DIS Packet Efficiency Using Bundling

William Oliver, Peter Ross and Peter Ryan

Defence Science & Technology Group
506 Lorimer St, Fishermans Bend, Victoria 3207, Australia
william.oliver@dst.defence.gov.au
peter.ross@dst.defence.gov.au
peter.ryan@dst.defence.gov.au

Keywords:

Distributed Interactive Simulation (DIS), Protocol Data Unit (PDU), DIS Communication Services & Profiles, Bundling

Abstract. IEEE 1278.1 Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs) are typically transmitted within UDP packets over Ethernet on the Local Area Network (LAN). The proposals for DIS version 8 include implementing simpler smaller PDUs which will result in more PDUs being sent on the network. Bundling of PDUs, introduced in IEEE 1278.2 Communication Services and Profiles, has been proposed as a way of increasing the efficiency of transmitting a larger number of smaller PDUs. Bundling concatenates several PDUs into a single datagram that can be transmitted and relayed through the network in one operation. The effect of bundling has been examined using PDUs logged from large scale international training exercises. This logged PDU data was used as the input to a model to quantify the effect of bundling on current DIS exercises. Results show that bundling would have provided considerable network efficiencies for these exercises with considerably smaller bandwidth required for the same PDU throughput. These findings provide some background data that can inform the DIS v8 discussions on PDU bundling.

1 Introduction

In preparation for DIS version 8, where it is expected that PDU bundling will be encouraged, it is useful to present some data relating to bundling and its possible impacts. PDU bundling is described in [1], as:

> *Network efficiency may be enhanced with PDU bundling. This is the process of concatenating two or more PDUs into a single network datagram so that they may be transmitted and relayed through the network in a single operation.*

The requirements and further details are given in [2] and [3]. Each packet sent by the operating system will have some part of the packet *not* being DIS data, in the form of network packet headers. If we bundle PDUs, *more* DIS data is going into each packet, and there is less overhead per unit of DIS data. This paper will examine the improvements in network utilisation that bundling can provide.

It is important to note what will *not* be examined. The nature of voice communications means that delaying packets to bundle will have negative consequences. Both the Transmitter and Signal PDUs from voice communications are excluded from the analysis. Compression [4] and the effects of bundling traffic over specific Wide Area Network (WAN) technologies [5] are not examined. The network model used in this analysis only considers Internet Protocol (IP) version 4 on an Ethernet Local Area Network (LAN). Finally our model uses meta-data logged from real events which are DIS version 6 or 7 PDUs. No analysis of the benefit of the proposed DIS version 8 structures is done.

2 Packet Efficiency

In order to calculate some metrics about packet efficiency we have to define some parameters. *Packet efficiency* (e) is defined as the *useful* payload divided by the packet length (l_p) or:

$$e = \frac{\text{useful payload}}{\text{packet length}} \tag{1}$$

The *packet overhead* (o) is defined as the amount of data we have to send as part of the packet format that does not include payload, that is:

$$o = \text{packet length} - \text{useful payload} \tag{2}$$

To get an idea of how much of the network packet is *useful* data we need to understand the packet formats. There are too many technologies used for the WANs so only the most common LAN technology is considered: DIS transmitted via User Datagram Protocol (UDP) over IP over Ethernet. Figure 1 shows how a DIS PDU is packaged into an Ethernet frame. Note that the whole DIS packet (including header) is considered the *useful* payload.

