数理解析研究所講究録
第625巻 1987年 38-47

優先度付きトークンリング方式ＬＡＮの「進行性」の検証について

大阪大学基礎工学部情報工学科　住谷忠雄　（ Tadao Sumitani ）
大阪大学基礎工学部情報工学科　東野輝夫　（ Teruo Higashino ）
大阪大学基礎工学部情報工学科　谷口健一　（ Kenichi Taniguchi ）

あらまし　本稿では，まずトークンリング方式ＬＡＮプロトコルの規格をパケットレベルで整理し，リング全体を抽象的な”順序機械”とみなして形式的な仕様記述を行っている．次にその仕様が，「相互排他性」「一巡性」「進行性」を満たすことをテンポラルロジックの概念を用いて証明している．証明に際しては，まず，（１）ライン上にパケットが存在すればいつかは受信動作が行われること等，各性質の証明の前提となる条件をテンポラルロジックの式の形で表す．次に，（２）仕様から直接証明できる性質を一定の規則に従ってテンポラルロジックの式に変換し，（３）「進行性」など各性質を表すテンポラルロジックの式が真であることを，（１），（２）で導入した式およびテンポラルロジックの推論則，ならびに非負整数に関するいくつかの性質のみを用いて証明する．

１．まえがき

近年ネットワーク技術の発展に伴い，ローカルエリアネットワーク（ＬＡＮ）の重要性が高まってきている．トークンリング方式ＬＡＮには，緊急度の高いデータを優先的に伝送させるためのメカニズムが設けられており，そのメカニズムは複雑である．また，提案された一つのプロトコルにおいて，それが満たすべき性質が成立しているか否かは，一見単純そうなプロトコルの場合でも，必ずしも簡単ではない．従来，我々の研究グループでは，トークンリング方式ＬＡＮにおいて成立すべき静的な性質の検証を行ってきた[1]．また，Stenningのデータ転送プロトコルを例にして，動的な性質の検証に有効な手段を検討してきた[2]．本稿では，[2]で検討した手法を応用して，トークンリング方式ＬＡＮにおいて成立すべき動的な性質の検証を行っている．検証に際しては，まずトークンリング方式ＬＡＮプロトコルの規格[3]をパケットレベルで整理し，リング上の各局を抽象的な”順序機械”，局と局を結ぶラインをＦＩＦＯのキューとしてモデル化し，各局が送受信等の動作を行うことによって，”順序機械”やキューの内容がどのように変化するかを記述することによって形式的な仕様記述を行っている．次にその仕様が，「相互排他性」（リング上で発信権を持つ局は高々一つである）や「一巡性」（局から発信されたフレームは必ずリングを一巡し，目的の局にその内容が伝えられ，発信局によりリング上から取り除かれる）や「進行性」（ある時点以降，プライオリティがｋ以上のＰＤＵ（プロトコルの上位層から渡される発信すべきデータ）の追加がすべての局で行われなければ，各局が待ち行列中に保持していたプライオリティｋ以上のＰＤＵは，すべていつかはリングに送出される）等の性質を満たすことを証明している．

トークンリング方式ＬＡＮプロトコルでは，Ｎ個の局が一方向にリング状に接続されており，本稿では，リング全体を以下で述べるＮ個の局及びそれらの局間を結ぶＮ個のラインで構成されていると考え，すべての局及びラインの２Ｎ字組で表す．又，各局は，（Ａ）局の動作を制御する順序機械，（Ｂ）上位レベルから渡されるＰＤＵを保持するプライオリティ付キュー，（Ｃ）トークンのプライオリティの上昇の記憶，及び下降の制御に用いるスタック（レベル上昇スタック），（Ｄ）発信局が獲得したトークンのプライオリティ（PRIORITY）を保持するレジスタ，（Ｅ）発信局が受信したフレームのリザベーション（RESERVATION）を保持するレジスタ，の５字組で表し，各ラインはいわゆるFIFOのキューとしてモデル化している．

トークンリング方式ＬＡＮでは，各局がパケットの送信や受信，プライオリティ付キューへのＰＤＵの追加等の動作を行うことによって通信が進められる．そこで本稿では，各局のパケットの送受信動作，ＰＤＵ追加動作を「状態遷移関数」として表す．また，最後にある状態遷移関数を実行して現在のリングの２Ｎ字組の内容になったとする．このとき最後に実行した状態遷移関数を「最終状態遷移関数」と呼ぶ．

次にリングの「状態」，及び「初期状態」を定義する．２Ｎ字組のリングの内容，及び最終状態遷移関数の２Ｎ＋１字組をリングの「状態」と呼ぶ．通常，トークンリング方式ＬＡＮでは，リング上にトークンが１個だけあるような状況から動作を開始する．そこで，リング上にトークンが１個だけあるような状況を一つ定め，それをリングの「初期状態」とする．

