.
- **Design wizard:** Allow the user to specify some broad system requirements, and make the tool generate a valid network configuration indicating any missing parts.

8.2 Stakeholders

Considering the project objectives, three stakeholders have been identified for this platform.

1. Technicians looking to design, maintain, and implement networks.
2. Manufacturers and vendors who want their equipment and protocols included in the application.
3. The administrative team who host and manage the application in production.

Our primary stakeholder is group 1, the technicians, as these are the users we have access to for requirement elicitation and evaluation. Future work could involve consulting with users in groups 2 and 3 to establish their exact requirements for the platform and further develop the system.

8.3 Functional and non-functional requirements

After carrying out the context survey above and getting an improved understanding of the gaps in existing work in the field, the project objectives were translated into functional and

non-functional requirements. These are stated such that they can map directly to tests when evaluating the project.

8.3.1 Functional requirements

1. A data import process must allow metadata about arbitrary hardware and software to be added to the system.
2. A user must be able to select hardware and software from the system and add these entities to a network.
3. A user must be able to connect network ports on devices to each other to build a system.
4. The system must assign appropriate identifiers (addresses, hostnames, etc) to each entity in the network, and allow the user to customise them so long as the chosen customisations are valid within the system.
5. Using entity metadata and the configuration of entities in a network, the system must indicate potential errors to the user and recommend appropriate solutions (such as inappropriate configuration settings, etc).
6. The system must offer plugin hooks allowing it to be extended with additional functionality created by other developers.
7. In addition to the editor interface, the tool must generate printable documentation suitable for distribution to a crew installing the network in a venue.
8. A user must be able to freely browse the hardware and software the system knows about, with the database available as a ‘reference library’ of information.

8.3.2 Non-functional requirements

1. The system must be a web application, so it can be used across multiple platforms.
2. The system must assume the user knows nothing about networks, IP, or related technologies. (It can, however, assume a working knowledge of other venue or performance-related technologies, for example DMX luminaires and audio equipment).
3. Users must be able to operate the system without separate training.
4. When surveyed, users will report feeling more confident deploying a network in a venue than they would have done without using the system.

8.4 Requirements survey

A requirements survey was sent to a group of potential users. The users were recruited anonymously from within the University of St Andrews, invited through their work department (University or Students’ Union technical production) or a relevant student society. This let us reach people across a reasonable range of experience levels and backgrounds. There were 10 responses to the survey – a small sample size, but a reasonable turnout given the size of the target audience to begin with. The questionnaire is included in section 17.

The first part of the questionnaire aimed to enrich the context of the project by recording technicians’ experience level and confidence working with computer networks compared to

their ‘bread and butter’ work. Responses were received from people with 2-8 years of experience in the industry (median 5, mean 4.6). All respondents had experience with lighting with most indicating it as their specialism. The majority of people had experience with sound, and half had experience with video. Nobody indicated that video was their specialism (Figure 7, Figure 8).

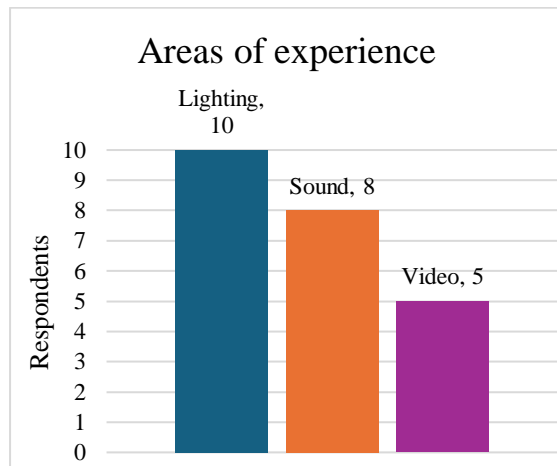


Figure 7: Respondents' areas of experience.



Figure 8: Respondents' self-stated specialism.

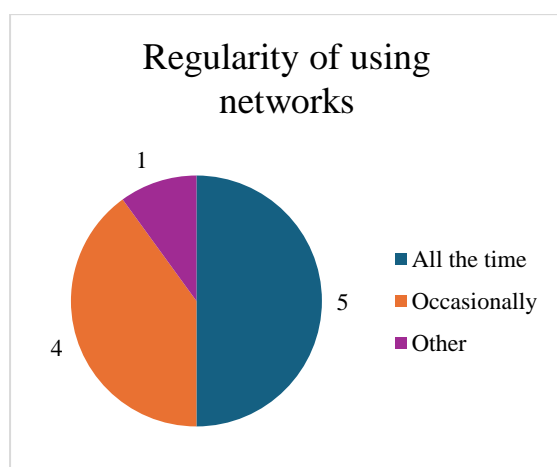


Figure 9: How regularly respondents work with computer networks.

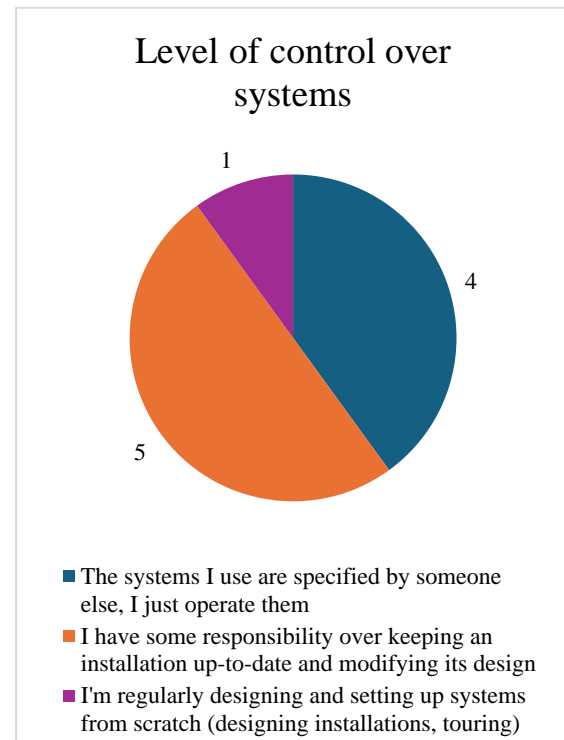


Figure 10: The level of control respondents have over the systems they work with.

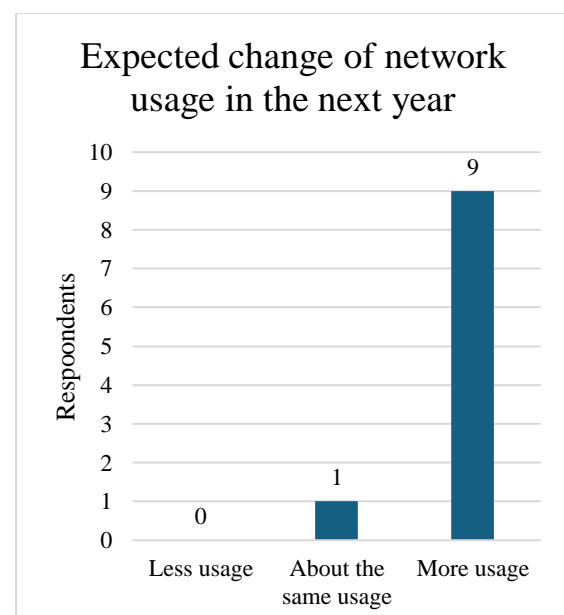


Figure 11: Whether respondents think they will use networks more, the same amount, or less in the next year.

Half of respondents stated they work with computer networks all the time (Figure 9), and half also stated they have at least some responsibility over maintaining an installed system (Figure 10). Only one respondent regularly sets up systems from scratch, which makes sense given the survey was distributed to people who primarily work in one or a small number of venues. Perhaps the tool will give people confidence to build from scratch more regularly. In response to the question about how regularly the participant uses networks, one person stated that the use of a network is often transparent to them. This could be because they work in a venue with a reasonably static core infrastructure, where the day-to-day operations do not require a lot of hands-on network management.

The vast majority of respondents think their usage of computer networks will increase in the next year; nobody expected to use networks less (Figure 11). This reinforces a key motivation of this project: that computer networking is becoming more and more important in the industry, and it is therefore an area worth researching.

Participants were asked to rate their confidence working in either their indicated specialism or general crew work (depending on whether they indicated a specialism or not), followed by their confidence setting up and using computer networks in their work. They rated their confidence on a sliding scale from -2 to +2 (Figure 12).

-2	Not confident, I would need lots of help to do tasks in this area.
-1	
0	No experience or not sure.
+1	
+2	Very confident, I could do most tasks in this area independently and help others too.

Figure 12: Confidence scale used in requirements survey.

From this, we can take a general sentiment of confidence by looking at the number of answers above and below 0, and more specific confidence ratings by comparing the magnitude of the numbers.

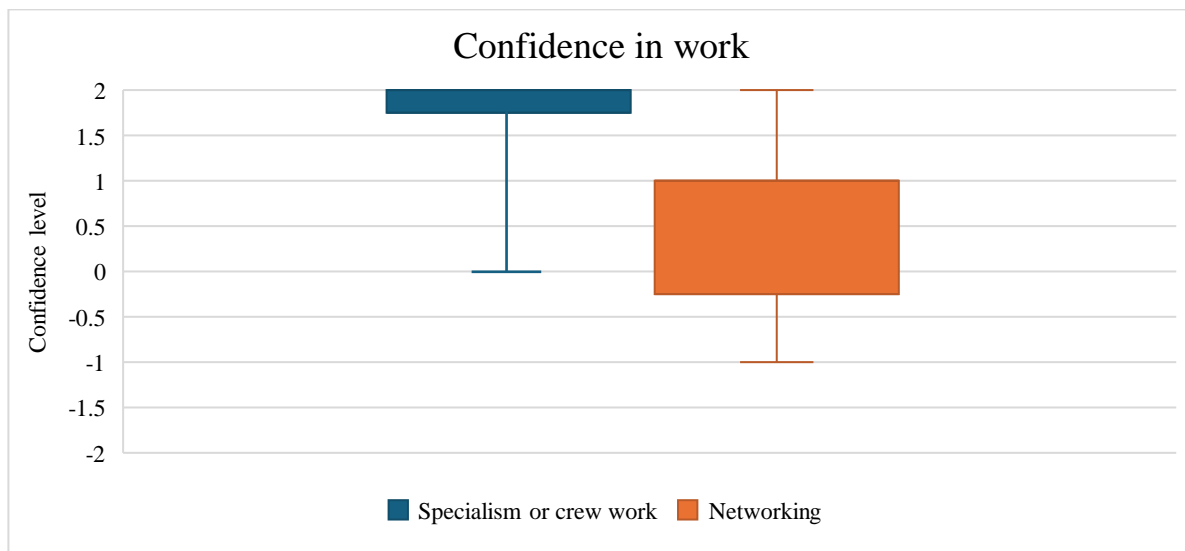


Figure 13: Confidence level in different tasks represented as a box plot. The whiskers are set to the minimum and maximum values.

Respondents were generally very confident in their main work, and this was consistent across the group (mean 1.7, median 2, IQR 0.25). Confidence levels in networking were generally lower and more varied, but people on average still felt more confident than not (mean 0.6, median 1, IQR 1.25). This is consistent with the other findings from the questionnaire, where around half of respondents are only occasionally using networks and primarily operate rather than manage and maintain systems.

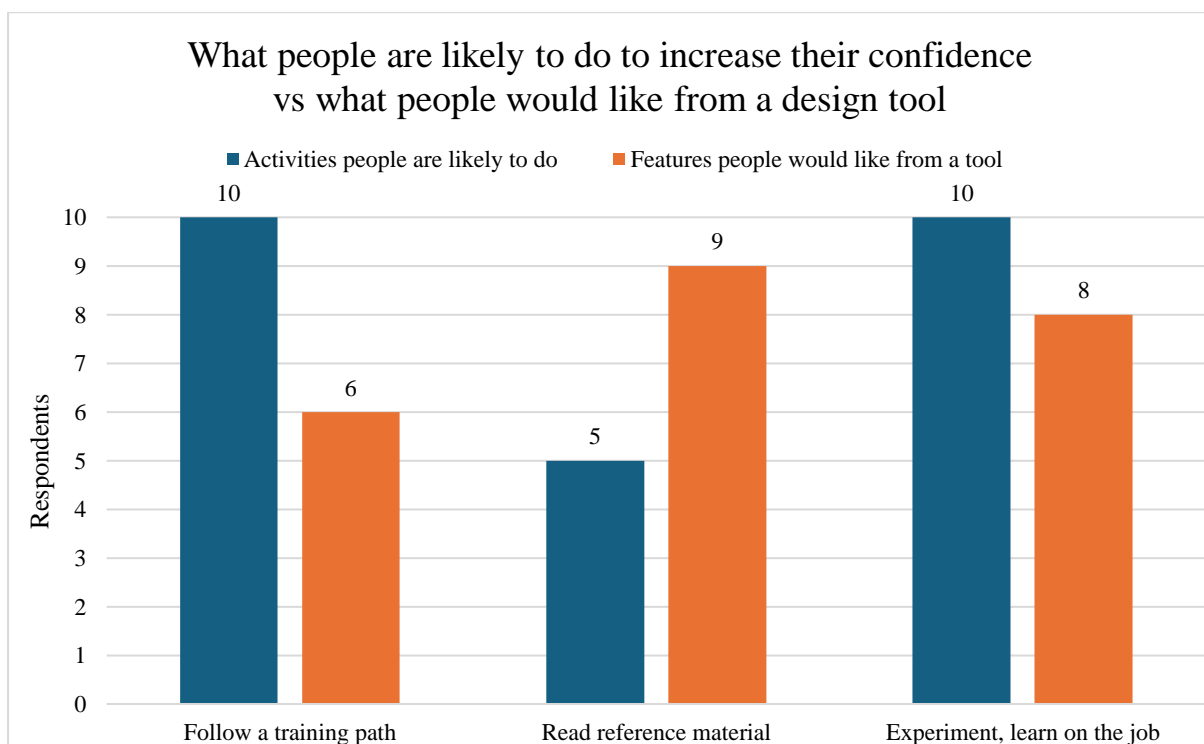


Figure 14: Activities people reported they would be likely to do to increase their confidence with networks, compared to the features they would like from a design tool. These were presented to respondents as two separate questions.

Finally, participants were asked separately what activities they would be likely to do to increase their confidence using computer networks at work, and what features they would like from a design tool (Figure 14). These two questions had answers which correlated, so we can identify participants' broader desires and how they think the tool fits into those desires.

Everyone said they would be likely to attend training and that they would want to practise and learn on the job, while only half said they would read reference material like books. This highlights a need for a hands-on approach to learning. In the context survey, it was predicted that appetite for training would be low due to the risk of lost income while not working, but this did not transpire in this survey. This may be due to the demographic of the survey: those in permanent work or part-time work while studying do not risk loss of income while learning in the same way as freelancers do, as training time would usually be paid by the employer.

It is interesting the reference material (in the form of a database of device and protocol information) was the most popular feature request but reading reference material was the least popular activity: this could be to do with the wording of the options, or it may indicate the challenge of ensuring heterogenous equipment from different manufacturers is compatible. It is also interesting that only 60% of respondents said they would be likely to use a design tool to learn more about networks, but 80% said they would like reassurance that plans they make will work. Perhaps users think a design tool would be too complex as a learning device, but still seek support in building systems.

With a longer project timescale, it would be beneficial to do a second round of user evaluation – perhaps with an expanded questionnaire, or a discussion with focus groups. This would enable a deeper understanding of some of the trends observed: for example, we could clarify the two points of uncertainty in the previous paragraph.

The main takeaways from this requirements survey are:

- The sampled group are generally very confident in their day-to-day work, but less confident in using networks, despite half of respondents using networks all the time, 60% regularly designing or maintaining networked systems, and 90% expecting to use networks more in the next year.
- Technicians prefer learning-by-doing over reading, and want a tool which provides access to data about their equipment, allowing them to experiment with ideas and get reassurance that their designs will work in the field.

From this survey, the 8th functional requirement was added, to include a dedicated data browser inside the application rather than only using the database to support design validation. This was based on the strong user demand for reference material in the application. The survey reinforced that the rest of the functional and non-functional requirements specified are all features which users want from the application.

8.5 Personas

A persona is a fictional character designed to represent the needs of a real user, incorporating information about a person's background, existing knowledge, personal preferences, needs, environment, and so on[9]. Compared to a list of user stories, personas highlight the 'human' element in the design process. In a group setting, they can help to facilitate communication, but even in this individual project they help to reason through problems by considering how these different people would interact with the application or talk about it between each other.

Personas are built from a combination of assumptions, the designers' prior knowledge and expertise in the field, and quantitative data. In this project, with a reasonably small number of responses to the requirements survey, assumptions and prior knowledge need to be relied on quite heavily when making design decisions, and personas allow this to happen in a more structured and describable manner².

Four personas were created to cover the different types of users the application is made for. The functional and non-functional requirements each persona encompasses are included for completeness.

Name	Lachlan
Age	23
Job title	Venue Technician
Education level	High school
Biography	Lachlan is new to the industry and works full-time in a concert venue. He's keen to update the venue systems so they're more reliable and simpler — he struggled to learn them when he arrived — and thinks the computer networking he learnt in school would be well-applied here.
Personality traits	Enthusiastic about new things, keen to teach others, enjoys video games in his spare time.
Frustrations	Resistance from older colleagues about trying new equipment and configurations which deviate from what they're used to.
Wants and needs	A way to express network designs to colleagues, showing them that the new way is easier and reassuring them that the configuration he has suggested will work.
Encompasses requirements	Functional: 2, 3, 4, 7

Figure 15: Lachlan's persona.

² Although the author's industry experience informed the writing of these personas, they are fictitious characters containing no real personal information and do not map to natural people.

Name	Fiona
Age	58
Job title	Production Manager
Education level	University degree
Biography	As a freelancer, Fiona works in a variety of venues (both proper theatres and function halls), usually bringing her own equipment and a small crew to set up and operate it. She has lots of experience, starting as a lighting technician and now operating her own production and hire company.
Personality traits	Approaching retirement and looking to keep work as quiet and simple as possible so she can spend more time with her family.
Frustrations	Hauling lots of heavy cabling and equipment around, wasting time while packing and during fit ups. Although she's heard of networked solutions being easier, she doesn't want to rely on it for a show because she doesn't trust she'll get it right.
Wants and needs	Guidance on how to set up networks with the equipment she has, a way to find out what else she needs to buy, and how to plug it all in when she gets to a venue. The ability to share setup plans easily with crew on site.
Encompasses requirements	Functional: 2, 3, 4, 5, 7 Non-functional: 2, 4

Figure 16: Fiona's persona.

Name	Isla
Age	38
Job title	Deputy Technical Manager
Education level	College
Biography	Isla works in a local theatre which produces its own shows. The theatre is growing and starting to put on larger and larger productions, so she is regularly hiring in extra equipment to support events and looking to purchase some items so they have to hire less.
Personality traits	Cautious and thorough, very professional. Likes to ensure that all plans are well-considered and clear so everyone's on the same page and there are minimal problems and delays during a show run.
Frustrations	It's hard to find compatible equipment across manufacturers and confirm what different devices and protocols require to work effectively together.
Wants and needs	A way to explore devices by the protocols they use, an easy reference for devices and protocols no matter the manufacturer so she can make informed decisions about what she needs to hire or will eventually invest in.
Encompasses requirements	Functional: 1, 7, 8

Figure 17: Isla's persona.

Name	Evan
Age	25
Job title	Part-Time Technician
Education level	In education while working
Biography	Evan has little experience in industry as he picked this job up to support his studies, so he's learning a lot from his colleagues. Because he's also in full-time education, he doesn't work many hours and doesn't have much time outside of work to dedicate to learning, so spends a lot of time learning on the job.
Personality traits	Hard-working, but focussed on his studies more than the venue he's working in so doesn't really care much about work.
Frustrations	The venue system is changing regularly as they transition to a more network-based approach, and since he's not in very often, he's struggling to keep up to date with how things work and therefore also struggling to fix problems he comes across.
Wants and needs	Access to the designs his full-time colleagues have made, so he can see what things should be like when he's troubleshooting, and record any changes he makes.
Encompasses requirements	Functional: 7 Non-functional: 1, 2, 3

Figure 18: Evan's persona.

The set of personas covers users with varying experience levels in both their primary work and computer networking, a variety of motivations and levels of investment for why they want to use networks in their work, and differing workplace remits. Intentionally excluded from these personas is someone who has no interest in using network-based approaches to their work. In many cases, analogue approaches are still a valid way to 'get the job done' to this day, so people who are looking at networked solutions are generally doing so because they already believe the benefits are worth the transition or are working in a venue where a network-based approach is prescribed for them. Someone who has no interest in networks would first need to be convinced that this new approach is worth the effort to learn and achieving that is probably a marketing or education task beyond objectives of this project.

9 Measurement and data collection

A primary objective of the project was to create an extendible dataset of information about how different devices and protocols use the network. In a way, this section is a ‘mini-project’ of its own, with its results feeding into the later software engineering. With this in mind, we set out a brief hypothesis and goals, explain the methodology, then present and analyse the results.

From the research carried out in the context survey, we know that despite there being a wide variety of hardware products from different manufacturers, there are a comparatively small number of communication protocols used across them (hence why they can be interoperable). These protocols are either implemented by each device manufacturer according to a specification, or the protocol vendor gives the hardware vendor an SDK or hardware chipset to include in their device. This suggests that across different hardware products, we can expect the same protocol to have roughly the same network utilisation characteristics – minus any additional functionality the device manufacturer adds on top.

The goals of this section are therefore:

- To show – insofar as possible with the resources available to the project – that different hardware products using the same communication protocols use the network in a similar way.
- To analyse how different devices and protocols use the network, and therefore what sort of validation the design tool needs to do.
- To centrally record appropriate data (based on the previous analysis step) in an extendible format usable by the design tool software.

9.1 Methodology

Thanks to the generosity of other University departments and affiliates, a selection of professional-grade entertainment equipment was available to the project for measurement (Table 2). We focus on measuring how devices transmit data rather than receive it, and on measuring protocols which are supported by devices from different manufacturers (as opposed to strictly vendor-specific ones).

Device	Type	Protocols implemented
Avolites Tiger Touch II[5]	Lighting console	Art-Net, sACN
ETC Ion Classic[32]	Lighting console	Art-Net, sACN
Behringer X32[33] with X-DANTE[34] card	Sound desk	Dante
DiGiCo A168D STAGE[35]	Stage box	Dante
Audinate AVIO-DAI2[36]	Stage box (input only)	Dante
Resolume Arena 7[37]	Computer software for video design	NDI
NDI HX Camera[38]	Mobile device software for camera streaming	NDI-HX

Table 2: Devices available for measurement, including their type and the protocols they implement.

As we have access to multiple devices which implement sACN, Art-Net, and Dante, we will be able to check how their implementations differ across device manufacturers.

Unfortunately, we only have one implementation of NDI and NDI-HX, so we cannot perform comparisons, but we take the measurements anyway so the design tool has some information about networked video solutions.

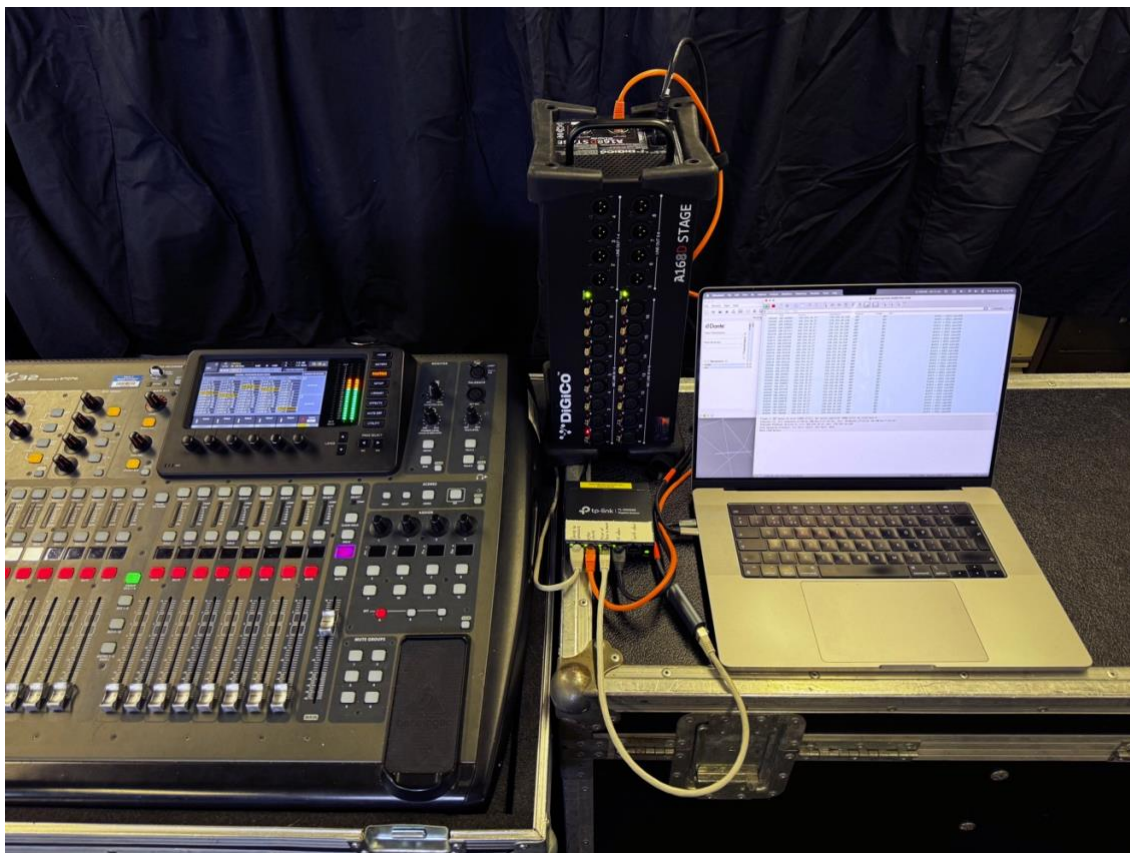


Figure 19: Behringer X32 and DiGiCo A168D STAGE connected to the measurement setup.