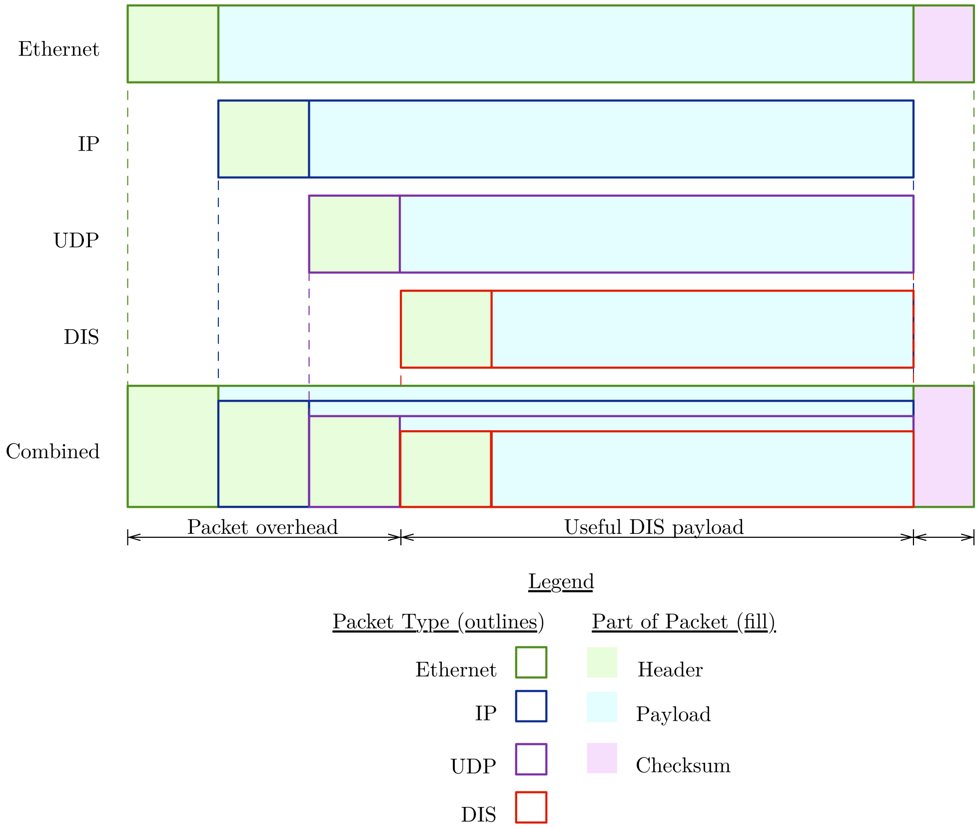


Figure 1: Encapsulation of a DIS PDU inside a UDP packet, IP packet and Ethernet frame.

2.1 Ethernet

An Ethernet frame is formatted in accordance with the rules in [6]. There are two variable length fields, the 802.1Q (Virtual LAN (VLAN)) field, which is 0 or 4 octets, and the payload. All other fields combined are 38 octets in length.

Let the length of the Ethernet payload be l_{e_p}, and the length of the VLAN header field be l_{e_v}, then the length of the Ethernet frame (l_e) is:

$$l_e = 38 + l_{e_v} + l_{e_p} \tag{3}$$

where

$$l_{e_v} = \begin{cases} 0\,(\text{min}) \\ 4\,(\text{max}) \end{cases} \tag{4}$$

using the definition of packet overhead from (2) we have an overhead for Ethernet (o_e) of:

$$o_e = \begin{cases} 38 + l_{e_{v_{\min}}} & = 38 + 0 & = 38 & (\text{min}) \\ 38 + l_{e_{v_{\max}}} & = 38 + 4 & = 42 & (\text{max}) \end{cases} \tag{5}$$

giving

$$l_e = \begin{cases} o_{e_{\min}} + l_{e_p} & (\text{min}) \\ o_{e_{\max}} + l_{e_p} & (\text{max}) \end{cases} \tag{6}$$

That is, the *packet overhead* for Ethernet is a minimum of 38 octets and a maximum of 42 octets.

2.2 Internet Protocol v4

The format of an IP packet is given in [7]. The IPv4 packet header has 14 fields. The mandatory fields take up 20 octets and the options fields can be up to 40 octets. If we define the length of the options as l_{i_o} and the payload as l_{i_p} we can calculate the length of an IP packet (in octets) as:

$$l_i = 20 + l_{i_o} + l_{i_p} \tag{7}$$

where

$$l_{i_o} = \begin{cases} 0 & (\text{min}) \\ 40 & (\text{max}) \end{cases} \tag{8}$$

using the definition of overhead from (2) we have an overhead for IP (o_i) of:

$$o_i = \begin{cases} 20 + l_{i_{o_{\min}}} & = 20 + 0 & = 20 & (\text{min}) \\ 20 + l_{i_{o_{\max}}} & = 20 + 40 & = 60 & (\text{max}) \end{cases} \tag{9}$$

giving

$$l_i = \begin{cases} o_{i_{\min}} + l_{i_p} & (\text{min}) \\ o_{i_{\max}} + l_{i_p} & (\text{max}) \end{cases} \tag{10}$$

2.3 UDP

The format of a UDP packet is given in [8]. UDP packets have an 8 octet header, so the overhead (o_u) is eight octets.

