ON ROUTING CONSTRAINTS IN PACKET SWITCHING NETWORKS

WILFRIED BRAUER and WALTER VOGLER

Institut für Informatik, Technische Universität München
Arcisstrasse 21, D – 8000 München 2

(Received December 18, 1985)

Abstract. In his doctoral dissertation W. Wimmer has developed and implemented (in the DESYNET) a method of imposing static routing constraints upon packet switching networks which allows store-and-forward lock-up prevention in addition to avoiding ping-ponging and looping behaviour. The method is based on graph theoretic considerations, which are studied in more detail in the present paper; in particular the problems of constructing appropriate constraints and of minimizing their number are treated and some conjectures due to W. Wimmer are proved.

1. Introduction

The work about deadlock prevention in computer networks has been summarized, classified, and has been given a unified theoretical background by Günther [2]. Using his results Wimmer ([3], [4]) introduced two closely related concepts of static routing constraints which, combined with certain resource access constraints, allow the prevention of store-and-forward lock-ups. If routing tables and routing algorithms are constructed in accordance with the routing constraints no packet will travel around a cycle more than once. Both concepts are graph-theoretically based, and in the following we will deal with the graph-theoretic problems only. In fact, the *routing constrained digraphs* (RCD), as we call them, are just a special case of the *routing constrained switching networks* (RCSN). They deserve our attention because they allow better use of the resources (i. e. buffers). In Section 2 we introduce RCSN's and RCD's, in Sections 3 and 4 we study their relation and the question of minimizing the routing constraints, As is shown in [1] it is important that the routing constraints are placed in such a way that between any two nodes a shortest path of the unconstrained graph may be used. This property of distance preservation will be studied in Section 5.

2. Routing constrained switching networks and routing constrained digraphs

2.1. Definition. A graph with barriers is a pair (G, S) where G is a graph and S a reflexive, symmetric relation on the edges of G holding for adjacent edges only. For $(e, e') \in S$ we write $e | e'$ as well and if $e \neq e'$ we say that there is a *barrier* between e and e', which is the same as the one between e' and e. An *S-walk* in (G, S) is a walk in G such that for consecutive edges e, e' $(e, e') \notin S$. (G, S) is *S-connected* if for any two nodes v, w there is an S-walk from v to w. (G, S) is *cycle-separated* if no S-walk uses the same edge twice in the same direction. An S-connected, cycle-separated graph (G, S) is called a *routing constrained switching network (RCSN)*. □

Remark. Cycle-separated means exactly that one cannot walk around a cycle (or rather closed walk) twice.

2.2. Example. One might conjecture that a graph with barriers, (G, S), for which S is minimal among the relations that give a cycle-separated graph (G, S) is automatically S-connected. This is not true as the graph in Figure 1 shows.

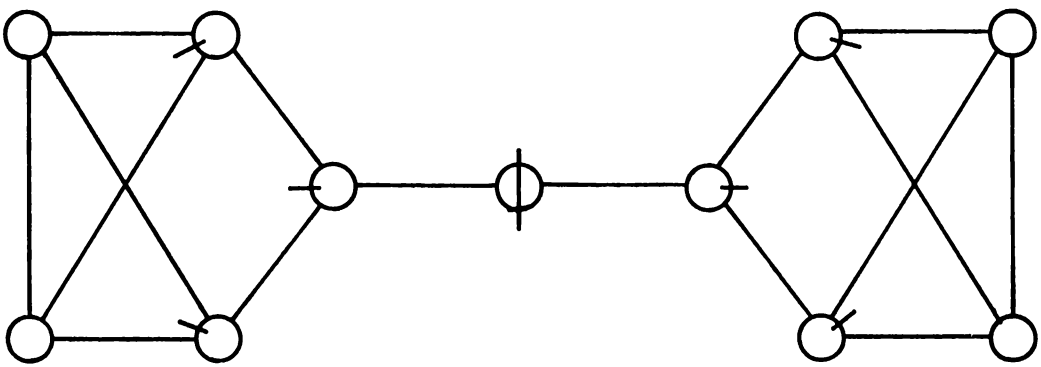

Fig. 1.