リング全体の仕様は，上述のようなモデルに対して，次の（１）ー（３）を定義することによって表す．

（１）初期状態における各局の順序機械の状態，ＰＤＵキュー，スタック，レジスタの内容（初期値）及びラインキューの内容

（２）各局の状態，ＰＤＵキュー，スタック，レジスタ及びラインキューの内容がどのような場合に各状態遷移関数の実行が可能か

（３）実行可能な状態遷移関数を実行したときに，各局の状態，ＰＤＵキュー，スタック，レジスタ及びラインキューの内容がどのように変化するか

これより初期状態から実行可能な動作（状態遷移関数）を実行していった状態に対して，局及びラインの内容が定まる．

トークンリング方式ＬＡＮでは，高いプライオリティのＰＤＵを持つ局が早くトークンを獲得できるメカニズムをもつ．各局はトークンのプライオリティの上昇・下降のためにレベル上昇スタックを使う．プライオリティを上昇させる局は，どの値からどの値へ上昇させたかを対にして自局のレベル上昇スタックにプッシュし，プライオリティを下降させるときにレベル上昇スタックのトップの要素をポップし，その内容を見てプライオリティをどの値まで下げるかを決める．リング上のＮ個の局が，このようなスタック操作を独立・並行して行うことによってリング全体におけるプライオリティの上昇・下降の制御を行っている．進行性の証明は，このスタック操作が関連しているのでかなり複雑になっている．

このモデルにおいて，「相互排他性」や「一巡性」や「進行性」といった性質は初期状態から実行可能な状態遷移関数を１個ずつ実行していったような状態の系列に対する性質である．これらの性質は，「ずっと・・・」や「いつかは・・・」や「ある時点以降・・・」といった性質の表現であるので，テンポラルロジックの概念を導入し，「相互排他性」や「一巡性」や「進行性」を□（always），◇（eventually），○（next）等のオペレータを用いてテンポラルロジックの式の形で表す．又，ライン上にパケットが存在すれば，いつかは受信動作が行われること等，「一巡性」や「進行性」を証明するために当然前提とすべきリング全体に課す条件についてもテンポラルロジックの式の形で表す．そして，仕様から直接証明できる性質（基礎的補題）を状態遷移に関する帰納法を用いて証明し，一定規則に従ってテンポラルロジックの式に変換した後に，リング全体に課す条件の下で本稿で与えた仕様が上述の３つの性質をみたすことを，導入した基礎的補題とテンポラルロジックのオペレータに関する推論則，及び非負整数の性質のみを用いて形式的に証明する．

２．トークンリング方式ＬＡＮ

２．１　動作概略

以下，トークンリング方式ＬＡＮにおける送信権の割当等アクセス制御に関する部分の概略について述べる．このＬＡＮは一本の閉じたケーブル（データが一方向にしか流れない）とそれに直列に接続された複数の局からなる．各局は上位レベルからプロトコル・データ・ユニット（ＰＤＵ）と呼ばれるデータを渡され，ＰＤＵは局内で待ち行列をつくる（以下，この待ち行列をＰＤＵキューと呼ぶ）．送信局は一つのＰＤＵから一つのフレームを構成し，そのように構成した一連のフレームを次々と下流に向かって送信する．他の局（中継局）は上流から流れてきたデータをそのまま下流に流す（但し，もし自局宛のデータであれば，そのデータを自局上にコピーしておく必要がある）．リングを一巡して送信局まで戻って来るデータは送信局によりリングから除去される．ここで，送信局は一つの局に固定されているわけではなく動的に変化すべきものであり，複数の局が同時に送信局になることがないように，送信権の割当てを行っている．

トークンリング方式のＬＡＮでは，ＰＤＵから生成される伝送単位「フレーム」の他に，送信権を表わす伝送単位「トークン」を導入し，以下のように送信権の割当てを行なう．

(1) どの局も送信権を持っていないとき，一つのトークンがリング上を巡回している．

(2) 発信要求を持つ局がトークンを受信すると，これを取込み送信権を得る．

(3) 送信権を得た局は，待ち行列に保持されているＰ

ＤＵをそれぞれフレームの形にして下流に向かって送信する．

(4) 幾つかのフレームを送信した後，トークンを下流に向かって送信することにより送信権を解放し，(1) の状況を再現する．ここで，トークンを送信した後においても，自局が送信したフレームについては，リングを一巡して戻ってきたときリングから除去する．

２．２　プライオリテイを用いた送信権の割当て

上位レベルから渡されるＰＤＵには，そのデータの送信緊急度を示すプライオリテイ（PRIORITY）が設定されており，プライオリテイの高いデータを優先的に伝送させるためのメカニズムが要求される．

IEEE 802.5では，各局でのＰＤＵの待ち行列（ＰＤＵキュー）としてプライオリテイの高いものが先頭にくる，いわゆるプライオリテイ付き待ち行列を採用し，各局ではプライオリテイの大きいＰＤＵから順に送信する．トークンにPRIORITYの値を保持するプライオリテイ表示部を設け，PRIORITYがN のトークンはN 以上のＰＤＵの送信にのみ利用できるとする．即ち，N-1 以下のプライオリテイのＰＤＵしか持たない局は，そのトークンを取り込むことが許されず，PRIORITYがNのトークンを獲得した局は，自局中にあるＰＤＵのうちプライオリテイＮ以上のものが送信できる．トークンのPRIORITYの値の設定は以下に述べるプライオリテイの予約を用いる方法によって定められる．