$$o_u = 8 \tag{11}$$

If the length of the UDP payload is l_{u_p} then the length (l_u) is given as

$$l_u = o_u + l_{u_p} \tag{12}$$

2.4 Putting it together

In (2) the overhead is defined to be anything that is not payload. Figure 1 shows how the Ethernet frame carries the IP packet as its payload, the IP packet carries the the UDP packet as its payload, and the UDP packet carries the DIS PDU as its payload, i.e. $l_{e_p} = l_i$, $l_{i_p} = l_u$, and $l_{u_p} = l_{dis}$. Substituting into (12), (10) and (6)

$$l_e = \begin{cases} o_{e_{\min}} + (o_{i_{\min}} + (o_u + l_{dis})) & (\text{min}) \\ o_{e_{\max}} + (o_{i_{\max}} + (o_u + l_{dis})) & (\text{max}) \end{cases} \tag{13}$$

substituting the overheads from (5), (9) and (11) gives

$$l_e = \begin{cases} 38 + (20 + (8 + l_{dis})) & = 66 + l_{dis} \quad \text{(min)} \\ 42 + (60 + (8 + l_{dis})) & = 110 + l_{dis} \quad \text{(max)} \end{cases} \tag{14}$$

Substituting (14) into the definition of efficiency in (1)

$$e = \begin{cases} \frac{l_{dis}}{110+l_{dis}} & \text{(min)} \\ \frac{l_{dis}}{66+l_{dis}} & \text{(max)} \end{cases} \tag{15}$$

Figure 2 plots efficiency as a function of the payload length, as described by equation (15). Shown are payload sizes for an entity Entity state PDU with no articulated parts (144 octets) and the standard Maximum Transmission Unit (MTU) for Ethernet (1500). It can be seen that efficiency increases considerably with payload size, from a value for the Entity State PDU (ES PDU) of 68.6% to a maximum possible (over Ethernet) of 95.8%.

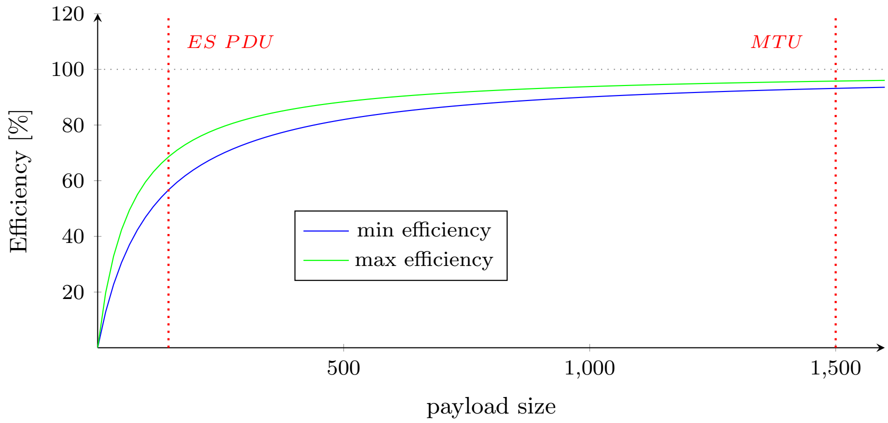


Figure 2: Efficiency versus payload size. MTU represents the largest payload possible in a standard Ethernet frame.

2.5 Choosing Bundle Sizes and Bundle Delays

To study the effect of bundling four values for the bundle size were chosen. The maximum size of a bundle is defined in [3] to be `MAX_PDU_SIZE_OCTETS`, which is 8192. The IP protocol [7] states:

> *All hosts must be prepared to accept datagrams of up to 576 octets . . .*

Assuming the smallest packet overhead for IP and UDP (equations 9 and 11), this leaves 548 octets for DIS data. This was chosen as the minimum value to use. Two further bundle sizes were chosen, based on *rules of thumb*, that the MTU on the LAN should be either 1400 or 1472 to allow for various network problems encountered in practice.

