TAO TIME 認知アルゴリズムによる、ネットワークアクセス制御の原理。

Principle of Network Access Control by TAO TIME Cognition

杉　山　　彰
Akira Sugiyama
ジェーシーエヌ株式会社
〒102-0083　東京都千代田区麹町２丁目１２番６号
sugiyama@jcn9000.co.jp

1. Abstract

As for the TAO TIME Cognition algorithm, the physical clock value, which the system clock included in the computer as standard equipment counts, is replaced by the logical clock, and in the Client (computer to access) and the Server (computer to be accessed) systems which exist in any two points on network, the Server cognizes the Client, simultaneously the Client does the server. Thus, this cognition algorithm was researched, developed, and commercialized for the purpose of building peer-to-peer mutual cognition system

１．はじめに

TAO TIME 認知アルゴリズムは、コンピュータに標準装備されているシステムクロックがカウントする物理的なクロックの値（clock tick の割込数）を論理的なクロックの値に置換。ネットワーク上の任意の２地点に存在する Client(ｱｸｾｽ側ｺﾝﾋﾟｭｰﾀ)＆Server(被ｱｸｾｽ側ｺﾝﾋﾟｭｰﾀ)システムにおいて、Server は Client を認知し、同時に Client は Server を認知。Peer to Peer な相互認知システムの構築を目的として研究・開発・商品化された認知アルゴリズムである。

論理的なクロックの定義としては、「２つのプロセスが相互作用し合うことがなければ、両方の物理的なクロックは同期化する必要がない。すべてのプロセスが正確な時刻について一致するのではなく、イベントの発生の順番が一致するか、どうかが問題である（分散システムの同期化におけるクロックの同期化等：Lamport,1978）」を前提としている。しかし、本文は、Lamport の論理的なクロックの定義をさらに発展させ、「たとえ２つのプロセスが相互作用し合うことがあっても、両方の物理的なクロックを同期化する必要がない」という定義を追加。

新しく、TAO TIME 認知アルゴリズムとして実現したロジックの解説を行うものである。

※1.[認知]の意味：わきまえること。他と区別してそうであるとすること　(出典：広辞苑)

※2.[認証]の意味：文書の成立、内容の正しさの公的証明(出典：広辞苑)

２．TAO TIME 認知アルゴリズムの基本原理。

TAO TIME 認知アルゴリズムは、時間とは「前と後に関する運動の数(ｱﾘｽﾄﾃﾚｽ)」であり、さらに時間とは「過去と未来が現在に必須であることを前提とし、現在とは、未来についての予測と、過去の記憶である(C.M.ｻﾍﾞｰｼﾞ)」を遵守することを前提とする。

また、TAO TIME 認知アルゴリズムに使用される「時計装置」は Server の側に標準装備されている「システムクロック(タイマ)」を使用し、刻時精度は、本文が記述する Client＆Server 間の環境においては、物理的に毎秒 1000 回の割込みを発生させるように設定。従って、1/1000sec を最小刻時１単位とする。

３．最小通信プロセス単位におけるイベントの定義。

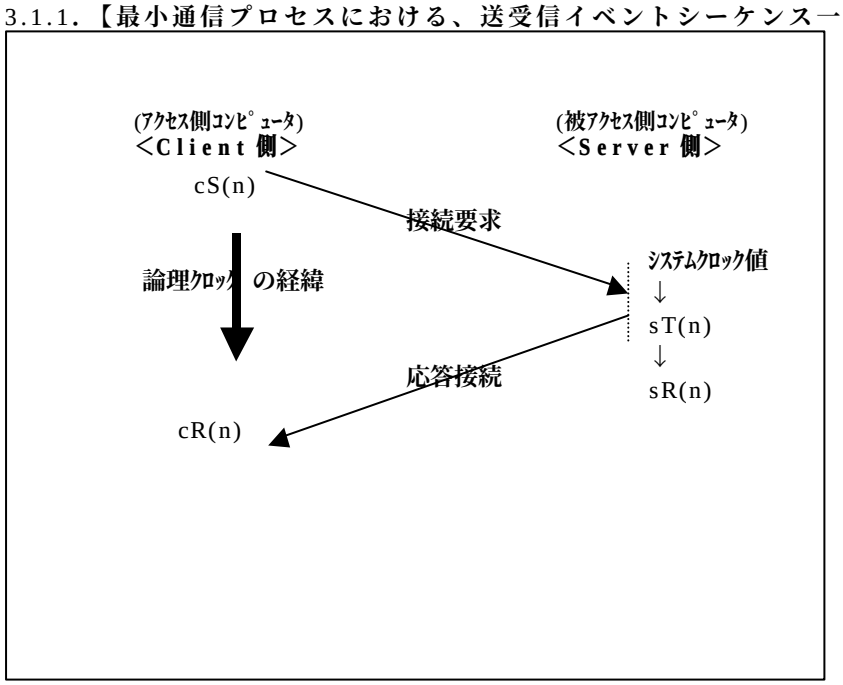
Server＆Client 間における最小通信プロセス単位を、Client の送信イベントの発生から始まり、Server の受信イベントの発生→送信イベントの発生を経て、Client の受信イベントの発生をもって終了すると定義する。また、前述のイベントには変数名を設定し、その変数が意味する目的を、任意の計測始点から任意の計測終点までのクロックチックのカウント量をイベント値としてあらわすこととする。なお、このイベント値は、後述する「Client＆Server 識別子データ」を計算するために必要な「要素データ」となる。

