大規模通信システムの開発手法とP2Pサービスへの適用に関する研究

菊間, 一宏

https://doi.org/10.15017/2534460

大規模通信システムの開発手法と
P2P サービスへの適用に関する研究

令和元年 **6** 月

菊間 一宏

目次

図目次

表目次

概要

ブロードバンド通信市場の世界的な拡大は IoT の牽引もあり，モバイル市場だけでなく，固定ブロードバンド市場の拡大を継続させている．日本の固定ブロードバンド通信は世界的にも高速，安価であり，B フレッツやフレッツ光プレミアムに代表される FTTH(Fiber To The Home) サービスの展開は日本のブロードバンド通信の本格的な普及を支えている．

しかしながら，インターネットを基盤とした映像コンテンツ配信やオンライン商取引等の世界的拡大は，競争をさらに加速させ，電話サービスとインターネットサービスの両方を提供するキャリアネットワークにおけるネットワークサービス開発のローコスト化を余儀なくしている．

キャリアの通信ネットワークを構築するハードウェアは，コモディティ化や仮想化により，設備面でのローコスト化は進んだが，各種通信サービスを提供するソフトウェアはサービスバリエーションが豊富で独自性が高く，大規模であるため，公衆網適用時に継続的安定運用を可能とするために長期化する開発期間や開発工数の高止まりの改善が課題となっている．

固定ブロードバンド通信を支える NGN(Next Generation Network) に代表される大規模通信システムは，社会の基礎的なインフラであるため，信頼性や安全性の保証や社会的な依存性を保証しなくてはならないが，早期のサービス提供のために短期開発にしたりコストカットをおこなうと，ソフトウェアに内在する不具合を公衆網適用時にまで残し継続的安定運用を維持することが困難となる．逆に，ソフトウェアの品質向上を目的とした開発中の不具合摘出のための施策の積み上げは開発期間の長延化や開発コストの高止まりを引き起こす．

このため，大規模で且つ信頼性，安全性の保障が必要であり，災害時を含めサービスの継続性を要求されるソフトウェアの開発においては，開発工数や開発期間を増加させることなく，ソフトウェアに内在する不具合を開発期間中に

発見し，公衆網適用時の継続的安定運用を維持する手法が必要となる．

大規模通信システム開発においては，ウォーターフォールモデル（Ｖ字モデル）による開発を進めている．実際の開発においては詳細に各工程の実施内容やアウトプットを定め，工程ごとに品質の判定を行う事で，手戻りの少ない高品質なソフトウェア開発を目指している．公衆網適用時に不具合が発生しない様にソフトウェアに内在する不具合を開発期間中に発見し品質を高めつつ，開発工数／期間の増加を抑止，減少させるためには，これら開発プロセスの定式化および，その自動化が必要であり，その検討は開発の一連の工程に渡り進める必要がある．

ソフトウェア品質を高めつつ，開発工数／期間の増加を抑止，減少させるためには，以下の課題がある．

(1) 開発工数を増やすことなく品質の向上を実現する手法の確立．
(2) 品質を維持したまま，開発工数を減らす手法の確立．

一つ目の課題に対しては，本論文では公衆網運用中に発生する不具合対処毎に増加する稼働削減への効果も期待できる不具合の「数」に着目し，開発時の試験項目の「数」を調整する事で，開発時不具合発見数を増加させ，公衆網適用時の不具合発生数を減少させる手法について提案した．

開発工数の観点においても，試験工程で機能部毎の試験項目数を調整する事でシステム全体としての試験項目数を変えない，すなわち開発工数を増加させない試験項目選定手法を提案し，その手法をキャリアネットワークである NGN のシステム開発に適用した結果を報告する．（3 章）

二つ目の課題に対しては，本論文では開発のための要求仕様書を教師データとし，機械学習により要求仕様書から自動的に試験項目を抽出する手法について提案した．

高いソフトウェア品質を継続的に維持するためには，試験工程において確実に不具合を発見する手法にかかっている．不具合を発見するためには仕様を正確に理解し，検証すべき試験項目を網羅的に且つ正確に実施しなくてはならないが，検証すべき試験項目が正しく選定されなかったり，網羅的に検証が行われない場合は，不具合を内包したままシステムを提供しサービスの継続性を損ねてしまう事がある．

検証すべき試験項目は，開発時に作成される要求仕様書を基に作成される

が，適切な試験項目を作成できるかどうかは作業者のスキルやノウハウに大きく依存している．このスキルやノウハウの偏りによる不均質な試験項目作成を避けるため，試験項目作成基準を定めたり，試験項目レビュー等に大きく時間を割いており，この稼働がソフト開発のコスト増につながってしまっている．

このコストを増大させる課題を解決するため，機械学習により試験項目を自動的に作成する手法を提案した．試験項目自動作成のため，まず，これまでの開発における試験項目を参考に，試験項目箇所にマーキングされた要求仕様書を大量に作成し，マーキングされた大量のドキュメントを教師データとして学習機にて学習させる．その後，学習データを基に，新規開発の要求仕様書に対し学習結果を適用して自動的に試験箇所にマーキングする．最終的には，このマーキングされた要求仕様書から試験項目を自動的に作成されるが，この結果を有スキル者によって作成された試験項目と比較し評価を行った．本アプローチに関しても一つ目と同様，キャリアネットワークである NGN のシステム開発に適用した結果を報告する．（4 章）

この様な開発手法により高いソフトウェア品質を維持することで，次世代ネットワーク (NGN) は SIP ベースのセッションによる帯域幅と品質 (損失，遅延，揺らぎ) を保証した通信機能を継続的に提供している．加えて，物理 IP ネットワークのセッションを制御するインタフェースを公開しており，NGN では SIP ベースのセッションを切換る Application Network Interface (ANI) 機能を提供している．

帯域幅と品質が保障される NGN のセッション制御機能は，継続的に安定してデータが流れることが望ましいデータストリーム型の通信サービスに有益であると考えられる．データストリーム型の通信サービスを行う場合，データストリーム配信サーバからエンドユーザ端末へ配信を行うより，多数のエンドユーザ端末が中継点となり再度配信を行う形態のほうがデータ転送負荷の集中が避けられ望ましい．これは，ユーザ端末間の Peer-to-Peer(P2P) ネットワークを NGN のセッションを用いて構成することがより望ましい形態であるといえる．しかしながら，P2P 通信サービスは，エンドユーザの各端末がアプリケーションレベルのクライアントでありサーバであり中継ルータでもあるため，トラヒック集中が偏在する可能性がある．輻輳した端末がボトルネックとなりサービス品質が低下しないよう，P2P ネットワークの中継ルートを効率化させる P2P ネットワークトポロジー制御方法が必要である．

本論文ではユーザ端末アプリケーションの負担が少なく P2P ネットワークトポロジーを効率のよい構成へ変える方式を提案する．接続切換機能についてはいくつか方式が考えられ，これらの方式の特性を比較分析する．各方式の比較結果をシミュレーションによって検証し，実装ソフトウェアアーキテクチャのフィージビリティを試作検証によって確認した結果を報告する．（5 章）