フレーム及びトークン中に RESERVATIONの値を保持するプライオリテイ予約部を設けるが，まず，フレーム中の RESERVATIONの使い方を述べる．送信局はフレームを送信するとき， RESERVATIONの値としてプライオリテイの下限１をプライオリテイ予約部にセットしておく．各中継局は自局に到達したフレーム中の RESERVATIONの値とその時点で自局で待たされているＰＤＵの最高プライオリテイ(maxlevel)を比較し，maxlevelの値が RESERVATIONの値より大きい場合，その RESERVATIONの値をmaxlevelの値に書き換えてから，下流に流す．このようにすると，フレームがリングを一巡して送信局に戻ってきたとき，リング上の各局で待たされているＰＤＵ中でプライオリテイが RESERVATION以上のものが必ず存在する（フレームが通過した以降さらに大きいプライオリテイのＰＤＵが追加された場合を除いて，各局で待たされているＰＤＵ中でプライオリテイの最大のもののプライオリテイが書き込まれている）．送信局が送信権を解放するときに送信するトークン中のPRIORITYの値は，その局が送信権を得た（トークンを獲得した）時のPRIORITYの値（pri で表わす），最近に到着したフレーム中の RESERVATIONの値（res で表わす），及び自局のＰＤＵキュー中にあるＰＤＵの最大プライオリティー（maxlevelで表す）の最大値とする． resとmaxlevelの最大値 （req で表わす）はその時点で要求されるプライオリテイであり，pri < req のとき，その局が「リングのサービスレベルをpri からreq へ上昇させた」という．PRIORITYの値がpri のトークンを出すときは，サービスレベルは変化しない．一般に，サービスレベルの上昇は，「局a が 1から2 に上昇させる」「局b が 2から4 に上昇させる」といった具合にネストして起る．リングのサービスレベルを上げた局は，そのレベルで送信できるＰＤＵがどの局にも存在しなくなったとき，サービスレベルを下げなければならない．サービスレベルを上昇・下降の制御のため，各局にレベル上昇スタックを用いる．サービスレベルを上昇させる局は，どの値からどの値へサービスレベルを上昇させたかを対にして，自局のレベル上昇スタックに記憶し，サービスレベルを下げなければならない時に，スタックのトップの要素をポップし，その内容をみてサービスレベルをどの値まで下げるかをきめる．

以上がトークンリング方式ＬＡＮの概略である．

３．局，リングの仕様

３．１　リングの仕様

トークンリング方式ＬＡＮは，Ｎ個の局をリング状に接続している．本稿では，リング全体を抽象的に次のように定義する．

まずリングに接続されているＮ個の局をそれぞれ局１，局２，…，局Ｎと呼ぶ．各局ｎは，次の５字組Ｓｎ，ＰＱｎ，ＳＴｎ，Ｐｒｎ，Ｒｒｎで表す．

Ｓｎ：局の動作を制御する順序機械で，順序機械の状態は以下の８状態のいずれかである．

①REPEAT：中継局であることを表す状態．

②F_TRANS_NO_RECEIVED： 発信局になって以降，フレームを受信していなく，かつ最終フレームを発信していない状態．

③F_TRANS_SOME_RECEIVED： 発信局になって以降，中間フレームを１つ以上受信，かつ最終フレーム

を発信していない状態.

④FRAME_WAITING： 発信局になって以降，フレームを受信していなく，かつ最終フレームを発信済みの状態.

⑤T_TRANS_L_NOTRECEIVED： 発信局になって以降，中間フレームを１つ以上受信，かつ最終フレームを発信済み，かつ最終フレームを受信していなく，かつトークンを発信していない状態.

⑥T_TRANS_ALL_RECEIVED：発信局になって以降，最終フレームを発信済み，かつ最終フレームを受信済み，かつトークンを発信していない状態.

⑦STRIP： 発信局から中継局への遷移局であり，発信局になって以降，トークンを発信済みであるが最終フレームを受信していない状態.

⑧ERROR： 予期しないパケットが入力された場合に陥る特別な状態.

ＰＱn： 上位レベルから渡されるＰＤＵを保持するプライオリティ付キュー.

プライオリティ付キューは，プライオリティが非負整数（実際には１から８までの８段階）で構成されているとし，プライオリティの高いＰＤＵから順に取り除かれる.

プライオリティ付キューに対する動作として次の３つを考える.

① EMPTY_P_QUEUE：空プライオリティ付キューを表す.

② ENQUE_P(q,d,p)： 現在のプライオリティ付キューの内容がｑのとき，プライオリティｐ，データｄのＰＤＵをキューに付け加えたあとのキューの内容を表す.

③ DEQUE_P(q)： 現在のプライオリティ付キューの内容がｑのとき，最もプライオリティが高く且つもっとも以前に付け加えられたＰＤＵを一個取り除いたあとのキューの内容を表す.

また，プライオリティ付きキューに対する関数として次の２つを考える.

① maxlevel(q)：プライオリティ付きキューに保持されているＰＤＵの最高プライオリティを表す関数.