3.1.【TAO TIME 認知システムで使用される変数名：(要素データ)】

- Client の送信イベント発生時を論理ｸﾛｯｸ値に置換した変数名：cS(**c**lient **S** end)
- Server の受信イベント発生時を論理ｸﾛｯｸ値に置換した変数名：sR(**s**erver **R**eceive)
- 通信プロセスごとの基準時を論理ｸﾛｯｸ値に置換した変数名：sT(**s**erver **T**ime)
- Client の受信イベント発生時を論理ｸﾛｯｸ値に置換した変数名：cR(**c**lient **R**eceive)

※3. 基準時の意味：１回の通信ﾌﾟﾛｾｽ内において、Servr＆Client 間で共有使用される始点時

3.1.1.【最小通信プロセスにおける、送受信イベントシーケンス一覧図】



＜イベントの意味＞

・cS(n)：Client 送信イベント発生時

・sT(n)：通信ﾌﾟﾛｾｽごとの基準時

・sR(n)：Server 受信イベント発生時

・cR(n)：Client 受信イベント発生時

＜クロックの種類＞

・cS(n)：論理的なクロック

・sT(n)：論理的なクロック

・sR(n)：論理的なクロック

・cR(n)：論理的なクロック

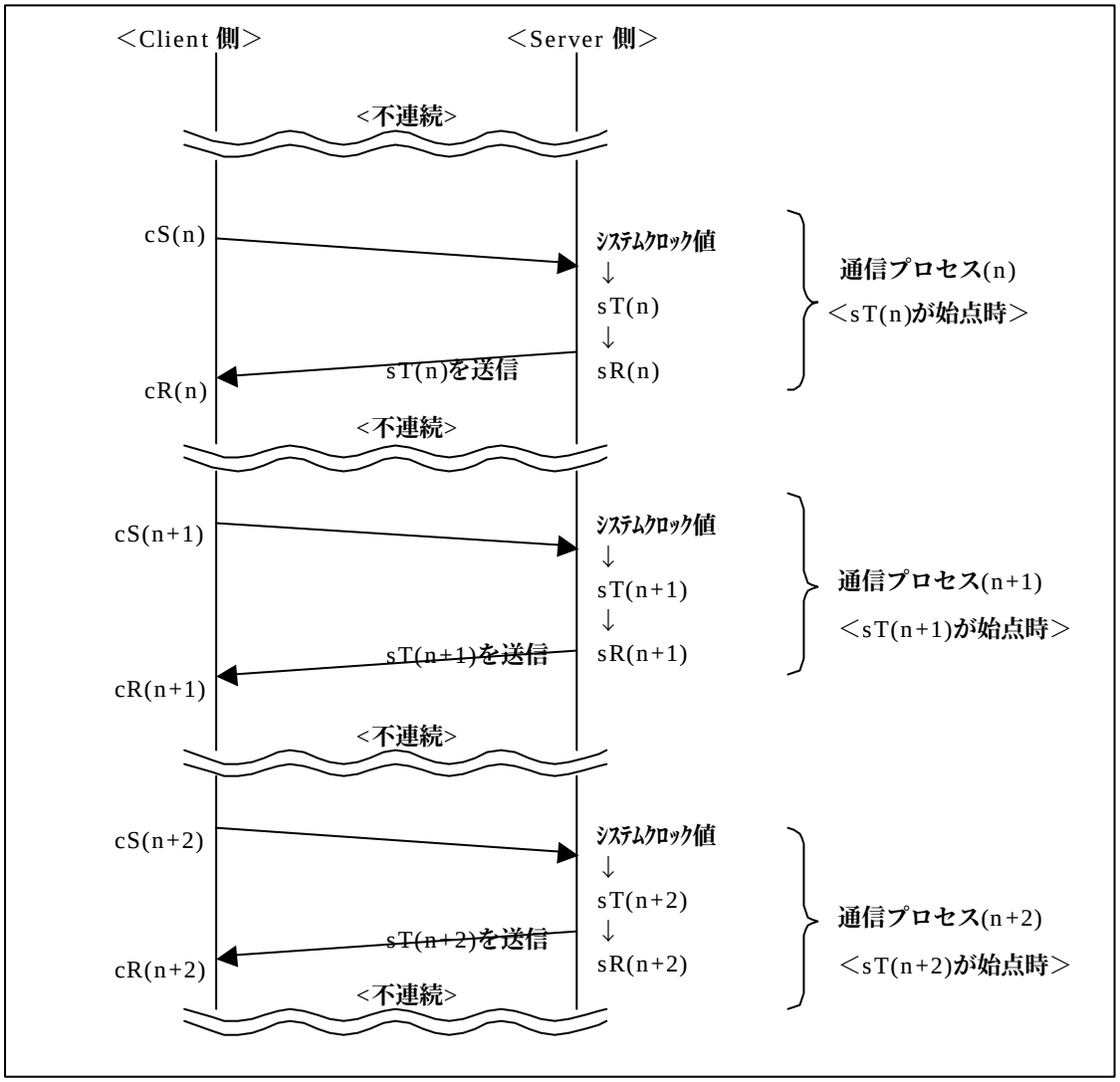
また、さらに Server&Client 間の通信における最小認知プロセス単位を過去２回の通信プロセスの成立を前提とする。認知処理は３回目の通信プロセスから、順次、行うものとする。従って、最小通信プロセス単位（①-1、①-2、①-3）と、最小認知プロセス単位（①～③）における、それぞれのイベントの発生順序は、以下の通りとなる。イベントの発生する順番は決して前後することなく、重なることもなく、発生順序が一致する。