以上，要約すると本論文では，信頼性や安全性の保証や社会的な依存性を保証しなくてはならない基礎的なインフラである大規模通信システムの開発において，ソフトウェア品質を高めつつ，開発工数／期間の増加を抑止，減少させるために，「開発工数（試験工数）を増やすことなく品質の向上を実現する手法」や「品質を維持したまま，開発工数を減らす手法」を提案し，実際の開発に適用しその効果を明らかにした．

また，開発された通信システムに対して適用される P2P サービスにおいて，中継ルートを効率化させる P2P ネットワークトポロジー制御方法を提案すると共に，フィージビリティを試作検証によって確認し，効果を明らかにした．

本成果である開発手法や開発されたソフトウェアによって，様々な端末がネットワーク機能と連携し，より多くの社会基盤となりえるネットワークサービスが実現されることが期待される．

第 1 章

はじめに

1.1　背景

ブロードバンド通信市場の世界的な拡大は IoT の牽引もあり，モバイル市場だけでなく，固定ブロードバンド市場の拡大を継続させている．

日本の固定ブロードバンド通信は世界的にも高速，安価であり，B フレッツやフレッツ光プレミアムに代表される FTTH(Fiber To The Home) サービスの展開は日本のブロードバンド通信の本格的な普及を支えている．

しかしながら，インターネットを基盤とした映像コンテンツ配信やオンライン商取引等の世界的拡大は，競争をさらに加速させ，電話サービスとインターネットサービスの両方を提供するキャリアネットワークにおけるネットワークサービス開発のローコスト化を余儀なくしている．

キャリアの通信ネットワークを構築するハードウェアは，コモディティ化や仮想化により，設備面でのローコスト化は進んだが，各種通信サービスを提供するソフトウェアはサービスバリエーションが豊富で独自性が高く，大規模であるため，公衆網適用時に継続的安定運用を可能とするために必要な開発期間や開発工数の高止まりの改善が課題となっている．

キャリアの大規模通信システム開発においては，ウォーターフォールモデル（Ｖ字モデル）による開発を進めている．実際の開発においては各工程の実施内容やアウトプットを詳細に定め，工程ごとに品質の判定を行う事で，手戻りの少ない高品質なソフトウェア開発を目指している．公衆網適用時に不具合が発生しない様にソフトウェアに内在する不具合を開発期間中に発見し品質を高めつつ，開発工数／期間の増加を抑止，減少させるためには，これら開発プロ

セスの定式化および，その自動化が必要となっている．

このように高品質／高信頼を目指した開発手法を用い，より高いソフトウェア品質を維持することで，次世代ネットワーク (NGN) は SIP ベースのセッションによる帯域幅と品質 (損失，遅延，揺らぎ) を保証した通信機能を継続的に提供している．加えて，NGN では物理 IP ネットワークのセッションを制御するインタフェースを公開しており，SIP ベースのセッションを切換る Application Network Interface (ANI) 機能を提供している．

帯域幅と品質が保障される NGN のセッション制御機能は，継続的に安定してデータが流れることが望ましいデータストリーム型の通信サービスに有益であると考えられる．映像等のデータストリーム型の通信はインターネットの主要トラヒックであり今後もトラヒックが増大していくと予想されており [1]，より多くのユーザに利用されるサービスである．

データストリーム型の通信サービスを行う場合，データストリーム配信サーバからエンドユーザ端末へ配信を行うより，多数のエンドユーザ端末が中継点となり再度配信を行う形態のほうがデータ転送負荷の集中が避けられ望ましい．これは，ユーザ端末間の Peer-to-Peer(P2P) ネットワークを NGN のセッションを用いて構成することがより望ましい形態であるといえ，NGN を利用した多くの P2P 型のサービスアプリケーションの展開が期待されている．

1.2　課題

固定ブロードバンド通信を支える NGN(Next Generation Network) に代表される大規模通信システムは，社会の基礎的なインフラであるため，信頼性や安全性の保証や社会的な依存性を保証しなくてはならない．また，ブロードバンド通信の世界的な拡大により競争が激化し，キャリアのサービスもこれまで以上にタイムリーな提供が求められている．

しかしながら，早期サービス提供や価格競争のために短期開発にしたりコストカットをおこなうと，ソフトウェアに内在する不具合を公衆網適用時にまで残し継続的安定運用を維持することが困難となる．逆に，ソフトウェアの品質向上を目的とした開発中の不具合摘出のための施策の積み上げは開発期間の長延化や開発コストの高止まりを引き起こしている．

これらのソフトウェア開発における課題を解決しつつ，高品質を維持するこ

とで，次世代ネットワーク (NGN) は SIP ベースのセッションによる帯域幅と品質 (損失，遅延，揺らぎ) を保証した通信機能を継続的に提供している．安定的な通信の保証は，継続的に安定してデータが流れることが望ましいデータストリーム型の通信サービスに有益であり，さらにはデータ転送負荷の集中が避けられるユーザ端末間の Peer-to-Peer(P2P) ネットワークを NGN のセッションを用いて構成することがより望ましい形態であるといえる．しかしながら，輻輳した端末がボトルネックとなりサービス品質が低下しないよう，P2P ネットワークの中継ルートを効率化させる P2P ネットワークトポロジー制御方法が必要となっている．

1.2.1　高品質なソフトウェア開発のための課題

大規模通信システムは社会の基礎的なインフラであるため信頼性や安全性の保証や社会的な依存性を保証しなくてはならない．一方でブロードバンド通信の世界的な拡大による競争に追従するためにも早期のサービス提供や価格競争力を持つことが必要となっている．早期サービス提供や価格競争のために短期開発にしたりコストカットをおこなうと，ソフトウェアに内在する不具合を公衆網適用時にまで残し継続的安定運用を維持することが困難となるが，逆に，ソフトウェアの品質向上を目的とした開発中の不具合摘出のための施策の積み上げは開発期間の長延化や開発コストの高止まりを引き起こしてしまう（図 1.1）．

このため，大規模で且つ信頼性，安全性の保障が必要であり，災害時を含めサービスの継続性を要求されるソフトウェアの開発においては，開発工数や開発期間を増加させることなく，ソフトウェアに内在する不具合を開発期間中に発見し，公衆網適用時の継続的安定運用を維持する手法が必要となる．

公衆網適用時に不具合が発生しない様にソフトウェアに内在する不具合を開発期間中に発見しつつ，開発工数／期間の増加を抑止するためには，開発プロセスの定式化および，その自動化が必要であり，その検討は開発の一連の工程に渡り進める必要がある [2, 3]．また，公衆網運用中の不具合発生によるサービス中断時間を少なくするためには，開発期間中に発見する不具合の「数」を増やすだけでなく，サービス継続に影響する不具合の「質」に着目し，開発期間中により多くの不具合を発見する手法が必要である．

