A Parametric State Space for the Analysis of the Infinite Class of Stop-and-Wait Protocols*

Guy Edward Gallasch and Jonathan Billington

Computer Systems Engineering Centre, University of South Australia,
Mawson Lakes Campus, SA 5095, Australia
guy.gallasch@postgrads.unisa.edu.au
jonathan.billington@unisa.edu.au

Abstract. The Stop-and-Wait protocol (SWP) has two (unbounded) parameters: the maximum sequence number (MaxSeqNo) and the maximum number of retransmissions (MaxRetrans). This paper presents an algebraic method for analysis of the SWP for all possible values of these parameters. Model checking such a system requires considering an infinite family of models, one for each combination of parameter values, and thus an infinite family of state spaces (reachability graphs). These reachability graphs are represented symbolically by a set of algebraic formulas in MaxSeqNo and MaxRetrans. This result is significant as it provides a complete characterisation of the infinite set of reachability graphs of our SWP model in both parameters, allowing properties to be verified for the infinite class. Verification of a number of properties is described.

Keywords: Stop and Wait Protocols, Infinite Families of Systems, Parametric Reachability Graphs, Coloured Petri Nets.

1 Introduction

Stop-and-Wait is an elementary and well-known form of flow control [20,22] used by communication protocols to prevent buffer overflow in the receiver. In practice Stop-and-Wait is often used with checksums to detect transmission errors and a timeout/retransmission scheme using sequence numbers, such as Automatic Repeat ReQuest [22], for error recovery.

The Stop-and-Wait mechanism forms the basis of many practical data transfer protocols, such as the Internet's Transmission Control Protocol (TCP) [19]. An understanding of how these mechanisms work and how they may fail is thus useful for the verification of more complex protocols like TCP. These protocols have a number of parameters, such as the maximum sequence number (MaxSeqNo) or the maximum number of retransmissions (MaxRetrans). The value of these parameters may vary depending on the application (e.g. TCP has a 32 bit sequence number, whereas others may use a 3 bit sequence number). It is thus of interest to verify these protocols for all values of these parameters.

* Partially supported by an Australian Research Council (ARC) Discovery Grant, DP0559927, and a University of South Australia Divisional Small Grant, SP04.

A. Valmari (Ed.): SPIN 2006, LNCS 3925, pp. 201–218, 2006.

Petri nets have proven to be a suitable formal method for protocol verification [2,3,6,15,17]. A Coloured Petri net (CPN) [14,16] model of the SWP, parameterised by MaxSeqNo and MaxRetrans, was developed and analysed in [4,5,6] following the *protocol verification methodology* presented in [6]. Because the model parameters are unbounded there is an infinite set of CPN models to verify, and state explosion [23] prevents analysis for all but small parameter values. Thus we were motivated to find a way to verify the SWP for any finite (but unbounded) value of the parameters. In [12] we presented a novel technique of representing the reachability graphs (RGs) of the SWP CPN symbolically in the MaxSeqNo parameter (with MaxRetrans=0) using algebraic expressions, and verified a number of properties directly from the expressions, including language equivalence to the service, for all values of the unbounded MaxSeqNo parameter.

Related work on symbolic verification considers only the MaxRetrans parameter. Abdulla et al [1] verify the Alternating Bit Protocol (ABP) (MaxSeqNo=1) with unbounded retransmissions and a variant called the Bounded Retransmission Protocol in which MaxRetrans is modelled nondeterministically. In [7,8] we used a tool called FAST (Fast Acceleration of Symbolic Transition Systems) [9] to model the SWP and analyse it symbolically. We were successful when MaxRetrans was an unbounded parameter with MaxSeqNo fixed to small values (1 to 5), and when MaxSeqNo was an unbounded parameter but with MaxRetrans fixed to 0. FAST did not return a result when both MaxSeqNo and MaxRetrans were unbounded parameters. In [24] a variant of the ABP with arbitrary MaxRetrans and operating over channels with a capacity of one message only, was verified using Valmari's Chaos-Free-Failures-Divergences (CFFD) equivalence. In contrast, our model operates over unbounded lossy ordered channels (similar to [1]) and explicitly considers any maximum sequence number (not just the alternating bit) and any maximum number of retransmissions.

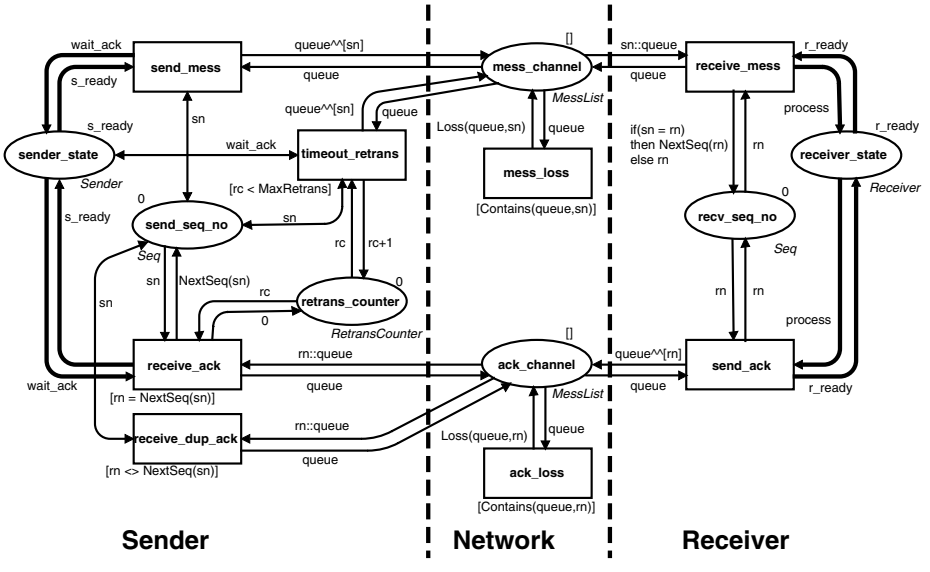
In this paper, the work in [12] is significantly extended by obtaining algebraic expressions for the infinite set of RGs of the SWP operating over an ordered medium over both the MaxSeqNo and MaxRetrans parameters. A sketch of the proof of correctness is given, details of which can be found in [11]. The contribution of this paper is threefold. Firstly, we further develop the novel algebraic representation method from [12]. Secondly, we provide the aforementioned symbolic representation. Inclusion of the MaxRetrans parameter represents a substantial increase in the complexity of the algebraic expressions. This can be gauged by the size of the RG, which grows linearly in MaxSeqNo but quartically in MaxRetrans [10, 12]. Previous work dealt with the linear growth in MaxSeqNo only, whereas this paper also deals with the quartic growth in MaxRetrans. Thirdly, we sketch the verification of a number of properties directly from the algebraic expressions. The authors are not aware of any previous attempts to obtain an explicit algebraic representation for the family of RGs for arbitrary unbounded values of the MaxSeqNo and MaxRetrans parameters for the class of Stop-and-Wait protocols.

The rest of this paper is organised as follows. Section 2 presents our parametric SWP CPN model. The necessary notational constructs and lemmas regarding model behaviour are presented in Section 3. The parametric algebraic expressions

of the RG are presented in Section 4, followed by a description of the verification of a number of properties. Conclusions and future work are presented in Section 5. Familiarity with basic CPN concepts and terminology is assumed. For introductions to CPNs the reader is referred to [14, 16].