A TP-Link TL-SG605E managed gigabit switch was used for measuring. Port 1 was used to connect the device being measured (i.e. the transmitter). Port 2 was used for any other AV device required for the measured device to work correctly (i.e. a receiver for the transmitter to direct its stream to). Port 3 was a mirror of port 1, used to take a packet capture of the device working, and port 4 was used to connect a laptop for any administration or setup tasks required to get the measured device to start transmitting. A schematic of this system is provided in Figure 20. The packet capture and analysis was carried out in Wireshark[39] (Figure 21).

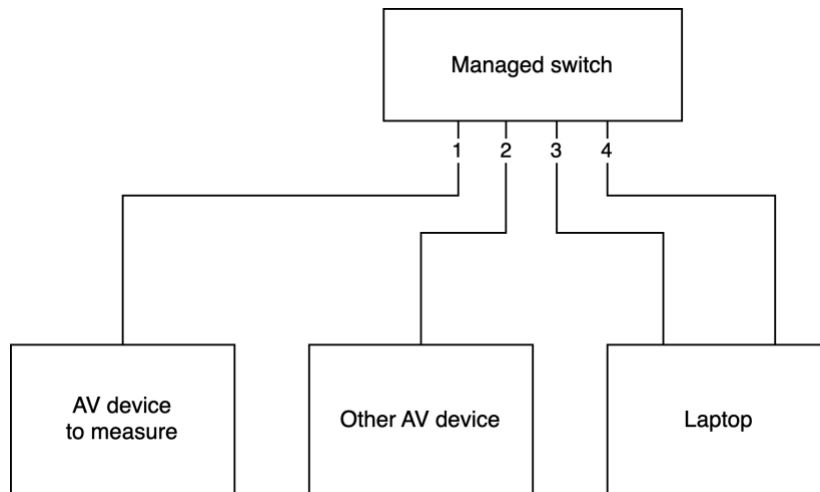


Figure 20: Schematic of the device measurement setup.

This setup was transported to the equipment in its home venue. The equipment was disconnected from its normal set up and connected to the measurement switch, then transmission started. Where possible, a ‘data-intensive’ signal was sent – for example, constantly changing lighting commands, random noise, and colourful complicated video. This was to record the worst-case network utilisation, where compression or other optimisations may not perform so well. Each configuration was recorded for 60 seconds, then the packet capture was analysed qualitatively by hand and statistics stored quantitatively in a spreadsheet.

File

Name:

[file path]

Length:

59 MB

Format:

Wireshark/tcpdump/... - pcap

Encapsulation:

Ethernet

Snapshot length:

262144

Time

First packet:

2024-02-20 10:32:23

Last packet:

2024-02-20 10:33:24

Elapsed:

00:01:00

Capture

Hardware:

Unknown

OS:

Unknown

Application:

Unknown

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	364889	364889 (100.0%)	—
Time span, s	60.585	60.585	—
Average pps	6022.8	6022.8	—
Average packet size, B	147.5	147.5	—
Bytes	53650346	53650346 (100.0%)	0
Average bytes/s	885 k	885 k	—
Average bits/s	7084 k	7084 k	—

Figure 21: Sample statistical report from a packet capture.

9.2 Results and analysis

9.2.1 Art-Net

We measure Art-Net transmission in broadcast mode, which is not advised by the specification[3], but is a widely implemented option for convenience when setting up a system. Art-Net communicates entirely using UDP, using DHCP or static IP addresses. By default, Art-Net devices are supposed to self-assign a class A address based on the device MAC address. Traditional DMX retransmits data constantly, but Art-Net should only transmit a packet when the DMX values change (or approximately every 4 seconds if the values are not changing). So, to capture the worst-case bandwidth usage, the transmitting lighting console was set to fade its channels up and down constantly, thus changing the DMX values

all the time. This was done by patching 512 dimmer channels³ at a time (since there are 512 addresses in a universe), then running an intensity effect across them.

Two transmitters were measured: Avolites Tiger Touch II (running Titan version 15.1), and ETC Ion Classic (running Eos version 3.1.5). Tiger Touch was measured up to 32 universes transmitted and Ion up to 8 universes. Avolites do not recommend running more than 16 universes from one Tiger Touch console to avoid overloading its processing engine, but they do not enforce this limit – some sluggish UI performance was experienced on the console in the last few tests but the data transmitted still seemed to be good. The Ion console used had a hard limit of 6144 addresses in use at a time, which is why testing on that console stopped at 8 universes.

Universes transmitted	Packets	Time (s)	Mean packets/sec	Mean packet size (B)	Total bytes	Mean Bytes/sec	Mean bits/sec
0	711	60.282	11.8	160.5	114129	1893	15 k
1	3107	60.477	51.4	481.5	1495149	24 k	197 k
2	5526	60.45	91.4	521.5	2879637	47 k	381 k
3	7944	60.449	131.4	536.5	4262507	70 k	564 k
4	10412	60.699	171.5	545.5	5674568	93 k	747 k
8	20061	60.528	331.4	558.5	11194751	184 k	1479 k
16	39617	60.798	651.6	564.5	22379400	368 k	2944 k
24	59017	60.827	970.2	567.5	33472242	550 k	4402 k
32	77380	60.859	1271.5	569.5	44038617	723 k	5788 k

Table 3: Packet capture summary data for Avolites Tiger Touch II transmitting Art-Net.

³ A dimmer channel refers to a lighting fixture which is controlled by a single DMX address. The name ‘dimmer’ comes from the fact this single control channel is usually light intensity controlled by a dimmer unit separate to the lantern itself. Patching a fixture refers to telling the console the about the existence of a fixture and its address, so the console can control it. DMX as standard does not have a way for lighting consoles to check whether what is patched is correct or not, so it does not matter that 512 fixtures were patched but no lights were actually plugged in – the console will transmit control data nonetheless.

Universes transmitted	Packets	Time (s)	Mean packets/sec	Mean packet size (B)	Total bytes	Mean Bytes/sec	Mean bits/sec
0	258	60.579	4.3	311.5	80401	1327	10 k
1	2785	60.736	45.9	547.5	1523775	25 k	200 k
2	5436	60.921	89.2	559.5	3043341	49 k	399 k
3	8056	60.626	132.9	563.5	4537356	74 k	598 k
4	10602	60.15	176.3	565.5	5996833	99 k	797 k
8	21200	60.58	349.9	568.5	12061585	199 k	1592 k

Table 4: Packet capture summary data for ETC Ion Classic transmitting Art-Net.

The results (Table 3, Table 4) indicate that Art-Net does behave the same way on different devices. The measurement when no DMX data transmission was happening was negligible at 10 kbit/sec, and the overall throughput remained modest throughout all tests, sitting at around 200 kbit/sec when transmitting 1 universe and reaching 5.7 Mbit/sec when transmitting 32 universes. Each time another universe was added, there was an increase in throughput of around 199 kbit/sec.

9.2.2 sACN

Selecting between Art-Net and sACN on a lighting console is a simple configuration option in the console settings, so the setup for sACN measurement is the same as Art-Net, with this protocol setting changed. The same consoles were used for measurement, again measuring up to 32 universes on Tiger Touch II and 8 universes on Ion Classic.

Universes transmitted	Packets	Time (s)	Mean packets/sec	Mean packet size (B)	Total bytes	Mean Bytes/sec	Mean bits/sec
0	361	60.681	5.9	238.5	86051	1418	11 k
1	2690	60.598	44.4	625.5	1682394	27 k	222 k
2	5086	60.628	83.9	653.5	3322106	54 k	438 k
3	7530	60.631	124.2	660.5	4975272	82 k	656 k
4	9956	60.747	163.9	666.5	6633866	109 k	873 k
8	19645	60.649	323.9	673.5	13223788	218 k	1744 k
16	38809	60.273	643.9	676.5	26245140	435 k	3483 k
24	58138	60.299	964.2	677.5	39400134	653 k	5227 k
32	78848	60.443	1304.5	668.5	52671912	871 k	6971 k

Table 5: Packet capture summary data for Avolites Tiger Touch II transmitting sACN.

Universes transmitted	Packets	Time (s)	Mean packets/sec	Mean packet size (B)	Total bytes	Mean Bytes/sec	Mean bits/sec
0	500	60.657	8.2	507.5	253797	4184	33 k
1	2835	60.579	46.8	650.5	1843596	30 k	243 k
2	5538	60.63	91.3	664.5	3680597	60 k	485 k
3	8250	60.744	135.8	669.5	5524983	90 k	727 k
4	10940	60.671	180.3	672.5	7352905	121 k	969 k
8	21752	60.646	358.7	674.5	14681933	242 k	1936 k

Table 6: Packet capture summary data for ETC Ion Classic transmitting sACN.

sACN throughput is again similar across different manufacturers' implementations, but slightly higher than the throughput of Art-Net (Figure 22) as each universe adds around 240 kbit/sec of data transfer. It is worth making clear though that this is only an increase of 40 kbit/sec, and the data rate is still very modest, so it is probably not worth factoring this difference into choosing between sACN and Art-Net. The mean packet size of sACN is larger, which makes sense as the sACN data packet header is 124 octets long[8] compared to Art-Net's 11 octets[3]. The differing size comes from sACN being built atop the broader ACN specification which has broader functionality, whereas Art-Net is first and foremost a lighting control system.

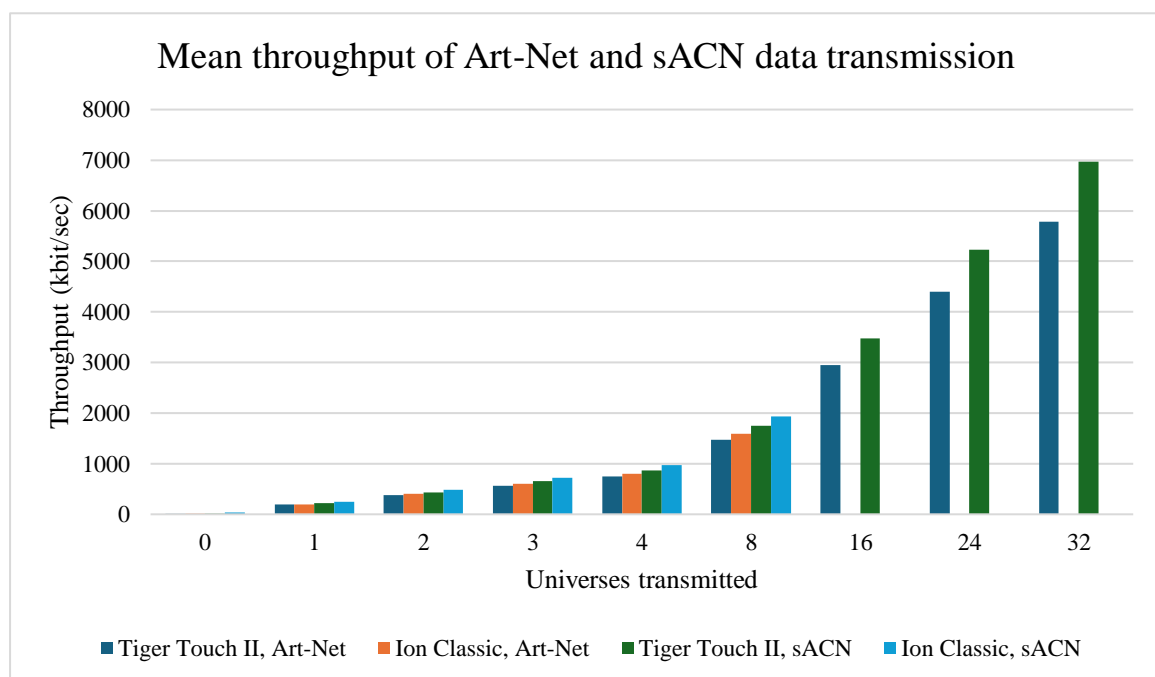


Figure 22: Mean throughput of Art-Net and sACN data transmission measured from Avolites Tiger Touch II and ETC Ion Classic.

9.2.3 Dante

A Dante flow contains four channels of audio[20], so we would expect the network utilisation to increase for every four additional channels of audio transmitted. Dante uses UDP for

audio, so we would expect a considerable volume of UDP traffic, with occasional PTP packets for clock sync across the network. We focus on unicast Dante flows, not multicast, as multicast is a reasonably advanced configuration and is not used often in small-to-medium scale systems, based on experience in venues.

The first setup tested was the DiGiCo A168D transmitting to X32. Firstly, a packet capture was taken of the device sitting on the network without setting up any audio flows. This showed a low throughput of 14 kbit/s and 14.1 packets per second, made up primarily of PTP clock messages and mDNS/ARP device discovery. This figure is low enough that it is treated as negligible in the rest of the Dante measurements, and the measured throughput is taken to be the result of audio transmission only.

Measurements were then taken of the device transmitting 1 channel of audio, then 2, 4, 6, 8, 10, 12, 14, and 16 channels (Table 7).

Channels transmitted	Packets	Time (s)	Mean packets/sec	Mean packet size (B)	Total bytes	Mean Bytes/sec	Mean bits/sec
0	851	60.567	14.1	129.5	110284	1820	14 k
1	361542	60.033	6022.4	147.5	53152979	885 k	7083 k
2	370817	61.57	6022.6	147.5	54518566	885 k	7083 k
4	364889	60.585	6022.8	147.5	53650346	885 k	7084 k
6	969867	80.642	12026.8	146.5	142557690	1767 k	14 M
8	731112	60.791	12026.7	146.5	107461555	1767 k	14 M
10	1091978	60.559	18031.5	146.5	160488850	2650 k	21 M
12	1108837	61.494	18031.6	146.5	162965045	2650 k	21 M
14	1464937	60.948	24035.7	146.5	215287982	3532 k	28 M
16	1634774	68.015	24035.3	146.5	240272312	3532 k	28 M

Table 7: Packet capture summary data for DiGiCo A168D transmitting Dante audio.

As hypothesised, the packet capture showed a lot of UDP audio transfer (which could be identified by its DSCP value, since Dante hardware tags audio and clocking packets differently), and a jump in throughput every 4 channels when a new Dante flow was added. Since each flow has space for 4 channels, the increase in data rate for each flow was consistent at approximately 7000 kbit/s, and the throughput figures within each ‘band’ of four channels were remarkably consistent too (Figure 23).

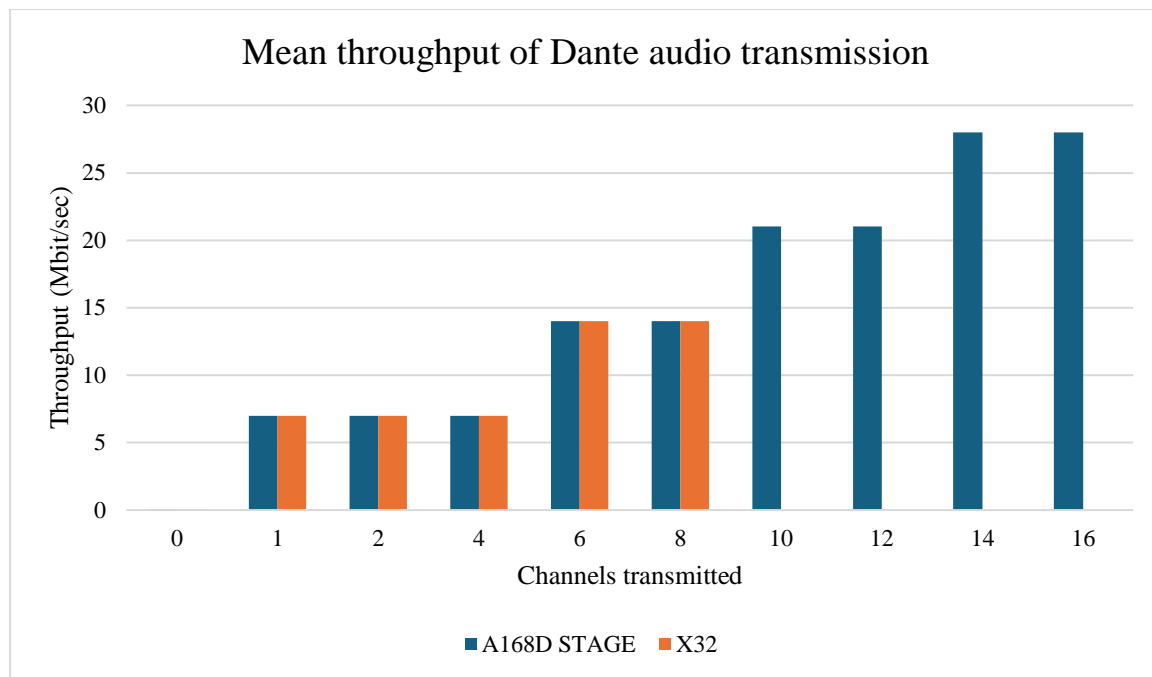


Figure 23: Mean throughput of Dante audio transmission, measured from the transmission of DiGiCo A168D STAGE and Behringer X32.

Next, the transmission of X32 was measured in a similar way (Figure 24). On X32, there is a separate network port for the Dante connectivity and the desk remote control feature. A common configuration is to connect the secondary Dante port to the desk remote control port – since the Dante secondary port acts as a switch, this means all networked functionality of the desk is available over one cable to the primary Dante port. Only 8 channels of transmission were measured from X32, because the receiver (A168D STAGE) can only receive 8 channels.

The throughput was slightly higher when there was no audio transmission, likely due to the desk remote control functionality, but still negligible. The rest of the audio transmission measurements were near identical. This supports the original hypothesis that for Dante, the expected throughput is the same no matter the specific device manufacturer.

Channels transmitted	Packets	Time (s)	Mean packets/sec	Mean packet size (B)	Total bytes	Mean Bytes/sec	Mean bits/sec
0	1398	70.327	19.9	185.5	259698	3692	29 k
1	365838	60.741	6023	147.5	53787275	885 k	7084 k
2	365070	60.613	6023	147.5	53674963	885 k	7084 k
4	364732	60.559	6022.7	147.5	53623700	885 k	7083 k
6	731203	60.796	12027.2	146.5	107470350	1767 k	14 M
8	753402	62.641	12027.2	146.5	110734046	1767 k	14 M

Figure 24: Packet capture summary data for Behringer X32 transmitting Dante audio.

9.2.4 NDI and NDI-HX

The project did not have access to any hardware which implements the NDI protocol, so we instead measure software implementations. We measure an implementation of the main NDI protocol from Resolume Arena 7 and the high-efficiency NDI-HX protocol from NDI HX Camera on an Apple iPhone. In both cases, the video transmitted was a colour-changing background with lines sliding across it in random directions, to minimise the efficiency of video compression. Resolume Arena ran this video as layered composition built in its editor for its measurement, and for NDI HX Camera the video was played on a screen which the camera recorded.

Measurements for Resolume Arena were taken at various video resolutions and frame rates (Table 8).

Resolution	Frame rate (fps)	Packets	Time (s)	Mean packets/sec	Mean packet size (B)	Total bytes	Mean Bytes/sec	Mean bits/sec
Video output off		371	60.379	6.1	160.5	59433	984	7874
1080p	1	4823	60.521	79.7	10858.5	52372156	865 k	6922 k
1080p	15	50738	61.329	827.3	15232.5	772844575	12 M	100 M
1080p	30	45577	60.652	751.5	13717.5	625224028	10 M	82 M
1080p	60	89960	60.683	1482.5	13930.5	1253217428	20 M	165 M
1080p	120	232439	60.924	3815.2	13756.5	3197542241	52 M	419 M
4K	30	117505	60.892	1929.7	13879.5	1630899927	26 M	214 M
4K	60	241504	60.566	3987.4	13275.5	3206030211	52 M	423 M
8K	30	285847	60.928	4691.6	13042.5	3728097542	61 M	489 M
8K	60	360015	61.068	5895.3	13531.5	4871623005	79 M	638 M

Table 8: Packet capture summary data for Resolume Arena 7 transmitting NDI video at various resolutions and frame rates.

Like previous measurements, the background data rate when no video transmission is enabled is negligible, but the rest of the throughput values are much larger than other protocols – a 1080p 30 fps video has a data rate almost 3x greater than 16 channels of Dante audio and 12x greater than 32 universes of sACN control data. It is interesting that NDI used TCP instead of UDP for its data transfer, since every other protocol measured so far relied on UDP and the network being inherently reliable. This may come from NDI's origins in studio usage, where every frame arriving for recording may be more important than low latency.

The 1 fps measurement was intended to find out if NDI was able to send a complete frame of video in each TCP packet or not: the mean packets per second for this measurement was 79.7, so it would appear not.

There is a drop in throughput between 1080p 15 fps and 1080p 30 fps (Figure 25). The data rate approximately doubles between 30 fps and 60 fps, but increases by around 2.5x when comparing 60 fps and 120 fps. Similarly, when comparing 1080p 30 fps to 4K 30 fps, the difference in throughput is around 2.5x, despite a 4K image containing four times as many

pixels as a 1080p one. These observations suggest that NDI performs some optimisation at different resolutions, but it is not clear from public specification documents what that optimisation may be.

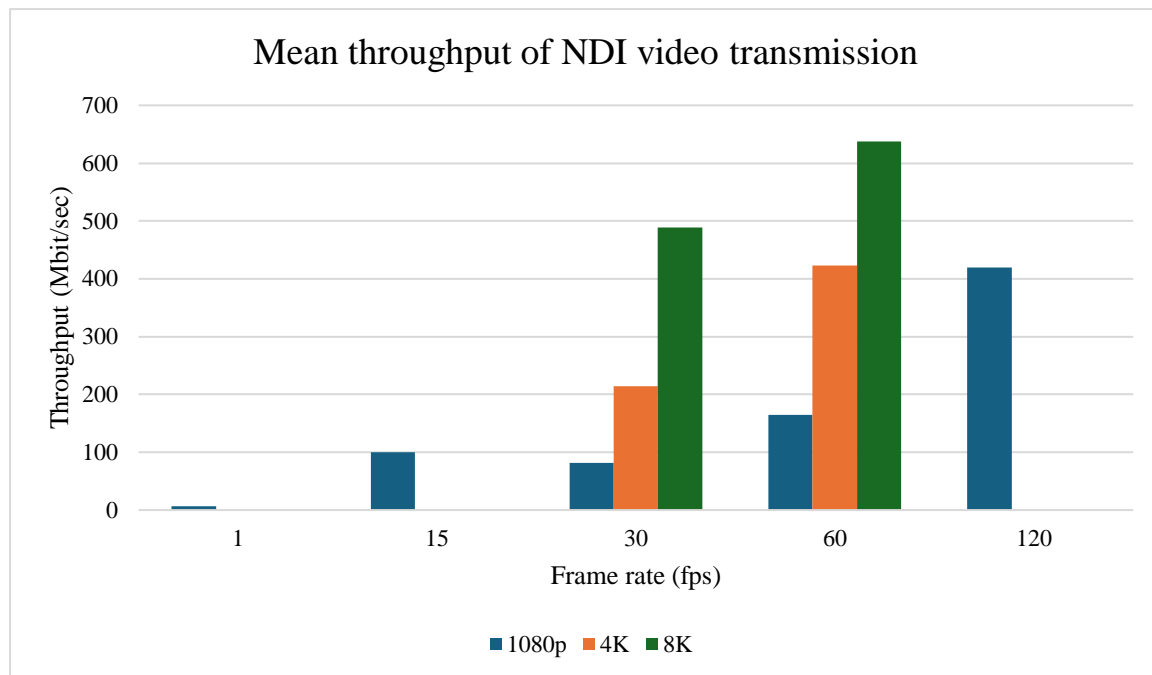


Figure 25: Mean throughput of NDI video transmission at different resolutions and frame rates, measured from Resolume Arena 7.