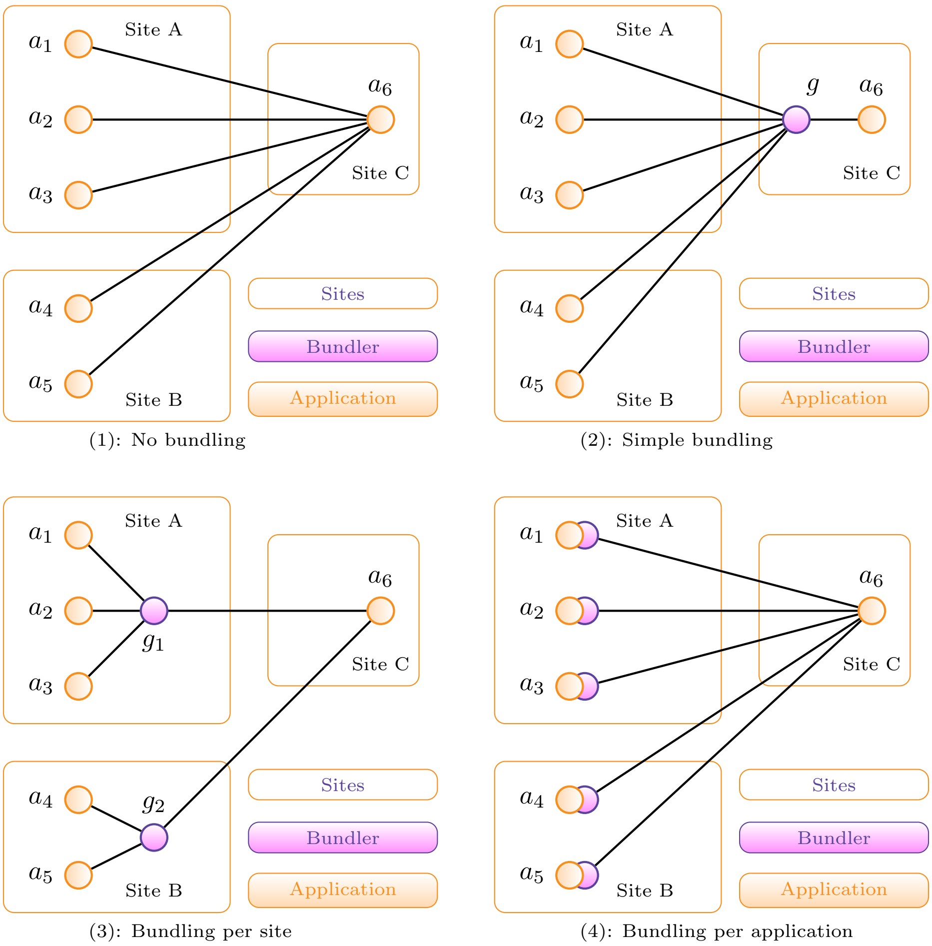
The amount of time to delay transmission of a PDU (referred to as *bundle delay*) is more subjective. [3] states

> *The amount of delay shall be limited to avoid an excess increase in latency.*

Values from 10 to 100 milliseconds were modelled to see how the bundle delay would effect efficiency (in sec. 3.1). The value of 50 milliseconds was then used for further analysis.

2.6 Modelling Bundling

A bundle model was constructed that takes a PDU, determines if it should be added to the current bundle by checking to see if the combined size would exceed the maximum bundle size, or if it has arrived within the



(1): No bundling (2): Simple bundling (3): Bundling per site (4): Bundling per application

Figure 3: Shows the different ways to model bundling. No. 1 is the *unbundled* case. No. 2 shows the simplest bundling model where, from the observer's perspective (a_6), simulation data is bundled at a central location. No. 3 shows the case where each site bundles traffic for all applications at that site, and No. 4 shows the case where each simulation application bundles its own traffic.

bundle delay time. Once the bundle is ready, the networking overhead is added to determine the total size of the transmitted datagram. Within this model there are three things that can be varied, the bundle size, the bundle delay and *where* to model the bundling process. Three cases were identified:

- *simple* or *central*, where all PDUs were included in the bundle model. This models a scenario where all sites send their data to a central point and this bundles the data. This is shown in Figure 3(2).
- *site*, where each site bundles data from all applications at that site. This models the scenario where there is a gateway that bundles outbound traffic from a site. This is shown in Figure 3(3)
- *application*, where each simulation application bundles its own data. This is shown in Figure 3(4)

The model gives us two metrics to assess the benefits of bundling; (a) the amount of data transmitted, and (b) the number of packets transmitted.