2 The Stop-and-Wait Protocol CPN Model

The SWP is modelled using Coloured Petri nets [14, 16], a form of Petri net in which tokens are arbitrarily complex data values. The CPN diagram is shown in




```
val MaxRetrans = 0;
val MaxSeqNo = 1;

color Sender = with s_ready | wait_ack;
color Receiver = with r_ready | process;
color Seq = int with 0..MaxSeqNo;
color RetransCounter = int with 0..MaxRetrans;
color Message = Seq;
color MessList = list Message;

var sn,rn : Seq;
var rc    : RetransCounter;
var queue : MessList;

fun NextSeq(n) = if(n = MaxSeqNo) then 0 else n+1;
fun Contains([],sn) = false
  | Contains(m::queue,sn) = if(sn=m) then true else Contains(queue,sn);
fun Loss(m::queue,sn) = if(sn=m) then queue else m::Loss(queue,sn);
```

Fig. 1. A CPN of the Stop-and-Wait Protocol operating over an in-order medium

Fig. 1 along with all the declarations used in the inscriptions of the CPN diagram. The inscription language is a variant of Standard ML [21]. The two parameters MaxRetrans and MaxSeqNo can be seen at the top of the declarations in Fig. 1. This model is the same as the one presented in [12], with the exception of loss. This change is motivated and described below. (The focus of this paper is not the modelling of SWP with CPNs. A detailed description of the model is given in [12] and hence omitted here.)

The channels are modelled as lists manipulated by the arc inscriptions as First-In-First-Out (FIFO) queues in places mess_channel and ack_channel. Transitions mess_loss and ack_loss model loss, both in the network (buffer overflow in a router) and by discarding messages and acknowledgements with transmission errors (checksum failures). Unlike the model in [12], loss can occur anywhere in the message and acknowledgement queues, not just from the head. This is done via nondeterministic binding of variables sn and rn and the function Contains in the guard of each loss transition, to ensure that sn and rn are only bound to values that are present in the channels. The removal of the message is via function Loss in the arc inscriptions.

Motivation is provided by it being a more general model, suited to the TCP environment, where loss can occur anywhere in the network due to e.g. router congestion, in addition to loss caused by detection of errors. It turns out that this model of loss is easier to formalise in the algebraic expressions in Section 4.

3 Notation and Model Properties

This section introduces notation and proves a number of properties of the SWP CPN model required for the proof of correctness of the algebraic formula presented in Section 4.

3.1 Marking and Arc Notation

We begin by defining the RG of a CPN. In CPN terminology, a reachability graph is often called an *occurrence graph* (OG).

Definition 1 (Reachability Graph). *The* **OG** *of a CPN with initial marking, M_0, and a set of binding elements, BE, is a labelled directed graph $OG = (V,A)$ where*

1. *$V = [M_0\rangle$ is the set of reachable markings of the CPN; and*
2. *$A = \{(M,(t,b),M') \in V \times BE \times V | M[(t,b)\rangle M'\}$ is the set of labelled directed arcs, where $M[(t,b)\rangle M'$ denotes that the marking of the CPN changes from M to M' on the occurrence of transition t with binding b, $(t,b) \in BE$.*

The parameterised CPN and its RG are denoted by $CPN_{(MS,MR)}$ and $OG_{(MS,MR)}$ given by the following definition:

Definition 2 (Parameterised CPN and Reachability Graph). *For* $MS \in \mathbb{N}^+$ *and* $MR \in \mathbb{N}, CPN_{(MS,MR)}$ *is defined as the Stop-and-Wait Protocol CPN of Fig. 1 with* MaxSeqNo $= MS$ *and* MaxRetrans $= MR$*. The reachability graph of* $CPN_{(MS,MR)}$ *is denoted by* $OG_{(MS,MR)} = (V_{(MS,MR)}, A_{(MS,MR)})$*.*

In order for the notation for markings and arcs defined below to be correct, we must prove that each place in the SWP CPN with initial marking M_0 as illustrated in Fig. 1 always contains exactly one token.

Lemma 1. *For all reachable markings of* $CPN_{(MS,MR)}$ *and all allowable values of* MS *and* MR*, each place in the CPN diagram contains exactly one token, i.e.* $\forall MS \in \mathbb{N}^+, \forall MR \in \mathbb{N}, \forall M \in V_{(MS,MR)}$, $|M(\textit{sender\_state})| = |M(\textit{receiver\_state})| = |M(\textit{retrans\_counter})| = |M(\textit{mess\_channel})| = |M(\textit{ack\_channel})| = |M(\textit{send\_seq\_no})| = |M(\textit{recv\_seq\_no})| = 1$*.*

Sketch of Proof. Proof is by direct inspection of Fig. 1. Consider the recv_seq_no place. M_0(recv_seq_no) = 1‘0 and so $|M_0(\text{recv\_seq\_no})| = 1$. The marking of this place can only be changed by transitions receive_mess and send_ack. The occurrence of these transitions either replaces one value by another (the receive_mess transition when $sn{=}rn$) or does not change the marking (the receive_mess transition when $sn{\neq}rn$ and the send_ack transition). The value of MS may affect the token value (via function NextSeq) but it does not affect the number of tokens removed or added (always 1). Hence $|M(\text{recv\_seq\_no})| = 1$ for all markings. Similar arguments reveal that this property also holds for the remaining 6 places. □

The following function converts a singleton multiset into its basis element:

Definition 3 (Singleton Multiset to Colour). *Let* S_{MS_1} *be the set of all singleton multisets over a basis set* $S : S_{MS_1} = \{\{(s, 1)\}|s \in S\}$*. A function that converts a singleton multiset to its basis element is given by* $f_c : S_{MS_1} \rightarrow S$*, where* $f_c(\{(s, 1)\}) = s$*.*

In addition, the following notational conventions are used throughout this paper:

- $M[t\rangle$ is used as shorthand to represent that transition t is enabled by marking M for some binding of variables b, such that $M[(t, b)\rangle, (t, b) \in BE$;
- $|f_c(M(p))|$ is the length of the list on places $p \in \{\text{mess\_channel}, \text{ack\_channel}\}$;
- i^j represents j repetitions of the message (or acknowledgement) with sequence number i in the message (or acknowledgement) channel;
- $\oplus_{MS}$ represents modulo $MS + 1$ addition; and
- $\ominus_{MS}$ represents modulo $MS + 1$ subtraction.

The markings of our SWP CPN can be classified into *types* based on the four possible combinations of the *major* state of the sender and receiver, i.e. the markings of places sender_state and receiver_state. The relationship between the sender sequence number (ssn) and receiver sequence number (rsn), either $rsn = ssn$ or $rsn = ssn \oplus_{MS} 1$, gives rise to subtypes for two of the four combinations of major state. Thus there are six combinations in total, giving the six types, 1, 2a, 2b, 3a, 3b and 4, shown in Table 1. An explanation of the significance of each type is given in [11].

Table 1. Classification of markings $M \in V_{(MS,MR)}$ into types based on the state of the sender and receiver

M(sender_state)	M(receiver_state)	M(send_seq_no)	M(recv_seq_no)	$Type_{MS}(M)$
1‘s_ready	1‘r_ready	1‘sn	1‘sn	1
1‘wait_ack	1‘r_ready	1‘sn	1‘sn	2a
1‘wait_ack	1‘r_ready	1‘sn	1‘$(sn \oplus_{MS} 1)$	2b
1‘wait_ack	1‘process	1‘sn	1‘sn	3a
1‘wait_ack	1‘process	1‘sn	1‘$(sn \oplus_{MS} 1)$	3b
1‘s_ready	1‘process	1‘sn	1‘sn	4

Definition 4 (Markings to Types). *We define the family of functions that classifies markings as* $Type_{MS} : V_{(MS,MR)} \rightarrow \{1, 2a, 2b, 3a, 3b, 4\}$ *where the body of* $Type_{MS}$ *is given in Table 1.*

In addition, the following assumptions are made about the content of the communication channels, all of which are proved valid at the end of Section 4.1.