ソフトウェア品質を高めつつ，開発工数／期間の増加を抑止，減少させるた

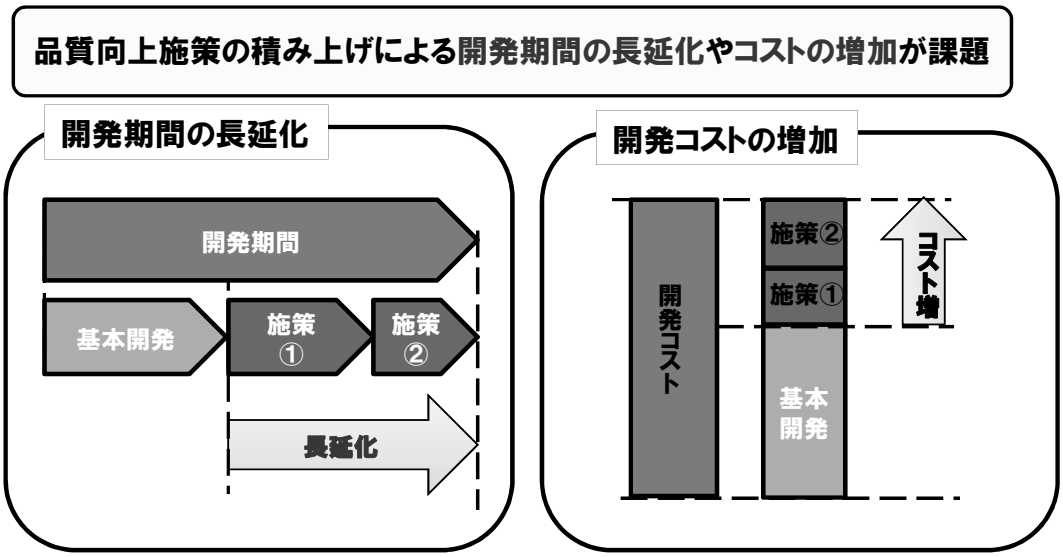


図 1.1　高品質なソフトウェア開発のための課題

めには，以下の課題がある．

(1) 開発工数（試験工数）を増やすことなく品質の向上を実現する手法の確立．
(2) 品質を維持したまま，開発工数を減らす手法の確立．

一つ目の課題に対しては，公衆網運用中に発生する不具合対処毎に増加する稼働削減への効果も期待できる不具合の「数」に着目し，開発時の試験項目の「数」を調整する事で，開発時不具合発見数を増加させ，公衆網適用時の不具合発生数を減少させる手法について提案した．

開発工数の観点においても，試験工程で機能部毎の試験項目数を調整する事でシステム全体としての試験項目数を変えない，すなわち開発工数を増加させない試験項目選定手法を提案し，その手法をキャリアネットワークである NGN のシステム開発に適用した．

二つ目の課題に対しては開発のための要求仕様書と試験項目を教師データとし，機械学習により自動的に試験項目を抽出する手法について提案した．

高いソフトウェア品質を継続的に維持するためには，試験工程において確実に不具合を発見する手法にかかっている．不具合を発見するためには仕様を正確に理解し，検証すべき試験項目を網羅的に且つ正確に実施しなくてはならないが，検証すべき試験項目が正しく選定されなかったり，網羅的に検証が行わ

れない場合は，不具合を内包したままシステムを提供しサービスの継続性を損ねてしまう事がある．検証すべき試験項目は，開発時に作成される要求仕様書を基に作成されるが，適切な試験項目を作成できるかどうかは作業者のスキルやノウハウに大きく依存している．このスキルやノウハウの偏りによる不均質な試験項目作成を避けるため，試験項目作成基準を定めるたり試験項目レビュー等に大きく時間を割いており，この稼働がソフト開発のコスト増につながってしまっている．

このコストを増大させる課題を解決するため，本論文では機械学習により試験項目を自動的に作成する手法を提案した．一つ目の課題同様，キャリアネットワークである NGN のシステム開発に適用した．

1.2.2　P2P 通信サービス提供における課題

高いソフトウェア品質を維持することで，次世代ネットワーク (NGN) は SIP ベースのセッションによる帯域幅と品質 (損失，遅延，揺らぎ) を保証した通信機能を継続的に提供している．加えて，物理 IP ネットワークのセッションを制御するインタフェースが積極的に公開されてきている．

例えば，各国の主要通信事業者のインタフェース提供 [4] に加え，現在は，ネットワークセッション制御の事業者として Twilio[5] などがでてきており，このセッション制御インタフェースの本格的な利用が注目されてきている．NGN にも SIP ベースのセッションを切換る Application Network Interface (ANI) 機能がある (例えば, Parlay-X Web Service API [6]).

SIP ベースのセッションによる帯域幅と品質 (損失，遅延，揺らぎ) を保証した通信機能の継続的且つ安定的な通信の提供は，継続的に安定してデータが流れることが望ましいデータストリーム型の通信サービスに有益であると考えられる．映像等のデータストリーム型の通信はインターネットの主要トラヒックであり今後もトラヒックが増大していくと予想されており [1]，より多くのユーザに利用されるサービスである．さらにはデータ転送負荷の集中が避けられるユーザ端末間の Peer-to-Peer(P2P) ネットワークを NGN のセッションを用いて構成することがより望ましい形態であるともいえる．

しかしながら，ネットワークリソースの観点からみると，データストリーム転送は，データストリーム配信サーバとエンドユーザ端末間の 1 対 N のセッション接続によりデータストリーム配信を行うと，データストリーム配信サー

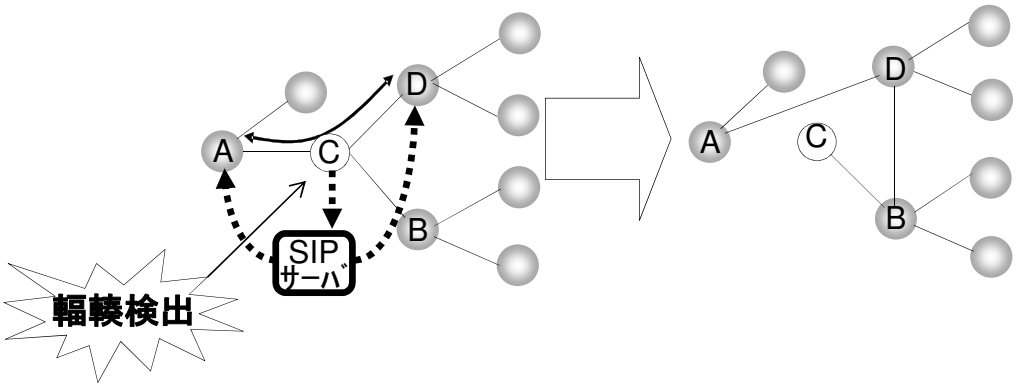


図 1.2　P2P 型のセッション制御

バ側のデータ転送負荷の集中が懸念される．

NGN の SIP による帯域幅と品質が保障されるセッション制御機能を利用しつつ IP マルチキャストのような方法に Session Description Protocol (SDP) でマルチセッション [7] を設定することが考えられる．しかし，実際の NGN では，帯域幅と品質を保証するセッションはユニキャスト通信機能のみを提供し，複数地点への再配信等の高度な機能は User Network Interface (UNI) で接続されるユーザ設備にて行っている．従って，多数のユーザ設備端末がデータストリームの中継点となりバケツリレー的に再度配信を端末間のセッションで行う形態がデータ転送負荷の集中を避けられ望ましいといえる．これは，ユーザ端末間のアプリケーションレベルのネットワーク，Peer-to-Peer(P2P) ネットワークを NGN のセッションを用いて構成することといえる．NGN において通信品質を保証した SIP ベースのセッションで端末間をつなぎ P2P ネットワークを構成することで，帯域幅と品質が保障された様々な P2P サービスへの展開が可能となる．