② top_P(q)： プライオリティ付きキューの内容がｑのとき，最もプライオリティが高くかつ最も以前に付け加えられたＰＤＵのデータを表す関数.

ＳＴn： トークンのpriorityの値を上昇させたことを記憶するスタックであり，スタックの内容は，＜上昇前のトークンのpriority，上昇後のトークンのpriority＞の２字組の要素からなる.このスタックのことをレベル上昇スタックと呼ぶ.

レベル上昇スタックに対する動作として次の４つを考える.

① NEWSTACK： 空スタックを表す.

② PUSH(s,<p,q>)：現在のスタックｓに２字組＜ｐ，ｑ＞を追加したあとのスタックの内容を表す.

③ POP(s)：現在のスタックｓから先頭の要素１個＜ｐ，ｑ＞を取り除いたあとのスタックの内容を表す.

④ EXCHANGE(s,q)：現在のスタックｓの先頭の要素＜ｐ，ｒ＞を＜ｐ，ｑ＞に交換した後のスタックの内容を表す.

また，レベル上昇スタックに対する関数として次の２つを考える.

①top1(s)：現在のスタックｓの先頭の要素＜ｐ，ｑ＞のうち第一要素ｐを表す関数.

②top2(s)：現在のスタックｓの先頭の要素＜ｐ，ｑ＞のうち第二要素ｑを表す関数.

Ｐrn：発信局が獲得したトークンのpriorityを保持するレジスタ.

Ｒrn：発信局が最近受信したフレームのreservationを保持するレジスタ.

又，順序機械Ｓnの状態ｓn，プライオリティ付キューＰＱnの内容ｐｑn，レベル上昇スタックＳＴnの内容ｓｔn，２つのレジスタＰrn，Ｒrnの内容ｐn，ｒnの５字組＜ｓn，ｐｑn，ｓｔn，ｐn，ｒn＞を局ｎの状態と呼ぶ.

次に局ｎから局ｎ＋１間のラインをラインｎ（但し，ラインＮは局Ｎから局１間）と呼ぶ．各ラインｎはいわゆるFIFOのキューＱｎで構成されていると考え，キューＱｎの内容ｑｎをラインｎの状態と呼ぶ．

以下キューに追加されるトークン，フレームを次のように表し，これをパケットと呼ぶ．

token(p,r)：プライオリティｐ，リザベーションｒのトークン．
Iframe(n,d,r)：局ｎから送信されたデータｄ，リザベーションｒの中間フレーム．
Lframe(n,d,r)：局ｎから送信されたデータｄ，リザベーションｒの最終フレーム．

ラインに対する動作として次の３つを考える．

① EMPTY_QUEUE：空キューを表す．
② ENQUE(q,p)：キューｑにパケットｐを追加したあとのキューの内容を表す．
③ DEQUE(q)：キューｑの先頭の要素を一個取り除いたあとのキューの内容を表す．

また，ラインに対する関数として次の３つを考える．

① first(q)：キューｑの先頭の要素を表す関数．
② exist(q,p)：キューｑに要素ｐが存在するか否かを表す述語．
③ token_number(q)：キューｑに存在するトークンの数を表す関数．

まえがきでも述べたように，リング全体は，Ｎ個の局（局１，…，局Ｎ）及びＮ個のライン（ライン１，…，ラインＮ）で構成されていると考え，すべての局，及びラインの状態の２Ｎ字組＜局１の状態，局２の状態，…，局Ｎの状態，ライン１の状態，ライン２の状態，…，ラインＮの状態＞で表す．

次に５種類の「状態遷移関数」を定義する．

状態遷移関数

RECEIVE(n) ： 局ｎがラインｎ－１の先頭パケットを受信する．
TRANS_TOKEN(n) ： 局ｎがトークンを発信する．
TRANS_IFRAME(n) ： 局ｎが中間フレームを発信する．
TRANS_LFRAME(n) ： 局ｎが最終フレームを発信する．
PDU_ENQUEUE(n,d,i) ： 局ｎがプライオリティ付キューにプライオリティｉ，内容ｄのＰＤＵを追加する．

次にリングの「状態」ならびに「初期状態」を定義する．２Ｎ字組のリングの内容，及び最終状態遷移関数αの２Ｎ＋１字組をリングの状態と呼ぶ．また，本稿では，ラインＮ上にトークンが１個だけあるような状況から通信を開始するとし，リングの初期状態を次のように定義する．

リングの初期状態

リングの初期状態（以下，INITIAL で表す）における各局ｎの状態＜ｓｎ，ｐｑｎ，ｓｔｎ，ｐｎ，ｒｎ＞，ラインｍの状態ｑｍ，及び最終状態遷移αを次のように定める．

ｓｎ ： REPEAT状態．
ｐｑｎ ： 空プライオリティ付キュー（EMPTY_PR_QUEUE）．
ｓｔｎ ： 空スタック（NEWSTACK）．
ｐｎ ： 任意のプライオリティ．
ｒｎ ： 〃 （以上１≦ｎ≦Ｎ）
ｑｎ ： １≦ｎ≦Ｎ－１なるｎに対しては空キュー（EMPTY_QUEUE）
ｎ＝Ｎなるｎに対してはｑｎ≙ENQUE(EMPTY_QUEUE,token(1,1))．
α ： 空系列