NDI HX Camera can operate in TCP or UDP mode, we measure both streaming 1080p video at 30 fps (Table 9).

Type	Packets	Time (s)	Mean packets/sec	Mean packet size (B)	Total bytes	Mean Bytes/sec	Mean bits/sec
UDP	98337	60.446	1626.8	586.5	57675013	954 k	7633 k
TCP	47846	60.69	788.4	1104.5	52831977	870 k	6964 k

Table 9: Packet capture summary data for NDI-HX transmission from the NDI HX Camera app.

UDP is the default mode and its mean throughput is slightly lower than the TCP variant's, which is probably why it is the default. The increased overhead of TCP communication seems to be compensated for by sending more data per packet across fewer packets, with almost half the mean packets per second but double the packet size. In both cases though, the “high efficiency” name appears to be accurate, as the mean bytes per second is about a tenth of the equivalent standard NDI feed, comparable to the throughput of 1-4 channels of Dante audio. So, in bandwidth constrained applications – such as portable cameras connected wirelessly to the network – selecting NDI-HX instead of NDI would be a sensible decision.

9.3 Measurement conclusion

From the measurements in this section, we conclude that different implementations of the same software protocol across different hardware devices use the network in a very similar – or even exactly the same – way. We can therefore use the measurements of a protocol’s behaviour on one device to represent the protocol’s behaviour across other devices, making it easier to support a broader range of hardware in our tool. Protocol throughput increases as more channels of data are added which makes intuitive sense, but for protocols like Dante, the throughput increases per four additional channels of data since each flow contains either four channels of data or fewer than four channels plus padding. These findings will inform the database schema designed in the next section.

10 Design

The main design tool application follows the three-tier approach[2] with the presentation layer running in the user's web browser, the application layer running on a web server and managing data access, and the data layer being the database server (Figure 26). Due to the selection of technologies (explained further in section 11), the application and presentation layers are tightly-coupled, with the line occasionally blurred between them in the name of improved application performance. Because of this coupling, after looking over some guiding design principles, we explain the design of the data layer followed by the application and presentation layer together in sections by system functionality.

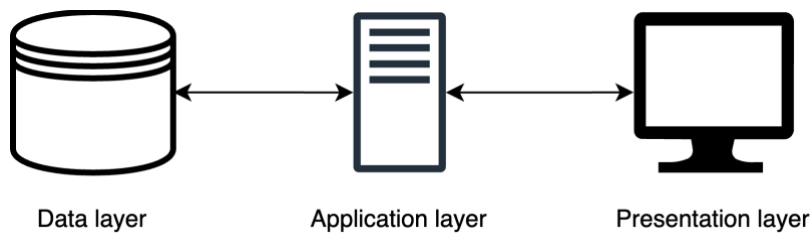


Figure 26: Three-tier software architecture.

10.1 Nielsen's usability heuristics

Jakob Nielsen's 10 usability heuristics (Figure 27) [15] were refined in 1994 and have not changed since, yet still are a reliable set of principles for user interface and experience design. There is a considerable amount of work based on these heuristics, either developing them further or applying them to various problems, with Nielsen's original papers on the topic gaining thousands of citations and more all the time. For this reason, these heuristics were selected as the main set of guidelines to follow when designing the application.

These heuristics augment the requirement specification with design guidance and expertise and are used to reason how best to fill gaps where the project timeline meant further stakeholder involvement was impractical.

1 Visibility of System Status

Designs should *keep users informed* about what is going on, through appropriate, timely feedback.



Interactive mall maps have to show people where they currently are, to help them understand where to go next.

2 Match between System and the Real World

The design should *speak the users' language*. Use words, phrases, and concepts *familiar to the user*, rather than internal jargon.



Users can quickly understand which stovetop control maps to each heating element.

5 Error Prevention

Good error messages are important, but the best designs *carefully prevent problems* from occurring in the first place.



Guard rails on curvy mountain roads prevent drivers from falling off cliffs.

8 Aesthetic and Minimalist Design

Interfaces should not contain information which is irrelevant. Every extra unit of information in an interface *competes* with the relevant units of information.



A minimalist three-legged stool is still a place to sit.

Nielsen Norman Group

Jakob's Ten Usability Heuristics

3 User Control and Freedom

Users often perform actions by mistake. They *need a clearly marked "emergency exit"* to leave the unwanted action.



Just like physical spaces, digital spaces need quick "emergency" exits too.

4 Consistency and Standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. *Follow platform conventions.*



Check-in counters are usually located at the front of hotels, which meets expectations.

6 Recognition Rather Than Recall

Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.



People are likely to correctly answer "Is Lisbon the capital of Portugal?".

7 Flexibility and Efficiency of Use

Shortcuts — hidden from novice users — may speed up the interaction for the expert user.



Regular routes are listed on maps, but locals with more knowledge of the area can take shortcuts.

9 Recognize, Diagnose, and Recover from Errors

Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.



Wrong-way signs on the road remind drivers that they are heading in the wrong direction.

10 Help and Documentation

It's best if the design *doesn't need any additional explanation*. However, it may be necessary to provide documentation to help users complete their tasks.



Information kiosks at airports are easily recognizable and solve customers' problems in context and immediately.

NN/g

www.nngroup.com/articles/ten-usability-heuristics/

Figure 27: Nielsen's usability heuristics.

These heuristics are woven through the design and mentioned throughout the remainder of the section, but in the interests of being explicit, we list some specific examples of each heuristic at work in the application.

1. **Visibility of system status:** The currently-active navigation item (within the application or tool pane) is clearly highlighted through to help the user orient themselves. (Figure 28)

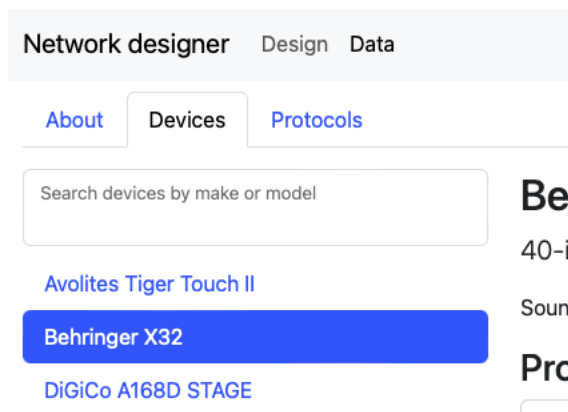


Figure 28: Partial screenshot showing the navigation hierarchy of the Data section. The data page is highlighted, as is the Devices tab and the Behringer X32 section.

2. **Match between system and the real world:** The system talks mostly about “devices” rather than “nodes” or another more technical term, matching the way people would speak in the industry. The main UI is a map where the user can lay out devices in a way which matches their mental model of their venue or setup (Figure 29).

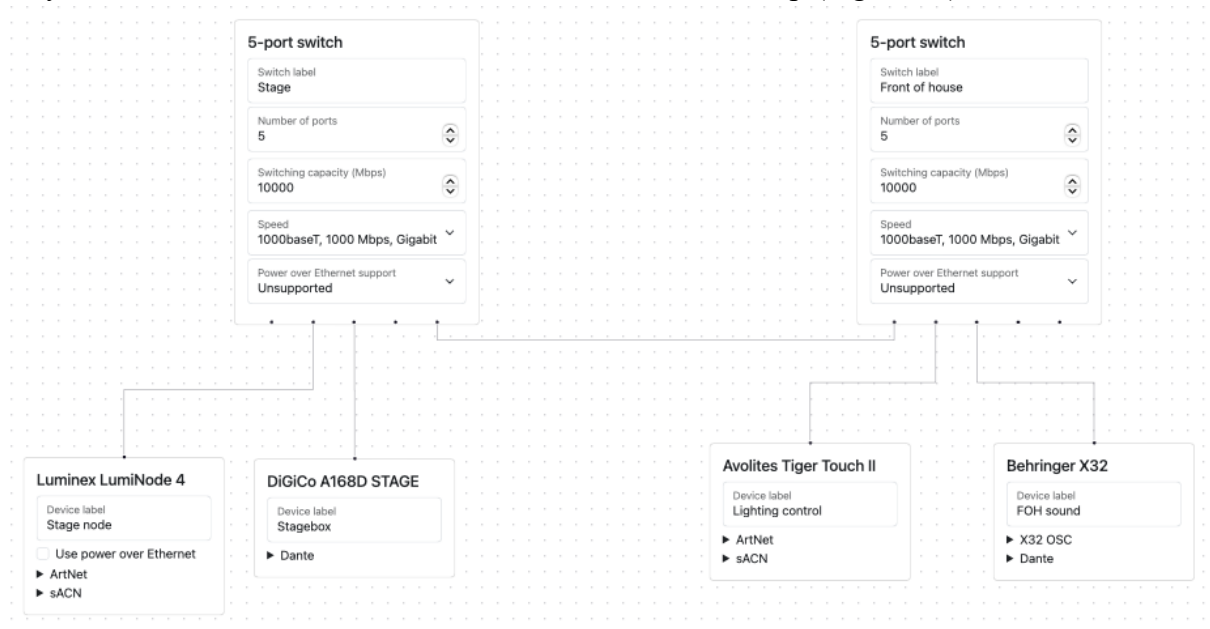


Figure 29: Partial screenshot showing a network map, where devices which will be in different areas of the venue are grouped together and separated from each other.

3. **User control and freedom:** The design wizard includes a “back” option which takes the user back a step and lets them fix any mistake they made, and a “cancel” option to abort the entire process (Figure 30).

Figure 30: Screenshot of the design wizard. Users have clear navigation options to go back or completely cancel the operation.

4. **Consistency and standards:** The componentised frontend architecture encourages re-use of design elements across the system so everything looks familiar. When finding information about a device, the same information pane is used in the data explore tab, the info modal in the design tool, and the info modal in the design wizard. The device filtering options are the same in the design tool and wizard.
5. **Error prevention:** Devices and protocols should be added to the database with help text and sensible defaults so users can enter data correctly first time, or not have to enter the data at all. The wizard prevents errors by arranging much of the core infrastructure for you.
6. **Recognition rather than recall:** When selecting devices to include in the design during the design wizard, the current system status including what the user said their requirements were and how well the currently-selected devices meet those requirements is constantly displayed on screen (Figure 31).

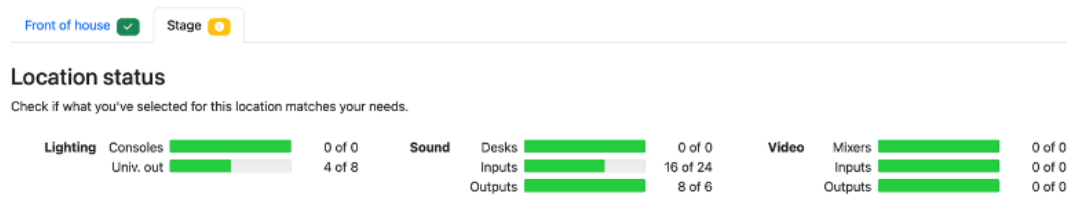


Figure 31: Partial screenshot of the device selection step of the design wizard. Each location's tab includes an indicator to show whether its requirements are met, and within each tab there are clear meters to show progress towards each requirement.

7. **Flexibility and efficiency of use:** Using standard operating system option keys on the keyboard, users can select multiple devices, switches, and connections on the map to move them at once.
8. **Aesthetic and minimalist design:** The design wizard shows each step on the screen one at a time, avoiding overloading the user with the whole procedure at once. The

device picker list shows the device make and model and its number of inputs and outputs, then hides all the rest of the details in a modal dialogue to ensure the list remains easy to skim through.

9. **Recognise, diagnose, and recover from errors:** Design validation errors are written in plain English, include advice for fixing the problem, and provide links so the user can quickly find the problematic option in the map (Figure 32). Error boundaries are included for each section of the application to limit the ‘blast radius’ of a faulty component.

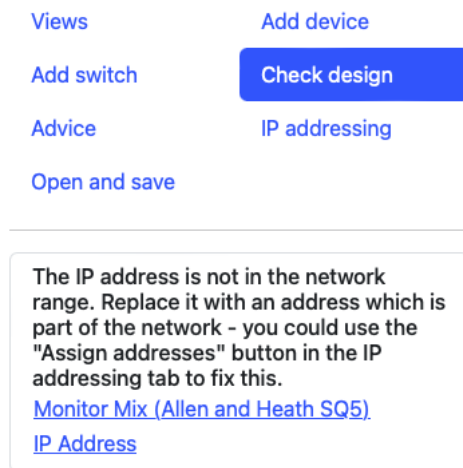


Figure 32: Partial screenshot of the "Check design" tool pane indicating a problem. The problem includes links to solve the issue, and descriptive text which explains what the user needs to do.

10. **Help and documentation:** All configuration options are accompanied by descriptive text explaining what the option should do and how the user should decide what to enter. The application includes a tour to help the user learn both how to use the design tool and set up a basic entertainment computer network.

10.2 Health and safety consequences

ISO 9241-210 requires potential adverse consequences for health and safety to be identified. There are no particular adverse consequences for the use of this tool other than those inherent with the usage of a computer system.

10.3 Database

Based on the findings of the measurement activities, a database schema was created. This schema is included in section 21.

For each protocol, we store:

- **A name and description.**
- What **functionality** the protocol offers (lighting control, sound, video, and/or device control).
- **Quality-of-service rules** which the protocol recommends.

- **Configuration options** which the protocol makes available. These are available in a variety of data types (text, number, select) and can be used for protocol configuration data (e.g. deciding what sample rate an audio device should work at) or to store metadata used for configuration validation (e.g. how many channels of data a device is transmitting). Depending on the option data type, various validation options are available, including minimum and maximum value, maximum length, regular expression pattern to match, whether the option is required or not, whether the option must be unique across the entire design or not, and the default value for the field. Where possible, every configuration option should have a sensible default value to guide users towards a functional setup with minimal effort.
- **A validation function.** This is a JavaScript function which can do any computation required for protocol-specific validation which cannot be covered by the basic input validation available on the configuration options. The function is run for every device which uses the protocol in a design and has access to that device's configuration values and the configuration of the nearest switch. For example, for Art-Net, this function compares the number of universes being transmitted and the speed of the nearest switch to ensure it meets the Art-Net specification.
- **Advice.** It is not practical for the design tool to validate *everything* which a protocol may recommend to make a network most efficient – some items are too complex and niche, or refer to external factors which cannot be sensibly represented in the system. In this case, the tool can instead offer advice to the user. Each piece of advice is a text field of information with a JavaScript function used to decide whether to give the advice or not. This means advice can be given as soon as a protocol is used (e.g. “disable energy-efficient Ethernet on all switches in the network”), or only appear based on some condition (e.g. if the device is configured to send more than 16 channels, “ensure this device is set to transmit-only mode to raise the limit from the default 16 channels”).
- **Channel calculation.** This is some JavaScript code which is used to calculate how many channels of data a given device is sending and receiving for a protocol. This way, each protocol can use any data it needs to find this value, including the device's configuration options.
- **Flows.** Each protocol can describe multiple flows which it uses (for example in Dante, its clocking data and its audio data). Within each flow, the system stores the flow's DSCP value, the number of channels of data a flow of some type can transport, and a set of measurements of that flow's throughput (including the throughput and where that measurement came from, e.g. device measurement or a specification document). All this data is available for exploration in the application, but more importantly, is used to calculate the link utilisation for each switch port, as described in section 11.3 below.

Functional requirement 6 (in section 8.3.1) refers to the system offering plugin hooks. On considering this requirement further, the three key areas a plugin could help with are:

1. Offering additional validation logic for devices and protocols.

2. Adding custom user interface elements for configuring devices and protocols.
3. Integrating with third-party applications and systems.

It was decided that the work required to support these three areas would not be proportional to the benefit users would get from it, and so the requirement adjusted. The first area, additional validation, would instead be implemented by storing a validation function in the database with each protocol. This offers the same extendible functionality, but with a lower overhead than having to include it in a ‘plugin’ as you may traditionally think of one (and so probably be less of a burden for system administrators too). The other two areas are moved to future work, although the system is still designed with them in mind so this future work is possible without significant application re-architecture.

While storing JavaScript functions in the database offers a very high level of flexibility and customisability, it also comes with security risks and the risk of introducing bugs in production which may be difficult to track down. To support the latter, usage of the database-stored functions is wrapped in appropriate error handling to minimise the reach of the problem. In support of both points, the ability to edit the code in the database must be restricted to trusted administrators only. A future development could be replacing the JavaScript with a more restricted environment – perhaps a visual, blocks-based language which would have the added benefit of making the creation of these functions more approachable to administrators who are less familiar with computer programming[12].

For each device, the system stores:

- **Make, model, and description.**
- A device **type** from a set of pre-defined categories, and the number of inputs and outputs the device offers. This is used for filtering to find devices and helping the wizard check that a system meets requirements.
- **Power over Ethernet information**, including whether the device can only be powered by PoE or if it is optional, and the maximum wattage drawn. This is used to ensure appropriate switches are used and nothing is overloaded.
- **Advice**, in the same way as protocols, allowing for any device-specific advice to be included.
- A **relationship to any protocols** which the device always uses or may optionally use. This covers devices like A168D STAGE which always and solely communicates using Dante, X32 which may use Dante if the appropriate expansion card is installed and otherwise uses non-networked solutions, and ETC Ion Classic which supports both Art-Net and sACN and the user can select which they wish to enable.
- A record of which **protocol configuration options the device supports** (for example, some devices may not support changing the sample rate), and the ability to override the validation for any configuration option (for example, current Dante chipsets can allow a device to receive up to 512 channels of audio[40], but a stage box with 8 output sockets can only receive 8).

Switches are not stored in the database. Instead, users enter the specification of their selected switches when they are working in the design tool. This is because switch specification information is usually readily-available – often written on the switch itself – and people are likely to use switches they already own from a wide range of manufacturers, so a switch database would need to be incredibly extensive to be useful.

This structure reflects the findings from the measurement section, where it was found that a protocol acts similarly across all devices which implement it. The data about a protocol is stored once, then linked to any device which implements it with the ability to override some parameters to handle any special-case devices.

The database also stores users' designs, so they can be shared with others and accessed on any device. We store a UUID to find the design, a user-defined file name, and a JSON object containing all data about the design. The format of the JSON object is defined by React Flow, described in section 11.3. Some schools of thought do not recommend storing JSON in a database, since it is inefficient to query the database based on values within that JSON object. This is not relevant in this case as we treat the JSON object like a file or blob – we only ever recall it in its entirety and it is found by its UUID only, never the contents.

IP addresses are included as protocol-level configuration option, rather than on a device-by-device basis. While this initially seems unusual, it increases the flexibility of how data is represented in the system. For example:

- A168D STAGE supports Dante only. The device's IP address is recorded as part of the Dante configuration options.
- X32 has two network ports (one for remote control and one for Dante), and each port needs its own IP address. The device is recorded as supporting both protocols and supporting the IP address configuration option of both, which allows a separate address to be stored under each protocol's configuration options.
- ETC Ion Classic uses one IP address for remote control, Art-Net, and sACN. The device is recorded as supporting all three protocols, but only supporting the IP address option of its remote control protocol. This means the device IP address is stored under the remote control configuration, and the option is not available under Art-Net or sACN.

10.4 Data management

Initially, it was planned that users would be able to submit information to the database directly from the user interface. This would allow maximum flexibility, since if a technician had access to a device the system didn't know about, they would simply add it. This feature was scrapped for three reasons:

- Getting the data is hard. After carrying out the measurement phase of this project, it was decided that a user unfamiliar with networks would struggle to do the task independently, and there is limited scope for this software-only development to help:

the tool could theoretically carry out packet capture and analysis, but the user would still need to be instructed on setting up a managed switch with port mirroring, create appropriate testing scenarios which would vary depending on the devices, and work out how many channels go into each flow.

- A publicly-editable resource like this would require moderation, which the project in its current form does not have capacity to do.
- Creating a data input user interface which meets the requirements of being easy to use, supporting a moderation flow, and fully integrated into the main application was time-consuming and made little progress towards the main project objectives.

Tesler's Law[11] states that every application has an inherent amount of complexity that cannot be removed or hidden. In this case, end-user device data management is just too inherently complex and would probably scare users not familiar with networking, like persona Fiona (Figure 16). So, we take this complexity out of the user interface and place it onto the tool's administration team. Looking at Nielsen's heuristic of consistency and standards, technicians are unlikely to be surprised by this design decision. In lighting, console manufacturers maintain a database of lighting fixtures which they update periodically so their consoles know how to control different devices. If you are using a lighting fixture which your console doesn't support, you contact the manufacturer and ask them to create a profile[41–43]. Doing the same for devices in the network design tool is therefore consistent with similar processes elsewhere in the industry.

The user interface still includes a way for users to browse the database, both in the context of making a design and generally. The UI is shared between the two browsing interfaces, keeping a familiar and consistent experience across the application (Nielsen 4).

10.5 Homepage

The application homepage (Figure 33) acts as a landing page and has information targeted both to new and returning users. The most prominent options are for creating something new, starting from a blank canvas, using the design wizard, or starting an onboarding tour of the application. Secondly, a list of all recently opened files is shown, followed by links to explore device and protocol data.

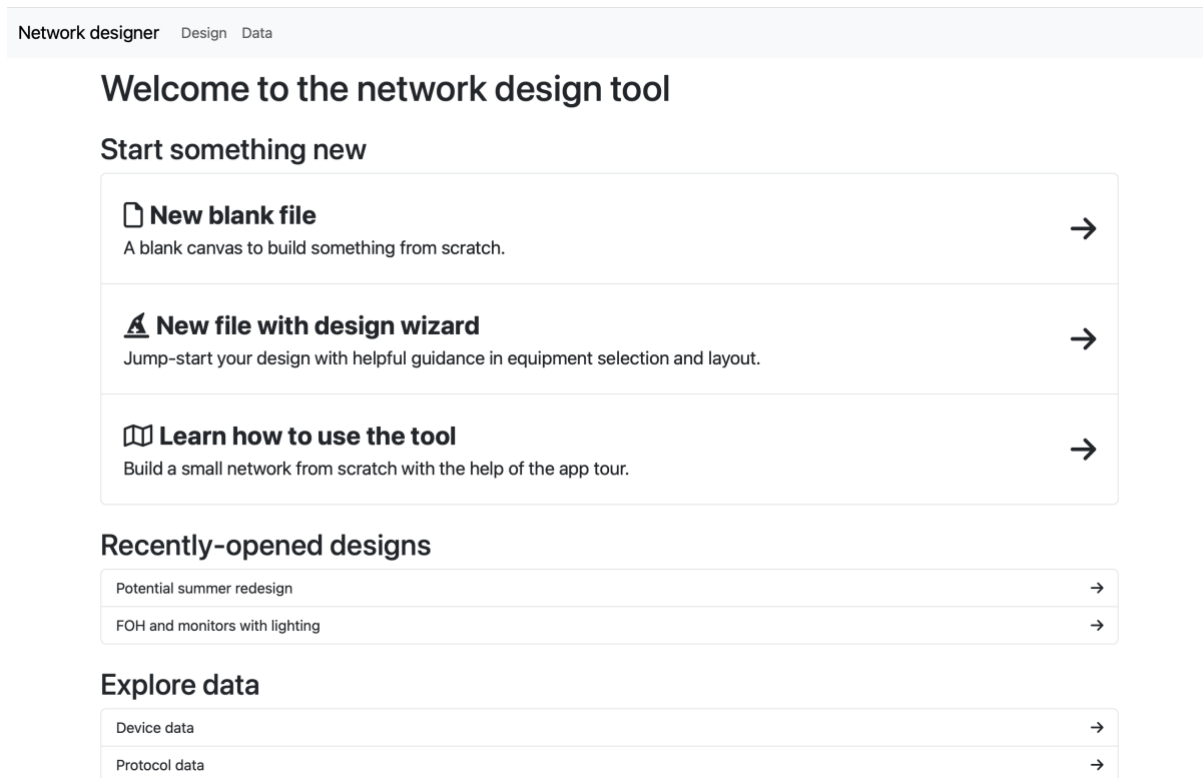


Figure 33: Application homepage, with large buttons to start new things, open recent files, or start browsing the database.

This prominent navigation makes it easy for users to find what they are looking to do, and the inclusion of descriptive text for the different ways users can start something new helps users with varying experience find the best way to begin.