Assumption 1. *The content of the message and acknowledgement channels is a list of contiguous integers of the form* i^*j^* *where* $i, j \in \{0, ..., MaxSeqNo\}$.

Assumption 2. *The message and acknowledgement channels contain at most two distinct consecutive integers, i.e. of the form* i^*j^* *where* $j = i \oplus_{MS} 1$.

Assumption 3. *All reachable markings* $M \in V_{(MS,MR)}$ *of* $CPN_{(MS,MR)}$ *can be classified into one of the 6 types in Table 1.*

Using Lemma 1, Assumptions 1, 2 and 3, and Table 1, every marking can be encoded and uniquely identified by the following marking notation:

Definition 5 (Shorthand Marking Notation). *For* $CPN_{(MS,MR)}$ *all markings* $M \in V_{(MS,MR)}$ *can be uniquely identified and represented by the notation* $M^{(MS,MR)}_{(type,ssn),(mo,ao,mn,an,ret)}$ *where the superscript contains the parameter values of the SWP CPN and the subscript contains the marking description, where:*

- $type = Type_{MS}(M)$*;*
- $ssn \in \{0, 1, ..., MS\}$ *is the sender sequence number;*
- $mo \in \mathbb{N}$ *is the number of* old *(duplicate)* messages *with sequence number* $ssn \ominus_{MS} 1$ *in the message channel;*
- $ao \in \mathbb{N}$ *is the number of* old *(duplicate)* acknowledgements *with sequence number ssn in the acknowledgement channel;*
- $mn \in \mathbb{N}$ *is the number of* new *(current) messages with sequence number ssn in the message channel;*
- $an \in \mathbb{N}$ *is the number of* new *(current) acknowledgements with sequence number* $ssn \oplus_{MS} 1$ *in the acknowledgement channel; and*
- $ret \in \{0, 1, ..., MR\}$ *is the value of the retransmission counter for the currently outstanding (unacknowledged) message;*

so that for a given $M \in V_{(MS,MR)}$ *represented by* $M^{(MS,MR)}_{(type,ssn),(mo,ao,mn,an,ret)}$ *the marking of places* sender_state, receiver_state, send_seq_no *and* recv_seq_no *is encoded in the pair* $(type, ssn)$ *as given by Table 1 and:*

$$\begin{aligned} M(\mathsf{mess\_channel}) &= 1`[(ssn \ominus_{MS} 1)^{mo} \quad ssn^{mn}] \\ M(\mathsf{ack\_channel}) &= 1`[ssn^{ao} \quad (ssn \oplus_{MS} 1)^{an}] \\ M(\mathsf{retrans\_counter}) &= 1`ret \end{aligned}$$

Analogously, a shorthand notation is defined for arcs in [11].

Sets of markings and sets of arcs are defined as follows:

Definition 6 (Sets of Markings). $V^{(MS,MR)}_{(type,ssn)} = \{M \in V_{(MS,MR)} \mid Type_{MS}(M) = type, M(\mathsf{send\_seq\_no}) = 1`ssn\}$ *represents the set of markings in which the sender sequence number is given by ssn, and the sender and receiver states and receiver sequence number are given by the type as specified in Table 1.*

Definition 7 (Sets of Arcs). $A^{(MS,MR)}_{(type,ssn)} = \{(M,(t,b),M') \in A_{(MS,MR)} \mid Type_{MS}(M) = type, M(\mathsf{send\_seq\_no}) = 1`ssn\}$ *represents the set of arcs with source nodes in* $V^{(MS,MR)}_{(type,ssn)}$.

3.2 Important Model Properties

There are several important behavioural properties of the SWP CPN model that are needed for the proof of correctness of the algebraic expressions:

Lemma 2. *For all* $M \in V_{(MS,MR)}$*, the enabling and subsequent firing of each transition is independent of the values of the sequence numbers in the binding.*

Sketch of Proof. (See [11] for the full proof.) Proof is from Lemma 1 and the standard enabling and firing rules of CPNs [14].

From Fig. 1 the enabling conditions of send_mess are: $f_c(M(\mathsf{sender\_state})) =$ s_ready; $|M(\mathsf{send\_seq\_no})| > 0$; and $|M(\mathsf{mess\_channel})| > 0$. All three conditions are independent of sequence numbers. send_mess is enabled with binding queue $= f_c(M(\mathsf{mess\_channel}))$ and sn $= f_c(M(\mathsf{send\_seq\_no}))$. When send_mess occurs, it:

- Removes 1`s_ready from sender_state and returns 1`wait_ack to this place;
- Leaves the marking of place send_seq_no unchanged; and
- Removes 1`queue from place mess_channel and returns 1`queue^^[sn] to this place (append a copy of sn to the end of the message channel queue).

None of these actions depend on or are affected by the particular values of queue or sn in the binding, thus the behaviour of send_mess is independent of the values of the sequence numbers with which it interacts. The same reasoning is used to prove this lemma for the other seven transitions. □

Lemma 3. *For all* $M \in V_{(MS,MR)}$ *in which* $M(\mathsf{receiver\_state}) = 1`r\_ready$ *and* $|f_c(M(\mathsf{mess\_channel}))| > 0$*, the message at the head of the queue in the message channel can always be converted into an acknowledgement, i.e.* $\exists M', M'' \in V_{(MS,MR)}$ *such that* $M[\mathsf{receive\_mess}\rangle M'[\mathsf{send\_ack}\rangle M''$, $|f_c(M''(\mathsf{mess\_channel}))| = |f_c(M(\mathsf{mess\_channel}))|-1$ *and* $|f_c(M''(\mathsf{ack\_channel}))| = |f_c(M(\mathsf{ack\_channel}))|+1$.

Sketch of Proof. (See [11] for the full proof.) Only reachable markings satisfying the enabling conditions of receive_mess need be considered. For each such marking M, Lemma 2 ensures that the enabling and action taken upon firing receive_mess is independent of the values of the sequence numbers involved. When receive_mess occurs from any such M we reach a marking M' in which the receiver state has changed to process and one message has been removed from the message channel. From the CPN diagram in Fig. 1, each such marking M' enables send_ack, the occurrence of which leads to a marking M'' such that the receiver has returned once again to the ready state, the message channel contains one less message than in M and the acknowledgement channel contains one more acknowledgement than in M. Thus the lemma is proved. □

Lemma 4. $\forall M \in V_{(MS,MR)}$, $|f_c(M(\mathsf{mess\_channel}))| > 0 \implies \exists M_1 \in V_{(MS,MR)}$ *such that* $M[\mathsf{mess\_loss}\rangle M_1$ *and* $|f_c(M_1(\mathsf{mess\_channel}))| = |f_c(M(\mathsf{mess\_channel}))| - 1$ *and* $|f_c(M(\mathsf{ack\_channel}))| > 0 \implies \exists M_2 \in V_{(MS,MR)}$ *such that* $M[\mathsf{ack\_loss}\rangle M_2$ *and* $|f_c(M_2(\mathsf{ack\_channel}))| = |f_c(M(\mathsf{ack\_channel}))| - 1$, *while the marking of all other places remains unchanged.*

Proof. The proof follows immediately from the CPN in Fig. 1. □

4 Algebraic Expressions for the SWP CPN RGs

Empirical evidence gathered in [12] for small parameter values reveals a regular structure in the RG that is linear in MaxSeqNo and quartic in MaxRetrans. This also holds true for the model presented in Section 2. Based on the intuition in [12] for the case where MaxRetrans=0, in this paper, we present an algebraic formula representing the family of RGs of our SWP CPN and prove it correct. We then discuss a number of properties that can be proved directly from the algebraic formula. Because of size limitations, only proof sketches are presented (see [11] for details).