3.2.【最小認知プロセス単位における送受信イベントの発生順序】

①-1．cS(n)：初回 Client 送信イベント発生
①-2．sR(n)：初回 Server 受信イベント発生
①-3．cR(n)：初回 Client 受信イベント発生
②-1．cS(n+1)：２回目 Client 送信イベント発生
②-2．sR(n+1)：２回目 Client 受信イベント発生
②-3．cR(n+1)：２回目 Client 受信イベント発生
③-1．cS(n+2)：３回目 Client 送信イベント発生
③-2．sR(n+2)：３回目 Client 受信イベント発生
③-3．cR(n+2)：３回目 Client 受信イベント発生

※4.上記、最小認知プロセス単位において、Server 送信イベント発生が存在しない理由は、Server にとって受信イベントが発生すれば、送信イベントの発生は、Server 機能としての不可欠イベントである。そのため、TAO TIME 認知アルゴリズムにおいては Server の側が当該 Client の認知を許諾すれば、Server の送信イベントは Server の受信イベント発生時と Client 受信イベント発生時の間に必然的に発生しているものと見なす。従って、送信イベント発生時を論理クロック値に置換した変数名の設定は不要とし、なおかつ、その変数値の利用も不要とする。

3.2.1.【最小認知プロセスにおける、通信プロセス＆イベントシーケンス一覧図】



3.3.【論理的なクロック値に置換された各送受信イベント値の前提条件】

- cS(n)→sR(n)→cR(n)→cS(n+1)→sR(n+1)→cR(n+1)→cS(n+2)→sR(n+2)→cR(n+2)
- cS(n)＜sR(n)＜cR(n)＜cS(n+1)＜sR(n+1)＜cR(n+1)＜cS(n+2)＜sR(n+2)＜cR(n+2)
- cS(n)≠sR(n)≠cR(n)≠{cS(n+1)≠sR(n+1)≠cR(n+1)≠cS(n+2)≠sR(n+2)≠cR(n+2)

・cS(n)とcR(n)が同じプロセス内のイベントであり、cS(n)がcR(n)より先に発生する場合、cS(n)→cR(n+1)、cS(n)＜cR(n+1)、cS(n)≠cR(n+1)が真になる。

・cS(n)が１つのプロセスからのメッセージ送信イベントであり、sR(n)が、別のプロセスのメッセージ受信イベントである場合、cS(n)→sR(n)、cS(n)＜sR(n)、cS(n)≠sR(n)が真になる。

・cR(n)が前回プロセスのメッセージ受信イベントであり、sR(n+1)が今回プロセスのメッセージ受信イベントである場合、前回プロセスと今回プロセスが相互作用し合うプロセスにおいても、cR(n)→sR(n+1)、cR(n)＜sR(n+1)、cR(n)≠sR(n+1)が真になる。

＊Lamportの定義においては、２つのﾌﾟﾛｾｽが相互作用し合う条件下では、論理ｸﾛｯｸの同期は困難としている。

4．TAO TIME 認知システムによる、論理的なクロックの同期化。

TAO TIME 認知システムによる論理的なクロックの同期化は「たとえ２つのプロセスが相互作用し合うことがあっても、両方の物理的なクロックを同期化する必要がない」という前提を新たに追加している。

そのために、TAO TIME 認知システムにおいてシステムクロックと呼ばれるクロックチック(clock tick)発生装置は Client の側に必ずしも装備されている必要はなく、Server の側にだけ装備されていればよい。その結果、複数の水晶の周波数が、すべて同じ周波数で振動するかどうかが保証されないことによって発生するクロックスキ

ュー(clock skew)の問題は不問となる。

例え、CPU が複数あり、それぞれに固有のクロックがある場合でも、TAO TIME 認知システムは「相対的な時刻」を利用しない。Server が生成するクロックのみを「絶対的な時刻」として利用する認知ロジックを開発。つまり、Server の数だけ、無数の「絶対的な時刻」が存在するという、新しい「論理的なクロックの同期化」を実現している。

5．TAO TIME 認知システムによる、論理的なクロック生成の計算式一覧。

5.1.【sT(n)の計算式】

sT(n)は、当該 Client からｎ回目の「接続要求」信号を受信した受信イベントの発生時を Server のシステムクロックが刻時した値に、最小刻時３単位(3/1000sec)を累積加算した値である。システムクロックが刻時した値の算出方法は、従来方式と同様に、1900 年 1 月 1 日の午前 12:00:01 を 1 として、1900 年 1 月 1 日の深夜 0 時からの秒数で測定される方式を採用。なお、sT(n)は、通信プロセスごとに Client に送信される。

sT(n)＝｛(1900/01/01 00:00:00～任意の yyyy/mm/dd hh:mm:ss)+(…+3+…)｝

5.2.【sR(n)の計算式】

sR(n)は、当該 Client からｎ回目の「接続要求」信号を受信したイベント発生時を論理クロック値に置換した変数である。sR(n)の計算は、sT(n)に、当該 Client から送信される前回送信イベント値：cS(n-1)と、前回受信イベント値：cR(n-1)の差分値を、通信プロセスごとに累積加算する。なお、cS(n-1)と cR(n-1)は、n 回目の「接続要求」信号受信時に、Client から通信プロセスごとに送信される。

sR(n)＝＜sT(n)…＋{cR(n－1)－cS(n－1)}＋…＞

5.3.【cS(n)の計算式】

cS(n)は、当該 Server へｎ回目の「接続要求」信号を送信したイベント発生時を論理的なクロック値に置換した変数である。cS(n)の計算は、sT(n)に、Client 自らが生成した、前回送信イベント値：cS(n-1)と、前回受信イベント値：cR(n-1)の差分値を、通信プロセスごとに累積加算し、なおかつ最小計測１単位(1/1000sec)をマイナスする。