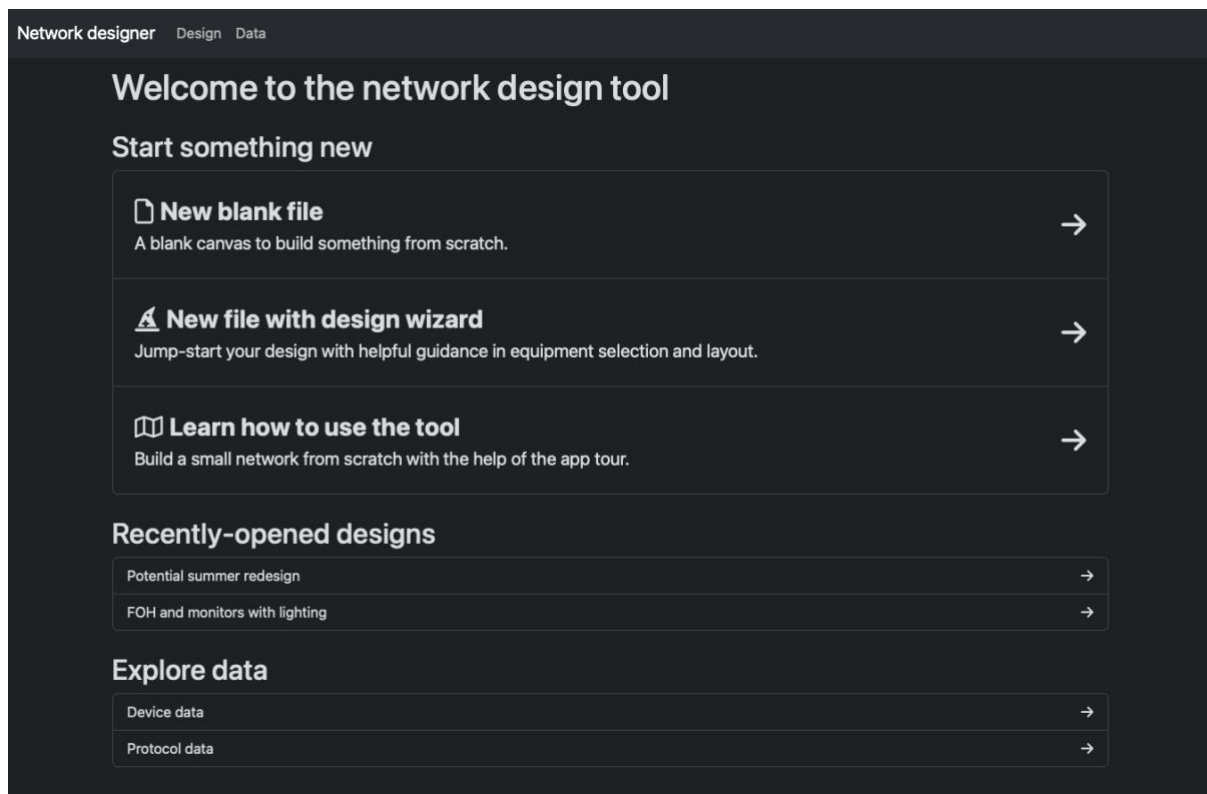


Figure 34: Application homepage in dark mode.

The whole application is available in dark mode (Figure 34) and switches automatically based on the user's operating system settings. While 'dark mode' may be seen as a gimmick sometimes, it is often a popular feature among users, and in this case has a functional benefit: this software will likely be operated in dark theatres and dimly-lit backstage areas, so dark mode will make it easier to read the screen and prevent someone's display lighting up the room during a show or rehearsal.

10.6 Network design interface

The requirements survey discovered people prefer a hands-on approach, so the main design app is based around a map of the network, showing all devices, their settings, and connections between each other (Figure 35). Using a map for design is supported by Nielsen's second heuristic, matching the system to the real world, since technicians can position devices anywhere on the canvas and mirror how the system will be physically deployed. This map appears in a primary workspace area on the left, with a tool pane on the right.

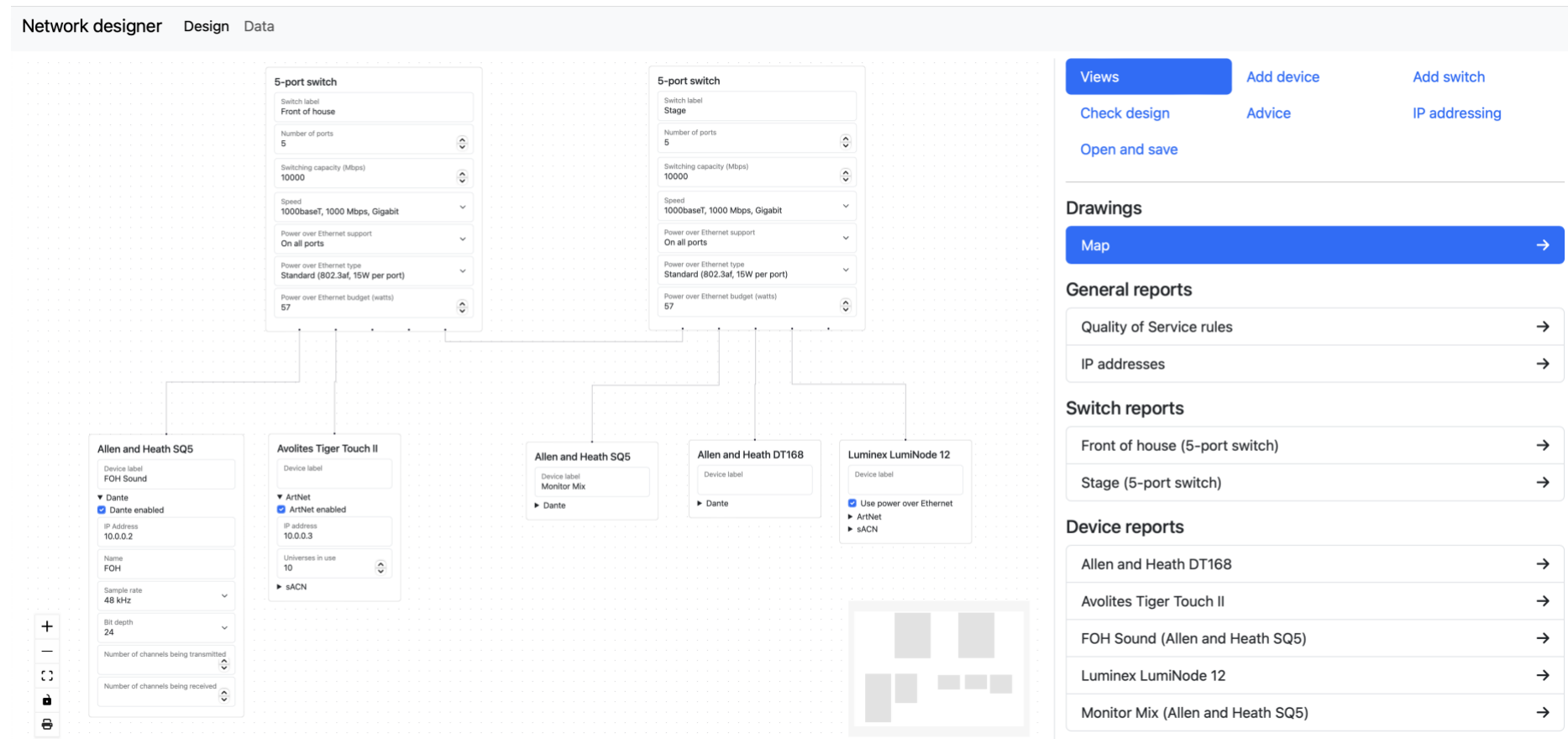


Figure 35: A small network design in the tool. The map on the left shows the devices and switches in the system, their settings, and connections. The tool pane on the right is currently showing the different ways data can be presented instead of the map.

A set of seven user tasks and relevant sub-tasks were identified, and these map to the tabs of the tool pane:

1. Views: See the design in various formats.
 - a. Explore the default map layout.
 - b. Modify device and switch settings.
 - c. Move devices around the map.
 - d. Connect devices and switches together.
 - e. Get a report of the necessary quality of service rules.
 - f. Get a report of IP addresses in use.
 - g. Get a report for each device and switch showing all the relevant configuration and specification information.
 - h. Print any report for distribution to a crew.
2. Add device: Find and add hardware to the design.
 - a. Browse information about specific devices and their features.
 - b. Filter devices the tool knows about by category, inputs, outputs, to find devices which meet the technician's needs.
 - c. Add devices to the design.
3. Add switch: Specify and add switches to the design.
 - a. Provide the details of a switch the user owns and add it to the system. The system asks the user to enter the specification of a switch they have because this data tends to be readily available – often written on the switch itself. This way, the system supports any switch the user owns instead of a limited set the system knows about.
4. Check design: Perform automated validation of the design.
 - a. Run checks: find devices and switches which are not connected to the network, validate configuration settings, run custom protocol validation functions, check throughput of each switch port, check PoE requirements are met and switches are not over budget.
 - b. Learn how to fix any problems which are found.
 - c. Quickly identify where in the system the problem is (link to the exact option or node with a problem).
5. Advice: Get advice on how best to deploy the system, helping to reduce the impact of potential problems which the system can't check automatically.
6. IP addressing: Configure addressing settings for the system.
 - a. Set the network address range, or reset to a sensible default if the user doesn't know how to pick their own. Any valid address range is permitted, including publicly-routable ones despite entertainment networks not usually being connected to the Internet. This is because some protocols ignore the standard link-local and private network ranges and pick their own, for example Art-Net[3] describes a procedure to generate an address in the 2.0.0.0/8 subnet based on device MAC address.
 - b. Auto-assign addresses to all devices in the network.
7. Open and save: File management.

- a. Save designs to the cloud.
- b. Open other designs.
- c. Share designs with other people.

Following Nielsen's tenth heuristic, all options in tool panes have support information inline (Figure 36). Depending on the option, this information can range from a simple line explaining what the options mean, telling the user where they would find the information they need, or recommending a default option to choose if the user isn't sure what to do.

The "Check design" and "Advice" panes follow Nielsen's ninth heuristic, helping users diagnose and recover from errors (Figure 37). All error messages are written using the simplest language possible, and where possible include a hint as to how the problem can be fixed. There is also a link which takes the user directly to the device, switch, or configuration option which needs to be changed, minimising the time they spend hunting around the map. Some types of errors are also indicated directly on the input element, with a red outline supported by a warning icon.

The set of reports was selected so that each report generated corresponds to one task in the real world. This makes it easy to communicate a design to a crew, since each setup task corresponds to a single sheet of paper – this feature is particularly desired by the persona Fiona (Figure 16). On each report, a Print button is included to make it obvious that this action is supported. Reports have print styles which remove all interactive features, app navigation, etc, leaving printed documents uncluttered and easy to read.

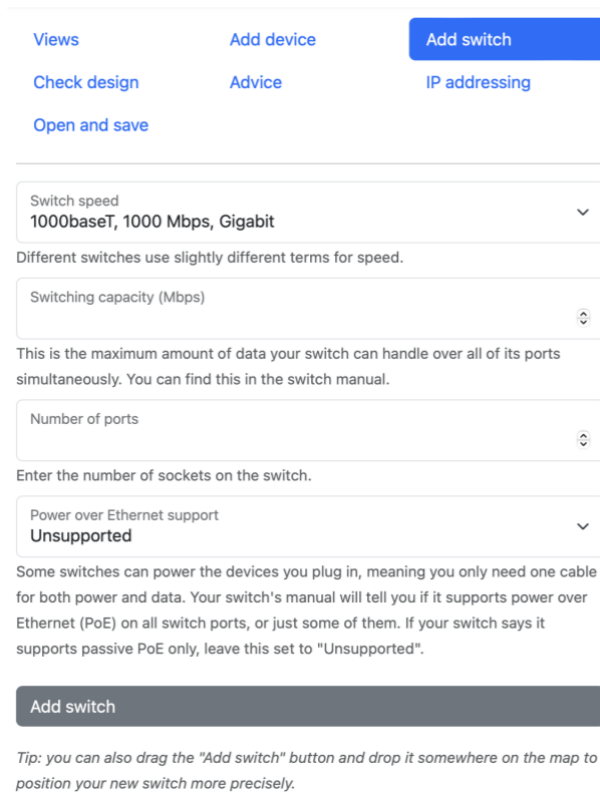


Figure 36: Partial screenshot of the "Add switch" tool pane. There is descriptive text next to every option, supporting users who might not be familiar with switch terminology.

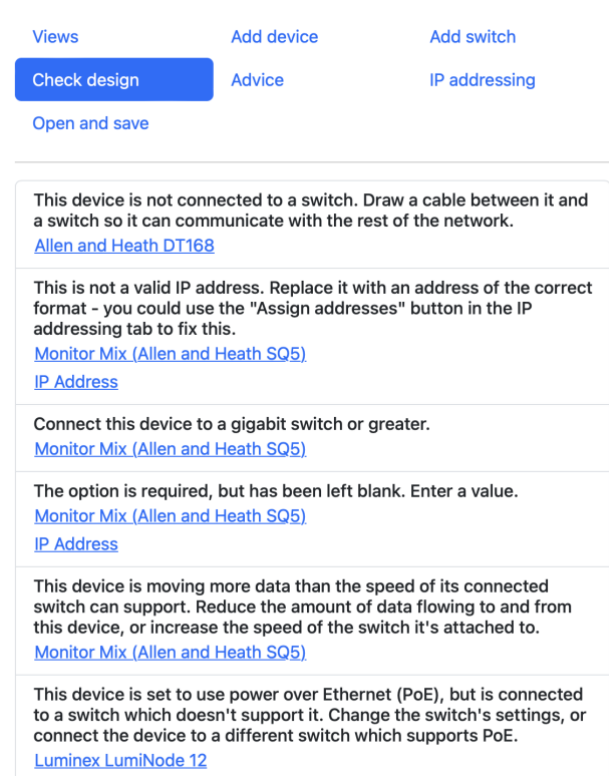


Figure 37: Partial screenshot of the "Check design" tool pane. The pane indicates any issues with the design which the system can detect, including links to the problem and help to fix it.

10.7 User onboarding

Technicians are introduced to the application by an optional tour. It is optional because users who have experience in other design software or networks (like Lachlan, Figure 15) may prefer to just get started and work it out independently, and forcing them through a tour would not be a good user experience (nor would it likely be necessary).

The tour is written to be an introduction to both the tool and the principles of building a small entertainment network. The user is guided through all the tool panes in the application and given instructions on how to use them to build a valid converged network sharing devices from multiple departments.

There is a balance to be found here. A new application user with great networking knowledge (like the persona Lachlan) may find it annoying that the tour wants them to build a specific network, but a user with low networking knowledge (like Evan, Figure 18) would probably appreciate the guiding hand. To work towards this balance, the tour suggests what a user may want to include in their network, but does not enforce it. Users can stray from the instructions whenever they want and to any degree they want, but can re-join the tour at any time without losing their work by clicking a pulsing icon which appears on the interface element which the tour covers next.

Each tour step is presented in a popup box and the relevant user interface element is spotlighted, with the rest of the page shaded. Some steps proceed automatically when the user carries out the action (Figure 38), some are progressed by the user when they are ready by pressing “Next” (Figure 39). The content of the tour is in section 19.

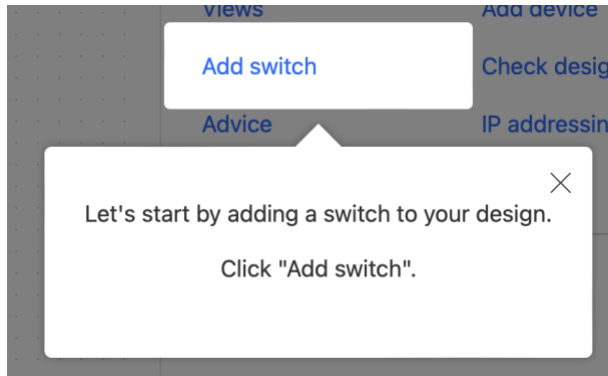


Figure 38: A tour step highlighting a specific button which progresses when that button is pressed.

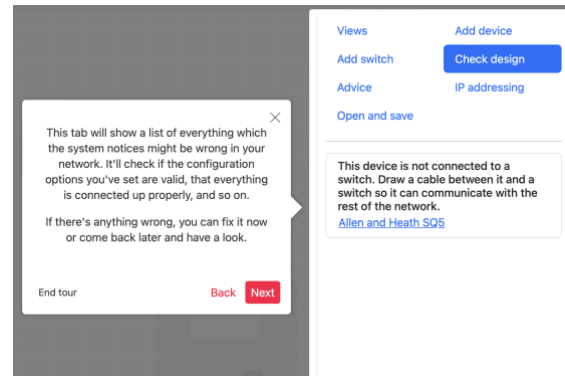


Figure 39: A tour step highlighting the tool pane which progresses when the user selects "Next".

At the end of the tour, users are invited to continue working on the system they started (so users who deviated from the prescribed tour can keep working without data loss), or start a new project from blank or using the design wizard. This caters to the needs of any user type who has followed through the tour, and nudges people towards continued application exploration in the wizard.

10.8 Design wizard

The design wizard offers more ‘handholding’ for users initially setting up their network. It is a guided process breaking the large task of “design and specify a system” into smaller chunks:

1. “How many locations do you need?”
This question starts the user off by thinking about where they need inputs and outputs into their system, and thus where they will eventually need devices and switches to be located.
2. “Give each location a name.”
This name helps users remember which location is which, and will eventually be used as switch labels.
3. “What do you need in each location?”
The technician now specifies how many consoles, inputs, and outputs they need in each location. This is a natural question to ask because all of this information will be available from other plans which will have been prepared for a show: for example, the necessary number of lighting universes will be specified on lighting patch plans, and the number of sound inputs will be included on band specifications.
4. “Which protocols do you want to use?”
This step is the first time the wizard directly addresses a networking concept. The user needs to select their preferred protocol for lighting, sound, and video data transfer.

This question is at a high enough level though that a user should be able to pick a name they are familiar, because they will probably know what at least some of the equipment they already own can support.

5. “Pick your kit.” (Figure 40)

A tab is presented for each location. In each location’s tab, a set of meters at the top indicate how much of the IO requirement the user specified has been satisfied by the equipment they have selected. The user adds devices to the location using the same user interface as the main design wizard (supporting Nielsen 4), but with a new filtering option to only show devices which support the protocols the user specified they want to use. Once a location’s requirements are met, the meters show as full and the tab label is ticked.

6. The wizard builds the network.

The wizard starts by creating an appropriately sized switch for each location, meeting the requirements for power over Ethernet and so on. It specifies a switch with enough ports for the location, rounded up to the common sizes for switches (5-port, 8-port, 24-port, or 48-port). It then adds all the devices needed in each location and connects them to the location’s switch, enabling required protocols if they are optional and assigning an IP address to everything. Finally, it reports what it has done to the user, and takes them to the “Check design” tool pane which will indicate any fields which the wizard wasn’t able to complete independently – such as filling in any mandatory protocol configuration options which don’t have a default value.

Network designer Design Data

Design wizard

Pick your kit.

Add equipment to each location to meet your requirements.

Front of house ☒ Stage ☐

Location status

Check if what you've selected for this location matches your needs.

Category	Item	Progress	Count
Lighting	Consoles	<div></div>	0 of 0
	Univ. out	<div></div>	4 of 8
Sound	Desks	<div></div>	0 of 0
	Inputs	<div></div>	16 of 24
	Outputs	<div></div>	8 of 6
Video	Mixers	<div></div>	0 of 0
	Inputs	<div></div>	0 of 0
	Outputs	<div></div>	0 of 0

Selected devices

This is the equipment you've added to this location.

Device	Inputs	Outputs
Luminex LumiNode 4	0	4
Allen and Heath DT168	16	8

Add devices

Press items from this list to add them to this location. You can have up to 47 devices in each location.

Search devices by make or model

Show device types: All devices

Min inputs: 8 Max inputs: 24

Min outputs: 8 Max outputs: 6

☒ Only show devices which support the protocols I chose

Device	Inputs	Outputs
Allen and Heath DT168	16	8
Allen and Heath SQ5	22	12
Audinate 2-ch output AVIO	0	2
Avolites Tiger Touch II	0	4
Behringer X32	38	22

Next Back Cancel

Figure 40: In the design wizard, users pick equipment to meet their requirements. There are two locations in the tabs at the top: front of house is satisfied (indicated by the tick), and stage is still missing parts (indicated by the yellow warning). Exactly what is missing in a given location is indicated by the meters.

Each step of the wizard is presented on its own screen with lots of descriptive text and help, making each stage as manageable and understandable as possible. When selecting equipment for a location, the requirements the user listed and how well the equipment they have specified meets those requirements stays on screen at all times, so users remember their target and what they're working towards.

All stages of the wizard have a back button which will show the contents of the previous step, so the user doesn't lose progress if they need to go back. There is also a Cancel option to stop the wizard and return home. Although the user is asked to select which protocols they want to use and this is used to filter the devices shown to them, they can disable the filter and add any devices they like during the wizard, or ignore the requirements they specified and build whatever they like. These options fulfil Nielsen 3, user freedom and control.

10.9 File management

Design files are stored in a cloud database, referenced by a UUID. This method was chosen to simplify sharing: a link to the page is all anyone needs to start collaborating on designs.

The lack of account system both reduces the barrier to entry for new users and invited collaborators and makes it easy to view designs on shared PCs which are often installed in venues – this would be particularly valuable for a persona like Evan, who is not regularly in the theatre but may need to refer to design documentation in the course of his work.

Simultaneously though, the fact any user with the link can edit the design is a downside of this approach. However, UUIDs are difficult to guess, so it is unlikely that someone would guess the link to a design, and users who have had the link shared with them are unlikely to make malicious edits, since they can probably sabotage an event anyway in much easier and more effective ways.

11 Implementation

One of the non-functional requirements of the system is that it is web-based. This requirement is borne out of the need for a cross-platform and collaborative solution, which can be run on any machine so files can be shared with colleagues easily. The web-based requirement and the rich ecosystem of libraries available is what informed the decision to use JavaScript for the implementation, with TypeScript used to improve the type-safety of the system making the system more resilient than if it were in JavaScript alone.

11.1 Libraries and frameworks

The first library which was chosen was React Flow[44]. React Flow is a React component which allows for the creation of a node-based UI⁴. Since the network map is a key user interface element, finding a library which offered a reliable and customisable way to build the map was the priority when picking the technologies to build the tool atop. React Flow is well-supported, used by a variety of major software development companies, and has a good library of documentation and examples, indicating it supported all the features needed and would be reliable.

React Flow recommends the use of Zustand[45] for state management, so Zustand was used in this project. Zustand makes it easy for any component in the application to access the current state of the flow (and thus of the user's network design), making it easier to componentise behaviour and reduce the amount of manual data passing which needs to happen between parts of the app.

Since React Flow was selected, the wider application framework needed to be based around React[46]. Remix[47] is a React framework which adds a more prescriptive application structure over the React foundations. Remix was selected because its nested routes and strong data management features work well with the application design. It sits across the application and presentation layers of the software architecture, as it includes both a client and server component (Figure 41), and in the name of improved performance and optimisation it can spread its computation across both layers. For this reason, again, we will review the implementation of both of these layers together. Each tool in the design tool pane is its own route, and therefore manages all its own state and data loading independently, while still appearing as one coherent application to the user. React components are used to separate user interface elements and their associated logic into sensible composable units, and these components could provide a foundation for user interface plugins in future work.

⁴ “Node-based UI” refers to a user interface based around a graph containing nodes and edges, not the JavaScript runtime Node.js (although Node.js is used in the project).

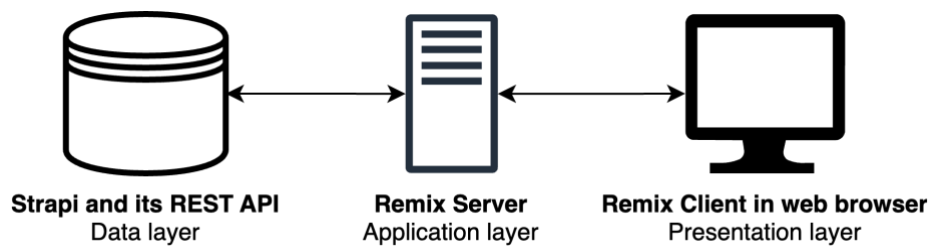


Figure 41: Three-tier software architecture annotated with the key technology used at each layer.

For user interface elements, Bootstrap[14] was used. Bootstrap is a well-used and well-tested framework, so by using its components and following its guidance, fulfilling Nielsen's fourth heuristic becomes easier and we make good progress towards a simple, understandable, and accessible user interface.

The database is managed by Strapi[48]. Originally, Prisma ORM[49] was used, but as described in section 10.2, this was replaced with Strapi mid-way through the project. Strapi is highly-customisable so could easily handle the intricacies of our data model, but most importantly it includes user-friendly data management facilities as standard. This lifted a huge development burden off the project, allowing us to focus on features which support the core stakeholders of technicians trying to build networks, rather than administrative teams managing the application database.