P2P ネットワークはエンドユーザの各端末がアプリケーションレベルのクライアントでありサーバであり中継ルータでもあり，トラヒック集中が偏在する可能性がある．また，各端末は Windows Update 等のピア内他 APL 処理負荷が要因で予想できない輻輳状態に陥る可能性もある．このような課題に対し，本論文では輻輳した端末がボトルネックとなりサービス品質が低下しないよう，P2P ネットワークの中継ルートを効率化させる P2P ネットワークトポロジー制御方法を提案する（図 1.2）．

1.3　目的

大規模通信システム開発においては，ウォーターフォールモデル（V字モデル）による開発を進めている．実際の開発においては各工程の実施内容やアウトプットを詳細に定め，工程ごとに品質の判定を行う事で，手戻りの少ない高品質なソフトウェア開発を目指している．

本論文では，大規模通信システムを公衆網適用した際に不具合が発生しない様にソフトウェアに内在する不具合を開発期間中に発見し品質を高めつつ，開発工数／期間の増加を抑止，減少させるため，開発プロセスの定式化および，自動化を提案し，その効果を明らかにする事を目的とする．

また，高いソフトウェア品質を維持しつつ，次世代ネットワーク (NGN) は SIP ベースのセッションによる帯域幅と品質 (損失，遅延，揺らぎ) を保証した通信機能を継続的に提供しているが，この品質をアプリケーションレベルも維持しなくては，サービスとしての品質保証とならない．NGN を利用した P2P 通信サービスにおいて，輻輳した端末がボトルネックとなりサービス品質が低下しないよう，P2P ネットワークの中継ルートを効率化させる P2P ネットワークトポロジー制御方法が必要であり，本論文ではユーザ端末アプリケーションの負担が少なく P2P ネットワークトポロジーを効率のよい構成へ変える方式を提案し，各方式の比較結果をシミュレーションによって検証し，実装ソフトウェアアーキテクチャのフィージビリティを試作検証によって確認し有効性を明らかにする事を目的とする．

開発プロセスの定式化および，自動化に関する提案方式のねらいと有効性の観点 (評価の観点)，及び，P2P ネットワークトポロジーの効率的構成変更方式のねらいと有効性の観点 (評価の観点) を以下に示す．

(1) 開発プロセスの定式化および，自動化に関する提案方式のねらい

大規模通信システムを公衆網適用した際に不具合が発生しない様に，開発工数／期間の増加を抑止しつつ，ソフトウェアに内在する不具合を開発期間中に発見し品質を高めことができる提案手法である事が重要である．

また，システムの品質に大きく影響する試験項目について，高いスキルを保有する技術者と同等の内容を自動的に作成でき，開発期間を短縮できる事が重

要である．

【有効性の観点 (評価の観点)】

ソフトウェアに内在する不具合を，開発工数／期間の増加を抑止しつつ，開発期間中に発見する数を増加させ，公衆網適用後に発生する数を減少させることが重要な観点である．

また，システム開発時の試験項目について，高いスキルを保有する技術者と同等の内容を自動的に作成できる事が重要な観点である．

(2) 大規模通信システムを利用した P2P ネットワークにおける物理ネットワークセッション制御のねらい

大規模通信システムを利用した P2P ネットワークが物理ネットワークセッション制御機能を利用する際，ユーザ端末アプリケーションの負担が少なく P2P ネットワークトポロジーを効率のよい構成へ変える方式が重要である．

【有効性の観点 (評価の観点)】

ユーザ端末アプリケーションの負荷及び，P2P ネットワークトポロジー構成変更負荷を小さくすることが重要な観点である．

1.4　本論文の構成

第 2 章では，研究対象である NGN の概要，また，NGN システムを利用した P2P ネットワークの概要を述べると共に，関連するシステム開発手法及び P2P ネットワークサービス技術の研究・開発を紹介する．

第 3 章では，公衆網運用中に発生する不具合対処毎に増加する稼働削減への効果も期待できる不具合の「数」に着目し，開発時の試験項目の「数」を調整する事で，開発時不具合発見数を増加させ，公衆網適用時の不具合発生数を減少させる手法について提案する．開発工数の観点においては，提案手法は試験工程で機能部毎の試験項目数を調整する事でシステム全体としての試験項目数を変えない，すなわち開発工数を増加させない試験項目選定手法であり，その手法をキャリアネットワークである NGN のシステム開発に適用した結果について述べる．

第 4 章では，高いソフトウェア品質を継続的に維持するため，スキルやノウハウの偏りによる不均質な試験項目作成を避け，適切な試験項目作成を行う必要があるが，この試験項目作成稼働を増大させる課題解決に向け，機械学習により試験項目を自動的に作成する手法を提案する．提案手法によって自動的に作成された試験項目については，有スキル者によって作成された試験項目と比較し，キャリアネットワークである NGN のシステム開発に適用した結果について述べる．

この様な開発手法を用い高信頼／高品質なシステムによる通信サービスを提供する NGN を利用したとしても，P2P 通信サービスにおいてはエンドユーザの各端末がアプリケーションレベルのクライアントでありサーバであり中継ルータでもあるため，トラヒック集中が偏在する可能性がある．

第 5 章では，輻輳した端末がボトルネックとなりサービス品質が低下しないよう，ユーザ端末アプリケーションの負担が少なく P2P ネットワークトポロジーを効率のよい構成へ変える方式を提案する．接続切換機能についてはいくつか方式が考えられるが，これらの方式の特性を比較分析しシミュレーションによって検証し，実装ソフトウェアアーキテクチャのフィージビリティを試作検証によって確認した結果について述べる．

第 6 章では，本論文をまとめ，今後の課題を述べる．

第 2 章

大規模通信システム開発と P2P ネットワークに関する関連研究

本章では研究対象の大規模通信システムを代表する NGN の概要，また，NGN システムを利用した P2P ネットワークの概要を述べ，関連するシステム開発手法及び P2P ネットワークサービス基盤技術の研究・開発を紹介する．

2.1 NGN の概要

一般家庭における映像コンテンツ配信やオンライン商取引の利用に見られるように，ブロードバンド通信市場は世界的な拡大を見せており，これに伴い，日本においては安価で常時社会基盤として依存できるネットワーク（ディペンダブルネットワーク）の実現が期待されてきた．