The indicated barriers give a cycle-separated graph with barriers which is not S-connected. On the other hand in any RCSN there is at least one node v and one edge e, such that v is a dead end for e, i. e. an S-walk using e has to end at v. Otherwise there would be an S-walk using the same edge twice in the same direction. If we have an RCSN on the graph of Fig. 1 only a node of degree 3 might be a dead end for some e, hence 2 barriers are needed to make some v a dead end for some e; removal of e and the two barriers gives a cycle-separated graph with barriers and by the proof of (4.1) below there have to be at least 5 barriers. Therefore any RCSN on the graph of Fig. 1 must have at least 7 barriers.

2.3. Definition. Let D be an acyclic digraph having exactly one node of indegree 0 (the start node) and let the relation S on the edges of the underlying graph G be defined by $S := \{(e, e') \mid e, e' \text{ point to the same vertex in } D\}$. Then (G, S) is called a *routing constrained digraph* (RCD).

Remark. The underlying graph of an RCD is necessarily connected, since each component of an acyclic digraph has a node of indegree 0. The S-walks of an RCD are those walks of the underlying graph that after using some arc following its direction never use any arc opposite to its direction and that do not use the same arc twice in a row. Thus an S-walk may use an arc opposite to its direction but only in the beginning.

2.4. Proposition [3]. *An RCD is an RCSN, i. e.*

(i) *an RCD is S-connected,*

(ii) *an RCD is cycle-separated.*

Proof. (i) There are directed paths P_V, P_W from the start node to v and w respectively. Let x be the last common node of P_V, P_W. Then using P_V from v to x and P_W from x to w we get an S-walk from v to w.

(ii) If an S-walk uses the same arc twice in the same direction, then (see the remark above) it uses this arc and the arcs in between all following their direction or all opposite their direction, i. e. the digraph is not acyclic. □

3. Minimal and reduced RCD's and RCSN's

3.1. Definition. An RCSN (G, S) is called *transitive* if S is transitive. It is called *reduced (t-reduced),* if there is no proper subset (transitive subset) S' of S such that (G, S') is an RCSN (transitive RCSN). It is called *minimal (t-minimal)* if there is no RCSN (transitive RCSN) on G with a smaller number of barriers. □

We are interested in these types of RCSN's because RCD's, which allow better use of the resources (buffers), give rise to transitive RCSN's, and because we want to minimize the routing constraints. The following propositions deal with the relation between the various types of RCSN's.

3.2. Proposition. *For each transitive* RCSN N *an* RCD N' *can be constructed by deleting some barriers.*

Proof. (1) In any RCSN there is at least one node v and one edge e such that v is a dead end for e.

(2) In a transitive RCSN a node v, which is a dead end for one edge e, is also a dead end for all edges at v, i. e. it is a total dead end. This can be seen as follows. Take any edge e' at v. Then $e|e'$, and for any other edge e'' we have $e''|e$, since v is a dead end for e. By transitivity then $e''|e'$.

(3) If a total dead end v together with all edges at it is deleted from a transitive RCSN N the remaining graph N' is connected. N' together with the separation relation restricted to the remaining edges is again a transitive RCSN, since in N the transport of an object from one node $\neq v$ to another node $\neq v$ cannot pass through v.

(4) The following (nondeterministic) algorithm accomplishes the construction:

(i) Take a total dead end v of N.

(ii) Give all edges joining v an orientation towards v and delete the barriers in v.

(iii) Denote the RCSN obtained by deleting v from N and restricting the separation relation again by N and continue with (i) until N contains only one node.

(5) The algorithm in (4) obviously always terminates by delivering an acyclic digraph whose start node is the node considered last. If we put barriers between all the arcs pointing to a node we get barriers which were present in the original RCSN. □

Remark. The basic idea of the above proof is due to Ute Brauer.