11.2 Database: data layer

The first version of the database implementation used Prisma ORM. Prisma specifies a JSON-like schema language and uses this to build a SQL database, automatically managing schema updates and database upgrades as the project progresses. From this schema, it generates a client including functions for CRUD operations and type signatures for all data which the application can use to interact with the database in a type-safe way.

During this first version, the data exploration interface included functionality to create and edit devices and protocols, but as described in section 10.4 above, this added a lot of extra development work. The data model is reasonably complex, for example with how protocols may have any number of configuration options each with differing types, so the editor user interface was becoming equally complex, before even considering access management and moderating submissions. The way that the Prisma client generated type signatures was also not always entirely intuitive, leading to occasional subtle errors with understanding the type of data returned from queries. As none of this work supports the core project objectives, Prisma was removed.

Strapi was brought in to replace Prisma. Strapi is a content management system at its core, but is incredibly flexible. Like Prisma, the developer specifies a schema in a defined system (a web interface this time as opposed to a text file), and Strapi builds an appropriate SQL database, then manages updates and upgrades over the project lifecycle. Strapi has its own administration panel for managing the content of the database – removing the need for device and protocol editing to be part of the core application – and also provides a REST API for

CRUD operations on the data. The core application can therefore query Strapi using an API key which gives it read-only permission for all device and application data, and also the ability to write saved design information back to the database.

Manually, type signatures for Strapi's REST API responses have been written and included in the core application codebase, which allows for type safety to be achieved and in a more reliable and understandable way than Prisma allowed.

In terms of deployment, Strapi offers flexibility as it can be backed by SQLite, MySQL, or PostgreSQL by changing an environment variable.

The REST API offered also presents an opportunity for future work, where the device and protocol information could be exposed as a public API and support other projects and systems in the industry.

11.3 Frontend: application and presentation layers

React Flow allows you to specify any React component as a node or an edge in its diagram and stores these components in a JSON structure. This JSON is used throughout the entire application to represent the system design, so the system does not need to map data between the Flow UI and its own storage and manipulation data because they are one and the same. The flow JSON is also what is stored in the database for users to save their designs for later.

We have two custom flow nodes in use the application: one for devices and one for switches. Each node includes the user interface of the graph node itself, including the name of the device and the editing facilities for the various configuration options. In the switch node, there is an object with the metadata about the switch (port count, PoE information, etc). The device node has similar, but its data object has the same type as the device record in the database with an additional options field added (which includes the currently-set value for any available configuration options). This approach means the device data from the database is stored inside the node in the map, resulting in design files which are completely self-sufficient and can theoretically be loaded without needing to request device information from the database.

The self-contained node approach is valuable throughout the application for always having the relevant data at hand to generate reports and so on, but most clearly useful in the “check design” validation tool. Validation is implemented as a function for each validation task. Most validation functions then iterate through all the nodes and edges in the design and perform some check using the data stored within that node. Type guard functions are available so the validation function knows whether a given node is a device or a switch and can perform appropriate steps. This is probably not the most efficient implementation as one run of validation would iterate through the nodes in the design multiple times, but it does make the validation code clear and easy to extend with more checks over time.

The validation feature will:

- Find nodes not on the network – that is, devices which are not connected to switches and switches which are not connected to other switches. This step also identifies devices which were connected to a switch, but aren't any more because the switch was downsized. In this case, the user can fix the problem by connecting the device to another port, or upsizing the switch again – meaning a typo in the number of switch ports will not result in data loss.
- Validate configuration options – ensuring constraints like minimum or maximum value, uniqueness, and so on are met.
- Check specified IP addresses are both of a valid format and within the specified network range.
- Run protocol validation functions for any special validation checks. As described in section 10.2, these functions are wrapped in additional error handling in the form of try/catch blocks to minimise the impact of errors in the function. Depending on which protocol-specific function threw an error and how it went wrong, different actions are taken – a problem may be reported to the user, an error could be logged to the browser console, or the program may continue using some default data and log the error to console.
- Calculate the transmission throughput of each device by finding the number of data channels in use for each protocol and using the protocol flow and measurement data to calculate the expected throughput. This value checks that the switch port speed can handle the transmission expected and that the switch's overall switching capacity is not overloaded.
- Check the power over Ethernet requirements for each device, ensuring the switch can deliver enough power over a single port for the device and that the switch is not over its power budget across all ports.
- At the time of drawing connections between devices, the system checks that connections are only from device-to-device, device-to-switch, or switch-to-device, but not device-to-device. Device-to-device connections are appropriate in some contexts, but trivial to implement and restrict the number of nodes on the network to two, so it is very unlikely a user planning a network in the tool would want a device-to-device connection.

These validation steps identify a range of common misconfigurations, including some subtle ones which a new networking user may not foresee until it is too late. A future development could be to expand these features to cover even more potential problems, perhaps by more thoroughly modelling data flow across the entire network and taking into account how unicast, multicast, and broadcast traffic moves across the network in different ways. From this, as well as pointing out misconfigurations, the system could offer intelligent suggestions on configuring IGMP snooping and other, more advanced network management features.

Following Remix's philosophy of "use the platform", the application relies on standard browser technologies as far as possible to avoid re-inventing the wheel and ensure the user feels at home with a familiar user interface (related to Nielsen 4). Some examples include that many of the validation of configuration options use the HTML form validation API; the ID of

the currently-open file is included in the page URL so sharing the current page with others to send them a design ‘just works’; and the in-app “share design” button launches the browser or OS’ sharing pane. Dedicated print styles are included for all sections of the frontend (including the data explore area which doesn’t have any in-app print buttons), since printed paperwork is such a common way of sharing information on events – by including these as standard print style sheets, users get well-formatted print documents whether they use the in-app buttons or their browser’s print functionality.

Devices and switches can be added to the design by dragging and dropping them to the exact location on the map where they are desired, or a fallback “Add” button is available. The “Add” button places the new device in the centre of the current viewport so they don’t get lost somewhere else on the map, and it is easy to move the device to exactly where you want it in the design. The centre point which nodes are added has some randomness, to prevent nodes being placed directly on top of each other and the technician losing them.

Images of the implemented application are included in section 20.

12 Evaluation and critical appraisal

12.1 Meeting project requirements

To begin, we review the requirements laid out at the start of the project (section 8) and perform an initial, high-level description of how well each requirement has been met. These descriptions are then expanded throughout this section.

12.1.1 Primary objectives

These are outlined fully in section 8.1.

- **Database construction:** The database has been designed and implemented. The dataset can be added to and updated through an administrator interface (Figure 42), and accessed via a REST API which opens an opportunity for future interoperability.
- **Design tool:** The tool has been created backed by the database and can validate a range of features (tested in section 22.2).
- **Accessibility of information:** A user evaluation was carried out, and the overall sentiment from the testers was positive. These results are described in section 12.2.

The screenshot shows a web interface for editing a device entry. At the top left, there is a 'Back' link. The device name 'X32' is prominently displayed, with 'API ID: device' below it. A 'Save' button is in the top right. The main form is divided into several sections:

- Make* and Model*:** Two input fields containing 'Behringer' and 'X32' respectively.
- Description:** A text area containing '40-input digital sound mixer'.
- Functionality*:** A section with a 'Device type*' dropdown set to 'Sound desk', and two numeric input fields for 'Inputs' (38) and 'Outputs' (22). Below these are 'min. 0' labels.
- Power over Ethernet requirement:** A section with a '+' icon and the text 'No entry yet. Click on the button below to add one.'
- Advice (1):** A section with a dropdown set to '1' and an 'Add an entry' button.
- Required protocols (1):** A section with an 'Add relation' dropdown and a list containing 'X32 OSC'.
- Optional protocols (1):** A section with an 'Add relation' dropdown and a list containing 'Dante'.
- Supported protocol configuration options (11):** A list of two UUIDs: '0b171ed6-6bc5-4ddd-a297-91e121216085' and 'b11e10a2-3ab1-4265-8cd9-ea57b65935ea'.

On the right side, there is an 'INFORMATION' panel showing 'Created' (last month) and 'By' (Andrew Barron), and 'Last update' (22 days ago) and 'By' (Andrew Barron). Below this are three buttons: 'Edit the model', 'Configure the view', and 'Delete this entry'.

Figure 42: Editing a device in the database administrator interface.

12.1.2 Secondary objectives

These are described fully in section 8.1.

- Extendibility:** The dataset of device and protocol information can be added to via a web interface, although this ability was restricted to system administrators due to complexities around collecting data. A variety of future validation conditions and requirements can be described inside the database without needing to update the application. The database is accessible via a REST API which opens more future development opportunities, and the main codebase is componentised so can be updated easily in future. A plugin architecture was not explored in this version of the

tool, instead the ability to store validation functions etc in the database was implemented.

- **Documentation generation:** The tool generates reports of device and switch configuration, quality of service rules, IP address usage, and can print its maps (tested in section 22.1).
- **Design wizard:** Users specify where they need IO in their system and how much, then are supported to choose devices which fit their needs. The devices they choose are configured into a network by the system, and the user is pointed towards any gaps they need to fill in themselves.

12.1.3 Functional requirements

These are described fully in section 8.3.1.

1. **Fully met:** The data import process has been implemented and the database schema is expected to support a wide range of devices and protocols.
2. **Fully met:** Users can select devices from a tool pane and drag them into their design.
3. **Fully met:** Users can draw connections between devices and switches on the map.
4. **Partially met:** The system can automatically assign valid IP addresses to devices in the system and allows the user to customise them. Protocol-specific identifiers may need to be completed manually by the user if they cannot be pre-populated by a consistent value, but these are usually not required.
5. **Fully met:** The system validates the user's design and reports any problems, along with advice to fix them. For other things which cannot be automatically detected, the system offers a list of advice. Future development is still possible in this area, to check for more potential misconfigurations.
6. **Not met:** This requirement was adjusted during the design stage and thus not implemented as originally described. Plugin-like functionality in the form of extendible validation rules has been implemented through storing validation functions in the database, and other plugin features has been moved to future work which focusses more on the system administrator stakeholder. The application is written in a modular enough way that this future development should be possible.
7. **Fully met:** Printable documentation is generated by the application.
8. **Fully met:** The database of devices and protocols can be browsed from within the application.

12.1.4 Non-functional requirements

These are described fully in section 8.3.2.

1. **Fully met:** The system is a cross-platform web application.
2. **Met:** Users who participated in the survey had a range of networking experience and were still able to use the tool (section 12.2).
3. **Fully met:** Users who took part in the evaluation were able to design networks with no guidance from outside the tool (section 12.2).

4. **Met:** Users' overall sentiment and confidence level after using the tool was higher than when they designed a network unsupported (section 12.2).

12.2 User evaluation

An evaluation questionnaire, included in section 1818, was distributed via the same channels as the requirements survey described previously (section 8.4). This questionnaire was intended to assess whether users felt more confident using the design tool than working without it, and get some feedback to inform future development work and improvements.

12.2.1 Respondent demographics

The demographic of the respondents was similar to that of the evaluation questionnaire, so we do not explore it in such detail here. There were 5 respondents, ranging from 2-6 years of experience in the industry. Everyone had experience with lighting with three listing it as their specialism (the rest gave sound as their specialism). All but one respondent had experience with sound, two mentioned it as their specialism. Two respondents had experience with video.

60% of respondents occasionally use networks in their work and the remainder use them all the time. 80% of respondents think they will use networks more next year, 20% will use them the same amount. Three out of five respondents have some responsibility over keeping an installation up to date, and the rest operate systems specified for them.

People generally felt confident about working in their specialism, giving a mean score of 1.6 (min 1, max 2) on the scale described in Figure 12. As before, there was a much wider range of confidence ratings for networking, with a mean of 0.8, median 1, min -2, max 2.

This shows that the evaluation has reached the target audience, spanning a range of experience levels and expertise.

12.2.2 Sentiment and confidence

Respondents were asked to first design a network for a venue in their normal way, using whatever drawings or notes they would normally make, then do the same thing again using the design tool. After each task, they were asked generally how they found the exercise, and how confident they were that their solution would work. For the first question, they responded on a scale of a very sad face to a very happy face with a neutral face in the middle, which we map onto the -2 to +2 scale used elsewhere (Figure 12). For the second, they used the same -2 to +2 confidence slider as they did when they reported how confident they felt about their specialism and using networks in general.

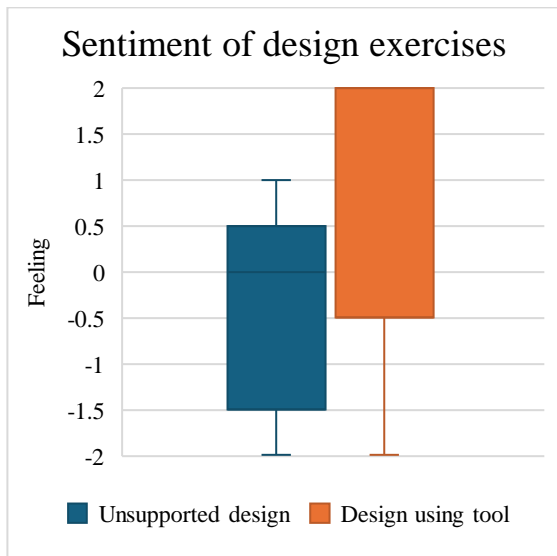


Figure 43: Respondents' overall feeling after designing a network in their normal way, then supported by the tool.

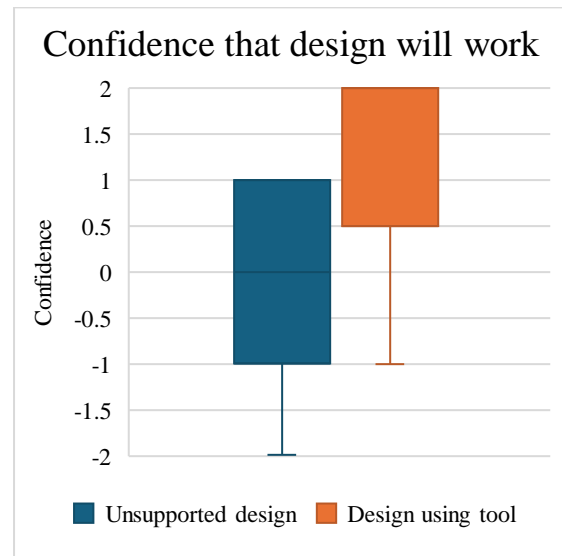


Figure 44: Respondents' confidence that their design will work after designing in their normal way, then supported by the tool.

In terms of overall sentiment, respondents were feeling negative or only slightly positive after making their design without any support (mean -0.4, median 0). After using the design tool, respondents were more positive, although there is a spread of responses across the entire scale (mean 1, median 2) (Figure 43).

No respondent was completely confident that their design would work in reality when working without the support of the tool, with a spread of confidence ratings ranging from not at all confident to slightly confident (mean 0, median 1) (Figure 44)⁵. After using the tool, all respondents were more confident, with most feeling completely confident their design will work and giving a +2 rating (mean 1, median 2).

These results suggest that the project has been successful in its objective of inspiring confidence in technicians, but that there is still further work possible to improve confidence.

12.2.3 Qualitative responses

The qualitative responses to the evaluation can be sorted into four categories:

- **Fixed bugs:**
 - Pressing enter in the device search box during the design wizard would progress the wizard to the next step instead of running the search.

⁵ Two people did not respond when asked for their confidence after designing without the tool. Everybody gave a confidence rating after designing with the tool. This means a different number of data points are being used for each part of this comparison.

- Some icons and buttons did not change colour when the application changed from light mode to dark mode, so were not visible.
- **Unfixed bugs:**
 - The application tour can cause some unexpected scrolling behaviour. This is a bug related to how the library used to run the tour interacts with the rest of the application, so more testing would be needed to find exactly what this issue is and how to resolve it.
- **UI improvements:** The majority of these refer to non-functional requirement 3, where users should be able to operate the system without separate training.
 - A user requested a print button to print maps. This already exists, so needs to be made more obvious in the UI.
 - A user requested the ability to see devices and IP addresses together in the map and change IP addresses more easily. Currently, this is included under device protocol options (Figure 45). This feedback sounds like this user did not find these options properly, so this UI could be reviewed or included more clearly in the tour.

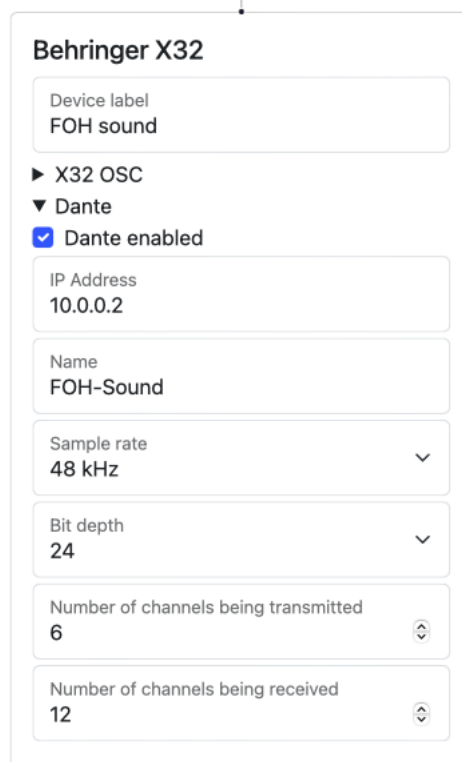


Figure 45: Partial screenshot of a device node and its configuration options.

- More zoom range on the map, so you can zoom out enough to see a large network in its entirety.
- Some steps of the tour do not have a back button, one user requested this to be added.

- **Feature requests:**

- A user requested more customisability. This request would be best informed by further user research to find exactly what users would like from the application and new requirements generated.
- Include a manual toggle between dark and light mode, instead of following the operating system theme. This links to the previous new requirement of greater customisability.
- The application as users tested included some generic “Other” devices (e.g. “Other Lighting Console”, “Other Sound Desk”) intended as a solution to help users design systems including hardware which this project didn’t have the ability to measure and add. One user requested the ability to customise these “other” devices. Theoretically, this would not be necessary in a final application as it would have a larger set of devices and protocols available in the first place, but it may be worth investigating this nonetheless so there is an ‘escape hatch’ if a technician needs to include a device not in the system and does not have time to wait for it to be properly added to the system by the administrators. This may result in a new functional requirement.
- A broader range of device types available, specifically a user requested radio microphone receivers. This would be possible in a larger project where more device data could be compiled.
- One user asked for the ability to mark devices as not counting towards a location’s requirements in the design wizard. The use case given for this was to prevent things like the local input and output sockets on a control console from contributing to the amount of IO in a location, since the normally-installed console in a venue may be swapped out for a touring company’s console so the IO available may change regularly. This is a new functional requirement, and user testing would be required to check that this doesn’t cause confusion.

One respondent reported that they still felt confused about how to create a network, and would like more explanation of the different hardware and the likely IO requirements in different locations. It is difficult to decide the best way for this to be included in the application: this need may best be met by a more comprehensive version of the tour, integration with other design tools to explore IO requirements and hardware options, or perhaps was a consequence of the open-ended nature of the exercise presented as part of the questionnaire.

Overall, users seemed generally happy with the system and the evaluation has given some good guidance for future user research focuses and development activities. The fact users had a range of feature requests suggests they could envisage using this tool again in future, so it is filling a current gap in the market.

13 Future work

Following the ISO 9241-210 standard, human-centred design requires an iterative approach informed by user feedback. From the requirements elicitation, design, and user evaluation processes carried out in this project, some suggestions of future work for the next iteration of this project include:

- A project focussing on the requirements of device manufacturers, protocol-designing organisations, and potential system administrators of this application, to get a clearer picture of their requirements from a tool like this.
- Adding a plugin system, so different devices and protocols can create custom UI elements for configuring their devices and protocols.
- Improving the user experience on mobile devices, so it is easier to reference the tool in the field. This would involve prototyping different ways to manipulate the potentially-large network map on a small screen and getting feedback from users.
- Measuring more protocols and devices, expanding the dataset to support more equipment.
- Exposing the database API publicly, encouraging its use by projects with similar goals to ours. If there is commercial interest, access to this API could be monetised.
- Further user involvement through more detailed and targeted questionnaires and focus groups. Now that one round of user requirement elicitation and evaluation have happened, the project has a clearer idea of what users may want. More interaction with stakeholders will help to focus these ideas further by clarifying open questions and prioritising different features.

14 Conclusion

This project met its objectives, following the principles of ISO 9241-210 to develop a tool to design networks in entertainment venues. By measuring real devices and reviewing protocol specifications, a database to store useful information about networked entertainment technologies was created and populated, with avenues to expand the dataset with more information in future. Through two rounds of interaction with real stakeholders, a requirement specification was drawn up, the system designed and implemented, then assessed against the project requirements. Users reported feeling more positive about network design after using the tool and offered helpful feedback. The feedback and evaluation can inform future work to develop this project further, continuing to make computer networks and their benefits available to more people in the entertainment industry.

15 Bibliography

- [1] Allen & Heath. SQ MixPad. *Allen & Heath*. Retrieved February 5, 2024 from <https://www.allen-heath.com/hardware/sq/sq-mixpad/>
- [2] Ariel Ortiz Ramirez. 2000. Three-Tier Architecture. *Linux Journal* 2000, 75es (July 2000), 7-es.
- [3] Artistic Licence. 2023. Art-Net 4 Specification.
- [4] Audinate. 2018. Dante Audio System Application Diagrams.
- [5] Avolites. Tiger Touch II. *Avolites*. Retrieved April 18, 2024 from <https://www.avolites.com/product/the-tiger-touch-ii/>
- [6] Richard Cabot. High-resolution multi-channel audio interconnection (HRMAI).
- [7] Richard C Cabot. Audio applications of networks - High-performance streaming audio-over-IP interoperability.
- [8] ESTA. 2018. ANSI E1.31 — 2018 Entertainment Technology Lightweight streaming protocol for transport of DMX512 using ACN.
- [9] Jennifer L. Ward. 2010. Persona Development and Use, or, How to Make Imaginary People Work for You. October 2010. . Retrieved April 17, 2024 from <https://digital.lib.washington.edu/researchworks/bitstream/handle/1773/19303/personas-Paper-2010LAC-ward.pdf?sequence=1&isAllowed=y>
- [10] John Huntington. 2007. *Control Systems for Live Entertainment*.
- [11] Jon Yablonski. 2020. *Laws of UX: Using Psychology to Design Better Products & Services*. O'Reilly.
- [12] Mohammad Amin Kuhail, Shahbano Farooq, Rawad Hammad, and Mohammed Bahja. 2021. Characterizing Visual Programming Approaches for End-User Developers: A Systematic Review. *IEEE Access* 9, (2021), 14181–14202. <https://doi.org/10.1109/ACCESS.2021.3051043>
- [13] Magnolia International Ltd. Midas | Product | M32C. Retrieved February 5, 2024 from <https://www.midasconsoles.com/product.html?modelCode=P0BM8>
- [14] Mark Otto, Jacob Thornton, and Bootstrap contributors. Bootstrap. Retrieved April 23, 2024 from <https://getbootstrap.com/>
- [15] Jakob Nielsen. 1994. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94*, 1994. ACM Press, Boston, Massachusetts, United States, 152–158. <https://doi.org/10.1145/191666.191729>
- [16] Rick Kreifeldt. *AVB for Professional A/V Use*. AVnu Alliance.
- [17] Wayne Howell. 2009. *Rock Solid Ethernet* (2nd ed.). Entertainment Technology Press.
- [18] 2018. American National Standard ANSI E1.11 – 2008 (R2018) Entertainment Technology—USITT DMX512-A Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories. Retrieved January 28, 2024 from https://tsp.esta.org/tsp/documents/docs/ANSI-ESTA_E1-11_2008R2018.pdf
- [19] 2018. DiGiCo Opens Up the SD Range with Generic OSC control. *DiGiCo*. Retrieved February 5, 2024 from <https://digico.biz/digico-opens-up-the-sd-range-with-generic-osc-control/>
- [20] 2020. So you're adding Dante to your network?