4.1 The Algebraic Formula in Both Parameters

When defining the markings and arcs of $OG_{(MS,MR)}$ we specify sets of markings and arcs using the notation from Definitions 5, 6 and 7 and by specifying allowable ranges of the five variables (mo, ao, mn, an, ret). All variables are assumed to be greater than or equal to 0, unless otherwise indicated.

All of the markings of $OG_{(MS,MR)}$ are described in this way in Table 2, by evaluating the expressions in this table for $0 \leq i \leq MS$. The first column gives the name of the set of markings for each subset of the partition according to its type. Column 2 defines the set of markings by specifying the allowed ranges of variable values. If a variable is restricted to a specific value, e.g. 0, we write this directly in the label of the marking. Note that because of the expression $0 \leq mo + ao \leq MR - 1$, the markings of type 3a and type 4 (rows 4 and 6) are defined only when $MR > 0$. Hence $V^{(MS,0)}_{(3a,i)} = V^{(MS,0)}_{(4,i)} = \emptyset$, the empty set, when $MR = 0$.

Table 2. $V^{(MS,MR)}_{(type,i)}$, for $0 \leq i \leq MS$ and $type \in \{1, 2a, 2b, 3a, 3b, 4\}$

Name	Set Definition
$V^{(MS,MR)}_{(1,i)}$	$\{M^{(MS,MR)}_{(1,i),(mo,ao,0,0,0)} \mid 0 \leq mo + ao \leq MR\}$
$V^{(MS,MR)}_{(2a,i)}$	$\{M^{(MS,MR)}_{(2a,i),(mo,ao,mn,0,ret)} \mid 0 \leq mo + ao \leq MR, 0 \leq ret \leq MR,$ $0 \leq mn \leq ret + 1\}$
$V^{(MS,MR)}_{(2b,i)}$	$\{M^{(MS,MR)}_{(2b,i),(0,ao,mn,an,ret)} \mid 0 \leq ao \leq MR, 0 \leq ret \leq MR, 0 \leq mn \leq ret,$ $0 \leq mn + an \leq ret + 1\}$
$V^{(MS,MR)}_{(3a,i)}$	$\{\}$, for $MR = 0$; or $\{M^{(MS,MR)}_{(3a,i),(mo,ao,mn,0,ret)} \mid 0 \leq mo + ao \leq MR - 1, 0 \leq ret \leq MR,$ $0 \leq mn \leq ret + 1\}$, for $MR > 0$.
$V^{(MS,MR)}_{(3b,i)}$	$\{M^{(MS,MR)}_{(3b,i),(0,ao,mn,an,ret)} \mid 0 \leq ao \leq MR, 0 \leq ret \leq MR, 0 \leq mn + an \leq ret\}$
$V^{(MS,MR)}_{(4,i)}$	$\{\}$, for $MR = 0$; or $\{M^{(MS,MR)}_{(4,i),(mo,ao,0,0,0)} \mid 0 \leq mo + ao \leq MR - 1\}$, for $MR > 0$.

All of the arcs of $OG_{(MS,MR)}$ are described in Tables 3 to 8 by evaluating each table for $0 \leq i \leq MS$. There is one table of arcs per set of markings (i.e. per row) in Table 2, describing the set of outgoing arcs of that set of markings. Correspondingly, $A^{(MS,0)}_{(3a,i)}$ and $A^{(MS,0)}_{(4,i)} = \emptyset$ when $MR = 0$. The first column of each arc table gives any additional restrictions that must be placed on the variables mo, ao, mn, an and ret. For example, loss of an old message cannot occur when $mo = 0$. The second, third and fourth columns list the source marking, binding element and destination marking respectively.

We now state the theorem for our parametric RG over both parameters and prove its correctness.

Theorem 1. *For all* $MS \in \mathbb{N}^+, MR \in \mathbb{N}$ *and for* $Type = \{1, 2a, 2b, 3a, 3b, 4\}$, $OG_{(MS,MR)}$=$(V_{(MS,MR)}, A_{(MS,MR)})$ *where*

$$V_{(MS,MR)} = \bigcup_{\substack{0 \leq i \leq MS \\ t \in Type}} V^{(MS,MR)}_{(t,i)}$$

and

$$A_{(MS,MR)} = \bigcup_{\substack{0 \leq i \leq MS \\ t \in Type}} A^{(MS,MR)}_{(t,i)}$$

where all nodes and arcs are defined in Tables 2 to 8.

Proof. The proof is in two parts. The first part proves that all states in $V_{(MS,MR)}$ are reachable from the initial marking using a connected spanning graph. The second part proves that every arc from every state in $V_{(MS,MR)}$ leads to a state in $V_{(MS,MR)}$ and that this set of arcs equals $A_{(MS,MR)}$. The two parts of the proof each describe a necessary condition, which together are sufficient to show that Theorem 1 is correct.

Table 3. The set of arcs $A^{(MS,MR)}_{(1,i)}$ with source markings in $V^{(MS,MR)}_{(1,i)}$

Condition	Source Marking	Binding Element	Destination Marking
none	$M^{(MS,MR)}_{(1,i),(mo,ao,0,0,0)}$	send_mess<queue = $[(i \ominus_{MS} 1)^{mo}]$, sn=$i$>	$M^{(MS,MR)}_{(2a,i),(mo,ao,1,0,0)}$
$mo \geq 1$	$M^{(MS,MR)}_{(1,i),(mo,ao,0,0,0)}$	mess_loss<queue = $[(i \ominus_{MS} 1)^{mo}]$, sn=$i \ominus_{MS} 1$>	$M^{(MS,MR)}_{(1,i),(mo-1,ao,0,0,0)}$
$mo \geq 1$	$M^{(MS,MR)}_{(1,i),(mo,ao,0,0,0)}$	receive_mess<queue = $[(i \ominus_{MS} 1)^{mo-1}]$, sn=$i \ominus_{MS} 1$, rn=$i$>	$M^{(MS,MR)}_{(4,i),(mo-1,ao,0,0,0)}$
$ao \geq 1$	$M^{(MS,MR)}_{(1,i),(mo,ao,0,0,0)}$	ack_loss<queue = $[i^{ao}]$, rn=i>	$M^{(MS,MR)}_{(1,i),(mo,ao-1,0,0,0)}$
$ao \geq 1$	$M^{(MS,MR)}_{(1,i),(mo,ao,0,0,0)}$	receive_dup_ack<queue = $[i^{ao-1}]$, sn=i, rn=i>	$M^{(MS,MR)}_{(1,i),(mo,ao-1,0,0,0)}$

Table 4. The set of arcs $A^{(MS,MR)}_{(2a,i)}$ with source markings in $V^{(MS,MR)}_{(2a,i)}$