3.3. Corollary. (i) *A t-reduced* RCSN *is an* RCD.

(ii) *Each t-reduced* RCSN *can be constructed by the following method. Start with some node* v_1. *Add some node* v *and edges incident with* v *to the* RCSN N *and put barriers in* v *between all these edges.*

Remark. Not all RCSN's obtained by the construction in (ii) are t-minimal.

Obviously in the corollary instead of "t-reduced" we can also write "t-minimal"; in this version the first part of the corollary was conjectured by W. Wimmer.

Viewing the algorithm of (3.3) from a different angle we get an algorithm to construct a t-reduced RCSN for a given graph G.

3.4. Corollary. (i) *The following construction gives exactly the t-reduced* RCSN's *on a given graph G: If G has one node only we are done. As long as there is more than one node choose a node v such that G-v is connected. Put barriers between all edges at v and put G:* $= G - v$.

(ii) *For any graph G there is an* RCD *on G.*

The sum over the degrees of the chosen nodes is the number of edges of G, the number of barriers added in each step is quadratic in the degree of v. In order to minimize the number of barriers one has to have the degrees equally distributed (see Wimmer [3]). One might conjecture that it cannot be wrong to choose any node of minimal degree first to get a t-minimal RCSN – deletion of some nodes reduces the degrees of the remaining nodes. The following example (Fig. 2.) proves this to be wrong.

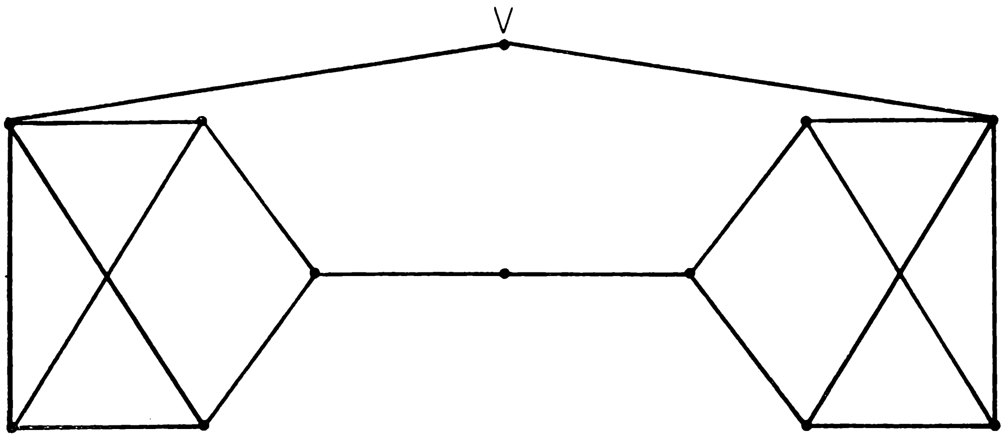

Fig. 2.

3.5. Example. If we choose v first we will need 7 barriers for the remaining graph (compare Example (2.2)), that is we have 8 barriers all in all. But the 7 barriers shown in Fig. 1 give a t-minimal RCSN.

This example indicates that "local reasoning" is not enough and therefore no "good" algorithm exists.

3.6. Problem. Find a good algorithm to determine a t-minimal RCSN for a given graph G or show that there is none.

3.7. Proposition. *An* RCD (G, S) *is reduced.*

Proof. The proof is by induction on the number of nodes of G. The proposition is trivial for $|V(G)| = 1$. For a given (G, S) we can find a total dead end v as above, $(G, S)-v$ is an RCD, hence reduced by induction. An edge vw has barriers at v only. Let w, w' be neighbours of v. Then there is an S-walk from w to w' in the RCD $G-v$. This S-walk together with edges $w'v$, vw forms a closed walk which is only separated by the barrier between $w'v$ and vw. Hence no barrier is superflous. □

3.8. Corollary. *An* RCSN *is t-reduced if and only if it is an* RCD. *A t-reduced* RCSN *is an* RCD.

Remark. The above proposition does not hold with (t-) reduced replaced by (t-) minimal. Consider the graph of Fig. 3.