- [21] 2021. OSC index. Retrieved February 5, 2024 from <https://opensoundcontrol.stanford.edu/index.html>
- [22] 2023. *The Big Freelancer Survey 2023*. Freelancers Make Theatre Work.
- [23] 2023. *NDI 5.6 White Paper 2023*. Vizrt NDI AB. Retrieved February 2, 2024 from <https://ndi.video/wp-content/uploads/2023/09/NDI-5.6-White-Paper-2023.pdf>
- [24] Meet Dante. *Dante*. Retrieved February 10, 2024 from <https://www.getdante.com/meet-dante/>
- [25] Why MIDI Matters. *The MIDI Association*. Retrieved February 5, 2024 from <https://midi.org/midi-articles/why-midi-matters>
- [26] Dante Certification Program | Audinate | Dante Pro AV Networking. Retrieved January 29, 2024 from <https://www.audinate.com/learning/training-certification/dante-certification-program>
- [27] Cisco Modeling Labs. *Cisco*. Retrieved February 4, 2024 from <https://www.cisco.com/c/en/us/products/cloud-systems-management/modeling-labs/index.html>
- [28] GigaCore switches - Luminex. Retrieved February 4, 2024 from <https://www.luminex.be/products/gigacore/>
- [29] Araneo - AV network topology viewer. *Luminex*. Retrieved February 4, 2024 from <https://www.luminex.be/products/software/araneo/>
- [30] Entertainment & Lighting - Design Software | Vectorworks Spotlight. Retrieved April 28, 2024 from <https://www.vectorworks.net/en-GB/spotlight>
- [31] Ergonomics of human-system interaction - Human-centred design for interactive systems.
- [32] Ion Features. *Electronic Theatre Controls*. Retrieved April 18, 2024 from <https://www.etcconnect.com/Products/Legacy/Console/Eos-Family/Ion/Features.aspx>
- [33] Behringer | Product | X32. *Behringer*. Retrieved April 18, 2024 from <https://www.behringer.com/behringer/product?modelCode=0603-ACE>
- [34] Behringer | Product | X-DANTE. *Behringer*. Retrieved April 18, 2024 from <https://www.behringer.com/behringer/product?modelCode=0606-ABX>
- [35] Dante A168D STAGE. *DiGiCo*. Retrieved April 18, 2024 from https://digico.biz/4rea4_components/a168d-stage-copy/
- [36] Audinate - Dante AVIO 2-Channel Analog Input. *Dante*. Retrieved April 18, 2024 from <https://www.getdante.com/product/audinate-dante-avio-2-channel-analog-input/>
- [37] Avenue & Arena - Resolume. *Resolume*. Retrieved April 18, 2024 from https://resolume.com/software/avenue_arena
- [38] NDI HX Camera: Turn your phone into a live video source. *NDI*. Retrieved April 18, 2024 from <https://ndi.video/tools/ndi-hx-camera/>
- [39] Wireshark. Retrieved April 16, 2024 from <https://www.wireshark.org>
- [40] How to Implement Dante for Audio. *Dante*. Retrieved April 20, 2024 from <https://www.getdante.com/products/solutions-for-manufacturers/dante-for-audio/>
- [41] Fixture Library Hub. Retrieved April 20, 2024 from <https://fixture-library-hub.etcconnect.com/help>
- [42] Avolites Personality Library. Retrieved April 20, 2024 from <https://personalities.avolites.com/?mainPage=Downloads.asp&>

- [43] ChamSys MagicQ Downloads. *ChamSys*. Retrieved April 20, 2024 from <https://chamsyslighting.com/pages/magicq-downloads>
- [44] Node-Based UIs in React – React Flow. Retrieved April 23, 2024 from <https://reactflow.dev/>
- [45] Zustand. Retrieved April 28, 2024 from <https://zustand-demo.pmnd.rs/>
- [46] React. Retrieved April 23, 2024 from <https://react.dev/>
- [47] Remix - Build Better Websites. Retrieved April 23, 2024 from <https://remix.run>
- [48] Strapi - Open source Node.js Headless CMS 🚀. Retrieved April 23, 2024 from <https://strapi.io/>
- [49] Prisma | Simplify working and interacting with databases. *Prisma*. Retrieved April 23, 2024 from <https://www.prisma.io>

16 Appendix: Ethics approval



University of
St Andrews | FOUNDED
1413

School of Computer Science Ethics Committee

09 February 2024

Dear Andrew,

Thank you for submitting your ethical application which was considered by the School Ethics Committee.

The School of Computer Science Ethics Committee, acting on behalf of the University Teaching and Research Ethics Committee (UTREC), has approved this application:

Approval Code:	CS17539	Approved on:	09.02.2024	Approval Expiry:	09.02.2029
Project Title:	A tool to design networks for entertainment venues				
Researcher(s):	Andrew Barron				
Supervisor(s):	Prof Saleem Bhatti				

The following supporting documents are also acknowledged and approved:

1. Application Form
2. Participant Information Sheet(s)
3. Consent Form
4. Questionnaire

Approval is awarded for 5 years, see the approval expiry data above.

If your project has not commenced within 2 years of approval, you must submit a new and updated ethical application to your School Ethics Committee.

If you are unable to complete your research by the approval expiry date you must request an extension to the approval period. You can write to your School Ethics Committee who may grant a discretionary extension of up to 6 months. For longer extensions, or for any other changes, you must submit an ethical amendment application.

You must report any serious adverse events, or significant changes not covered by this approval, related to this study immediately to the School Ethics Committee.

Approval is given on the following conditions:

- ☐ that you conduct your research in line with:
 - o the details provided in your ethical application
 - o the University's [Principles of Good Research Conduct](#)
 - o the conditions of any funding associated with your work
- ☐ that you obtain all applicable additional documents (see the ['additional documents' webpage](#) for guidance) before research commences.

You should retain this approval letter with your study paperwork.

Yours sincerely,

Natasha Bazeley

SEC Administrator

School of Computer Science Ethics Committee

Dr Olexandr Kononov/Convenor, Jack Cole Building, North Haugh, St Andrews, Fife, KY16 9SX

Telephone: 01334 463273 Email: ethics-cs@st-andrews.ac.uk

The University of St Andrews is a charity registered in Scotland: No SC013532



School of Computer Science Ethics Committee

14 March 2024

Dear Andrew,

Thank you for submitting your ethical amendment application.

The School of Computer Science Ethics Committee has approved this ethical amendment application:

Original Approval Code:	CS17539	Original Approval Date:	09.02.2024
Amendment Approval Date:	13.03.2024	Approval Expiry Date:	10.02.2029
Project Title:	A tool to design networks for entertainment venues		
Researcher(s):	Andrew Barron	Supervisor/PI:	Saleem Bhatti
School/Unit:	Computer Science		

The following supporting documents are also acknowledged and approved:

1. Ethical Amendment Application Form

This approval does not extend the originally granted approval period. If you require an extension to the approval period, you can write to your School Ethics Committee who may grant a discretionary extension of no greater than 6 months. For longer extensions, or for any further changes, you must submit an additional ethical amendment application. For all extensions, you should inform the School Ethics Committee when your study is complete.

You must report any serious adverse events, or significant changes not covered by this approval, related to this study immediately to the School Ethics Committee.

Approval is given on the following conditions:

- ☐ that you conduct your research in line with:
 - the details provided in your ethical amendment application (and the original ethical application where still relevant)
 - the University's [Principles of Good Research Conduct](#)
 - the conditions of any funding associated with your work
- ☐ that you obtain all applicable additional documents (see the ['additional documents' webpage](#) for guidance) before research commences.

You should retain this approval letter with your study paperwork.

Yours sincerely,

SEC Administrator

School of Computer Science Ethics Committee

Dr Olexandr Konovalov/Convenor, Jack Cole Building, North Haugh, St Andrews, Fife, KY16 9SX

T: 01334 463273 E: ethics-cs@st-andrews.ac.uk

The University of St Andrews is a charity registered in Scotland: No SC013532

17 Appendix: Requirements survey

1. How long have you been working in the entertainment industry in a technical role?
 - a. *[Answer by slider from 0-40 years.]*
2. Which technical areas do you have experience in? Select as many as you like.
 - a. Lighting
 - b. Sound
 - c. Video
 - d. Other *[free text]*
3. Which technical department would you say is your specialism?
 - a. Lighting
 - b. Sound
 - c. Video
 - d. Other *[free text]*
 - e. I don't have a specialism
4. Please rate your confidence in each of these areas.
[using a slider from -2 to +2, where -2 is labelled "Not confident, I would need lots of help to do tasks in this area", 0 is labelled "No experience or not sure", and +2 is labelled "Very confident, I could do most tasks in this area independently and help others too"]
 - a. *[The specialism selected in question 4, or "General technical crew work" if not answered or no specialism indicated]*
 - b. Setting up and using computer networks in your work
5. How regularly do you use computer networks in your work?
 - a. Never
 - b. Occasionally (e.g. to remote control a console or connect one lighting console to one node), but I mostly rely on analogue and point to point-solutions like multicores, AES50, SLink, SDI, etc
 - c. All the time, computer networks are a core part of the system design where I work
 - d. Other *[free text]*
6. Do you think your use of computer networks will change over the next year?
 - a. I think I'll use networks less
 - b. I think I'll use networks about the same amount
 - c. I think I'll use networks more
 - d. Other *[free text]*
7. How much choice and control do you have over the equipment you use in your work?
 - a. The systems I use are specified by someone else, I just operate them
 - b. I have some responsibility over keeping an installation up-to-date and modifying its design
 - c. I'm regularly designing and setting up systems from scratch (designing installations, touring)
 - d. Other *[free text]*
8. To increase your confidence using computer networks at work, what activities would you be likely to do? You can select as many options as you like or add your own.
 - a. Attend training courses
 - b. Read reference material (books, etc)
 - c. Use design tools
 - d. Practice and try to learn on the job
 - e. Other *[free text]*

9. What features would you like from a network design tool intended for entertainment venues?
 - a. Access to a database of information about protocols and devices
 - b. Options to experiment with and test different setups
 - c. Ways to learn about new topics and concepts
 - d. Reassurance that plans I have made will work in reality
 - e. Other *[free text]*

18 Appendix: Evaluation questionnaire

1. To confirm you're eligible to participate in this questionnaire, please select the appropriate statement from the options. *[required]*
 - a. I am 18 years old or older and work or volunteer in the entertainment industry
 - b. I am younger than 18 years old and/or do not work nor volunteer in the entertainment industry *[leads to termination of survey with no response recorded]*
2. How long have you been working in the entertainment industry in a technical role?
 - a. *[Answer by slider from 0-40 years.]*
3. Which technical areas do you have experience in? Select as many as you like.
 - a. Lighting
 - b. Sound
 - c. Video
 - d. Other *[free text]*
4. Which technical department would you say is your specialism?
 - a. Lighting
 - b. Sound
 - c. Video
 - d. Other *[free text]*
 - e. I don't have a specialism
5. Please rate your confidence in each of these areas.
[using a slider from -2 to +2, where -2 is labelled "Not confident, I would need lots of help to do tasks in this area", 0 is labelled "No experience or not sure", and +2 is labelled "Very confident, I could do most tasks in this area independently and help others too"]
 - a. *[The specialism selected in question 4, or "General technical crew work" if not answered or no specialism indicated]*
 - b. Setting up and using computer networks in your work
6. How regularly do you use computer networks in your work?
 - a. Never
 - b. Occasionally (e.g. to remote control a console or connect one lighting console to one node), but I mostly rely on analogue and point to point-solutions like multicores, AES50, SLink, SDI, etc
 - c. All the time, computer networks are a core part of the system design where I work
 - d. Other *[free text]*
7. Do you think your use of computer networks will change over the next year?
 - a. I think I'll use networks less
 - b. I think I'll use networks about the same amount
 - c. I think I'll use networks more
 - d. Other *[free text]*
8. How much choice and control do you have over the equipment you use in your work?
 - a. The systems I use are specified by someone else, I just operate them
 - b. I have some responsibility over keeping an installation up-to-date and modifying its design
 - c. I'm regularly designing and setting up systems from scratch (designing installations, touring)
 - d. Other *[free text]*

9. Please take some time to think through a network you might set up in your work. Feel free to note your ideas down on paper and do any drawings or plans you'd normally do while setting up a network, you won't need to submit them.

Include control consoles from different departments, lighting nodes, stage boxes, and any other digital equipment you're familiar with. Think about cable runs, addresses, and how devices will communicate.

How did you find that exercise?

- a. *[Answer by smiley face from very sad to very happy.]*
10. How confident are you that your solution will work?
- a. *[Answer using a slider from -2 to +2, where -2 is labelled "I didn't finish or I know my solution won't work", 0 is labelled "It might work, but I'm not sure", and +2 is labelled "My solution will definitely work"]*
11. Now, try copying your system design from the previous step into our design tool. Open the tool in a new tab so you can come back to this questionnaire when you're finished.

If you didn't finish a design in the last step, feel free to start something new now.

What device did you use the tool on?

- a. Desktop computer
- b. Laptop
- c. Tablet
- d. Mobile phone
- e. Other *[free text]*
12. Which web browser did you use the tool on?
- a. Chrome
- b. Firefox
- c. Safari
- d. Edge
- e. Other *[free text]*
13. How did you find that exercise?
- a. *[Answer by smiley face from very sad to very happy.]*
14. How confident are you that your solution will work?
- a. *[Answer using a slider from -2 to +2, where -2 is labelled "I didn't finish or I know my solution won't work", 0 is labelled "It might work, but I'm not sure", and +2 is labelled "My solution will definitely work"]*
15. Did you have any problems using the tool?
- a. *[Free text]*
16. Is there anything missing from the tool that you think would make the task easier?
- a. *[Free text]*
17. Do you have any other feedback about the tool, in general or compared to the previous design exercise?
- a. *[Free text]*

19 Appendix: Application tour

Tour dialogue	Highlighted UI element	Progress by pressing
Welcome to the tour! This will guide you through the main features of the app as we build a small network together.	None	Next
Let's start by adding a switch to your design. Click "Add switch".	"Add switch" tool button	Add switch
To make a switch, enter the details of a real switch you have, or type in the following example values: Switch speed 1000baseT, 1000 Mbps, Gigabit Switching capacity 10000 Number of ports 5 Power over Ethernet support Unsupported Press "Add switch" twice to create two switches, then press "Next" to continue.	Tool pane	Next
In the network map on the left, the switches we've just made have appeared. You can edit any of the settings you chose when you made the switch using the options in the box. You can click and drag anything on the map to move it around. Try labelling each switch so you know which is which as we move through the tour - perhaps name one "Stage" and one "Front of house".	Network map	Next

Press "Next" when you're ready to continue.		
We'll add some devices next. Click "Add device"	"Add device" tool button	Add device
<p>This panel shows you all the devices the system knows about. You can search and filter at the top, or scroll to browse.</p> <p>Pick some devices to add to your network by pressing + or dragging the item onto the map. Include some devices you'd use at front of house and some you'd use on stage. Here's some ideas:</p> <ul style="list-style-type: none"> • A sound desk, like Behringer X32. • A lighting desk, like ETC Ion Classic. • An sACN/ArtNet node, like Luminex Luminode 12. • A stage box, like DiGiCo A168D. <p>Once you've added all the devices you want, press "Next" to continue.</p>	Tool pane	Next
<p>Drag the devices you've made around in the map so you can see everything. You can also add some labels if you like.</p> <p>Any network protocols supported by a device are presented in a dropdown on the device. You can open any of these dropdowns to configure the protocol.</p>	Network map	Next
These dots are ports, used to connect devices and switches together.	A row of dots at the bottom of one of the switch nodes	Next
Use your mouse to draw a cable between the ports on each device and a port on one of the switches.	Network map	Next

<p>Connect all the devices at front of house to the front of house switch, and all the devices on stage to the stage switch.</p> <p>Then, connect the two switches together.</p>		
<p>Every device in our network is identified by an address, known as an IP address. We could manually give every device an address, but the tool can do it for us instead.</p> <p>Press "IP Addressing".</p>	"IP addressing" tool button	IP addressing
<p>Press "Assign addresses" to assign an address to every device in the network.</p>	"Assign addresses" section of the tool pane	Assign addresses
<p>The "Check design" tab lets you check your work. Press the tab to open it.</p>	"Check design" tool button	Check design
<p>This tab will show a list of everything which the system notices might be wrong in your network. It'll check if the configuration options you've set are valid, that everything is connected up properly, and so on.</p> <p>If there's anything wrong, you can fix it now or come back later and have a look.</p>	Tool pane	Next
<p>The "Advice" tab is another helper to ensure your design works as well as possible. Press the tab to open it.</p>	"Advice" tool button	Advice
<p>This tool can't check everything about your design. The Advice tab includes pointers towards anything you should check manually, or any extra steps you should do when setting up your network.</p>	Tool pane	Next
<p>When it comes time to install your network, the "Views" tab comes in handy. Press the tab to open it.</p>	"Views" tool button	Views

These reports can be printed out and include information about what is plugged into every switch, and the configuration you plan to do on each device.	Tool pane	Next
Finally, the "Open and save" tab is where you can manage your designs and share them with others. Press the tab to open it.	"Open and save" button	Open and save
Here, you can save your work, open other designs you've made, and share your work with others.	Tool pane	Next
And that's the end of the tour. You can re-do it any time you like from the homepage. Press "Done" to continue experimenting with the network we started here, or [start a new blank file], or [jump-start a design with the design wizard].	None	Finish

20 Appendix: Screenshots of application

This section includes some screenshots of representative sections of the application, showing some screens which otherwise may not have been included in the main body of the report.

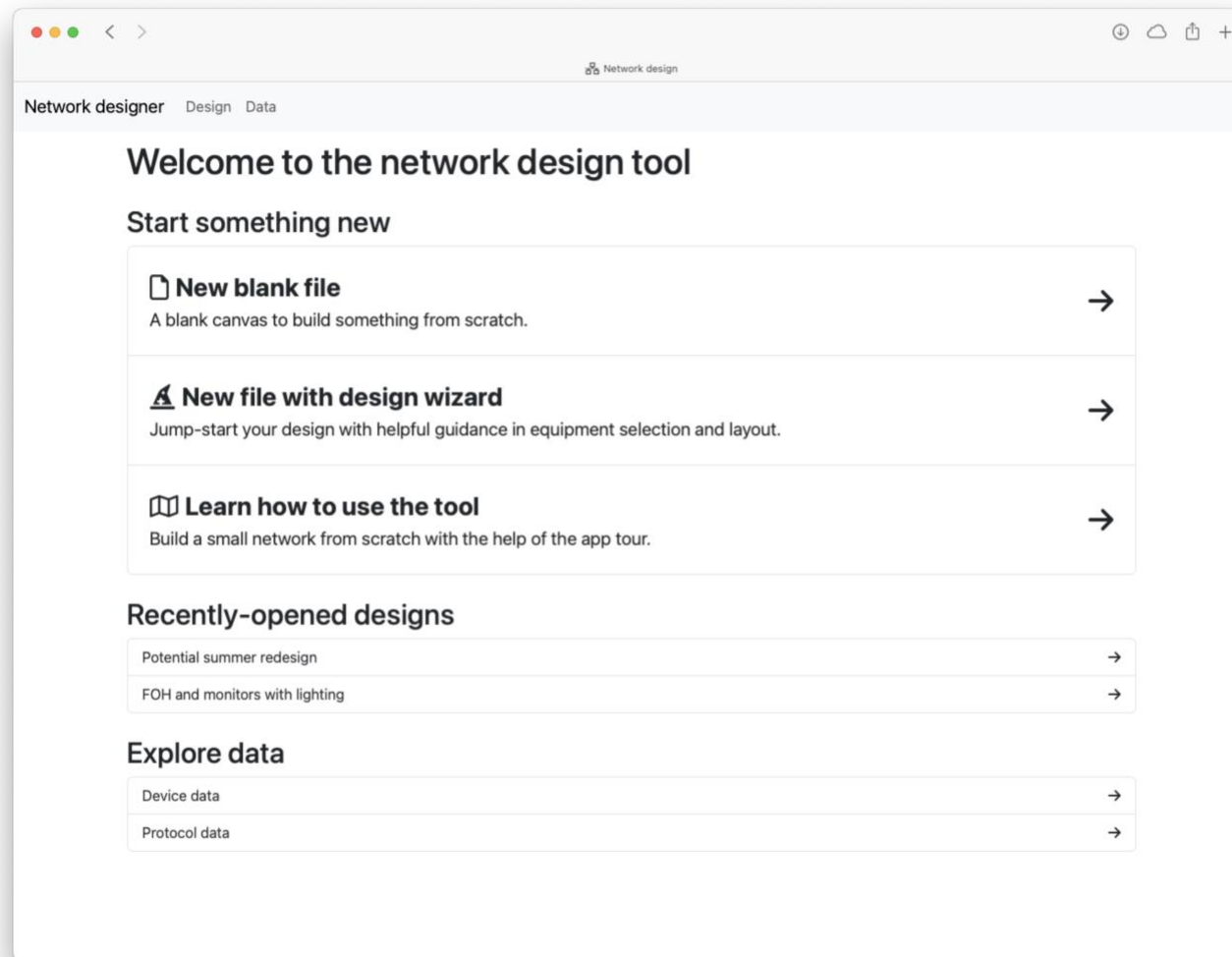


Figure 46: Application homepage. There are two recently-opened files available.

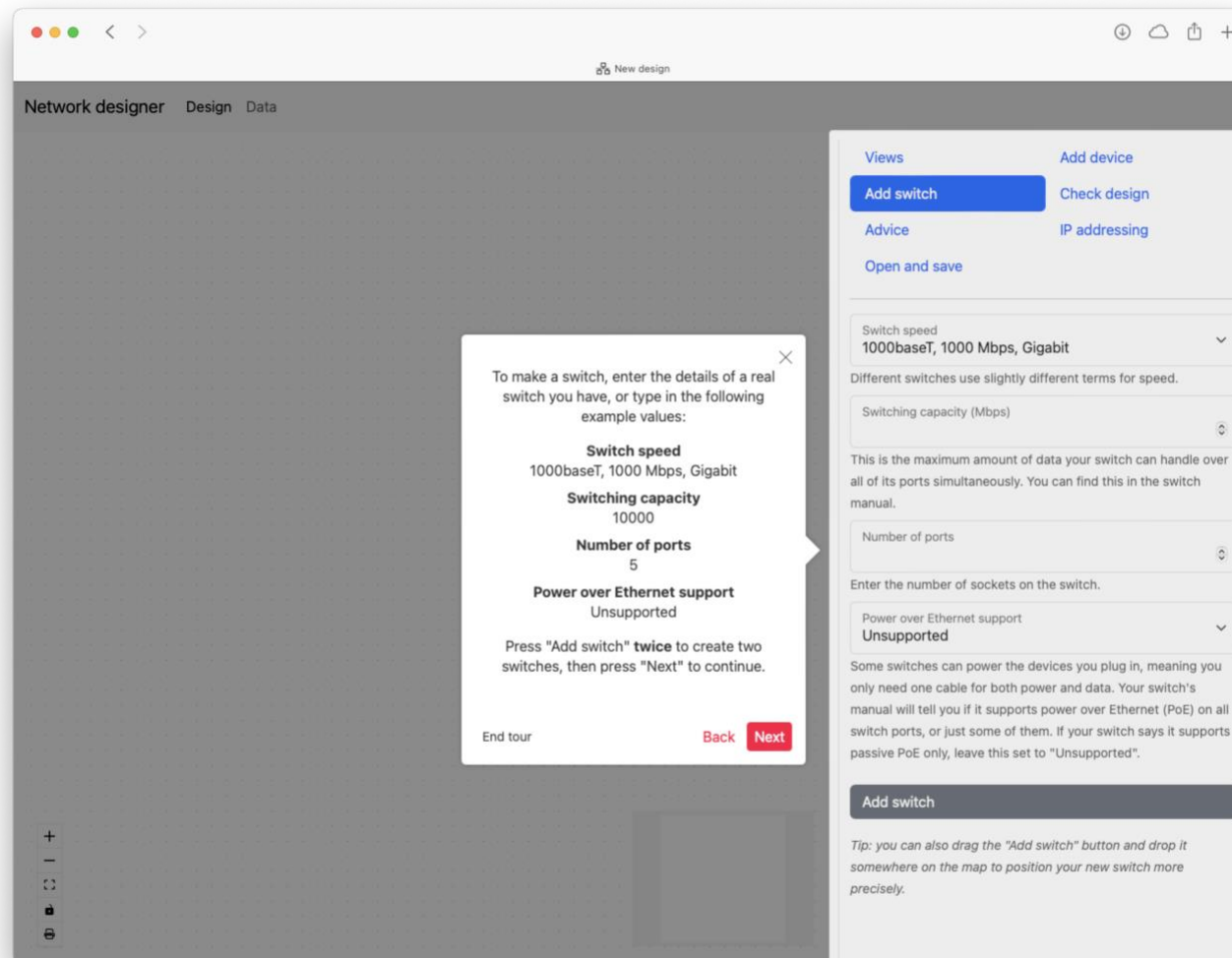


Figure 47: Application tour. The user is being guided to configure a switch.