上で述べた状態遷移が実行可能かどうか，実行できるとき実行後の各局の状態＜ｓｎ，ｐｑｎ，ｓｔｎ，ｐｎ，ｒｎ＞（１≦ｎ≦Ｎ）及びラインの状態ｑｍ（１≦ｍ≦Ｎ）がどのように変化するかについての詳細については表１に付す．

表１では，第一列，第二列で，リングの状態がｓのときの局ｎの順序機械の状態ｓｎが８状態のうちのいずれであるかに対して，それぞれ実行可能な状態遷移を記述している．１つの状態遷移に対しても，リングの状態ｓによって状態遷移後のリングの状態が異なる場合がある．このときのリングの状態ｓの場合わけを第三列に記述している．第四列から第八列で，状態遷移後の局ｎの状態を，第九列，第十列で，状態遷移後のラインｎ－１，ラインｎの状態を記述している．ただし，状態遷移後の空白は，状態遷移前と変化がない

ことを，＊は，任意の順序機械の状態，及び任意の値を表している．

表１の仕様では，予期しないパケットが入力された場合（例えば，発信局でトークンを受信した場合など），局の順序機械の状態はERROR という特別な状態に陥り，以降，どんな勝手な動作でも許される（何を受信してもその状態に留まり，又，その状態に留まったまま何を発信してもよい）ように記述されている．

以上がリングの仕様である．

表１の仕様からわかるように，禁止されている動作は次の３種類である．

（A）パケットの発信が許されない時に（即ち，発信局でないのに）パケットの発信を行うこと．
（B）発信できるパケットがトークン（又はフレーム）に制限されているにもかかわらず，フレーム（又はトークン）を発信すること．
（C）ライン上にパケットが存在しないときに受信を行うこと．

初期状態から許された動作のみを何回か実行することによって到達できる状態であることを表す述語VALI

（１）VALID(INITIAL)は真，
（２）VALID(s)が真で，且つ状態ｓで動作αが許されている時のみαを実行したあとの状態（α・ｓで表す）で，VALID(α・ｓ）は真，そうでない時VALID(α・ｓ）は偽，

と定める．又，VALID(ｓ）が真，かつVALID(α・ｓ)が真の時，ｓ =>α・ｓと書く．

３．２　補助関数の導入

本稿では相互排他性や一巡性や進行性などの性質の検証のために，リングの状態ｓを引数に持つ次のような補助関数を導入する．前節で述べたように，リングの状態は，すべての局，ラインの状態，及び最終状態遷移関数の２Ｎ＋１字組＜局１の状態，局２の状態，…，局Ｎの状態，ライン１の状態，ライン２の状態，…，ラインＮの状態，最終状態遷移関数＞で表される．また，各局ｎの状態は，順序機械 Ｓnの状態sn，プライオリテイ付きキュー ＰＱnの内容 pqn，レベル上昇スタック ＳＴnの内容 stn，２つのレジスタＰrn，Ｒrnの内容pn，rnの５字組＜sn,pqn,stn,pn,rn＞で表され，各ラインｎの状態は，キュー Ｑnの内容qnで表される．そこで，以下では，stationS(s,n) で，リングの状態ｓにおける局ｎの順序機械 Ｓnの状態snを，stationPQ(s,n)で，リングの状態ｓにおける局ｎのプライオリテイ付きキュー ＰＱnの内容 pqnを，stationST(s,n)で，リングの状態ｓにおける局ｎのレベル上昇スタック ＳＴnの内容 stnを，stationP(s,n) で，リングの状態ｓにおける局ｎのレジスタ Ｐrnの内容pnを，stationR(s,n) で，リングの状態ｓにおける局ｎのレジスタ Ｒrnの内容rnを，line(s,n)で，リングの状態ｓにおけるラインｎのキュー Ｑnの内容qnを，last_action(s)で，最終状態遷移関数を表す． 以下，主な補助関数について説明する．なお，これらの補助関数はすべて３．１の仕様で記述した関数を用いて定義している（すでに定めた補助関数を組み合わせることもある）．

トークン所有？(s,n)：状態ｓで，局ｎがトークンを所有しているか否かを表す述語．

```
トークン所有？(s,n) =
  { stationS(s,n) = F_TRANS_NO_RECEIVED or
    stationS(s,n) = F_TRANS_SOME_RECEIVED or
    stationS(s,n) = FRAME_WAITING or
    stationS(s,n) = T_TRANS_L_NOT_RECEIVED or
    stationS(s,n) = T_TRANS_ALL_RECEIVED }
```