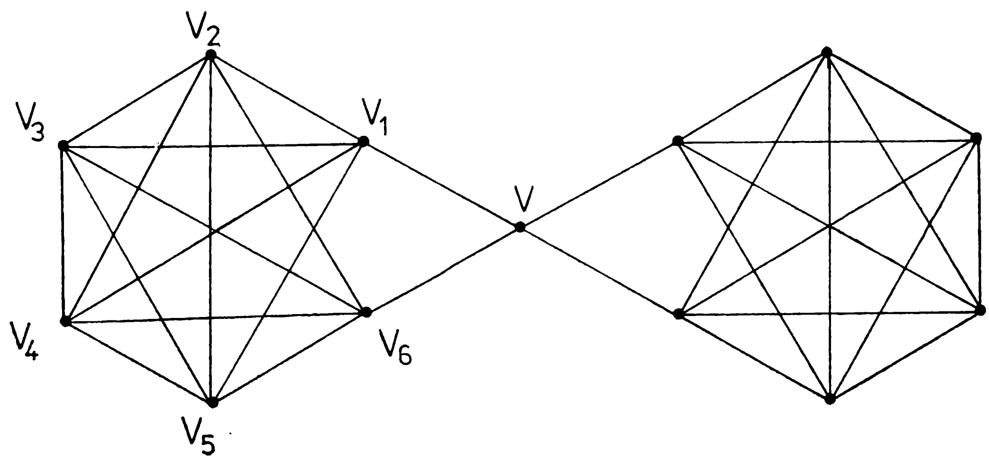

Fig. 3.

Apply (3.4). Before choosing v we have to choose all nodes "on the left" or all "on the right". So we can do nothing better than choosing $v_1, \ldots, v_6$ and go on in an optimal way. The result can be improved by deleting the 4 barriers of vv_1 at v_1 and introducing 3 barriers for vv_1 at v instead.

4. Absolutely minimal RCSN's

It is well-known from graph theory that the minimal number of independent (elementary) cycles in a graph with n nodes and m edges is the cyclomatic number $m-n+1$. This gives a lower bound for the number of barriers of an RCSN for some given graph.

4.1. Proposition. Let G be a graph with n nodes and m edges, (G, S) an RCSN. Then the number of barriers is at least $m-n+1$.

Proof. We prove more generally by induction on m. If (G, S) is cycle-separated then there are at least $m-n+1$ barriers. For $m=0$ this is true. For $m \geq 1$ choose an edge e that is a dead end at v. $(G, S)-e$ is cycle-separated as well.

Either v has degree one, and then $(G, S)-v$ has at least $(m-1)-(n-1)+1 = m-n+1$ barriers or e has at least one barrier at v and $(G, S)-e$ has at least $m-1-n+1$ barriers. In both cases we are done. □

4.2. Definition. An RCSN (G, S) with G having n nodes and m edges is called *absolutely minimal* if its number of barriers is $m-n+1$.

W. Wimmer has conjectured the following

4.3. Proposition. *An* RCSN *is absolutely minimal if and only if it can be constructed in the following way. Start with one node and connect new nodes to the* RCSN *by at most two edges and in case of two edges separate them by a barrier in the new node.*

Proof. This method obviously gives an absolutely minimal RCSN. On the other hand given an absolutely minimal RCSN we can find a node v as in the proof of (4.1). This node must have degree 1 or 2, since otherwise e would have more than one barrier and (G, S) would have at least $(m-1-n+1)+2 = m-n+2$ barriers. Hence v is a total dead end and we can build up $(G, S)-v$ and add v according to the above algorithm. □

Remark. The difficulty of constructing a good algorithm for finding an absolutely minimal RCSN for a given graph is again shown by example (3.5).