cS(n)＝＜sT(n)…＋{cR(n－1)－cS(n－1)}＋…＞－1

5.4.【cR(n)の計算式】

cR(n)は、当該 Server からｎ回目の「応答接続」信号を受信したイベント発生時を論理的なクロック値に置換した変数である。cR(n)の計算は、sT(n)に、Client 自らが生成した、前回送信イベント値：cS(n-1)と前回受信イベント値：cR(n-1)の差分値を、通信プロセスごとに累積加算し、さらに、今回通信プロセスにおいて生成した乱数値：a(n)をプラスし、なおかつ最小計測１単位(1/1000sec)をプラスする。

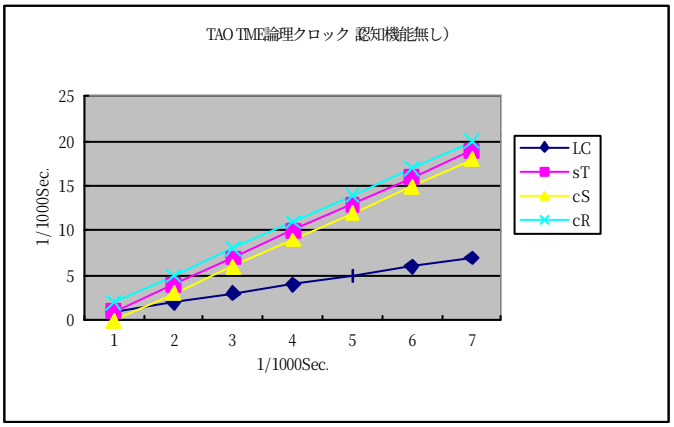
cR(n)＝＜ST(n)…＋{cR(n－1)－cS(n－1)}＋…＋a(n)＞＋1

5.5.【乱数値：a(n)の計算式】

a(n)は、通信プロセスごとに毎回生成され、オン・メモリにテンポラリに保持される乱数値である。乱数の生成方式は既存の任意の乱数生成方式を利用して生成する。a(n)の値はファイルに保存されることが無く、また、直接、外部に流出することがないデータであり、なおかつ、乱数を生成するパラメータも通信プロセスごとに変更。そのパラメータも外部に流出することがない。本文では、rand による乱数生成方式を紹介する。

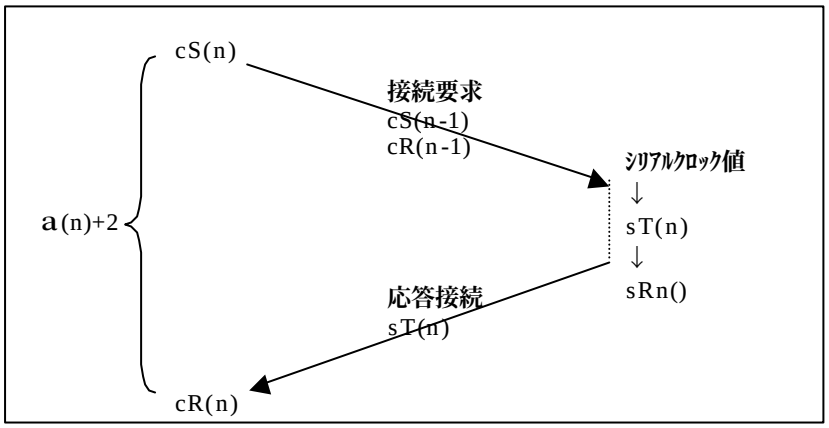
＊【rand()の初期値計算式】：a(n-1)＊1000＋CTIME(n) mod 1000
＊ 上記計算は、a(n)を生成する直前に毎回行う。
＊ a(n)＝a(n-1)＋rand()＊100000＋CTIME mod 100000
＊【CTIME(n)の計算式】：CTIME(n)＝1900/01/01 00:00:00～yyyy/mm/dd hh/mm/ss
＊ yyyy/mm/dd hh:mm:ss は、Client のシステムクロックが刻時した「接続要求」送信時
＊ CTIME は 1/1000sec 単位まで求められるが、乱数パラメータとして利用するのは下３桁の値。