3 Real World Data

Data from simulated training exercises was used as input to the model. The data is from a simulation of three hours in duration and had no bundled PDUs. A breakdown of the data is given in Table 1. Plots of the packets

(PDUs) per second and bandwidth utilisation (kB/sec) are given in Figure 4. Voice communications (both the Transmitter and Signal PDUs) are not included as part of our modelling but are shown in the table and figure for illustration.

Table 1: Breakdown of the exercise data.

PDU Type	Number of PDUs		Size (MB)	
Main Dataset				
ES	4095357	(60.89%)	623.98	(70.12%)
Fire	2510	(0.04%)	0.24	(0.03%)
Det.	2467	(0.04%)	0.26	(0.03%)
EE	730693	(10.86%)	97.14	(10.92%)
TX	669173	(9.95%)	74.95	(8.42%)
SIG	680618	(10.12%)	60.57	(6.81%)
IFF	545242	(8.11%)	32.71	(3.68%)
Total	6726060		889.85	
Voice Dataset				
Voice TX	498918	(25.12%)	51.89	(13.65%)
Voice SIG	1481529	(74.58%)	328.00	(86.29%)
Voice RX	5997	(0.30%)	0.22	(0.06%)
Total	1986444		380.11	
Whole exercise				
Total	8712504		1269.96	

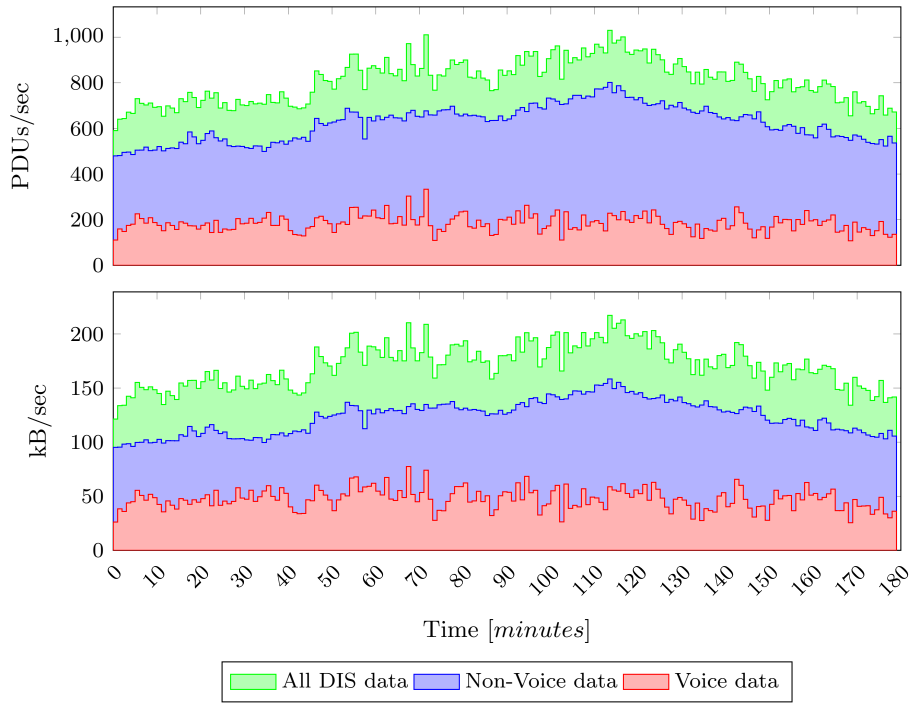


Figure 4: Packets per second and bandwidth utilisation for the duration of the exercises.

3.1 Effects Due to Bundle Delay

Figure 5 shows the improvement in efficiency as a function of the bundle delay. As the bundle size decreases the effect of delay becomes smaller. For our modelling we have chosen 50 milliseconds but any value greater than 20 sees a diminishing return.