W. Wimmer also conjectured the following

4.4. Corollary. (i) *In an absolutely minimal* RCSN *no node has more than one barrier and for each edge there is at most one node containing a barrier for it. In particular each absolutely minimal* RCSN *is an* RCD *such that in the corresponding digraph there are never more than two arcs pointing to a node.*

(ii) *If successive deletion of nodes of degree 1 from a graph finally gives a graph which has no node of degree 2, then there is no absolutely minimal* RCSN *over that graph.*

5. Distance-preserving RCSN's

5.1. Definition. We call an RCSN (G, S) *distance-preserving* if for any two nodes v, w there is a shortest path in G that is an S-walk in (G, S).

Wimmer gave the following construction method.

5.2. Proposition. *A t-reduced* RCSN *is distance-preserving if and only if it can be constructed in the following way. Start with one node. Then repeatedly choose a non-empty set of nodes that pairwise have a common neighbour or are adjacent and add a new node adjacent to all nodes of the set. Separate all edges at the new node by barriers.*

Remark. This method can be combined with that of (4.3) to give a necessary and sufficient construction method for RCSN's which are at the same time distance-preserving and absolutely minimal.

5.3. Example. In (3.3) we desrcibed an algorithm for constructing a t-reduced RCSN that viewed "the other way round" in (3.4) gave an algorithm to construct a t-reduced RCSN for a given graph. We would like to convert the algorithm of (5.2) similarly. But the graph of Fig. 4 shows that it is difficult to find a distance-preserving t-reduced RCSN for a given graph.

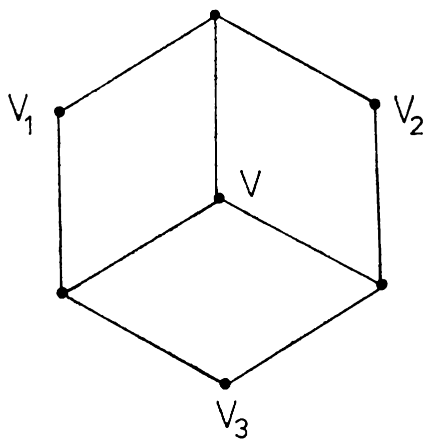

Fig. 4.

Locally it seems to be perfectly all right to choose v as a total dead end since its neighbours have distance 2 in $G-v$. But in $G-v$ there is no node with this property. Had we chosen v_1, v_2 and v_3, we would have got a t-reduced distance-preserving RCSN.
The following two propositions may be of some help in this situation.

One might expect that if there is a distance-preserving RCSN for a given graph then any minimal RCSN will be distance-preserving. This is not true in general. But we have the following result.

5.4. Proposition. *Let G be a graph for which a distance-preserving t-reduced* RCSN *R exists. Then every absolutely minimal* RCSN *R' on G is distance-preserving.*

Proof. The proof will be by induction on the number of nodes. Let R be constructed according to (5.2) and v a total dead end of R'. First we show that we may assume that v is a total end of R.

W. l. o. g. v has degree 2 by (4.3), let w, w' be the neighbours. We have three cases:

i) Constructing R according to (5.2) v is added before w and w'. This means that v is the first node and w, say, the second. We can reverse the order of v and w in the construction and thus reduce this case to ii).

ii) w is added before v, w' after v.

We have two subcases:

α) As w' is added it gets another neighbour x. This either equals w or has a common neighbour with v, i. e. it is adjacent to w. Therefore we can add v after w' and have case iii).

β) As w' is added it gets v as its only neighbour. Hence v is a separating node for some of the graphs constructed intermediately. Since v is a total dead end of R' it is not a separating node of G. Let x be the first node added such that v does not separate the resulting graph, and let G' be the graph x is added to. $G'-v$ has two components C_1, C_2 and x has neighbours in both. By (5.2) these neighbours have to be w and w'. Hence we can add x instead of v and v instead of x and again have case iii).

iii) v is added after w and w'.