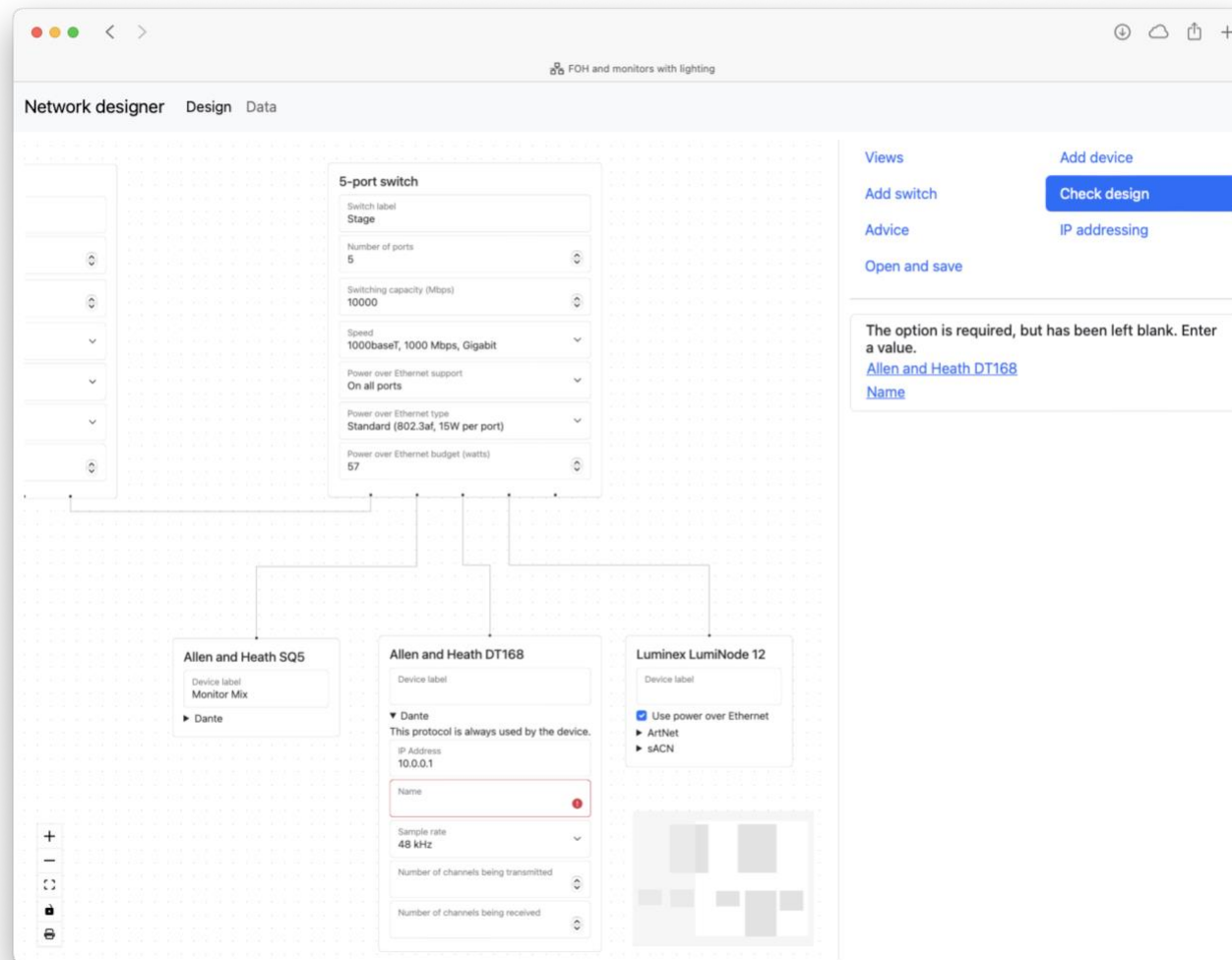


Figure 48: After creating a design, the technician is using the "Check design" tool pane to find potential problems. A missing configuration option is being indicated in the pane and on the input itself.

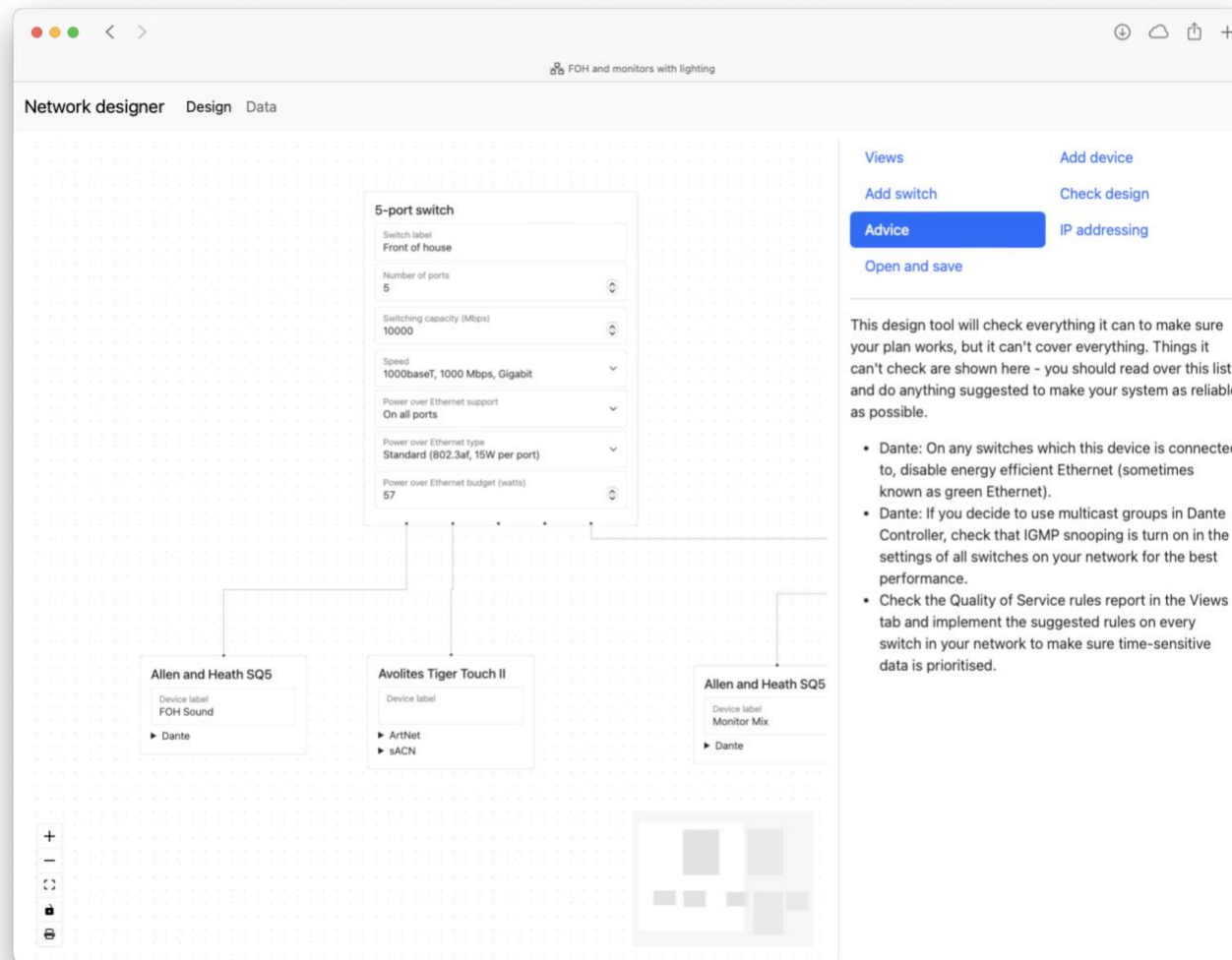


Figure 49: The Advice tool pane is giving some suggestions on the most efficient way to configure the network.

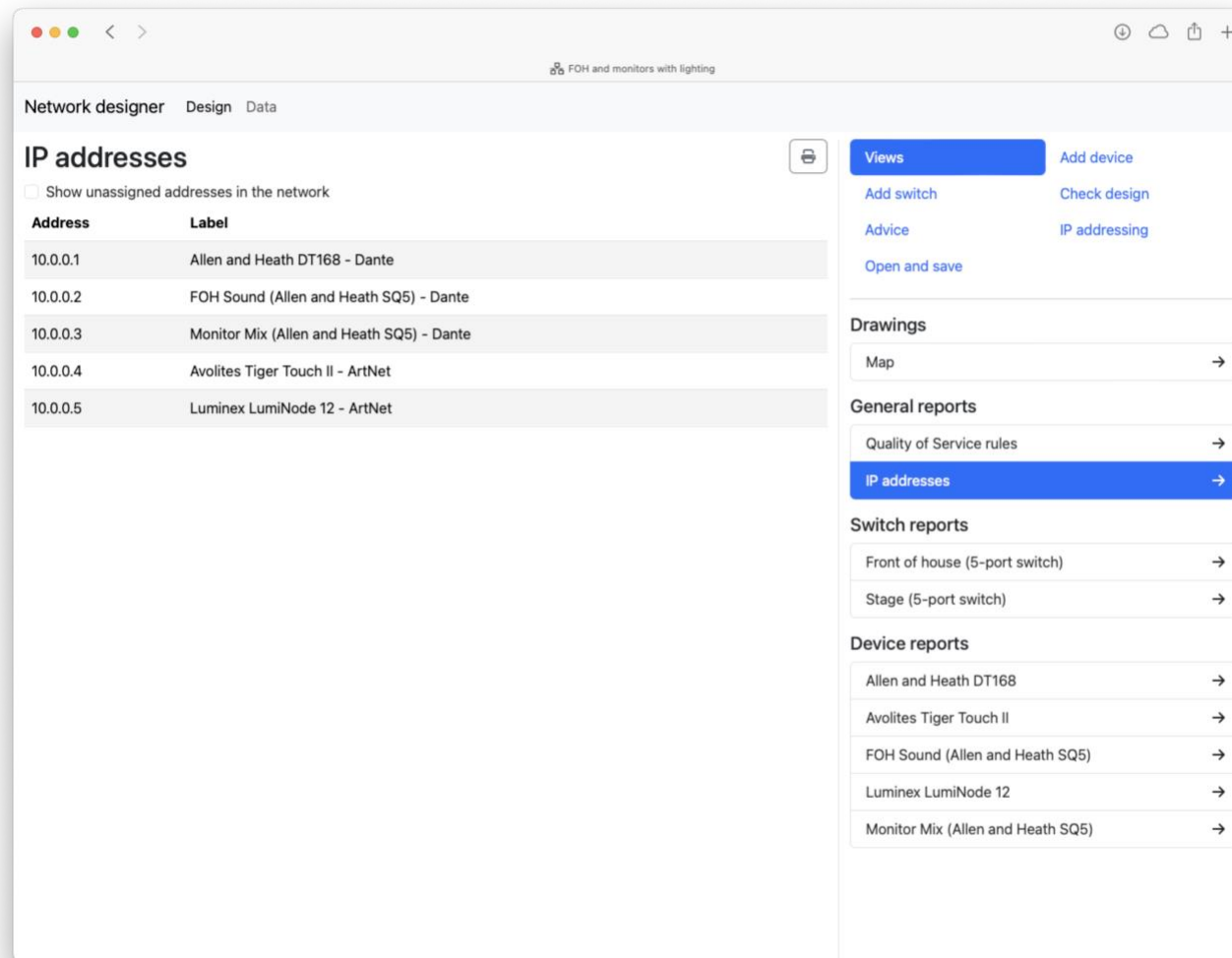


Figure 50: The IP address report displays the address of every device in the system in one convenient reference list. This can be printed with the button at the top.

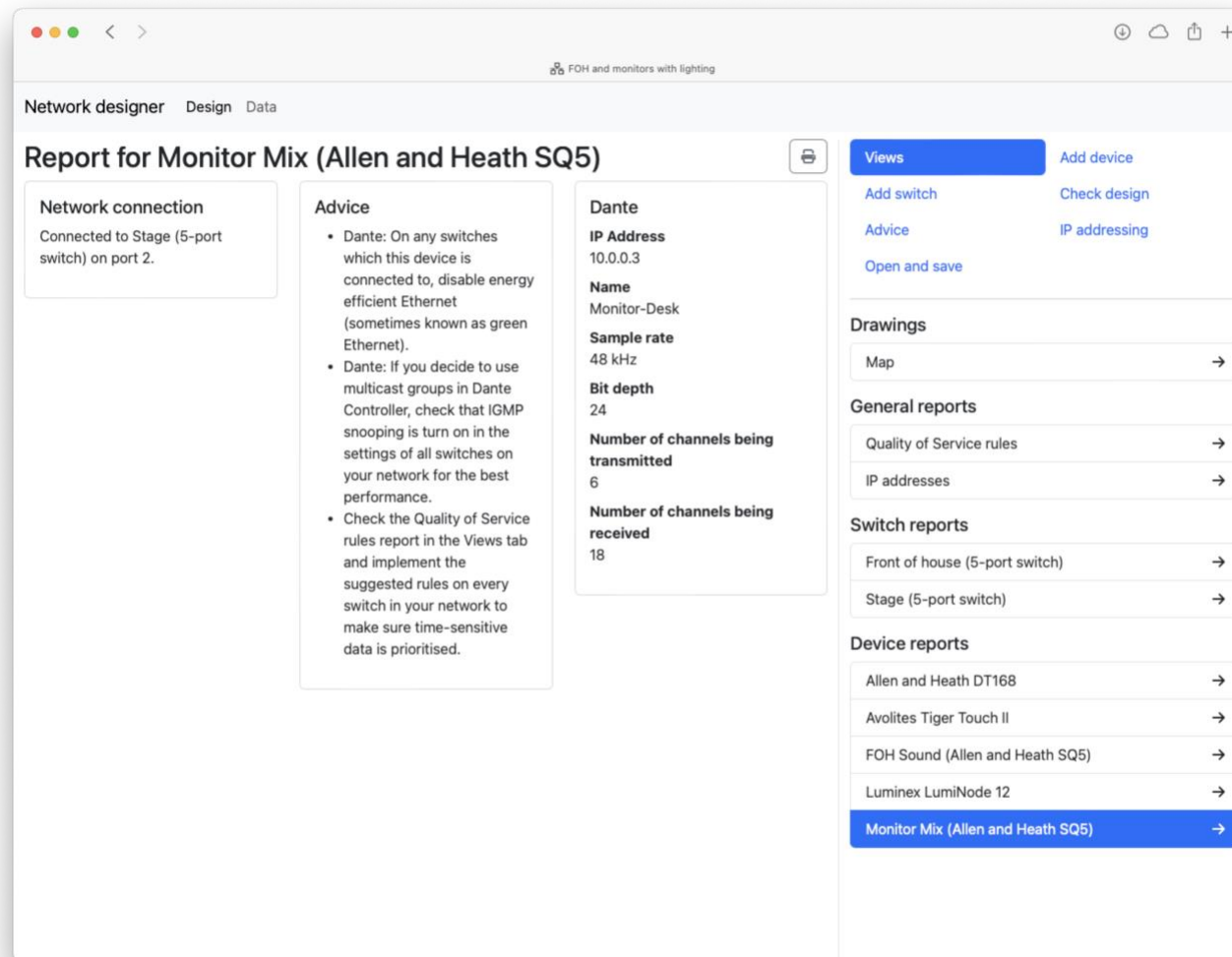


Figure 51: The device report includes information about where to connect the device, setup advice, and the necessary protocol configuration options. This can be printed with the button at the top.

Design wizard

Pick your kit.

Add equipment to each location to meet your requirements.

Front of house Stage

Location status

Check if what you've selected for this location matches your needs.

Lighting Consoles 1 of 1 Univ. out 4 of 0 Sound Desks 0 of 1 Inputs 0 of 3 Outputs 0 of 0 Video Mixers 0 of 0 Inputs 0 of 0 Outputs 0 of 0

Selected devices

This is the equipment you've added to this location.

Device	Inputs	Outputs
Avolites Tiger Touch II	0	4

Add devices

Press items from this list to add them to this location. You can have up to 47 devices in each location.

Search devices by make or model

Show device types Sound desk

Min inputs 3 Max inputs Max outputs

☒ Only show devices which support the protocols I chose

Device	Inputs	Outputs
Allen and Heath SQ5	22	12
Behringer X32	38	22
Behringer X32 Compact	22	14
Behringer X32	22	14

Next Back Cancel

Figure 52: The final step of the design wizard, where the user is selecting which devices they would like in each location. In this screenshot, the stage is complete and they are working on front of house. They have filtered the list of available devices to find things which meet their sound requirements..

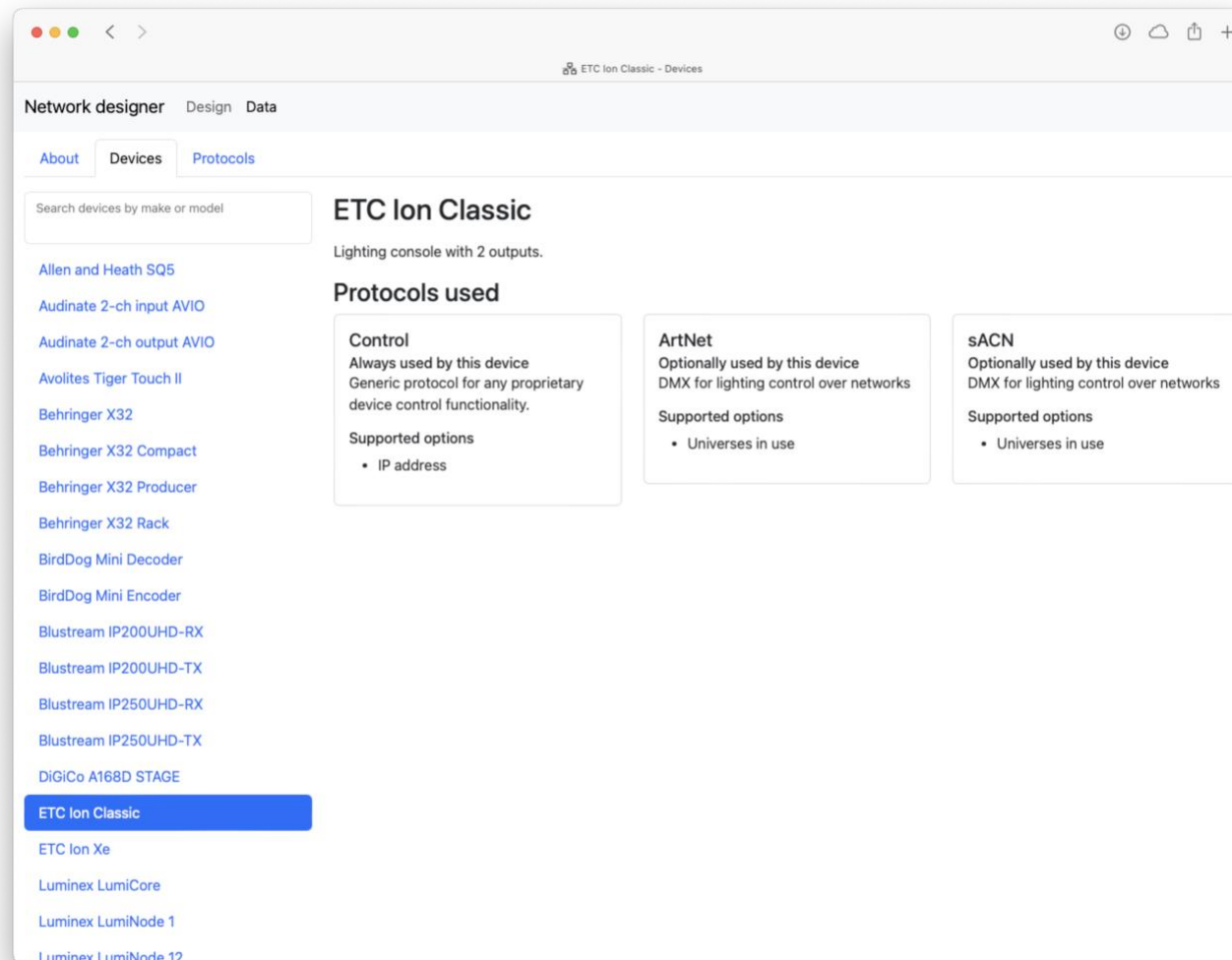


Figure 53: In the data browser, information about ETC Ion Classic is being displayed. The same layout is used in the design area when the "info" button is pressed on a device.

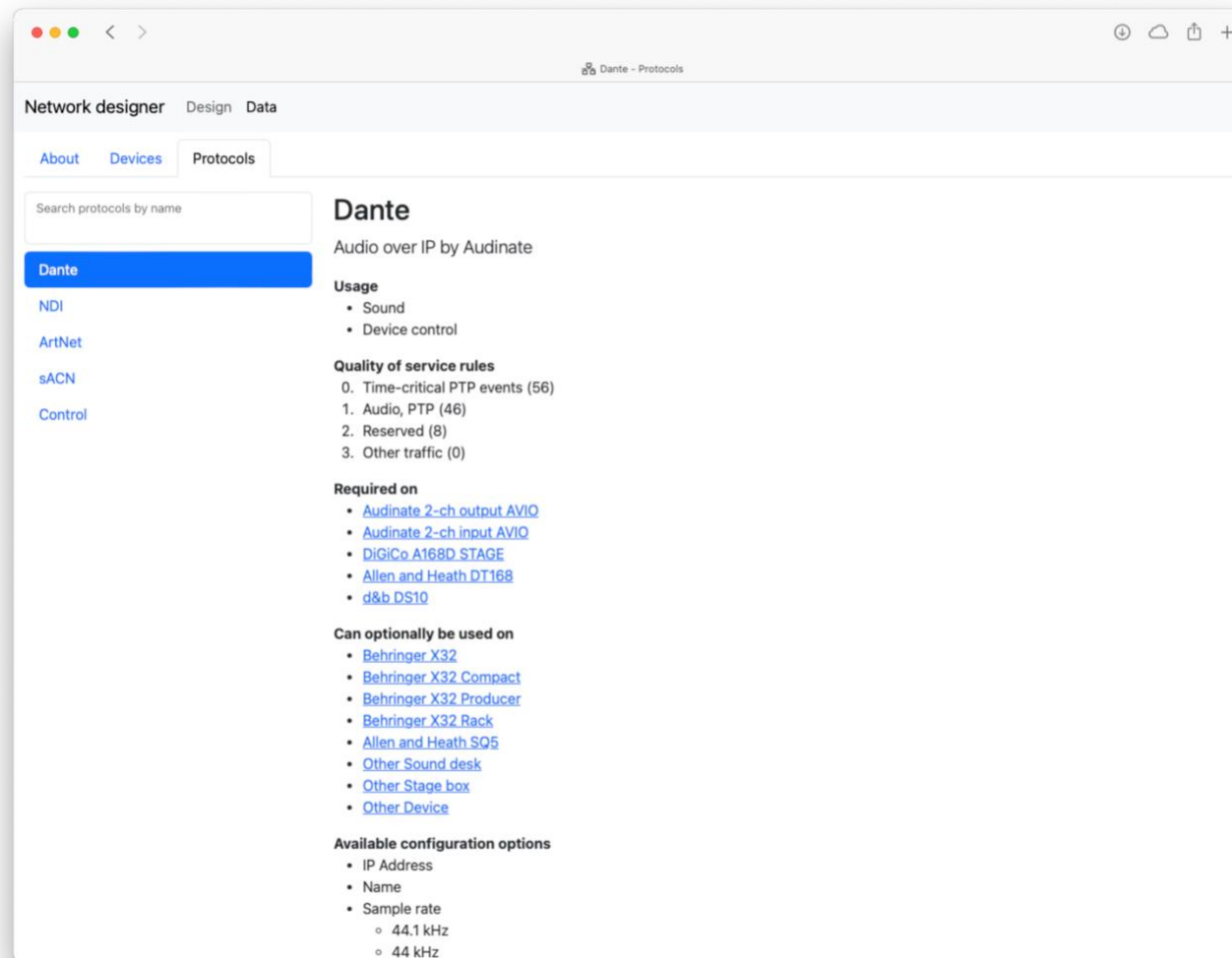


Figure 54: In the data viewer, information about a protocol is being displayed.

21 Appendix: Database schema

This is a logical schema for the database. The physical schema you would see in the MySQL database itself will vary from this, as Strapi generates a schema based on this higher-level specification and takes advantage of components for repeating fields, etc. This is also where some non-standard data types like ‘enum’ come from.

Similarly, under the ‘configOptions’ relation, there are many fields. Under Strapi, the exact fields available will change depending on whether the option type is set to text, number, select (dropdown), or IP address, so only validation options which make sense for the given type are available. Only if the option type is ‘select’ will the ability to create a relationship with ‘selectOption’ records become available.