トークン発信可能(s,n)：状態ｓで，発信局ｎが，自局が発信したフレームを一つ以上受信済みか否かを表す述語．

```
トークン発信可能(s,n) =
  { stationS(s,n) = F_TRANS_SOME_RECEIVED or
    stationS(s,n) = T_TRANS_L_NOT_RECEIVED or
    stationS(s,n) = T_TRANS_ALL_RECEIVED }
```

token_number_in_station(s,n) ：状態ｓで，局１から局ｎの間でトークンを所有している局の数を表す関数．

```
token_number_in_station(s,n) =
  if n = 1
  then if トークン所有？(s,n)
       then 1 else 0
  else if トークン所有？(s,n)
       then token_number_in_station(s,n-1)+1
       else token_number_in_station(s,n-1)
```

token_number_in_line(s,n)：状態ｓで，ライン１からラインｎの間にあるトークンの数を表す関数．

```
token_number_in_line(s,n) =
  if n = 1 then token_number(s,n)
           else token_number_in_line(s,n-1) +
                token_number(s,n)
```

トークン所有局有？(s)：状態ｓで，リング上にトークンを所有している局が１つだけ存在し，かつ，ライン上にトークンがないか否かを表す述語．

```
トークン所有局有？(s) =
  { token_number_in_station(s,N) = 1 and
    token_number_in_line(s,N) = 0 }
```

トークンライン上？(s)：状態ｓで，リング上にトークンを所有している局が存在せず，かつ，ライン上にトークンが１つあるか否かを表す述語．

```
トークンライン上？(s) =
  { token_number_in_station(s,N) = 0 and
    token_number_in_line(s,N) = 1 }
```

バケット受信(s,n)：状態ｓで，局ｎがバケットを受信した直後であるか否かを表す述語．

```
バケット受信(s,n) =
  { last_action(s) = RECEIVE(n) }
```

トークン発信(s,n)：状態ｓで，局ｎがトークンを発信した直後であるかを否かを表す述語．

```
トークン発信(s,n) =
  { last_action(s) = TRANS_TOKEN(n) }
```

プライオリテイq以上のPDUの追加なし(s,q)：
　状態ｓで，すべての局に対して，プライオリテイｑ以上のＰＤＵの追加を行った直後でないか否かを表す述語．

```
プライオリテイq以上のPDUの追加なし(s,q) =
  ∀n,d,i[ ( i >= q ) ⊃
          { not( last_action(s) =
            PDU_ENQUEUE(n,d,i) ) } ]
```

max.sub(s,m,n)：状態ｓで，局ｍから局ｎまでのプライオリティ付キューに保持されているＰＤＵのプライオリティの最高値を表す関数．

```
max.sub(s,m,n) =
  if m = n then maxlevel(stationPQ(s,n))
           else max(max.sub(s,m,n-1),
                    maxlevel(stationPQ(s,n)))
```

max.r(s)：状態ｓで，リング上のすべての局のプライオリティ付キューに保持されているＰＤＵのプライオリティの最高値を表す関数．

```
max.r(s) =
  max.sub(s,1,N)
```

プライオリテイq以上のPDUなし(s,q)：状態ｓで，リング上のすべての局のプライオリテイ付きキューに保持されているＰＤＵのプライオリテイの最高値がｑ未満であるか否かを表す述語．

```
プライオリテイq以上のPDUなし(s,q) =
  { max.r(s) < q }
```

ラインにバケット存在(s,n)：状態ｓで，ラインｎにバケットが存在するか否かを表す述語．

```
ラインにバケット存在(s,n) =
  not( line(s,n) = EMPTY_QUEUE )
```

ライン上にトークン(s,n,p,r)：状態ｓで，ラインｎがプライオリテイｐ，リザベーシヨンｒのトークンが存在するか否かを表す述語．

```
ライン上にトークン(s,n,p,r) =
  exist(line(s,n),token(p,r))
```

ライン上にフレーム(s,n,m,d,r)：状態ｓで，ラインｎに局ｍが発信したデータｄ，リザベーシヨンｒのフレームが存在するか否かを表す述語．

```
ライン上にフレーム(s,n,m,d,r) =
  { exist(line(s,n),lframe(m,d,r)) or
    exist(line(s,n),Lframe(m,d,r)) }
```

ラインの先頭要素がフレーム(s,n,m,d,r)：
　状態ｓで，ラインｎの先頭要素が，局ｍが発信したデータｄ，リザベーシヨンｒのフレームであるか否かを表す述語．

```
ラインの先頭要素がフレーム(s,n,m,d,r) =
  { first(line(s,n)) = lframe(m,d,r) or
    first(line(s,n)) = Lframe(m,d,r) }
```

他の補助関数の詳細については文献［４］参照のこと．

４．検証

４．１　状態の系列

「進行性」等の性質は，「ずっと．．．」や「いつかは．．．」や「ある時点以降．．．」といった表現で表されるために，リングの状態の系列（無限系列）に対して成り立つ性質であると考える．一般に，無限系列に対する性質を形式的に証明するためには，□(always)，◇(evntually)，○(next) 等のオペレータをもつテンポラルロジックの概念を導入することが有効である．

初期状態から許された動作のみを何回か行って到達できる状態の無限系列σを次のように定義する．

σ≙s0,s1,..,sn,...