Condition	Source Marking	Binding Element	Destination Marking
ret<MR	$M^{(MS,MR)}_{(2a,i),(mo,ao,mn,0,ret)}$	timeout_retrans<queue=$[(i \ominus_{MS} 1)^{mo}\ i^{mn}]$, sn=$i$, rc=$ret$>	$M^{(MS,MR)}_{(2a,i),(mo,ao,mn+1,0,ret+1)}$
$mo \geq 1$	$M^{(MS,MR)}_{(2a,i),(mo,ao,mn,0,ret)}$	mess_loss<queue=$[(i \ominus_{MS} 1)^{mo}\ i^{mn}]$, sn=$i \ominus_{MS} 1$>	$M^{(MS,MR)}_{(2a,i),(mo-1,ao,mn,0,ret)}$
$mn \geq 1$	$M^{(MS,MR)}_{(2a,i),(mo,ao,mn,0,ret)}$	mess_loss<queue = $[(i \ominus_{MS} 1)^{mo}\ i^{mn}]$, sn=$i$>	$M^{(MS,MR)}_{(2a,i),(mo,ao,mn-1,0,ret)}$
$mo \geq 1$	$M^{(MS,MR)}_{(2a,i),(mo,ao,mn,0,ret)}$	receive_mess<queue = $[(i \ominus_{MS} 1)^{mo-1}\ i^{mn}]$, sn=$i \ominus_{MS} 1$, rn=$i$>	$M^{(MS,MR)}_{(3a,i),(mo-1,ao,mn,0,ret)}$
$mn \geq 1$	$M^{(MS,MR)}_{(2a,i),(0,ao,mn,0,ret)}$	receive_mess<queue = $[i^{mn-1}]$,sn=i, rn=i>	$M^{(MS,MR)}_{(3b,i),(0,ao,mn-1,0,ret)}$
$ao \geq 1$	$M^{(MS,MR)}_{(2a,i),(mo,ao,mn,0,ret)}$	ack_loss<queue = $[i^{ao}]$, rn=i>	$M^{(MS,MR)}_{(2a,i),(mo,ao-1,mn,0,ret)}$
$ao \geq 1$	$M^{(MS,MR)}_{(2a,i),(mo,ao,mn,0,ret)}$	receive_dup_ack<queue = $[i^{ao-1}]$, sn=i, rn=i>	$M^{(MS,MR)}_{(2a,i),(mo,ao-1,mn,0,ret)}$

Table 5. The set of arcs $A^{(MS,MR)}_{(2b,i)}$ with source markings in $V^{(MS,MR)}_{(2b,i)}$

Condition	Source Marking	Binding Element	Destination Marking
ret<MR	$M^{(MS,MR)}_{(2b,i),(0,ao,mn,an,ret)}$	timeout_retrans<queue = $[i^{mn}]$, sn=i,rc=ret>	$M^{(MS,MR)}_{(2b,i),(0,ao,mn+1,an,ret+1)}$
$mn \geq 1$	$M^{(MS,MR)}_{(2b,i),(0,ao,mn,an,ret)}$	mess_loss<queue = $[i^{mn}]$, sn=i>	$M^{(MS,MR)}_{(2b,i),(0,ao,mn-1,an,ret)}$
$mn \geq 1$	$M^{(MS,MR)}_{(2b,i),(0,ao,mn,an,ret)}$	receive_mess<queue = $[i^{mn-1}]$,sn=i, rn=$i \oplus_{MS} 1$>	$M^{(MS,MR)}_{(3b,i),(0,ao,mn-1,an,ret)}$
$ao \geq 1$	$M^{(MS,MR)}_{(2b,i),(0,ao,mn,an,ret)}$	ack_loss<queue = $[i^{ao}\ (i \oplus_{MS} 1)^{an}]$, rn=$i$>	$M^{(MS,MR)}_{(2b,i),(0,ao-1,mn,an,ret)}$
$an \geq 1$	$M^{(MS,MR)}_{(2b,i),(0,ao,mn,an,ret)}$	ack_loss<queue = $[i^{ao}\ (i \oplus_{MS} 1)^{an}]$, rn=$i \oplus_{MS} 1$>	$M^{(MS,MR)}_{(2b,i),(0,ao,mn,an-1,ret)}$
$ao \geq 1$	$M^{(MS,MR)}_{(2b,i),(0,ao,mn,an,ret)}$	receive_dup_ack<queue = $[i^{ao-1}\ (i \oplus_{MS} 1)^{an}]$, sn=$i$, rn=$i$>	$M^{(MS,MR)}_{(2b,i),(0,ao-1,mn,an,ret)}$
$an \geq 1$	$M^{(MS,MR)}_{(2b,i),(0,0,mn,an,ret)}$	receive_ack<queue = $[(i \oplus_{MS} 1)^{an-1}]$, sn=$i$, rn=$i \oplus_{MS} 1$, rc = ret>	$M^{(MS,MR)}_{(1,i\oplus_{MS}1),(mn,an-1,0,0,0)}$

Table 6. The set of arcs $A^{(MS,MR)}_{(3a,i)}$ with source markings in $V^{(MS,MR)}_{(3a,i)}$, for $MR > 0$

Condition	Source Marking	Binding Element	Destination Marking
ret<MR	$M^{(MS,MR)}_{(3a,i),(mo,ao,mn,0,ret)}$	timeout_retrans<queue = $[(i \ominus_{MS} 1)^{mo}\ i^{mn}]$,sn=$i$,rc=$ret$>	$M^{(MS,MR)}_{(3a,i),(mo,ao,mn+1,0,ret+1)}$
$mo \geq 1$	$M^{(MS,MR)}_{(3a,i),(mo,ao,mn,0,ret)}$	mess_loss<queue = $[(i \ominus_{MS} 1)^{mo}\ i^{mn}]$, sn=$i \ominus_{MS} 1$>	$M^{(MS,MR)}_{(3a,i),(mo-1,ao,mn,0,ret)}$
$mn \geq 1$	$M^{(MS,MR)}_{(3a,i),(mo,ao,mn,0,ret)}$	mess_loss<queue = $[(i \ominus_{MS} 1)^{mo}\ i^{mn}]$, sn=$i$>	$M^{(MS,MR)}_{(3a,i),(mo,ao,mn-1,0,ret)}$
$ao \geq 1$	$M^{(MS,MR)}_{(3a,i),(mo,ao,mn,0,ret)}$	ack_loss<queue = $[i^{ao}]$, rn=i>	$M^{(MS,MR)}_{(3a,i),(mo,ao-1,mn,0,ret)}$
$ao \geq 1$	$M^{(MS,MR)}_{(3a,i),(mo,ao,mn,0,ret)}$	receive_dup_ack<queue = $[i^{ao-1}]$, sn=i, rn=i>	$M^{(MS,MR)}_{(3a,i),(mo,ao-1,mn,0,ret)}$
none	$M^{(MS,MR)}_{(3a,i),(mo,ao,mn,0,ret)}$	send_ack<queue = $[i^{ao}]$, rn=i>	$M^{(MS,MR)}_{(2a,i),(mo,ao+1,mn,0,ret)}$

Table 7. The set of arcs $A^{(MS,MR)}_{(3b,i)}$ with source markings in $V^{(MS,MR)}_{(3b,i)}$

Condition	Source Marking	Binding Element	Destination Marking
ret<MR	$M^{(MS,MR)}_{(3b,i),(0,ao,mn,an,ret)}$	timeout_retrans<queue = $[i^{mn}]$, sn=i, rc=ret>	$M^{(MS,MR)}_{(3b,i),(0,ao,mn+1,an,ret+1)}$
$mn \geq 1$	$M^{(MS,MR)}_{(3b,i),(0,ao,mn,an,ret)}$	mess_loss<queue = $[i^{mn}]$, sn=i>	$M^{(MS,MR)}_{(3b,i),(0,ao,mn-1,an,ret)}$
$ao \geq 1$	$M^{(MS,MR)}_{(3b,i),(0,ao,mn,an,ret)}$	ack_loss<queue = $[i^{ao}\ (i \oplus_{MS} 1)^{an}]$, rn=$i$>	$M^{(MS,MR)}_{(3b,i),(0,ao-1,mn,an,ret)}$
$an \geq 1$	$M^{(MS,MR)}_{(3b,i),(0,ao,mn,an,ret)}$	ack_loss<queue = $[i^{ao}\ (i \oplus_{MS} 1)^{an}]$, rn=$i \oplus_{MS} 1$>	$M^{(MS,MR)}_{(3b,i),(0,ao,mn,an-1,ret)}$
$ao \geq 1$	$M^{(MS,MR)}_{(3b,i),(0,ao,mn,an,ret)}$	receive_dup_ack<queue = $[i^{ao-1}\ (i \oplus_{MS} 1)^{an}]$, sn=$i$,rn=$i$>	$M^{(MS,MR)}_{(3b,i),(0,ao-1,mn,an,ret)}$
$an \geq 1$	$M^{(MS,MR)}_{(3b,i),(0,0,mn,an,ret)}$	receive_ack<queue = $[(i \oplus_{MS} 1)^{an-1}]$, sn=$i$,rn=$i \oplus_{MS} 1$, rc = ret>	$M^{(MS,MR)}_{(4,i\oplus_{MS}1),(mn,an-1,0,0,0)}$
none	$M^{(MS,MR)}_{(3b,i),(0,ao,mn,an,ret)}$	send_ack<queue = $[i^{ao}\ (i \oplus_{MS} 1)^{an}]$, rn=$i \oplus_{MS} 1$>	$M^{(MS,MR)}_{(2b,i),(0,ao,mn,an+1,ret)}$