IP マルチメディアサブシステム (IMS)[8] に準拠し，Session Initiation Protocol(SIP) ベースのセッションによる帯域幅と品質 (損失，遅延，揺らぎ) を保証した通信機能を提供する NGN(Next Generation Network)[9] は，この様な社会的背景から，従来の音声通信の中心であった PSTN（Public Switched Telephone Networks）が持つ信頼性や安定性と，IP(Internet Protocol) 網の柔軟性を併せ持つ次世代の情報通信ネットワークとして誕生した．NGN のアーキテクチャを図 2.1 に示す．

インターネットは自律分散協調型のネットワークであるため，個々のドメイ

ンでのシステム機能が完結しており，各々のシステムのドメイン間連携はゆるやかな結合性を持つことが多い．このため，個々のシステムサイズは小規模化し，小規模であるが故に旧来の大規模開発では実現が困難であったアジャイル型で安価な開発を比較的容易に可能としている．

これと比較し，従来型の電話サービス（PSTN）をも提供する NGN はインターネットの特徴と PSTN の両方の特徴を合わせ持っている．このため，従来型の電話による通信サービスに関しては信頼性や安全性の保証を要求され，社会的な依存性を求められており，結果として NGN システムの開発は高信頼，高品質を求められていた PSTN のような大規模システム開発と同等の開発モデルを採用している．

NGN における電話サービスは 2007 年度のサービス開始から段階的な機能の追加開発を行い，機能拡充をはかってきた [10, 11].

NGN のサービス開始前のトライアルサービス実施においては，トライアル実施中に不具合が多発した事から，実際の NGN 公衆網に適用するソフトウェアの開発工程においては不具合摘出数の向上施策を継続的且つ積み上げて実施することで，公衆網適用時の残存不具合数を極力少なくする事とした．しかしながら，開発中の不具合摘出のための施策の積み上げは開発期間の長延化や開発コストの高止まりを引き起こし，課題となっていた．

2.2　P2P ネットワークの概要

世界的な IoT/M2M サービスの拡大は，人と人，物と物といった個々のレベルでの情報交換の爆発的増加を引き起こし，そのような情報交換を行う様々なコミュニティも増加していく．この様な人と人，物と物といった個々の情報受発信の基盤技術として，個々の端末の自由で自律的な参加・離脱により組織化される Peer-to-Peer (P2P) ネットワークが有利である．

P2P ネットワークは図 2.2 に示すように，物理的な IP(Internet Protocol) ネットワークに接続している各端末に搭載されたソフトウェアがアプリケーションレベルの中継ルータの役割を果たし仮想的なオーバレイネットワークとして構成するものである．

P2P ネットワークの利点は，スケーラビリティの高さである．クライアント-サーバシステムでは，クライアント端末の数が増えていくと，サーバの負荷

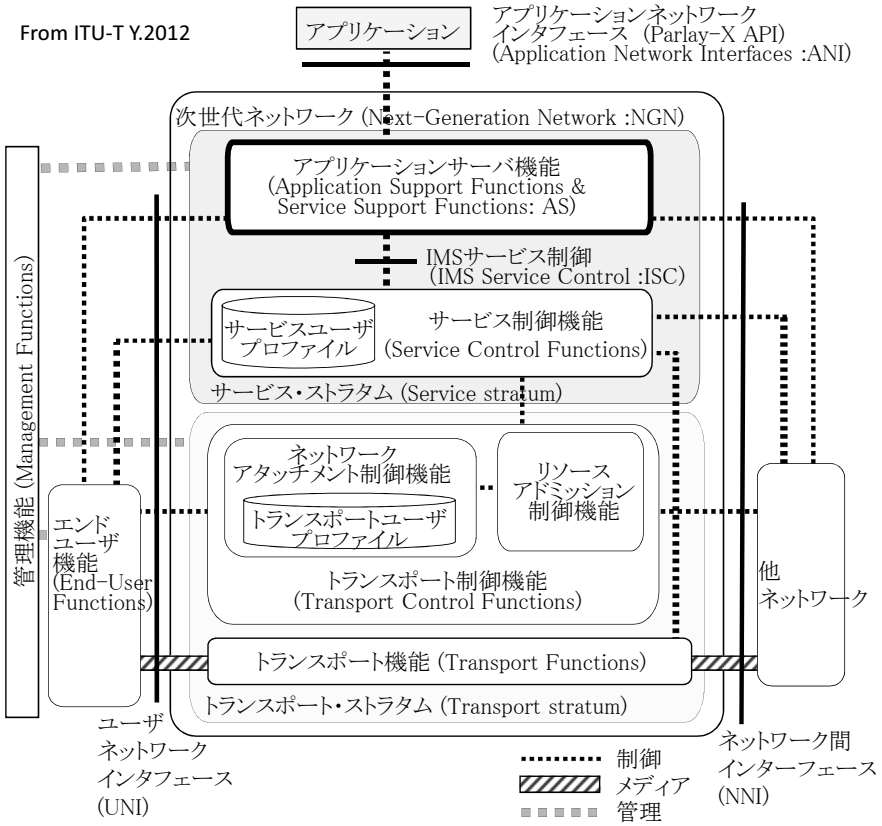


図 2.1　NGN アーキテクチャ (ITU-T Y.2012)

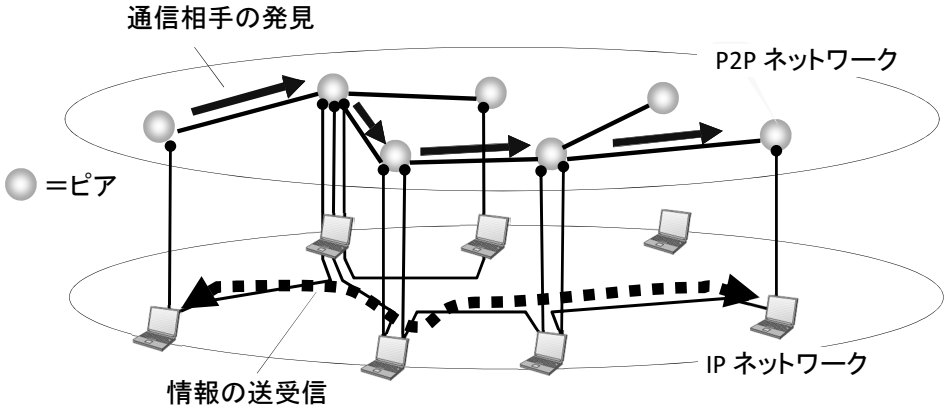


図 2.2　P2P ネットワーク

もそれに比例して増えていくため，クライアント端末数が膨大になったとき，サーバの処理能力，あるいは，サーバにつながっているネットワーク回線の能力が限界に達して，実用的な配信ができなくなるという問題がある (スケーラビリティが低い)．特に動画データの配信を行う場合等サーバから送るデータのサイズが大きい場合に，問題が顕著になる．