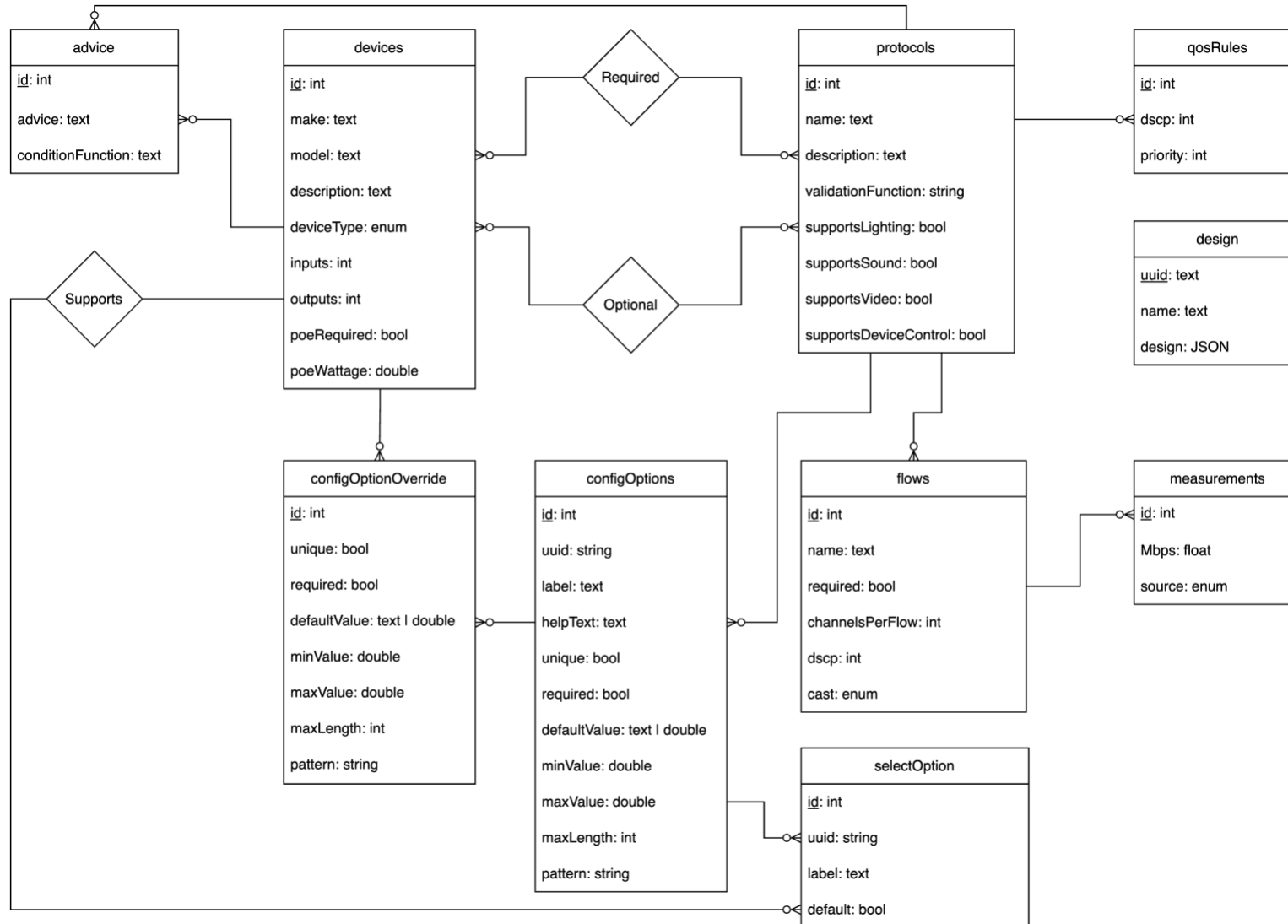


Figure 55: Entity-relationship diagram for the database.

22 Appendix: Testing summary

These tests are informed by the project objects and requirements, as described in section 12.1.

In the tables below, “FR” stands for functional requirement. The numbers map to those used in section 8.3.1.

22.1 Map and reports

ID	FR	Description	Expected output	Actual output	Status
1.	1	Create a new device in the administration interface.	The device appears in the Explore Devices section of the main application.	As expected.	Pass
2.	1	Create a new protocol in the administration interface.	The protocol appears in the Explore Protocols section of the main application.	As expected.	Pass
3.	1	Edit an existing device in the administration interface.	The changes are reflected in the Explore Devices section of the main application.	As expected.	Pass
4.	1	Edit an existing protocol in the administration interface.	The changes are reflected in the Explore Protocols section of the main application.	As expected.	Pass
5.	1	Delete a device in the administration interface.	The device disappears from the Explore Devices section of the main application.	As expected.	Pass
6.	1	Delete a protocol in the administration interface.	The protocol disappears from the Explore Protocols section of the main application.	As expected.	Pass
7.	2	In a new design file, drag and drop a device from the tool pane into the map area.	The device appears on the map where the cursor is released.	As expected.	Pass
8.	2	In a new design file, press the plus button on a device in the tool pane.	The device appears on the map, roughly in the middle of the viewport.	As expected.	Pass

9.	2	In a new design file, configure a switch as -- <i>Speed: Gigabit</i> <i>Switching capacity: 10000</i> <i>Number of ports: 5</i> <i>PoE support: Unsupported</i> Then add to the design with the “Add switch” button.	The switch appears roughly in the centre of the design with the same settings configured.	As expected.	Pass
10.	2	As test 9, with config -- <i>PoE support: On some ports</i> <i>PoE type: PoE+</i> <i>PoE budget: 42</i> <i>PoE ports: 1, 2, 3</i>	The PoE type, budget, and ports options appear during configuration. When adding the switch, it appears on the map configured appropriately.	As expected.	Pass
11.	2	As test 9, with config -- <i>PoE support: On all ports</i> <i>PoE type: Standard</i> <i>PoE budget: 75</i>	The PoE type, and budget options appear during configuration. When adding the switch, it appears on the map configured appropriately.	As expected.	Pass
12.	2	As test 9, with config -- <i>Number of ports: 1</i>	The add switch button rejects the number of ports.	As expected.	Pass
13.	2	As test 9, with config -- <i>Switching capacity: [blank]</i>	The add switch button rejects the blank switching capacity.	As expected.	Pass
14.	2	As test 9, with config -- <i>Number of ports: [blank]</i>	The add switch button rejects the blank number of ports.	As expected.	Pass
15.	3	Create two switches configured as test 9 and add any four devices as test 7. Connect the two switches together by drawing a line between a port on one to a port on the other.	The line appears and the switches connect.	As expected.	Pass

16.	3	Configure some devices and switches as test 15. Connect two devices to one switch and two to the other.	All four connections appear on the map and the devices connect to the switches.	As expected.	Pass
17.	3	Configure some devices and switches as test 15. Connect one device to another device.	The line does not appear.	As expected.	Pass
18.	4	Configure some devices and switches, including connections between them, as tests 15 and 16. Ensure each device has a protocol enabled which includes an IP address option. From the IP addressing tool pane, press 'Assign addresses'.	Each device is assigned an IP address in the range specified in the IP addressing pane.	As expected.	Pass
19.	7	Configure some devices, switches, and addresses as test 18. Include a device which uses a protocol which has QoS rules. Press the Print button on the map.	The browser print dialogue opens and displays the entire system map without application toolbars.	As expected.	Pass
20.	7	Configure some devices, switches, and addresses as test 19. Select "Quality of Service rules" from the Views tool pane.	A report of the quality of service rules from the protocols included on the map appears. When pressing the Print button, the browser print dialogue opens and displays report without application toolbars.	As expected.	Pass

21.	7	Configure some devices, switches, and addresses as test 19. Select “IP addresses” from the Views tool pane.	A report of the IP addresses in use appears. Toggling the “Show unassigned addresses in the network” checkbox hides and shows unassigned addresses. When pressing the Print button, the browser print dialogue opens and displays report without application toolbars.	As expected.	Pass
22.	7	Configure some devices, switches, and addresses as test 19. Select a switch report from the Views tool pane.	A report listing what is connected to each switch port is displayed. When pressing the Print button, the browser print dialogue opens and displays report without application toolbars.	As expected.	Pass
23.	7	Configure some devices, switches, and addresses as test 19. Select a device report from the Views tool pane.	A report including the switch port the device is connected to, any relevant advice, and the configuration options for all supported protocols is displayed. When pressing the Print button, the browser print dialogue opens and displays report without application toolbars.	As expected.	Pass
24.	8	In the Data tab, under Devices, enter a search term into the box.	Items in the list of devices are filtered down to only those which match the text entered.	As expected.	Pass

25.	8	In the Data tab, under Protocols, enter a search term into the box.	Items in the list of protocols are filtered down to only those which match the text entered.	As expected.	Pass
26.	8	In the Data tab, under Devices, select a device.	In the pane on the right, all data about that device is displayed and matches the content in the database administrator interface.	As expected.	Pass
27.	8	In the Data tab, under Protocols, select a device.	In the pane on the right, all data about that protocol is displayed and matches the content in the database administrator interface.	As expected.	Pass

22.2 Design validation

This section refers to FR4 and FR5. In each case, a dummy device with a dummy protocol is added to the application, then configured to be in the state described in the test description.

ID	Description	Expected output	Actual output	Status
1.	Device is not connected to any switches.	Error flagged in “Check design” tool pane.	As expected.	Pass
2.	Switch is not connected to another switch; there is one switch in the design.	No error flagged.	As expected.	Pass
3.	Switch is not connected to another switch; there are 2 or more switches in the design.	Error flagged in “Check design” tool pane.	As expected.	Pass
4.	Custom validation function indicated a problem.	Error flagged in “Check design” tool pane.	As expected.	Pass
5.	String option is too long.	No further text can be entered in the box.	As expected.	Pass
6.	String option does not match pattern	Error flagged in “Check design” tool pane and inline on the input.	As expected.	Pass

7.	Required string option left blank.	Error flagged in “Check design” tool pane and inline on the input.	As expected.	Pass
8.	String option not globally unique.	Error flagged in “Check design” tool pane.	As expected.	Pass
9.	Number option too small.	Error flagged in “Check design” tool pane and inline on the input.	As expected.	Pass
10.	Number option too big.	Error flagged in “Check design” tool pane and inline on the input.	As expected.	Pass
11.	Required number option left blank.	Error flagged in “Check design” tool pane and inline on the input.	As expected.	Pass
12.	IP address invalid format.	Error flagged in “Check design” tool pane and inline on the input.	As expected.	Pass
13.	IP address outside subnet.	Error flagged in “Check design” tool pane.	As expected.	Pass
14.	IP address not globally unique.	Error flagged in “Check design” tool pane.	As expected.	Pass
15.	Too much data for switch port based on switch speed.	Error flagged in “Check design” tool pane.	As expected.	Pass
16.	Switching capacity at greater than 75%.	Error flagged in “Check design” tool pane.	As expected.	Pass
17.	Switching capacity overloaded.	Error flagged in “Check design” tool pane.	As expected.	Pass
18.	Device needs PoE, connected to non-PoE switch.	Error flagged in “Check design” tool pane.	As expected.	Pass
19.	Device needs PoE, connected to PoE switch but a port which doesn’t support PoE.	Error flagged in “Check design” tool pane.	As expected.	Pass
20.	Device needs PoE, connected to PoE switch.	Error flagged in “Check design” tool pane.	As expected.	Pass

21.	Device needs PoE, connected to PoE port on partial-PoE switch.	No error flagged.	As expected.	Pass
22.	Device needs more PoE than switch supports.	Error flagged in “Check design” tool pane.	As expected.	Pass
23.	Switch delivering more power than PoE budget allows.	Error flagged in “Check design” tool pane.	As expected.	Pass

23 Appendix: Conformance to ISO 9241-210:2019

The below table is an annex from the ISO 9241 standard document used to check conformance with the standard. According to section 9 of the standard, conformance is achieved by “a) satisfying all the requirements; b) identifying applicable recommendations; c) explaining why particular recommendations are not applicable; d) stating whether or not the applicable recommendations have been followed”. These four steps are encapsulated in this table.

Columns “Clause/subclause” and “Requirement or recommendation” are taken directly from the standard. The “Applicability” and “Conformance” columns have been completed with information about this project.

23.1 Table B.10[31] — Checklist for assessing applicability and conformity with ISO 9241-210:2019

Clause/ subclause	Requirement or recommendation	Applicability		Conformance		
		Yes/No	Reason not applicable	Yes	No	Comments
5	Principles of human-centred design					
5.1	Whatever the design process and allocation of responsibilities and roles adopted, a human-centred approach should follow the principles listed [in 5.1].	Yes		✓		The principles align with the project objectives.
5.2	Products, systems and services should be designed to take account of the people who will use them as well as other stakeholder groups including those who can be affected (directly or indirectly) by their use.	Yes		✓		Using personal industry experience and formal questionnaires the stakeholders have been consulted.
5.2	All relevant user and stakeholder groups should be identified. [see also 7.2.2 a)]	Yes		✓		
5.3	User involvement should be active.	No	The project timescale is too short and pool of available users too small for constant involvement, but users can be consulted for requirements and evaluation of the project.			

5.3	The users who are involved should have capabilities, characteristics and experience that reflect the range of users for whom the system is being designed. [see also 7.2.2 b)]	Yes		✓		Recruited from relevant departments and affiliates of the University.
5.4	User-centred evaluation should take place as part of the final acceptance of the product to confirm that requirements have been met.	Yes		✓		
5.5	Iteration should be used to progressively eliminate uncertainty during the development of interactive systems.	No	See 5.3, although an iterative design process is used.			
5.6	The user's experience of previous or other systems and issues such as branding and advertising should also be considered.	Yes		✓		
5.6	Users' strengths, limitations, preferences and expectations should be taken into account when specifying which activities are carried out by the users and which functions are carried out by the technology.	Yes		✓		
5.6	Representative users should generally be involved in decisions related to the allocation of function.	No	Project timescale limits level of stakeholder involvement			
5.6	The human activities resulting from the allocation of function should form a set of tasks that is meaningful as a whole to the user.	Yes		✓		
5.7	Human-centred design teams do not have to be large but the team should be sufficiently diverse to collaborate over design and implementation trade-off decisions at appropriate times.	No	This is an individual project.	✓		Knowledge from a diverse offering of Computer Science modules was used throughout the project and supported by industry experience, along with input from users and the project supervisors.
6	Planning human-centred design					
6.1	Human-centred design shall be planned and integrated into all phases of the product life cycle.	Yes		✓		Primary objective of the project.
6.2	Those responsible for planning the project shall consider the relative importance of human factors/ergonomics in the project by evaluating:					
6.2 a)	how usability relates to the purpose and use of the product, system or service	Yes		✓		
6.2 b)	the levels of the various types of risk that can result from poor usability	Yes		✓		
6.2 c)	the nature of the development environment	Yes		✓		
6.3	The planning of human-centred design shall include:					

6.3 a)	identifying appropriate methods and resources for the activities described in Clause 7	Yes		✓		
6.3 b)	defining procedures for integrating these activities and their outputs with other system development activities	Yes		✓		
6.3 c)	identifying the individuals and the organization(s) responsible for the human-centred design activities and the range of skills and viewpoints they provide	Yes		✓		Individual project so author solely responsible.
6.3 d)	developing effective procedures for establishing feedback and communication on human-centred design activities as they affect other design activities and “trade-offs”, and methods for documenting outputs from these activities	Yes		✓		Individual project so communication limited, but progress reported regularly to supervisor and recorded for inclusion in this report.
6.3 e)	agreeing on appropriate milestones for human-centred activities that are integrated into the overall design and development process	Yes		✓		Release of user evaluation questionnaires was included in project timeline.
6.3 f)	agreeing on suitable timescales to allow iteration, use of feedback and possible design changes to be incorporated into the project schedule	Yes			✓	External deadline limits the ability to incorporate user evaluation into the artefact, but evaluation can be used to inform future work.
6.4	The plan for human-centred design shall form part of the overall system development project plan.	Yes		✓		
6.4	To ensure that it is followed through and implemented effectively, the plan for human-centred design should be subject to the same project disciplines (e.g. responsibilities, change control) as other key activities.	Yes		✓		
6.5	Project planning shall allocate time and resources to the human-centred activities.	Yes		✓		
6.5	[The plan] shall include time for iteration and the incorporation of user feedback, and for evaluating whether the design solution satisfies the user requirements.	Yes			✓	External deadline limits the ability to incorporate user evaluation into the artefact, but evaluation can be used to inform future work.
6.5	Additional time should be allocated to communication among design team participants and to reconciling potential conflicts and trade-offs that involve human–system issues.	No	External deadline imposed restricting available time.			
6.5	Human-centred design activities should start at the earliest stage of the project.	Yes		✓		

6.5	The human-centred design aspects of the project plan should be reviewed throughout the life of the project.	Yes		✓		
7	Human-centred design activities					
7.1	There are four linked human-centred design activities that shall take place during the design of any interactive system:					
7.1 a)	Understanding and specifying the context of use	Yes		✓		
7.1 b)	Specifying the user requirements	Yes		✓		
7.1 c)	Producing design solutions	Yes		✓		
7.1 d)	Evaluating the design	Yes		✓		
7.1	Human-centred design activities should be incorporated in the organization's design approach(es) for projects.	Yes		✓		
7.2.2	The context of use description shall include the following:	Yes				
7.2.2 a)	Relevant groups shall be identified and their relationship with the proposed development described in terms of key goals and constraints.	Yes		✓		
7.2.2 b)	Relevant characteristics of the users shall be identified.	Yes		✓		
7.2.2 b)	If necessary, the characteristics of different types of users should be defined.	Yes		✓		
7.2.2 b)	In order to achieve accessibility, products, systems and services should be designed to be used by people with the widest range of capabilities in intended user populations.	Yes		✓		This has not been fully tested.
7.2.2 c)	The characteristics of tasks that can influence usability and accessibility shall be described.	Yes		✓		
7.2.2 c)	Any potential adverse consequences for health and safety should be identified.	Yes		✓		
7.2.2 c)	If there is a risk that the task can be completed incorrectly, this should be identified.	Yes		✓		
7.2.2 c)	Tasks should not be described solely in terms of the functions or features provided by a product or system.	Yes		✓		
7.2.2 d)	The technical environment, including the hardware, software and materials, shall be identified.	Yes		✓		Included in non-functional requirements.

7.2.2 d)	The relevant characteristics of the physical, social , organizational and cultural environment shall be described.	Yes		✓		Included in personas.
7.2.3	The context of use of the system should be described in sufficient detail to support the requirements, design and evaluation activities.	Yes		✓		
7.2.4	The intended context of use should be specified as part of the user requirements specification to clearly identify the conditions under which the requirements apply.	Yes		✓		
7.3.1	Identifying user needs and specifying the functional and other requirements for the product or system shall be extended to create an explicit statement of user requirements in relation to the intended context of use and the business objectives of the system.	Yes		✓		
7.3.1	If it is known that the proposed interactive system will affect organizational practice, the development process should involve organizational stakeholders in the design process with the aim of optimizing both the organizational and technical systems.	No	Not relevant.			
7.3.2	User and other stakeholder needs should be identified, taking account of the context of use.	Yes		✓		Requirements survey.
7.3.2	User and other stakeholder needs should include that which users need to achieve (rather than how to achieve it) and any constraints imposed by the context of use.	Yes		✓		
7.3.3	The specification of user requirements shall include:					
7.3.3 a)	the intended context of use	Yes		✓		
7.3.3 b)	requirements derived from user needs and the context of use	Yes		✓		
7.3.3 c)	requirements arising from relevant ergonomics and user interface knowledge, standards and guidelines	Yes		✓		Nielsen's heuristics.
7.3.3 d)	usability requirements and objectives including measurable usability performance and satisfaction criteria in specific contexts of use	Yes				Partial conformance: Users will self-rate their confidence and happiness working with the tool compared to not, but ability to measure usability performance limited by timescale.
7.3.3 e)	requirements derived from organizational requirements that directly affect the user	No	Not relevant.			

7.3.4	Potential conflicts between user requirements should be resolved.	Yes		✓		
7.3.4	The rationales, the factors and the weighting of human–system issues for use in any trade-offs should be documented so that they can be understood in the future.	Yes		✓		
7.3.5	The user requirements specification should be:					
7.3.5 a)	stated in terms that permit subsequent testing	Yes		✓		
7.3.5 b)	verified by the relevant stakeholders	No	Project timescale limits level of stakeholder involvement.			
7.3.5 c)	internally consistent	Yes		✓		
7.3.5 d)	updated as necessary during the life of the project	Yes		✓		
7.4.1	Producing design solutions should include the following sub-activities:					
7.4.1 a)	designing user tasks, user-system interaction and user interface to meet the user requirements, taking into consideration the overall user experience	Yes		✓		As described in report.
7.4.1 b)	making the design solutions more concrete	Yes		✓		Through prototyping.
7.4.1 c)	altering the design solutions in response to user-centred evaluation and feedback	No	Project timescale limits level of stakeholder involvement, although recommendations made in evaluation.			
7.4.1 d)	communicating the design solutions to those responsible for their implementation	No	Not relevant: individual project.			
7.4.2.1	The following principles (taken from ISO 9241-110) should be taken into account when designing interactive systems:					
7.4.2.1 a)	suitability for the task	Yes		✓		
7.4.2.1 b)	self-descriptiveness	Yes		✓		
7.4.2.1 c)	conformity with user expectations	Yes		✓		
7.4.2.1 d)	suitability for learning	Yes		✓		
7.4.2.1 e)	controllability	Yes		✓		

7.4.2.1 f)	error tolerance	Yes		✓		
7.4.2.1 g)	suitability for individualization	Yes		✓		
7.4.2.2	Designing the interaction should include:					
7.4.2.2 a)	making high-level decisions	Yes		✓		
7.4.2.2 b)	identifying tasks and sub-tasks	Yes		✓		
7.4.2.2 c)	allocating tasks and sub-tasks to user and other parts of system	Yes		✓		
7.4.2.2 d)	identifying the interaction objects required for the completion of the tasks	Yes		✓		
7.4.2.2 e)	identifying appropriate dialogue techniques	Yes		✓		
7.4.2.2 f)	designing the sequence and timing (dynamics) of the interaction	Yes		✓		
7.4.2.2 g)	designing the information architecture of the user interface of an interactive system to allow efficient access to interaction objects	Yes		✓		
7.4.2.3	Ergonomics and user interface knowledge, standards and guidelines should be used to inform the design of both hardware and software of the user interface.	Yes		✓		
7.4.3	The level of detail and realism [of prototypes] should be appropriate to the issues that need to be investigated.	Yes		✓		
7.4.4	Feedback from evaluation should be used to improve and refine the system.	No	Project timescale limits level of stakeholder involvement, although recommendations made in evaluation.			
7.4.4	The costs and benefits of proposed changes should be evaluated and used to inform decisions about what will be modified.	Yes		✓		
7.4.5	There should be some sustained channel of communication between those responsible for human-centred design and other members of the project team.	No	Not relevant: individual project.			
7.4.5	When design solutions are communicated, they should be accompanied by an explanation and justification of the design decisions, especially where trade-offs are necessary	Yes		✓		

7.4.5	The communication [of details of the design] should take account of the constraints imposed by the project and the project team's knowledge and understanding about ergonomics and user interface design.	Yes		✓		
7.5.1	User-centred evaluation (evaluation based on the user's perspective) is a required activity in human-centred design.	Yes		✓		
7.5.1	Even at the earliest stages in the project, design concepts should be evaluated to obtain a better understanding of user needs.	Yes		✓		
7.5.1	If user-based testing is not practical or cost-effective at a particular stage of a project, design solutions should be evaluated in other ways.	Yes		✓		Tightly-coupled design and implementation allowing prototypes to be tested by the developer using industry knowledge.
7.5.2	User-centred evaluation should involve:					
7.5.2 a)	allocating resources both for obtaining early feedback to improve the product, and later for determining if requirements have been satisfied	No	Project timescale limits level of stakeholder involvement, although recommendations made in evaluation.			
7.5.2 b)	planning the user-centred evaluation so that it fits the project schedule	Yes	User evaluation included in project evaluation but not implemented due to limitation of schedule.	✓		
7.5.2 c)	carrying out sufficiently comprehensive testing to give meaningful results for the system as a whole	Yes		✓		
7.5.2 d)	analysing the results, prioritizing issues and proposing solutions	Yes		✓		
7.5.2 e)	communicating the solutions appropriately so that they can be used effectively by the design team	Yes		✓		
7.5.3	To obtain valid results, the evaluation should be carried out by experienced evaluators.	Yes	Evaluation guided by experienced project supervisor.	✓		
7.5.3	To obtain valid results, the evaluation should use appropriate methods.	Yes		✓		
7.5.3	The extent of the latter (summative) evaluation should depend on the extent of the risks associated with not meeting requirements.	Yes		✓		

7.5.4	When prototypes are being tested, users should carry out tasks using the prototype rather than just be shown demonstrations or a preview of the design.	Yes		✓		Users asked to build a system of their own accord.
7.5.6	A human-centred design process should include long-term monitoring of the use of the product, system or service.	No	Project is short-term.			
7.5.6	Criteria and measurements [for long-term monitoring] should be sensitive enough to identify system failure, or system problems, as early as possible.	No	Project is short-term.			