5.5.1.【ｲﾍﾞﾝﾄ発生順序の一致性の実現図、その１】

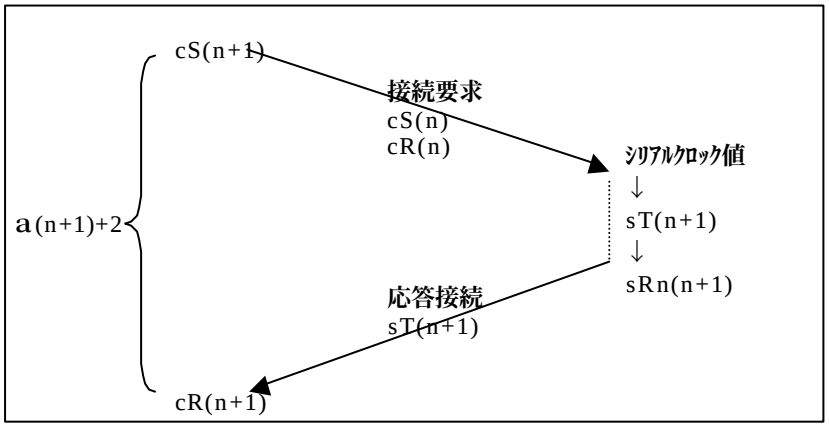


＊上記図表では、乱数値 a の値は付加されていない。

5.5.2.【論理的なクロック生成におけるイベントシーケンス一覧：（n）の場合】



5.6.2.【論理的なクロック生成におけるイベントシーケンス一覧：（n+1）の場合】



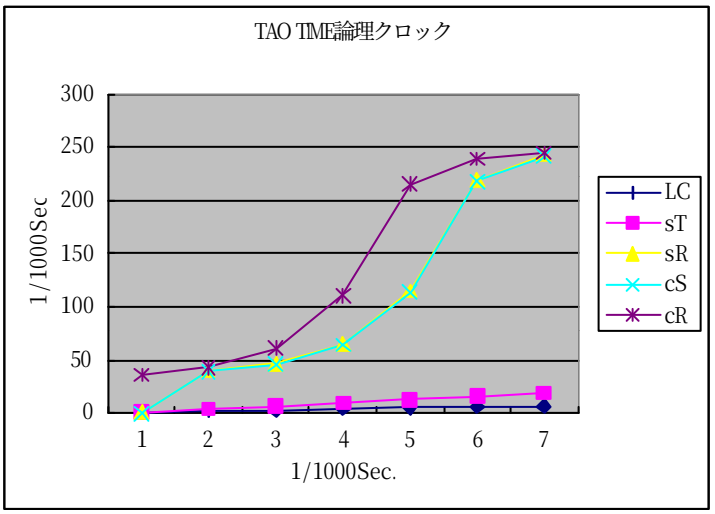
6．送受信イベントの順番の逆転、もしくは重なりの解決方法。

TAO TIME 認知システムによる論理的なクロックの同期化は「たとえ２つのプロセスが相互作用し合うことがあっても、両方のクロックを同期化する必要がない」という前提を新たに追加。そして、そのために必要な仕組みが、cS(n)の計算式にマイナス値として代入されている最小計測１単位と、cR(n)の計算式にプラス値として代入されている最小計測１単位と、sT(n)の計算式に累積加算値として代入されている最小計測３単位の設定。さらには、通信プロセスごとに、Client から Server に送信される cS(n-1)と cR(n-1)の差分によって生成される累積加算値である。

仮に、Client からの「接続要求」信号を、Server が最小計測１単位で連続して受信した場合、Client が Server からの「応答接続」信号を受信するイベント発生時と、Server が次回、Client からの「接続要求」信号を受信するイベント発生時のタイミングが逆転、もしくは同じ値になる可能性がある。論理的なクロックの同期の前提条件は、「イベントの発生する順番が一致していなくてはならない」としている。従って、Client からの「接続要求」信号を、Server が最小計測１単位で連続して受信した場合においても、各イベントの発生時の逆転、もしくは重なりは決して発生しない仕組みになる。

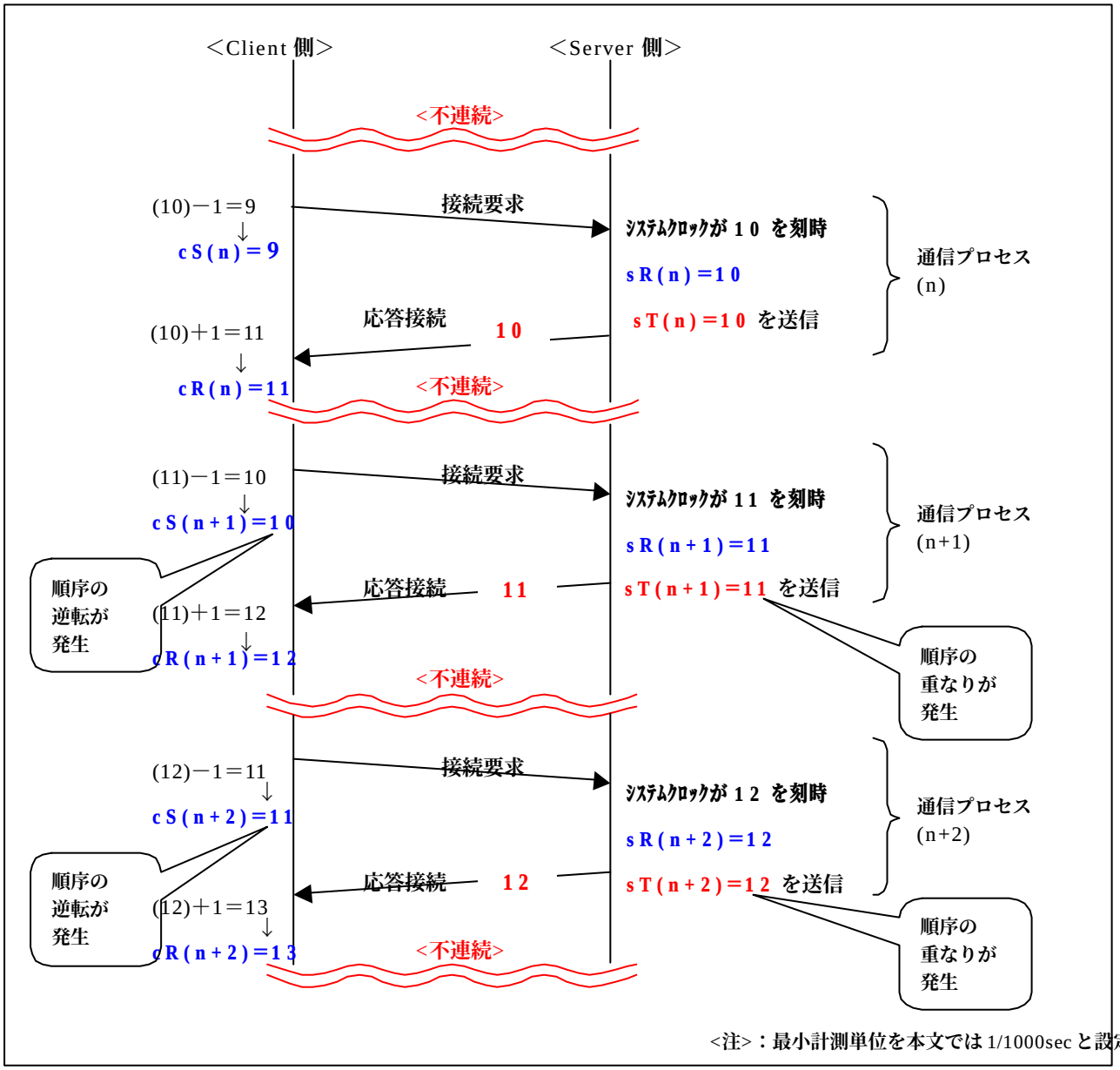
前述の、3.3.の項で述べている、論理的なクロック値に置換された各送受信イベント値の前提条件が完全に満たされるのである。