これに対して，P2P ネットワークでは，サーバを使わない (サーバを使わない方式はピュア (Pure)P2P と呼ばれる) 方式が採用できるため，ユーザ数が膨大になっても，サービスを提供しつづけることが可能である．クライアント-サーバシステムに比べて，サーバ装置に高性能な設備を用意しなくて済み，通信回線も通信帯域幅の細い安価な回線で済む．これは，端末数が増えれば増えるほど顕著な差となる．

もうひとつの P2P ネットワークの利点は，耐故障性である．クライアント-サーバシステムでは，サーバがダウンするとシステム全体が停止するが，P2P ネットワークでは，単一故障でシステム全体のサービスを停止させてしまう箇所を減らすことができるので，耐故障性が向上する．サーバに頼る箇所が少なく，たとえサーバ部分が故障しても，各ピアのキャッシュにあるデータに関してはサービスを続行できるため，この効果は顕著であり，端末数が多くなればなるほど，システムの耐故障性が向上する．

2.3　高品質なソフトウェア開発のための試験実施手法に関する関連研究

NGN はインターネットの特徴と従来型の電話サービス（PSTN）の両方の特徴を合わせ持っている．従来型の電話サービス提供において通信制御信号の紛失による通信途絶は社会的影響が大きく，これまでも信号集中等による輻輳対策方式 [12] を検討してきた．また，システム自身の不安定さによるサービス提供阻害を避けるため，高信頼，高品質を求められる大規模システム開発においては，段階的に行われる追加機能開発毎にそれまでに発生した不具合や，生じた手戻りを参考にし，運用時に不具合が残存しないようにするための施策を取り入れ，繰り返し実施が必要な施策については開発時に積み上げ的に施策実施を行っている．このため，開発が進むほど実運用におけるシステムの安定性は向上している反面，開発時に行う施策の積み上げは開発期間の延伸や開発工

数の増加を引き起こしている．

運用に不具合を残存させないための開発時の検証の強化施策が行われる事は一般的であり，不具合発見のため発生不具合に関連する試験を追加で実施しその評価を行った報告や，システム全体の試験項目に対し観点を設け優先度づけをすることで選択的試験を行った手法の評価について報告されている [13, 14, 15].

また，設計／製造／試験時の評価を行うにあたっては，SLOC/LOC(Source Lines Of Code / Lines Of Code) を用い不具合摘出密度の統計量を把握する事 [16] や評価軸の一つとして，不具合摘出の難易度を盛り込み工程単位に評価した報告がある [17].

しかし，システム開発において提案手法の効果の有無を確認するためには開発期間の不具合発生状況だけでなく，長期間の運用を通し，開発期間に発見できなかった不具合の数や不具合によって運用時にサービス継続を阻害したか否かを測る必要がある．

このため，3 章では大規模システム開発における提案手法の評価に向け，開発期間での不具合摘出から公衆網での運用期間まで含めた不具合発生状況の評価を行った．

また，複数の開発において不具合の管理情報を収集し，それぞれの運用期間における不具合発生状況からソフトウェアの機能部単位に特有の係数を求め，不具合発生の期待値との比較を行う事で提案手法の効果を評価するアプローチもとった．

2.4　高品質なソフトウェア開発のための試験項目作成手法に関する関連研究

システム開発における単体試験工程では，自動試験，自動試験項目抽出が研究開発され多くのサポートツールが一般的につかわれている [18]．また，ほとんどのツールでは，ソースコードを解析することで，試験項目を生成する．しかしながら，通信サービスを含めたシステム開発において，自然言語で記述される要求仕様書を基に試験項目を作成するシステムにおける安定化試験工程には，それらのツールを使うことができない．

様々な研究者は，試験項目生成のため，Unified ModelingLanguage（UML）

を用いた設計を使っている．UML は，図を用いる事を含めた仕様記述言語である [19]．しかし，大規模システムのソフトウェア開発においては，開発項目が多岐にわたるため，多くの関連者によって分担してアイデア，ビジネス・プロセス，ビジネス・ルール，およびその他の仕様を記述する．そして，関連者の全てが UML を十分に理解し同じレベルで使用する事が出来る訳では無い．このため，UML ではなく自然言語を使用している [20, 21].

参考研究 [20, 21] では，要求仕様書の分析技術として，文章を木構造化し，構成する要素をデシジョンテーブルテスト技法を用いることで，試験項目となるものを表に出力していた [22]．また分析のため，アルゴリズムを作成し，要求仕様書に適用している．

我々の研究対象とする大規模通信ソフトウェア開発における要求仕様書は複雑で省略の多い自然言語で記述されることが多い．そのため，4 章では定型的な文章構造分析ではなく，機械学習技術を利用することで要求仕様書を分析し，試験項目の自動生成を目指している．

2.5　大規模通信システムを利用した P2P ネットワーク制御に関する関連研究

これまで，P2P 技術を用いて低コストにコンテンツ配信を行うための取り組みがネットワーク高度利用推進協議会など多方面で進められてきた．近年，PPStream[23] や PPLive[24] に代表される P2PTV とよばれる P2P を用いたソフトウェアアプリケーションで動画ストリーミングを流すサービスが広がりをみせている．

現在，著作権上も問題ない商用サービスとして WebRTC[25] を使った P2P データ配信プラットフォームサービスである MistCDN[26] が提供されている．また，主要な P2P サービスである BitTorrent も P2P 技術を応用したストリーミングプラットフォーム BitTorrent Live[27] によるストリーミングサービスを提供した．P2P 型のコンテンツ配信，ストリーミング配信が有望であることを示しているが，これらは，ピアの輻輳の回避やネットワーク機能を利用して帯域幅や品質を保証するものではなかった．

IMS ベースの機能を活用して P2P 型のコンテンツ配信サービス行う検討も進められており，3GPP にて IMS based peer-to-peer content distribution services

が公開されている [28]．これは，無線網と固定網を横断的にピア間でコンテンツを転送することを狙い，コンテンツを保有するピアの検索をサポートし，また，コンテンツキャッシュサーバをアサインする機能を提供するものである．IMS をベースにしているが，SIP を使って物理ネットワークの通信リソースを P2P サービスに適応させるものではない．

従来から，P2P と SIP の連動については，SIP を用いて P2P を構成する P2P over SIP と，SIP サーバを P2P 型の分散アーキテクチャで構成する SIP using P2P が検討され，IETF の P2PSIP WG で標準化も議論されてきていた．SIP を用いて P2P を構成する P2P over SIP は，SIP の Location Service に Distributed Hash Tables (DHT) を適用し，主に Chord[29] 等の DHT を用いた Structured 型 P2P を実装するものであった [30]．ただし，本方式は，SIP の REGISTER 機能を活用するものであり，セッションによって確保される通信リソースを利用するものとはなっていない．

端末アプリケーションである P2P 基盤ソフトウェアの P2P ネットワークトポロジー制御機能によって P2P ネットワークトポロジーを制御することが考えられてきた．例えば，MSMT/MBST(Minimum Stress Multicast Tree/Maximum Bandwidth Multicast Tree) 法では，物理的なネットワーク構造を把握することによって，端末間の通信遅延を考慮し，特定の端末に通信負荷が集中することを避ける処理を各端末ソフトウェア機能が相互連携する処理で実現している [31]．これは，接続先端末と連動した切換処理中に輻輳状態に陥り切換処理が完了できず切換失敗となることが懸念される．