但し，s0≙INITIAL,∀i(si=>si+1)

状態の無限系列σの集合を SEQとする．σ∈SEQ なる任意のσに対して述語Ｐが真であるとき， SEQ⊢Ｐで表す．

４．２　相互排他性・一巡性・進行性の証明

相互排他性・一巡性・進行性の形式的な証明はTemporal logicの概念を導入して次のように行う．

相互排他性は，次の形のTemporal logicの式で表される性質である．

相互排他性

Ｐ = □[トークンライン上？ or
　　トークン所有局有？]

として，SEQ⊢Ｐ が成り立つ．

一般に，Ｐが□Ｑで表される性質は，Ｑを初期状態から許された動作のみを何回か行って到達できる任意の状態（関数VALID の値が真である状態）に対して成り立つ性質（不変式）と考えることができ， SEQ⊢□Ｑは，すべての状態ｓに対して VALID(s) ⊃ Q(s)が真であることに相当する．この性質は，状態遷移に関する帰納法を用いて，次の(1)，(2)を証明する．

(1) VALID(INITIAL) ⊃ Q(INITIAL)が真である．

(2) VALID(ｓ) ⊃ Q(ｓ)が真であると仮定すると，任意の状態遷移関数αを実行した後の状態α・ｓに対してもVALID(α・ｓ) ⊃ Q(α・ｓ)が真である．

一方，一巡性や進行性は，□や◇や○等を組み合わせて表される性質である．これらの性質の証明には，次の２つの条件を仮定する．

①ラインｎ－１にバケットが存在すれば，いつかは局ｎで受信動作が行われる（ラインｎ－１の先頭のバケットが受信される）．

②局はトークンの送信が可能（又は，最終フレームの送信すれば直ちにトークンの送信が可能）になれば(F_TRANS_SOME_RECEIVED,T_TRANS_L_NOTRECEIVED又はT_TRANS_ALL_RECEIVED状態），いつかはトークンを送信して自局の送信権を他の人に譲る．

これらの条件をここでは「リングに課す条件」と呼ぶ．リングに課す条件は，次の形のTemporal logicの式で表現する．

リングに課す条件 =
　□[ラインにバケット存在(n-1) ⊃
　　◇バケット受信(n)] and
　□[not{ □◇トークン発信可能(n) and
　　not(◇トークン発信(n)) }]

また，一巡性，及び進行性を次の形のTemporal logicの式で表現する．

一巡性

Ｐ = リングに課す条件 ⊃
　□[ライン上にフレーム(n,n,d,l) ⊃
　　◇{ ラインの先頭要素が
　　フレーム(n-1,n,d,r) and
　　○バケット受信(n) }]

として，SEQ⊢Ｐ が成り立つ．

進行性

Ｐ = リングに課す条件 ⊃
　□[□プライオリテイq以上の
　　PDUの追加なし(k) ⊃
　　◇プライオリテイq以上のPDUなし(k)]

として，SEQ⊢Ｐ が成り立つ．

一巡性や進行性の証明のため，まず，次の２つのタイプの性質（基礎的補題）だけを仕様から直接証明する．

1. typeA 任意の状態ｓに対して，VALID(s) ⊃ P(s) が真

相互排他性は typeAに属する性質である．このタイプの証明は，前述のように状態遷移に関する帰納法を用いて行う．進行性の証明に用いる typeAの基礎的補題として７個の補題を導入する．

2. typeB 任意の状態ｓに対して，

{ VALID(s) and P(s) and VALID(α・ｓ) } ⊃ Q(α・ｓ) が真

VALID(s) かつP(s)が真である任意の状態ｓ，及び状態遷移関数αを実行した後の状態α・ｓに対して，VALID(α・ｓ) ⊃ Q(α・ｓ) が真であることを仕様から直接証明する．進行性の証明に用いる typeBの基礎的補題として２５個の補題を導入する．

次に２つのタイプの基礎的補題をそれぞれ以下のようにTemporal logicの式に変換する．

1. typeA VALID(s) ⊃ P(s) の形の論理式は，すべて次の形のTemporal logic式に変換する．

SEQ ⊢ □P

2. typeB { VALID(s) and P(s) and VALID(α・ｓ) } ⊃ Q(α・ｓ) の形の論理式は，すべて次の形のTemporal logic式に変換する．

SEQ ⊢ □[{ P and ○(last_action = α) } ⊃ ○Q]

そして，これらの基礎的補題を用いて進行性等の証明を階層的に行う．すなわち，Temporal logicの式で表された基礎的補題とTemporal logicの推論則から一巡性など１１個の補題を証明する．そして，それらの補題といくつかの基礎的補題をさらに組み合わせることによって，進行性の証明を行う．なお，一部の性質の証明には，非負整数が整礎集合(well_founded_set)であること（任意の非負整数a0に対して，a0>a1>a2>...なる非負整数の無限列a0,a1,a2,...が存在しない）や等号・不等号に関する性質等，非負整数に関するいくつかの性質をTemporal logicの式で表し，それらを各性質の証明に利用している．