Table 8. The set of arcs $A^{(MS,MR)}_{(4,i)}$ with source markings in $V^{(MS,MR)}_{(4,i)}$, for $MR > 0$

Condition	Source Marking	Binding Element	Destination Marking
none	$M^{(MS,MR)}_{(4,i),(mo,ao,0,0,0)}$	send_mess<queue = $[(i \ominus_{MS} 1)^{mo}]$, sn=$i$>	$M^{(MS,MR)}_{(3a,i),(mo,ao,1,0,0)}$
$mo \geq 1$	$M^{(MS,MR)}_{(4,i),(mo,ao,0,0,0)}$	mess_loss<queue = $[(i \ominus_{MS} 1)^{mo}]$, sn=$i \ominus_{MS} 1$>	$M^{(MS,MR)}_{(4,i),(mo-1,ao,0,0,0)}$
$ao \geq 1$	$M^{(MS,MR)}_{(4,i),(mo,ao,0,0,0)}$	ack_loss<queue = $[i^{ao}]$, rn=i>	$M^{(MS,MR)}_{(4,i),(mo,ao-1,0,0,0)}$
$ao \geq 1$	$M^{(MS,MR)}_{(4,i),(mo,ao,0,0,0)}$	receive_dup_ack<queue = $[i^{ao-1}]$, sn=i, rn=i>	$M^{(MS,MR)}_{(4,i),(mo,ao-1,0,0,0)}$
none	$M^{(MS,MR)}_{(4,i),(mo,ao,0,0,0)}$	send_ack<queue = $[i^{ao}]$, rn=i>	$M^{(MS,MR)}_{(1,i),(mo,ao+1,0,0,0)}$

Lemma 5. Spanning Lemma. *For $MR \in \mathbb{N}$ and $MS \in \mathbb{N}^+$, and for $0 \leq i \leq MS$, and for $Type = \{1, 2a, 2b, 3a, 3b, 4\}$, all markings in $\cup_{t \in Type}(V^{(MS,MR)}_{(t,i)}) \cup \{M^{(MS,MR)}_{(1,i\oplus_{MS}1),(MR,0,0,0,0)}\}$ are reachable from $M^{(MS,MR)}_{(1,i),(MR,0,0,0,0)}$.*

Sketch of Proof. (See [11] for the full proof.) Lemma 2 allows this lemma to be proved directly, for any value of $i \in \{0, ..., MS\}$, rather than inductively over MS. The marking $M^{(MS,MR)}_{(1,i),(MR,0,0,0,0)}$, identical to the initial marking but for MR old duplicate messages with sequence number MS in the message channel, is chosen as the starting point, rather than the initial marking $M^{(MS,MR)}_{(1,i),(0,0,0,0,0)}$. This is because, as it turns out, it is easier to show that $M^{(MS,MR)}_{(1,i),(MR,0,0,0,0)}$ can reach all markings in $V^{(MS,MR)}_{(1,i)}$. (Had we started with the initial marking, we would need to complete a full cycle of the sequence number space in order to get old messages in the message channel when $ssn = 0$.)

Application of Lemma 3 MR number of times shows that $M^{(MS,MR)}_{(1,i),(MR,0,0,0,0)}$ can reach all markings in $V_{span1} = \{M^{(MS,MR)}_{(1,i),(mo,MR-mo,0,0,0)} \mid 0 \leq mo \leq MR\}$. Then by application of Lemma 4, V_{span1} can reach the markings in

$$V_{span2} = \{M^{(MS,MR)}_{(1,i),(mo',ao,0,0,0)} \mid M^{(MS,MR)}_{(1,i),(mo,MR-mo,0,0,0)} \in V_{span1}, \\ 0 \leq mo' \leq mo, 0 \leq ao \leq MR - mo\}$$

By a process of simplification of the inequalities in the set definition, we determine that V_{span2} equals $V^{(MS,MR)}_{(1,i)}$ (see Table 2).

From inspection of the CPN diagram in Fig. 1, $M^{(MS,MR)}_{(1,i),(MR,0,0,0,0)} \in V^{(MS,MR)}_{(1,i)}$ can reach $M^{(MS,MR)}_{(2a,i),(MR,0,1,0,0)}$ via occurrence of send_mess, regardless of the value of i. A similar process is then followed for marking $M^{(MS,MR)}_{(2a,i),(MR,0,1,0,0)}$ as was followed for $M^{(MS,MR)}_{(1,i),(MR,0,0,0,0)}$, to prove that $M^{(MS,MR)}_{(2a,i),(MR,0,1,0,0)}$ can reach all other markings in $V^{(MS,MR)}_{(2a,i)}$. This process continues for the markings in $V^{(MS,MR)}_{(2b,i)}$, $V^{(MS,MR)}_{(3a,i)}$, $V^{(MS,MR)}_{(3b,i)}$ and $V^{(MS,MR)}_{(4,i)}$, and for reachability from one set to another. The procedure for determining a spanning of markings in $V^{(MS,MR)}_{(type,i)}$ for $type \in \{2a, 2b, 3a, 3b\}$ is slightly more complicated, due to the fact that retransmissions can occur from these markings when $ret < MR$.

Finally, $M^{(MS,MR)}_{(1,i\oplus_{MS}1),(MR,0,0,0,0)}$ can be reached from $M^{(MS,MR)}_{(2b,i),(0,0,MR,1,MR)} \in V^{(MS,MR)}_{(2b,i)}$ by firing the receive_ack transition. (The MR new messages become MR old messages because ssn has incremented.) Thus the lemma is proved. □

Corollary 1. *All markings in $V_{(MS,MR)}$ are reachable from $M^{(MS,MR)}_{(1,0),(MR,0,0,0,0)}$. This follows directly from the Spanning Lemma by a trivial induction over MS.*

To complete the final step in the proof that all markings in $V_{(MS,MR)}$ are accessible from the initial marking, it is sufficient to show that the initial marking

$M^{(MS,MR)}_{(1,0),(0,0,0,0,0,0)}$ can reach one of the markings in $\{M^{(MS,MR)}_{(1,i),(MR,0,0,0,0)} \mid 0 \leq i \leq MS\}$. By repeated application of Lemma 5 this can reach $M^{(MS,MR)}_{(1,0),(MR,0,0,0,0)}$, which in turn, by Corollary 1, can reach all markings in $V_{(MS,MR)}$. The marking $M^{(MS,MR)}_{(1,1),(MR,0,0,0,0)}$ is chosen as it is the first suitable marking that can be reached from the initial marking. This is proved in the following lemma.