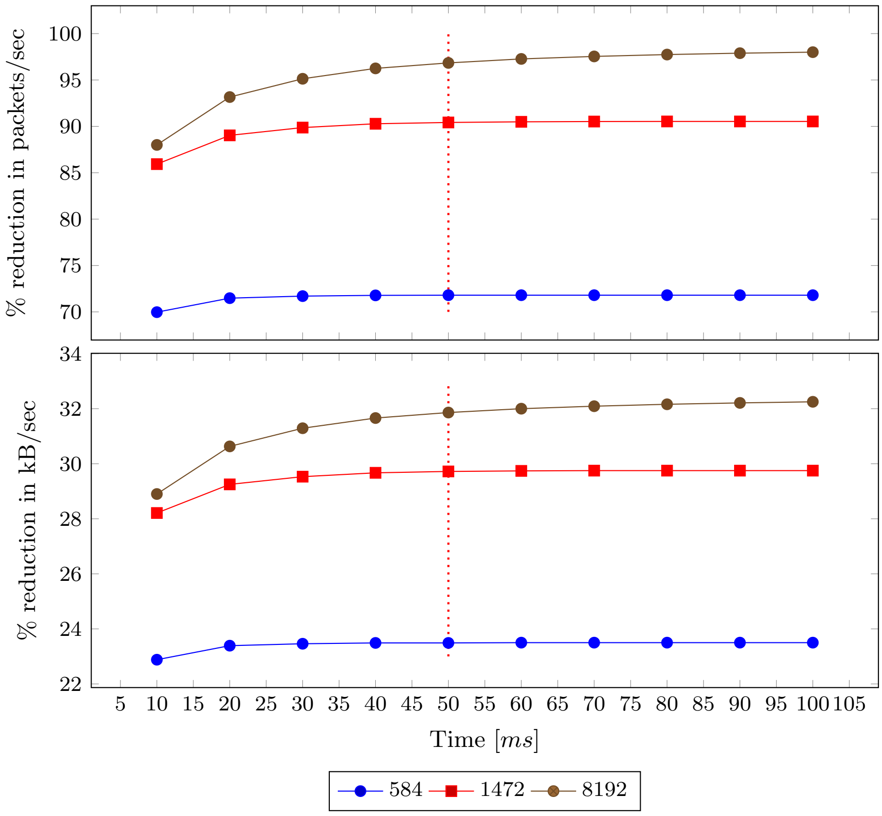


Figure 5: Percentage reduction in packets per second and bandwidth utilisation for three different maximum bundle sizes, as a function of the bundle delay. The vertical dotted line marks 50 ms.

3.2 Reduction in Packet Count and Bandwidth

Bundling provides an enormous reduction in packet count and a significant reduction in bandwidth utilisation due to greater packet efficiency. This can be seen in Table 2.

Table 2: Improvements in size and packet count due to bundling.

	Size (MB)		Number of Packets		
Bundle Size	Sent	Reduction	Sent	Reduction	PDUs/bundle
8192	908.83	31.86%	212277	96.84%	31.69
1472	937.40	29.72%	644047	90.42%	10.44
1400	939.88	29.53%	681503	89.87%	9.87
548	1020.46	23.49%	1897011	71.80%	3.55
unbundled	1333.78	0.00%	6726060	0.00%	1

The reduction (in percentage terms) of the packet count and the amount of data sent, as a function of bundle size, are plotted in Figure 7. These plots also show the difference between the three bundling locations, central bundling, site bundling and application bundling. The greatest increase in efficiency occurs when the bundling occurs in one place (shown in Figure 3(2)) and with a large bundle size.

Finally we plot the data for the duration of the exercise (in Figure 8), clearly showing the reductions in packet count and data transmitted.

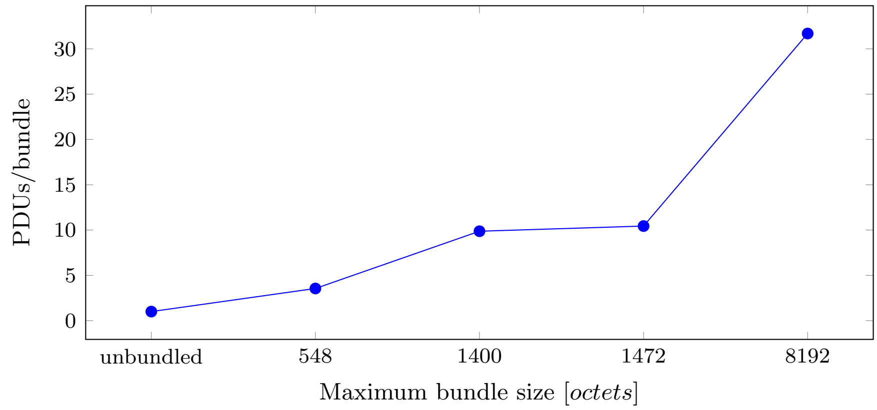


Figure 6: Number of PDUs per bundle as a function of maximum bundle size.