By construction v is a total dead end of R.
Now $R-v$ is a distance-preserving t-reduced RCSN, $R'-v$ an absolutely minimal RCSN. By induction $R'-v$ is distance-preserving. Since v is a total dead end of R, w and w' have distance at most 2 in $G-v$. Hence R' is distance-preserving by (5.2). □

Constructing a distance-preserving t-reduced RCSN for a given graph chordless cycles of length greater 4 can pose a problem. But such a construction is easy if there is no such cycle. To see this we need a lemma.

5.5. Lemma. *Let G be a graph without chordless cycles of length greater 4. Then there is a node v such that any two neighbours of v have distance at most 2 in $G-v$.*

Proof. Suppose the lemma fails. Then a node v has neighbours w, w' with distance greater than 2 in $G-v$. Any path from w to w' in $G-v$ contains a chordless path which together with $w'vw$ forms a cycle of length greater than 4, hence is contains another neighbour of v. Therefore $\{v\} \cup \{y \mid vy \in E(G) \wedge \wedge y \neq w \wedge y \neq w'\}$ separates G. Among the separating node sets consisting of one node, called the centre node, and some of its neighbours choose a set T with centre node x such that $|T| + |V(C)|$ is minimal, where C is a component of $G-T$.

i) All nodes of C are adjacent ot x. Choose v to be any of them. Then any neighbour of v is x or adjacent to x and we are done.

ii) There is a node v of C not adjacent to x. As above assume that two neighbours w, w' have distance greater than 2 in $G-v$.

$T' := \{v\} \cup \{y \mid vy \in E(G) \wedge y \neq w \wedge y \neq w'\}$ separates w and w' in G. Observe that $T' \subseteq T \cup V(C)$. W.l.o.g. the component C' of $G-T'$ containing w does not contain x. Then it does not contain any node of T either, so $C' \subsetneq C$. Thus $T' \cup V(C')$ is a proper subset of $T \cup V(C)$ since it does not contain x, contradicting the nimimality. Hence we can choose v. □

5.6. Proposition. *Let G be a connected graph without chordless cycles of length greater than* 4. *Then each distance-preserving t-reduced* RCSN *on G can be constructed by the following algorithm, which always terminates. As long as there is more than one node choose a node v such that for any two neighbours w, w' of v the distance of w and w' in $G-v$ is at most* 2. *Put barriers between all edges at v and put $G := G-v$.*

Proof. Since the assumption about G holds for all subgraphs as well there is always such a node v by (5.5). The result follows by (3.4) and (5.2). □

6. Conclusions

We have seen that a given graph can be converted into an RCD or an RCSN of various types and how t-reduced, absolutely minimal or distance-preserving RCSN's can be constructed. However, no good algorithm for finding such RCSN's for a given graph is known.

For practical applications additional considerations have to be made.

E.g. the flow of objects (packets) in the network, which might be different in different parts of it, must be taken into account by putting a minimal number of barriers into the regions with heavy traffic; also on should avoid cutting off shortest paths between important nodes. This poses some further graph theoretic problems – some of them have been solved by Wimmer [2].

And there is the problem that during operation the network may change because of breakdowns of some of its components of because of the need to add new nodes or edges. These questions can also be nicely handled by the methods presented above (see also [2]).

For the practical implementation of a method to construct an RCD for a given network and the corresponding algorithms for buffer management, flow control, routing, network changements etc. see [2], too.

REFERENCES

[1] *Frank H.* and *Chou W.*: Routing in computer networks. *Networks* **1** (1971), 99–112.

[2] *Günther K. D.*: Prevention of deadlocks in packet-switched data transport systems. *IEEE Trans. Commun.* **COM–29** (1981), 512–524.

[3] *Wimmer W.*: Ein Verfahren zu Verhinderung von Verklemmungen in Vermittlernetzen, Ph. D. Thesis, Univ. Hamburg, Hamburg, 1982.

[4] *Wimmer W.*: Using barrier graphs for deadlock prevention in communication networks. *IEEE Trans. Commun.* **COM–32** (1984), 897–901.