5 章ではこれら課題に対し，輻輳した端末がボトルネックとなりサービス品質が低下しないようにするため，NGN が保有する帯域幅と品質が保障されるセッション制御機能を利用し，P2P ネットワークの中継ルートを効率化させる P2P ネットワークトポロジー制御方法を提案し評価した．

第 3 章

高品質なソフトウェア開発のための試験実施手法と評価

本章では公衆網運用中に発生する不具合対処毎に増加する稼働削減への効果も期待できる不具合の「数」に着目し，開発時の試験項目の「数」を調整する事で，開発時不具合発見数を増加させ，公衆網適用時の不具合発生数を減少させる手法について提案する．

開発工数の観点においても，試験工程で機能部毎の試験項目数を調整する事でシステム全体としての試験項目数を変えない，すなわち開発工数を増加させない試験項目選定手法を提案し，その手法をキャリアネットワークである NGN のシステム開発に適用した結果を報告する．

3.1　背景と目的

ブロードバンド通信市場の世界的な拡大は IoT の牽引もあり，モバイル市場だけでなく，固定ブロードバンド市場の拡大を継続させている [32, 33].

日本の固定ブロードバンド通信は世界的にも高速，安価 [34] であり，B フレッツやフレッツ光プレミアムに代表される FTTH(Fiber To The Home) サービスの展開は日本のブロードバンド通信の本格的な普及を支えている．

しかしながら，インターネットを基盤とした映像コンテンツ配信やオンライン商取引等の世界的拡大は，競争をさらに加速させ，電話サービスとインターネットサービスの両方を提供するキャリアネットワークにおけるネットワークサービス開発のローコスト化を余儀なくしている．

キャリアの通信ネットワークを構築するハードウェアは，コモディティ化や仮想化により，設備面でのローコスト化は進んだが，各種通信サービスを提供するソフトウェアはサービスバリエーションが豊富で独自性が高く，大規模であるため，公衆網適用時に継続的安定運用を可能とするために必要な開発期間や開発工数の高止まりの改善が課題となっている．

通信サービスに代表される大規模システムは社会の基礎的なインフラであるため，信頼性や安全性の保証や社会的な依存性を保証しなくてはならないが，早期のサービス提供のために短期開発にしたりコストカットをおこなうと，ソフトウェアに内在する不具合を公衆網適用時にまで残し継続的安定運用を維持することが困難となる．

このため，大規模で且つ信頼性，安全性の保障が必要であり，災害時を含めサービスの継続性を要求されるソフトウェアの開発においては，開発工数や開発期間を増加させることなく，ソフトウェアに内在する不具合を開発期間中に発見し，公衆網適用時の継続的安定運用を維持する手法が必要となる．

公衆網適用時に不具合が発生しない様にソフトウェアに内在する不具合を開発期間中に発見しつつ，開発工数／期間の増加を抑止するためには，開発プロセスの定式化および，その自動化が必要であり，その検討は開発の一連の工程に渡り進める必要がある [2, 3]．また，公衆網運用中の不具合発生によるサービス中断時間を少なくするためには，開発期間中に発見する不具合の「数」を増やすだけでなく，サービス継続に影響する不具合の「質」に着目し，開発期間中により多くの不具合を発見する手法が必要である．

本章では公衆網運用中に発生する不具合対処毎に増加する稼働削減への効果も期待できる不具合の「数」に着目し，開発時の試験項目の「数」を調整する事で，開発時不具合発見数を増加させ，公衆網適用時の不具合発生数を減少させる手法について報告する．開発工数の観点においても，試験工程で機能部毎の試験項目数を調整する事でシステム全体としての試験項目数を変えない，すなわち開発工数を増加させない試験項目選定手法を提案し，その手法をキャリアネットワークである NGN のシステム開発に適用した結果を報告する．

以降，第 3.2 節では，システム開発時の各工程の概要や不具合の管理方法，開発工数の考え方を示す．第 3.3 節では検証手法の提案，第 3.4 節では具体的に提案手法を NGN に適用した時の結果を示す．

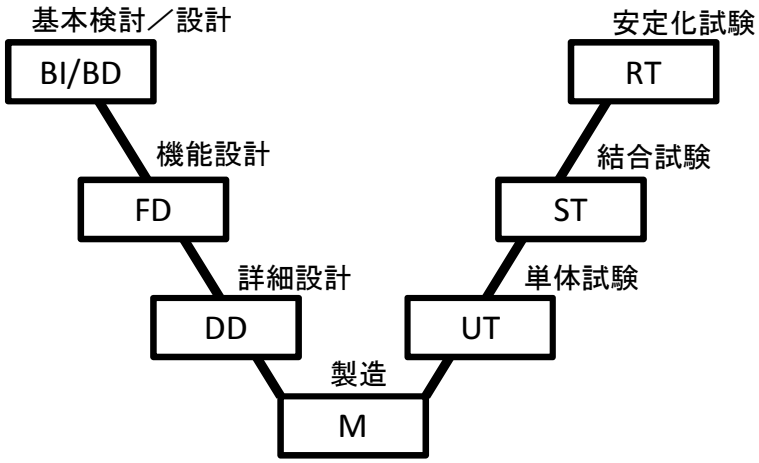


図 3.1　ソフトウェア開発モデル

3.2　システム開発モデルと管理手法

3.2.1　システム開発モデル

一般的に，大規模で高品質／高信頼／継続性を要求されるシステム開発は，手戻りやソフトウェアデグレードのリスクを考慮し，要求仕様の決定から開発完了まで各工程をシーケンシャルに進める V 字型のウォーターフォールモデルでの開発を行っている（図 3.1）.

また，大規模ソフトウェア開発においては開発の効率性向上のため，サービスを実現するソフトウェアをいくつもの機能部として分割して，機能部間のインタフェースを定義し，機能部毎の役割分担を明確にすることで分担可能なソフトウェア開発を行っている（図 3.2）.

ウォーターフォールモデル開発では，『前工程の不具合が完全に排除された事』を前提に各工程が進められるため，前工程が完了しないと後工程に進まず，前工程の終了時には，工程完了の審議を行い，不具合の排除が行われているかどうかを判断しながら進めることが一般的である.

言い換えれば，『後工程で発見された不具合に対する前工程に遡った対処の手戻りが大きい』ため，各工程毎に十分な不具合の発見ができる様に各種施策を行っている.