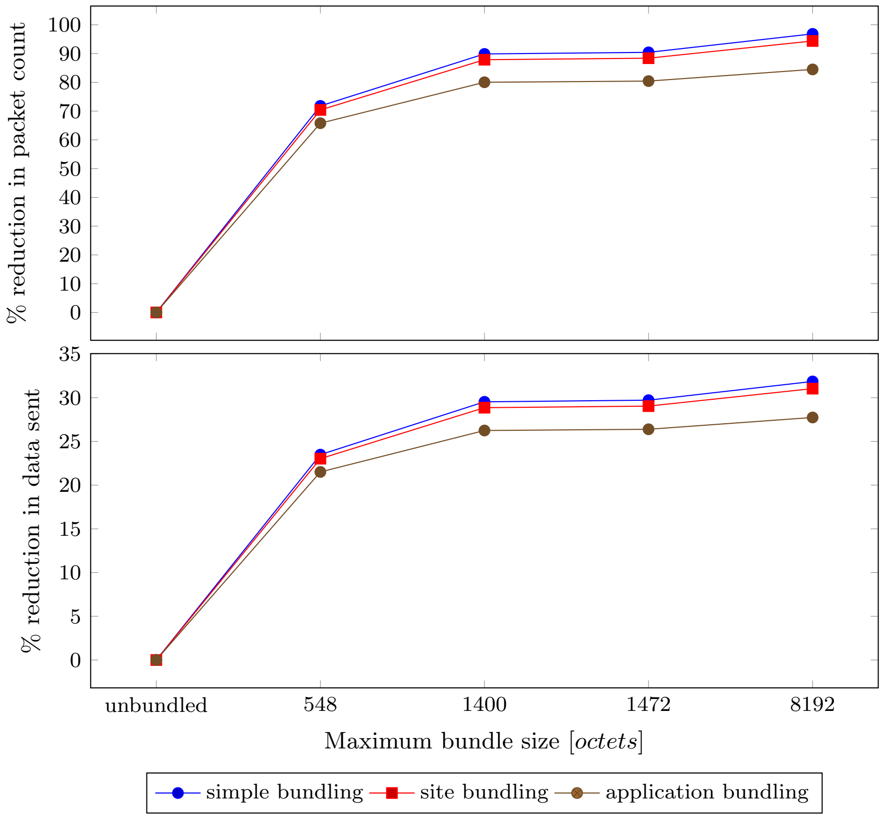


Figure 7: Percentage reduction in the number of packets and amount of data sent, as a function of the maximum bundle size. There are three plots, one for each bundling location (simple/central, site and application).

4 Discussion and Conclusion

PDU bundling is discussed in IEEE 1278.2-2015 as a way to reduce network traffic. Bundling concatenates several PDUs into a single datagram that can be transmitted through the network. The effects of PDU bundling have been studied both from a theoretical consideration of DIS PDU encapsulation into Ethernet IP/UDP packets and also by modelling the effects of bundling on real world simulation exercise data. Only non-voice PDUs were considered. Voice PDUs may not be suitable for bundling since coherence of the voice signal may be impacted by delaying PDUs. Bundle sizes chosen for the analysis were based on common networking MTUs (548, 1400, 1472, and 8192 octets).