Lemma 6. *$M^{(MS,MR)}_{(1,1),(MR,0,0,0,0)}$ is reachable from $M^{(MS,MR)}_{(1,0),(0,0,0,0,0)}$.*

Sketch of Proof. (See [11] for the full proof.) Proof is by direct inspection of the CPN diagram in Fig. 1. The initial marking enables transition send_mess with binding $<queue = [], sn = 0>$. This results in the marking $M^{(MS,MR)}_{(2a,0),(0,0,1,0,0)}$. From this marking, transition timeout_retrans can occur consecutively MR number of times. The resulting marking is $M^{(MS,MR)}_{(2a,0),(0,0,MR+1,0,MR)}$ in which $MR+1$ copies of the message with sequence number 0 are in the message channel. From this marking, receive_mess can occur, leading to marking $M^{(MS,MR)}_{(3b,0),(0,0,MR,0,MR)}$. From this marking, send_ack can occur, leading to $M^{(MS,MR)}_{(2b,0),(0,0,MR,1,MR)}$. The single acknowledgement with sequence number 1 is the acknowledgement for which the sender is waiting. The occurrence of receive_ack with binding $<queue = [], sn = 0, rn = 1, rc = MR>$ results in marking $M^{(MS,MR)}_{(1,1),(MR,0,0,0,0)}$. (Again, the new messages are now old messages because ssn has incremented.) Thus $M^{(MS,MR)}_{(1,0),(0,0,0,0,0)}$ can reach $M^{(MS,MR)}_{(1,1),(MR,0,0,0,0)}$ and the lemma is proved. □

From Corollary 1 and Lemma 6, all markings in $V_{(MS,MR)}$ are reachable from $M^{(MS,MR)}_{(1,0),(0,0,0,0,0)}$ and Part A of the proof of Theorem 1 is proved.

Part B of the proof of Theorem 1 is proved by the Successor Lemma:

Lemma 7. Successor Lemma. *For all $MR \in \mathbb{N}$, $MS \in \mathbb{N}^+$, $i \in \{0, ..., MS\}$ and $t \in \{1, 2a, 2b, 3a, 3b, 4\}$, $A^{(MS,MR)}_{(t,i)}$ describes exactly the enabled binding elements of all markings in $V^{(MS,MR)}_{(t,i)}$ and the destination marking of every arc in $A^{(MS,MR)}_{(t,i)}$ is in $V_{(MS,MR)}$.*

Sketch of Proof. (See [11] for a full proof.) Lemma 2 allows this lemma to be proved correct for any value of $i \in \{0, ..., MS\}$. Consider the markings in $V^{(MS,MR)}_{(1,i)}$ defined in row 1 of Table 2. From the CPN diagram in Fig. 1 and standard enabling rules of CPNs [14], all enabled binding elements (and thus associated arcs) can be identified. The send_mess transition is enabled by all markings in $V^{(MS,MR)}_{(1,i)}$. The mess_loss and receive_mess transitions are enabled only by markings in the subset of $V^{(MS,MR)}_{(1,i)}$ in which the message channel is non-empty. The ack_loss and receive_dup_ack transitions are enabled only by the subset of $V^{(MS,MR)}_{(1,i)}$ in which the acknowledgement channel is non-empty. No other transitions are enabled by any markings in $V^{(MS,MR)}_{(1,i)}$.

By systematically determining the destination marking for each pair of source marking and binding element, all arcs with source nodes in $V^{(MS,MR)}_{(1,i)}$ can be determined. For example, the occurrence of transition send_mess from marking $M^{(MS,MR)}_{(1,i),(mo,ao,0,0,0)} \in V^{(MS,MR)}_{(1,i)}$ with binding $< queue = [(i \ominus_{MS} 1)^{mo}], sn = 1 >$ leads to a marking $M^{(MS,MR)}_{(2a,i),(mo,ao,1,0,0)} \in V^{(MS,MR)}_{(2a,i)}$, for all $i \in \{0, ..., MS\}$. This corresponds to row 1 of Table 3. Rows 2 to 5 can be obtained by a similar procedure for the other enabled transitions.

This procedure can then be repeated for all markings in the other five sets of nodes defined in Table 2. This shows that all arcs with source markings in $V_{(MS,MR)}$ also have destination markings in $V_{(MS,MR)}$ and that these arcs correspond exactly to those defined in Tables 3 to 8. Thus the lemma is proved. □

For all $MS \in \mathbb{N}^+$ and all $MR \in \mathbb{N}$, Lemmas 5, 6 and 7 and Corollary 1 show that the markings in $V_{(MS,MR)}$ defined by Table 2 correspond exactly to the markings reachable from the initial marking. Lemma 7 also shows that the arcs captured by Tables 3 to 8 correspond exactly to the set of arcs with source markings in $V_{(MS,MR)}$. Thus, for all $MS \in \mathbb{N}^+$ and all $MR \in \mathbb{N}$, $OG_{(MS,MR)} = (V_{(MS,MR)}, A_{(MS,MR)})$ and hence Theorem 1 is proved. □

The validity of the three assumptions made in Section 3.1 is confirmed by the correctness of the algebraic expressions. No marking can be reached that violates any of the three assumptions, i.e. every marking has channel content of the form i^*j^* where $i, j \in \{0, ..., MS\}$ and $j = i \oplus_{MS} 1$, and every reachable marking can be classified into one of the 6 types in Table 1.

4.2 Analysis Results

Absence of Unexpected Deadlocks. Dead markings can be detected by subtracting from the corresponding set of markings in Table 2 the sets of markings defined as source markings in each table of arcs.

For all $MR \in \mathbb{N}$ and $MS \in \mathbb{N}^+$, the dead markings are $V^{(MS,MR)}_{dead} = \cup_{0 \le i \le MS} \{M^{(MS,MR)}_{(2a,i),(0,0,0,0,MR)}, M^{(MS,MR)}_{(2b,i),(0,0,0,0,MR)}\}$. All dead markings occur because of loss and a bounded retransmission scheme, and all are expected.

Channel Bounds. Channel bounds can be determined by direct examination of the set definitions in the rows of Table 2. Maximising $mo + mn$ gives the message channel bound for the markings in each row. The message channel bound of the SWP becomes the maximum of $mo + mn$ taken over all 6 rows. Similarly, the acknowledgement channel bound is found by maximising $ao + an$. The bound for both channels is $2MR + 1$, from row 2 (message channel) and row 3 (acknowledgement channel). These bounds are imposed by the SWP itself.

Size of the Reachability Graph. By direct inspection of Table 2 and Tables 3 to 8, Theorem 2 for the size of the RG in both parameters can be proved.

Theorem 2. *For $MR \in \mathbb{N}$ and $MS \in \mathbb{N}^+$, the number of nodes and arcs in $OG_{(MS,MR)}$ is given by*

$$|V_{(MS,MR)}| = ((MS+1)/6)(5MR^4 + 38MR^3 + 97MR^2 + 100MR + 36)$$

and

$$|A_{(MS,MR)}| = ((MS+1)/6)(30MR^4 + 175MR^3 + 306MR^2 + 179MR + 36).$$

Sketch of Proof. The nodes in $V_{(1,i)}^{(MS,MR)}$ and $V_{(4,i)}^{(MS,MR)}$ actually form a triangular structure, where the base contains the nodes where $mo + ao = MR$ and the apex is the node where $mo = ao = 0$. Using the formula for the n^{th} triangular number, $n(n+1)/2$, for $n = MR$ and $n = MR - 1$ respectively, we obtain $|V_{(1,i)}^{(MS,MR)}| = (MR^2+3MR+2)/2$ and $|V_{(4,i)}^{(MS,MR)}| = (MR^2+MR)/2$, for each value of $i \in \{0, ..., MS\}$.