6.1.【ｲﾍﾞﾝﾄ発生順序の一致性の実現図、その２】



＊上記図表では、乱数値 a の値が cR()に付加されている。

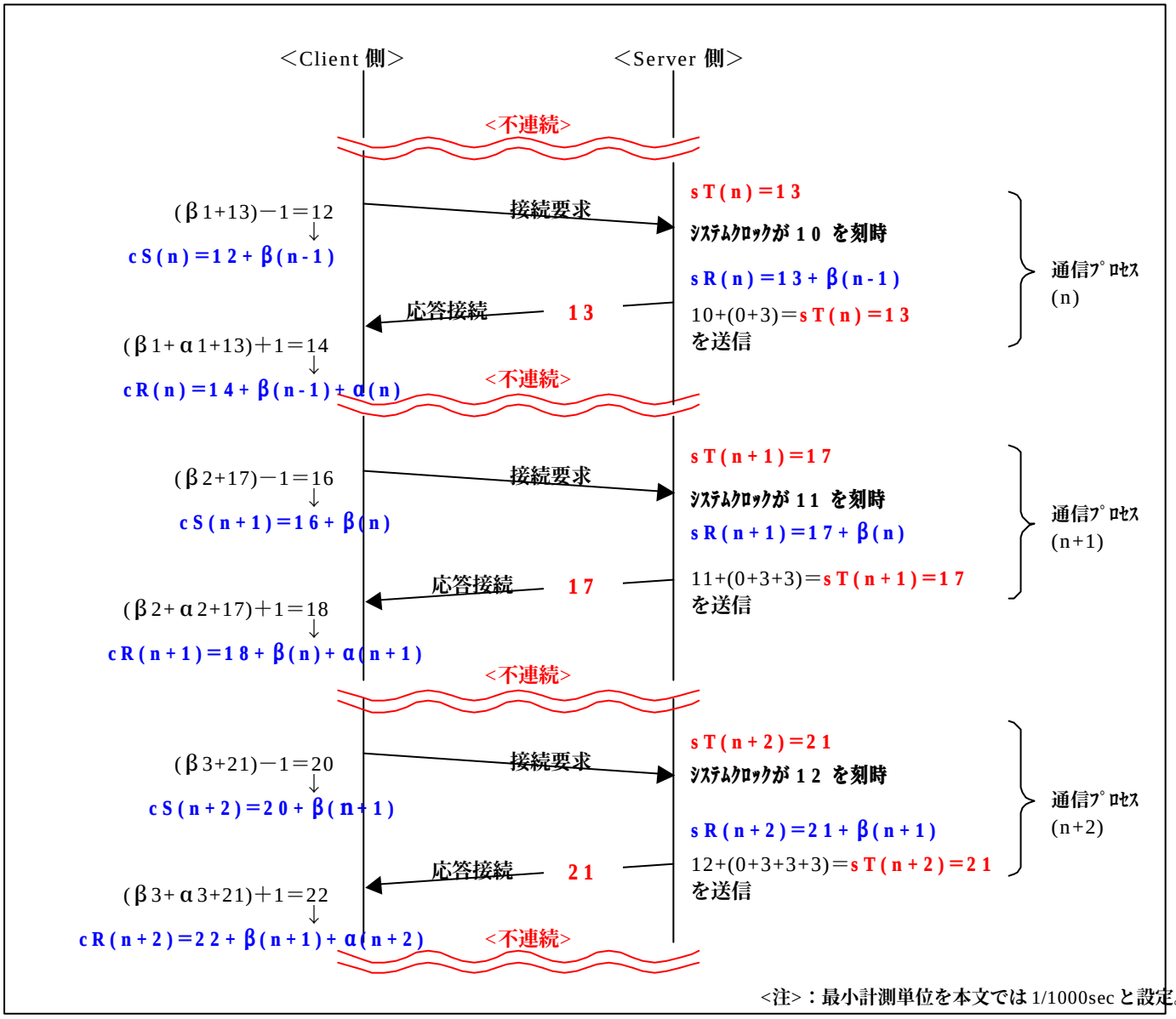
6.2. 2つのﾌﾟﾛｾｽが相互作用し合うことにより、各ｲﾍﾞﾝﾄの順序の逆転、重なりが発生する場合