5. あとがき

以上のように，トークンリング方式ＬＡＮにおける「進行性」の証明を，Temporal logicの概念を導入することにより形式的に行うことができた．ただ，実際のトークンリング方式ＬＡＮではビットごとに伝送を行っている．したがって，一つのパケットが複数の局にまたがる場合もあるが，本稿の仕様ではそれを考慮していない．今後の検討課題として，そのことも考慮した仕様を記述し，その仕様の上でも「進行性」が成り立つことを検証することが考えられる．その検証は，本稿で証明した基礎的補題の証明の部分のみを変更すればよいと思われる．

参 考 文 献

[1] 樋口，東野，谷口，嵩，藤井，森：”トークンリング方式ＬＡＮの代数的記述について”，信学技法，AL84-26(昭59-09)．

[2] 東野，谷口，嵩，藤井，森：”代数的に記述された通信プロトコルの動的性質の検証”，信学論（Ｄ），J69-D，10，pp.1471-1480(昭61-10)．

[3] IEEE PROJECT 802 LOCAL AREA NETWORK STANDARDS：”Draft IEEE Standard 802.5”，Working Draft (December 1983)．

[4] 住谷忠雄：”トークンリング方式ＬＡＮの「進行性」の検証”，大阪大学基礎工学研究科修士論文(昭62-02)．

表1　トークンリング方式ＬＡＮの仕様

sn	action	jouken	=> sn	pn	rn	stn	pqn	qn-1	qn
*	PE(d,i)						eq(d,i)		
1	R	fq=I(m,d,r)						dq	i(m,d,rm)
		fq=L(m,d,r)						dq	I(m,d,rm)
		fq=T(p,r)							
		DPm>=p	2	p			dqP	dq	i(n,tP,1)
		Pm>=p	4	p			dqP	dq	I(n,tP,1)
		Pm<p,p≠t2						dq	t(p,rm)
		Pm<p,p=t2,rm>t1				ex(rm)		dq	t(rm,1)
		Pm<p,p=t2,rm<=t1				pop		dq	t(t1,rm)
2	R	fq=I(m,d,r)	3		r			dq	
		fq=L(m,d,r)	D		r			dq	
		fq=T(p,r)	D		r			dq	
	TI	DPm>=pn					dqP		i(n,tP,1)
	TL		4				dqP		I(n,tP,1)
3	R	fq=I(m,d,r)			r			dq	
		fq=L(m,d,r)	D		r			dq	
		fq=T(p,r)	D		r			dq	
	TI	DPm>=pn					dqP		i(n,tP,1)
	TL		5				dqP		I(n,tP,1)
4	R	fq=I(m,d,r)	5		r			dq	
		fq=L(m,d,r)	6		r			dq	
		fq=T(p,r)	D		r			dq	
5	R	fq=I(m,d,r)			r			dq	
		fq=L(m,d,r)	6		r			dq	
		fq=T(p,r)	D		r			dq	
	TT	pn>=Rm	7						t(pn,Rm)
		pn<Rm,pn>t2	7			pu(pn,Rm)			t(Rm,1)
		pn<Rm,pn=t2	7			ex(Rm)			t(Rm,1)
6	R	fq=I(m,d,r)	D		r			dq	
		fq=L(m,d,r)	D		r			dq	
		fq=T(p,r)	D		r			dq	
	TT	pn>=Rm	1						t(pn,Rm)
		pn<Rm,pn>t2	1			pu(pn,Rm)			t(Rm,1)
		pn<Rm,pn=t2	1			ex(Rm)			t(Rm,1)
7	R	fq=I(m,d,r)						dq	
		fq=L(m,d,r)	1					dq	
		fq=T(p,r)	D					dq	
E	R	fq=I(m,d,r)						dq	
		fq=L(m,d,r)						dq	
		fq=T(p,r)						dq	
	TI								i(n,tP,*)
	TL								I(n,tP,*)
	TT								t(*,*)

略語の説明

sn;1:REPEAT,2:F_TRANS_NO_RECEIVED,3:F_TRANS_SOME_RECEIVED,4:FRAME_WAITING,
5:T_TRANS_L_NOT_RECEIVED,6:T_TRANS_ALL_RECEIVED,7:STRIP,E:ERROR

action;PE(d,i):PDU_ENQUEUE(n,d,i),R:RECEIVE(n),TI:TRANS_IFRAME(n),
TL:TRANS_LFRAME(n),TT:TRANS_TOKEN(n),
fq:first(qn-1),I(m,d,r):Iframe(m,d,r),L(m,d,r):Lframe(m,d,r),
T(p,r):token(p,r)

jouken;DPm:maxlevel(dequeueP(pqn)),Pm:maxlevel(pqn),t2:top2(stn),
rm:max(r,maxlevel(pqn))),t1:top1(stn),Rm:max(rn,maxlevel(pqn))

stn;ex(b):exchange(stn,b),pu(a,b):push(stn,<a,b>)

pqn;eq(d,i):enqueueP(pqn,d,i),dqP:dequeueP(pqn)

qn-1;dq:dequeue(qn-1)

qn;i(m,d,r):enqueue(qn,Iframe(m,d,r)),I(m,d,r):enqueue(qn,Lframe(m,d,r)),
t(p,r):enqueue(qn,token(p,r)),tP:top_P(pqn)