The nodes in the other four sets have a more complicated structure. Take $V_{(2a,i)}^{(MS,MR)}$ for example. The structure can be visualised as a succession of triangular structures over mo and ao, one for each value of $mn \in \{0, ..., ret\}$. A summation over $0 \leq ret \leq MR$ obtains $|V_{(2a,i)}^{(MS,MR)}| = (MR^4+8MR^3+21MR^2+22MR+8)/4$. Similar techniques are used to obtain the size of the other node sets. The total number of markings is given by a summation over all values of $i \in \{0, ..., MS\}$ and the result $V_{(MS,MR)} = (MS+1)(5MR^4+38MR^3+97MR^2+100MR+36)/6$ is obtained.

Determining the number of arcs requires a more complicated approach. The number of source markings for which each arc is defined is determined for each row in Tables 3 to 8. To do this in a way that prevents excessively copious summations, for each row, the number of markings that do not satisfy the conditions in column 1 of each arc table is determined. This is then subtracted from the total number of markings defined by the corresponding set in Table 2. The total number of arcs is then the summation over all rows in all arc tables of the number of arcs defined by each row. The result is as stated in the theorem. □

This theorem confirms our empirical results for small parameter values and matches RG size expressions obtained using methods to fit polynomials to data.

5 Conclusions and Future Work

We have proved a theorem which gives an algebraic expression for the infinite family of RGs of a parameterised CPN model of the class of Stop-and-Wait protocols. The parameters, MaxSeqNo and MaxRetrans, are both unbounded and the protocol operates over a lossy unbounded in-order medium. This is a considerable advance over previous work [12], which was restricted to the case where MaxRetrans = 0, and automatic verification attempts using the tool FAST [9] which were only successful when MaxRetrans was an unbounded parameter with MaxSeqNo restricted to small concrete values (1 to 5) [8], and when MaxSeqNo was an unbounded parameter with MaxRetrans fixed to 0 [7].

These symbolic expressions can be used for protocol verification. For example, we have shown how deadlocked states can be identified from the arc expressions as those markings that never appear as source nodes. Further, we have shown that the node table (Table 2) can be used to determine upper bounds on the channel capacity. This result (2MaxRetrans+1) confirms that previously obtained using a hand proof on the CPN in [5,6], but is much simpler (once the algebraic expressions are known). We have also derived formulae for the number of nodes and arcs in the state space as a function of the two parameters, proving they are linear in MaxSeqNo and quartic in MaxRetrans, an interesting complexity result. Proving language equivalence to a service of alternating send and receive events [6], as was done in [12] for the restricted case of MaxRetrans = 0, is currently being undertaken for the general case.

In the future, we would like to automate the procedure for obtaining algebraic expressions for the RGs of parametric systems based on finding structural regularities as a function of the parameters. Our experience with modelling other systems, including the Capability Exchange Signalling service [18] and TCP's data transfer service [13], also reveals repeating patterns in their RGs from which symbolic RGs representing the infinite family have been obtained. This provides evidence that our new parametric approach is promising and may be generalised to a larger class of systems.

References

1. P. Aziz Abdulla, A. Collomb-Annichini, A. Bouajjani, and B. Jonsson. Using Forward Reachability Analysis for Verification of Lossy Channel Systems. *Formal Methods in System Design*, 25(1):39–65, 2004.
2. J. Billington. Formal specification of protocols: Protocol Engineering. In *Encyclopedia of Microcomputers*, volume 7, pages 299–314. Marcel Dekker, New York, 1991.
3. J. Billington, M. Diaz, and G. Rozenberg, editors. *Application of Petri Nets to Communication Networks*, volume 1605 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999.
4. J. Billington and G. E. Gallasch. How Stop and Wait Protocols Can Fail Over The Internet. In *Proceedings of FORTE'03*, volume 2767 of *Lecture Notes in Computer Science*, pages 209–223. Springer-Verlag, 2003. (invited paper).
5. J. Billington and G. E. Gallasch. An Investigation of the Properties of Stop-and-Wait Protocols over Channels which can Re-order messages. Technical Report CSEC-15, Computer Systems Engineering Centre Report Series, University of South Australia, May 2004.
6. J. Billington, G. E. Gallasch, and B. Han. A Coloured Petri Net Approach to Protocol Verification. In *Lectures on Concurrency and Petri Nets, Advances in Petri Nets*, volume 3098 of *Lecture Notes in Computer Science*, pages 210–290. Springer-Verlag, 2004.
7. J. Billington, G.E. Gallasch, and L. Petrucci. FAST Verification of the Class of Stop-and-Wait Protocols modelled by Coloured Petri Nets. *Nordic Journal of Computing*, Vol. 12(3):251–274, 2005.

8. J. Billington, G.E. Gallasch, and L. Petrucci. Transforming Coloured Petri Nets to Counter Systems for Parametric Verification: A Stop-and-Wait Protocol Case Study. In *Proceedings of 2nd International Workshop on Model-Based Methodologies for Pervasive and Embedded Software (MOMPES'05), Rennes, France*, TUCS General Publication, No. 39, pages 37–55, May 2005.
9. FAST - Fast Acceleration of Symbolic Transition systems. `http://www.lsv.ens-cachan.fr/fast/`.
10. G. E. Gallasch and J. Billington. Towards the Parametric Verification of the Class of Stop-and-Wait Protocols over Ordered Channels. Technical Report CSEC-21, Computer Systems Engineering Centre Report Series, University of South Australia, March 2005, revised June 2005.
11. G. E. Gallasch and J. Billington. Parametric Verification of the Class of Stop-and-Wait Protocols over Ordered Channels. Technical Report CSEC-23, Computer Systems Engineering Centre Report Series, University of South Australia, Draft of January 2006.
12. G.E. Gallasch and J. Billington. Using Parametric Automata for the Verification of the Stop-and-Wait Class of Protocols. In *Proceedings of ATVA 2005*, volume 3707 of *Lecture Notes in Computer Science*, pages 457–473. Springer-Verlag, 2005.
13. B. Han. *Formal Specification of the TCP Service and Verification of TCP Connection Management*. PhD thesis, Computer Systems Engineering Centre, School of Electrical and Information Engineering, University of South Australia, Adelaide, Australia, December 2004.
14. K. Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Vol. 1, Basic Concepts*. Springer-Verlag, 2nd edition, 1997.
15. K. Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Vol. 3, Practical Use*. Springer-Verlag, 1997.
16. L.M. Kristensen, S. Christensen, and K. Jensen. The Practitioner's Guide to Coloured Petri Nets. *International Journal on Software Tools for Technology Transfer*, 2(2):98–132, 1998.
17. L.M. Kristensen, M. Westergaard, and Peder Christian Nørgaard. Model-Based Prototyping of an Interoperability Protocol for Mobile Ad-Hoc Networks. In *Proceedings of IFM 2005*, volume 3771 of *Lecture Notes in Computer Science*, pages 266–286. Springer-Verlag, 2005.
18. L. Liu and J. Billington. Tackling the Infinite State Space of a Multimedia Control Protocol Service Specification. In *Proceedings of ICATPN'02*, volume 2360 of *Lecture Notes in Computer Science*, pages 273–293. Springer-Verlag, 2002.
19. J. Postel. Transmission Control Protocol. RFC 793, September 1981.
20. W. Stallings. *Data and Computer Communications*. Prentice Hall, 7th edition, 2004.
21. Standard ML of New Jersey. `http://cm.bell-labs.com/cm/cs/what/smlnj/`.
22. A. Tanenbaum. *Computer Networks*. Prentice Hall, 4th edition, 2003.
23. A. Valmari. The State Explosion Problem. In *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*, pages 429–528. Springer-Verlag, 1998.
24. A. Valmari and I. Kokkarinen. Unbounded Verification Results by Finite-State Compositional Techniques: 10^{any} States and Beyond. In *Proceedings of International Conference on Application of Concurrency to System Design*, pages 75–85. IEEE Computer Society, March 1998.