<注>：最小計測単位を本文では 1/1000sec と設定。

＊2つのﾌﾟﾛｾｽが相互作用し合う場合、論理的ｸﾛｯｸの同期化は実現しない＜不連続値＝0～(∞±1)＞

6.3. 各ｲﾍﾞﾝﾄの順序逆転、重なりが発生しない仕組：ａ(n)の値をα(n)、cR(n-1)－cS(n-1)の差分値はβ(n-1)を適用



<注>：最小計測単位を本文では 1/1000sec と設定。

6.3.1.【TAO TIME 認知システムによる論理的なクロック同期化の実現図】＊不連続値＝0～(∞±1)

・cS(n)→sR(n)→cR(n)→cS(n+1)→sR(n+1)→cR(n+1)→cS(n＋2)→sR(n＋2)→cR(n＋2)
→不連続値→12→13→14→不連続値→16→17→18→不連続値→20→21→22→不連続値→

・cS(n)＜sR(n)＜cR(n)＜cS(n+1)＜sR(n+1)＜cR(n+1)＜cS(n＋2)＜sR(n＋2)＜cR(n＋2)
＜不連続値＜12＜13＜14＜不連続値＜16＜17＜18＜不連続値＜20＜21＜22＜不連続値＜

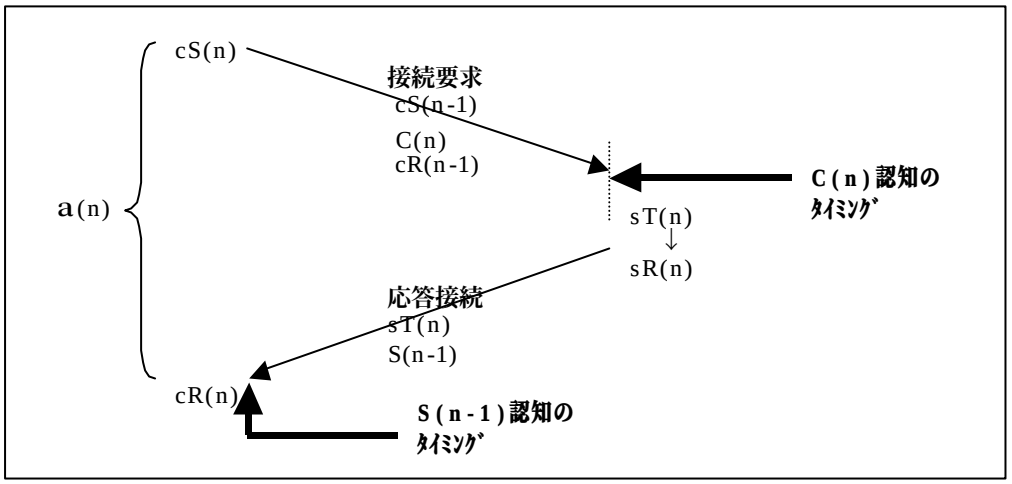
・cS(n)≠sR(n)≠cR(n)≠{cS(n+1)≠sR(n+1)≠cR(n+1)≠cS(n＋2)≠sR(n＋2)≠cR(n＋2)
≠不連続値≠12≠13≠14≠不連続値≠16≠17≠18≠不連続値≠20≠21≠22≠不連続値≠

7. TAO TIME 認知システムによる、識別子の生成方法。

TAO TIME 認知システムにおいては、Client 識別子の変数名として C()、Server 識別子の変数名として S()の、２種類の「識別子データ」が生成される。そして、この２種類の識別子を生成するために、既に説明済みの、cS()、cR()、sR()、sT()の４種類のイベント値が、「要素データ」として必要となる。

前述の、Client 識別子：C(n)は、n 回目の通信プロセスにおいて、Server への「接続要求」信号の送信時に生成され、cS(n-1)と cR(n-1)の要素データと一緒に Server へ送信されて、Server の認知許諾を要求する。また、Server 識別子：S(n-1)は、n 回目の通信プロセスにおいて、Client への「応答接続」信号の送信時に生成され、sT(n)の要素データと一緒に Server へ送信されて、Client の認知許諾を要求する。以下に、それぞれの識別子データの値を求める計算式を提示する。

7.1.【Server＆Client 間で、最終的に授受されるデータと、認知のタイミング】



7.2.【Client 識別子：C(n)の計算式】

TAO TIME 認知システムにおける、Client 識別子の生成は、最低限、過去３世代にわたって Server＆Client 間で授受される「要素データ」を必要とする。Client 側で付加する要素データは、前回送信イベント値と前々回受信イベント値であり、その差分値を S(n-1)よりマイナスする。

C(n)＝S(n-1)－{cS(n-1)－cR(n-2)}

7.3.【Server 識別子：S(n-1)の計算式】

TAO TIME 認知システムにおける、Server 識別子の生成は、最低限、過去３世代にわたって Server＆Client 間で授受される「要素データ」）を必要とする。Server 側で付加する要素データは、前回受信イベント値と前々回受信イベント値であり、その差分値を C(n-1)にプラスする。

S(n-1)＝C(n-1)＋{sR(n-1)－sR(n-2)}

7.4. イベント値(要素データ)、及び、識別子のデータベース構造。

7.4.1. Server 側の構築データベース（n=4 の場合）

＼	n-3 世代	n-2 世代	n-1 世代	n 世代
cS	cS(n-4)	cS(n-3)	cS(n-2)	cS(n-1)
sT	sT(n-3)	sT(n-2)	sT(n-1)	sT(n)
cR	cR(n-4)	cR(n-3)	cR(n-2)	cR(n-1)
C	C(n-3)	C(n-2)	C(n-1)	C(n)
S	S(n-4)	S(n-3)	S(n-2)	S(n-1)
sR	sR(n-3)	sR(n-2)	sR(n-1)	sR(n)