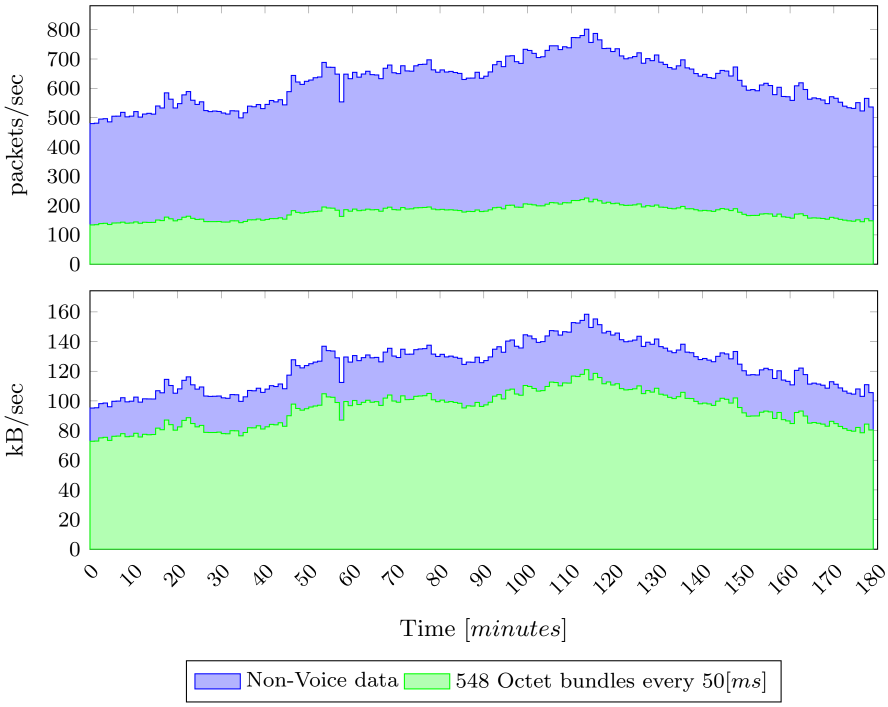


Figure 8: Packets per second and bandwidth for the duration of the exercise.

The effect of varying the bundle delay time was also investigated. However, the published IEEE standard does not address the issue of how long to wait for an incoming PDU before bundling. Investigations were carried out for delay times ranging from 10 to 100 ms. The results showed that a value of 50 ms may be optimal: higher values only made a marginal difference. These findings could be included as guidance for bundling in the proposed DIS v8 standard.

Bundling was shown to provide a large reduction in packet count and significant reduction in bandwidth utilisation. For the sample exercise data set the data transmitted can be reduced by 23 – 32%, while the number of packets can be reduced by 72 – 97% depending on the bundle size. These results can inform the DIS v8 discussions on PDU bundling as a means to reduce network traffic and increase efficiency.

5 References

[1] IEEE Standard for Distributed Interactive Simulation — Application Protocols. *IEEE Std 1278.1-2012*, pages 1–747, Dec 2012.

[2] R. E. Murray. Proposed PDU Bundling Specification for IEEE 1278.2 Rev A. In *Fall Simulation Interoperability Workshop*, Orlando, Florida, September 2009. SISO.

[3] IEEE Standard for Distributed Interactive Simulation (DIS) — Communication Services and Profiles. *IEEE Std 1278.2-2015 (Revision of IEEE Std 1278.2-1995)*, pages 1–42, November 2015.

[4] Lance Call, Scott Swigert, Mitchell Zamba, and David Noah. Compressed DIS. In *The Interservice/Industry Training, Simulation & Education Conference*, pages 2627–2638, Orlando, FL, November 2017. The Interservice/Industry Training, Simulation & Education Conference.

[5] Juan J. Vargas, Ronald F. DeMara, Michael Georgiopoulos, Avelino J. Gonzalez, and Henry Marshall. PDU Bundling and Replication for Reduction of Distributed Simulation Communication Traffic. *Journal of Defense Modeling and Simulation*, 1(3):167–185, August 2004.

[6] IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (csma/cd) Access Method and Physical Layer Specifications - Section One. *IEEE Std 802.3-2008 (Revision of IEEE Std 802.3-2005)*, pages c1 –597, Dec 2008.

[7] J. Postel. Internet Protocol. RFC 791, IETF, September 1981. Updated by RFC 1349.

[8] J. Postel. User Datagram Protocol. RFC 768, IETF, August 1980.

Author Biographies

WILLIAM OLIVER is a senior researcher in the Defence Science & Technology (DST) Group's Aerospace Division, specialising in interoperability issues and analysis techniques for advanced distributed simulation. Prior to joining DST, he developed software for flight simulators and simulated maintenance trainers, and studied Engineering and Mathematics.

PETER ROSS graduated from RMIT University in 2001 with a Bachelor of Applied Science, majoring in computer science. He joined DST's Aerospace Division in 2003, where he has undertaken a role in evaluating the use of advanced distributed simulation for collective training.

DR. PETER RYAN is an Honorary Research Fellow in DST Group's Aerospace Division. He has a 30 year background in the modelling and simulation of military operations. His main research interests include Advanced Distributed Simulation, real time simulation, synthetic environments, and their potential to provide enhanced training solutions for the Australian Defence Force.