7.4.2. Client 側の構築データベース（n=4 の場合）

＼	n-3 世代	n-2 世代	n-1 世代	n 世代
cS	cS(n-3)	cS(n-2)	cS(n-1)	cS(n)
sT	sT(n-3)	sT(n-2)	sT(n-1)	sT(n)
cR	cR(n-3)	cR(n-2)	cR(n-1)	cR(n)
C	C(n-3)	C(n-2)	C(n-1)	C(n)
S	S(n-4)	S(n-3)	S(n-2)	S(n-1)
a	a(n-3)	a(n-2)	a(n-1)	a(n)

8.TAO TIME 認知アルゴリズムにおける認知の仕組み。

8.1. Server が Client の識別子：C(n-1)を認知する手順。

(n-1)世代において、Server 側では Client から送信されてきた C(n-1)を認知するために、仮 C(n-1)を生成しなければならない。そして、認知が OK の場合、Server は Server 自らを Client に認知してもらうために Server 識別子として S(n-2)を生成しなければならない。

8.1.1. 仮 C(n-1)を生成するための計算式。＊ Server 側で入手した要素データで生成。

・仮 C(n-1)＝S(n-1)－{sR(n-1)－sR(n-2)}

＊Client が C(n-1)を生成した計算式：

・C(n-1)＝S(n-2)－{cS(n-2)－cR(n-3)}

8.1.2. Server の識別子：S(n-2)を生成するための計算式。

・S(n-2)＝C(n-2)＋{sR(n-2)－sR(n-3)}

8.1.3. 仮 C(n-1)と S(n-2)を生成するために必要な要素データの一覧。

・sR(n-1)＝＜sT(n-1)…＋{cR(n-2)－cS(n-2)}＋…＞
・sR(n-2)＝＜sT(n-2)…＋{cR(n-3)－cS(n-3)}＋…＞
・sR(n-3)＝＜sT(n-3)…＋{cR(n-4)－cS(n-4)}＋…＞

8.1.4. (n-1)世代において、不特定第三者が Client に成りすますために推測しなければならない変数値。

・sR(n-1)、cR(n-2)、cS(n-2)

8.1.5.【C(n-1)の認知式】

・if 仮 C(n-1)＝C(n-1) then 認知 OK

・if 仮 C(n-1)≠C(n-1) then 認知 NG

8.2. Client が Server の識別子：S(n-2)を認知する手順。

(n-1)世代において、Client 側では Server から送信されてきた S(n-2)を認知するために、仮 S(n-2)を生成しなければならない。そして、認知が OK の場合、Client は、Client 自らを Server に次回アクセス時に認知してもらうために、Client 識別子として、C(n)を生成しなければならない。

8.2.1. 仮 S(n-2)を生成するための計算式。

・仮 S(n-2)＝C(n-1)＋{cS(n-2)－cR(n-3)}

＊Server が S(n-2)を生成した計算式：

S(n-2)＝C(n-2)＋{sR(n-2)－sR(n-3)}

8.2.2. Client の識別子：C(n)を生成するための計算式。

・C(n)＝S(n-1)－{cS(n-1)－cR(n-2)}

＊上記式中、S(n-1)は、(n-1)世代では Server から送信されてこない Server 識別子である。しかし、Client の側では、S(n-1)＝C(n-1)＋{sR(n-1)－sR(n-2)}の計算式を使用して、Server 識別子：S(n-1)を事前に生成することができる。但し、Server は、Client からの次回アクセス時に送信される「cS(n-1)」が無ければ、S(n-1)を生成することはできない。

8.2.3. 仮 S(n-2)と C(n)を生成するための必要な要素データの一覧。

・cS(n-1)＝＜sT(n-1)…＋{cR(n-2)－cS(n-2)}＋…＞－1

・cS(n-2)＝＜sT(n-2)…＋{cR(n-3)－cS(n-3)}＋…＞－1

・cR(n-2)＝＜sT(n-2)…＋{cR(n-3)－cS(n-3)}＋…＋a(n-2)＞＋1

・cR(n-3)＝＜sT(n-3)…＋{cR(n-4)－cS(n-4)}＋…＋a(n-3)＞＋1

8.2.4. (n-1)世代において、不特定第三者が Server に成りすますために推測しなければならない変数値。

・sR(n-1)、cS(n-2)、a(n-1)、a(n-2)

8.2.5.【S(n-1)の認知式】

・if 仮 S(n－1)＝S(n－1) then 認知 OK

・if 仮 S(n－1)≠S(n－1) then 認知 NG

9. TAO TIME 認知システムのセキュリティ強度(偶然の一致率)。

【別紙資料参照】

「TAO TIME システム TAO識別子(ユニークパスワード)の【偶然の一致率】に関する考察報告書」

＜参考文献＞

1)「OS の基礎と応用」A.S.タネンバウム＝著／引地信之、引地美恵子＝訳
(株)ピアソン・エデュケーション

2)「Fifth Generation Management」 C.M.Savage：Digital Press

3)「TAO TIME システム、
TAO 識別子(ユニークパスワード)の【偶然の一致率】に関する考察報告書」

4) WO97/22047 「固有時間の生成装置」

5) 特願 2000-7796(P2000-77976) 「認証・管理装置」

6) 特願 2001-217918 「認証・管理方式、及び同装置」

以上