以下にウォーターフォールモデルでの開発工程を規定した開発工程の区分例

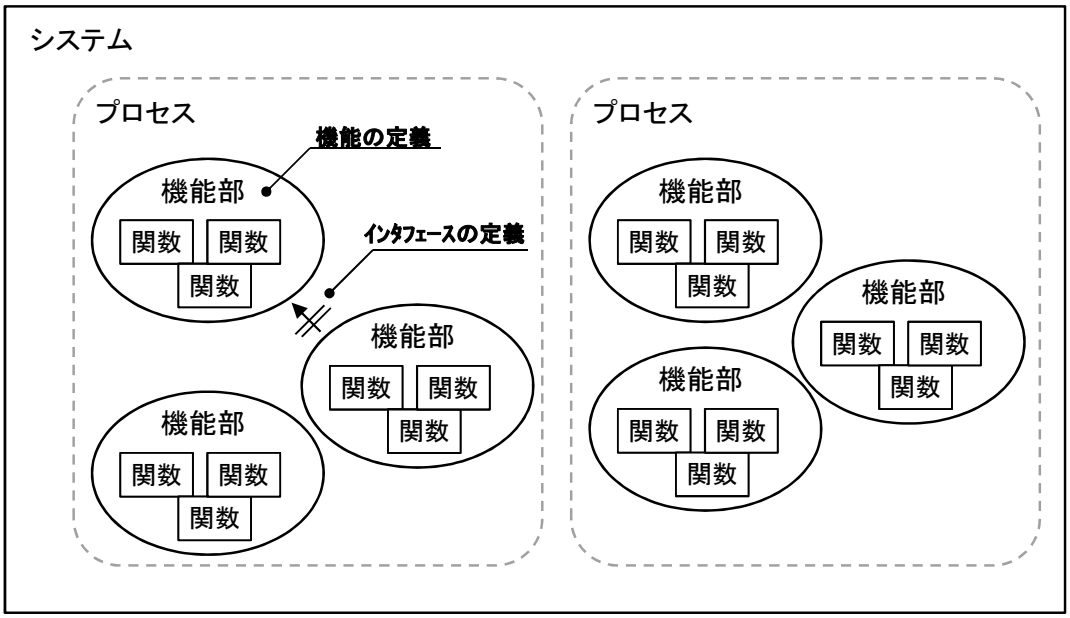


図 3.2　ソフトウェア構造

と作業及びアウトプットの詳細を示す.

■基本検討/設計 (BI/BD:Basic Investigation/Design)

- 要求条件をまとめ，開発機能の基本検討を行う
- 開発項目を取りまとめ各機能部へのマッピングを行う

【本工程でのアウトプット】

要求仕様書／基本設計書／外部条件仕様書 等

■機能設計 (FD: Function Design)

- 各機能部の設計（機能分担／処理概要）に関わる技術検討を行う

【本工程でのアウトプット】

開発項目設計書，機能部間インタフェース仕様，設計書 等

■詳細設計 (DD: Detail Design)

- 関数レベルの処理フローおよびデータ構造の設計を行う.

【本工程でのアウトプット】

機能部内詳細設計 等

■ 製造 (M: Manufacture)

- ソースコード作成を行う．

【本工程でのアウトプット】
ソースコード

■ 単体試験 (UT: Unit Test)

- 個別環境において，関数レベルでの動作確認と機能部内の関数を結合した試験を行う
- 以下の確認を行う．
  - 各関数単体の正常性
  - 機能部内での関数結合の正常性
  - 各データ構造確認 等

【本工程でのアウトプット】
単体試験項目

■ 結合試験 (ST: System Test)

- 複数の機能部間にまたがった機能確認を行う．
- 関連する全プロセスを実行した環境にて，単一外部イベント印加による動作確認を行う．
- 基本性能測定を行う．
- 対向システムと接続し，インタフェース動作の検証を行う．

【本工程でのアウトプット】
システム内の結合試験，一次性能測定試験，システム対向試験項目

■ 安定化試験 (RT: Running Test)

- 複数の外部イベント印加（競合）によるシステム動作確認を行う．
- システム全体の安定動作確認を行う
  （外部イベント連続印加，過負荷，障害耐性，性能評価）

【本工程でのアウトプット】

複数イベントによる競合試験，システム安定動作確認

（イベント連続／過負荷／障害等），長時間連続運転項目

3.2.2　システム不具合の管理手法

一般的に，大規模システム開発においては，その試験工程，及び，実際の商用フィールドでの不具合の管理および対処のために，不具合を BTS(Bug Tracking System) と呼ばれる一元的データベースで管理する．不具合は発見から対処完了まで複数のステータスで管理されると共に，不具合内容も含めて管理を行う．

ソフトウェア開発における代表的な不具合管理ツールとしては Mozilla Foundation の提供する Bugzilla[35] や Ruby on Rails で開発されている Redmine[36] があるが，後述する NGN システム開発においては独自のツールを用いている．不具合管理は対処を確実にするためだけでなく，以降の開発において，開発対象システムがどのような特徴を持つ不具合が発生しやすいか，また，どのような開発工程で不具合が混入しやすいかを明確にする事が可能であり，継続的なシステムソフトウェア開発においては，不具合管理は極めて重要である．以下には代表的な具体的管理項目を示す．

- 不具合発見工程
  不具合が発見された開発時の試験工程の何れかまたは，商用運用中か否かを管理
- 不具合発生箇所
  不具合を発見したソフトウェア機能部を管理
- 重要度
  不具合の影響度に応じて重要，通常，問題無しに分類し管理
- 不具合内容
  不具合の発生事象，原因，対処方法を管理
- 不具合を摘出すべき本来の工程
  本来発見し対処すべきであった開発工程について管理
- 不具合発生および解決日付

不具合の発生から対処ファイル作成完了までの期間を管理

上記の管理項目のうち，『重要度』は発見した不具合が与えるサービスに支障があった範囲により決定される．重要度が「重要」となるのは，主要な機能の損失に繋がる場合であり，NGN 開発のケースでは以下の不具合を「重要」とした．

- システムダウンにつながる不具合
- 大規模輻輳につながる不具合
- サービス停止につながる不具合
- 復旧にシステムリブートが必要な不具合
- 通信不可につながる不具合
- 料金の誤課金につながる不具合
- 性能を著しく低下させる不具合

管理項目「重要度」はシステムの継続的安定運用の可否を決定するため，本章では提案手法の効果を判断するために不具合数だけでなく，不具合の重要度についても評価を行った．

3.2.3　開発生産性

公衆網適用中の不具合発生を抑止するため開発期間内の不具合摘出数を増やしつつ，開発工数／期間を増加させないためには，一連のソフトウェア開発工程において工数の比重が大きい工程に対策を適用する事が重要となる．

中小規模のウォータフォール型のソフトウェア開発（ソフトウェア規模が 100KLine 以下，月あたりの開発要員数が 10 人以下）においては，設計工程 (基本設計～詳細設計)，製造工程（製造～単体試験），試験工程（結合試験～安定化試験）の工数比率がそれぞれ全体の 4:3:3 程度であることが報告されている [37].

しかしながら，大規模のシステム開発においては，システムの保有する機能

が多いため機能間連携を考慮する必要があり，基本設計から詳細設計期間の検討における工数が大きくなる．また，ソフトウェアを複数の機能部に分け並行して開発するため，製造期間（単体試験含む）に比べ，機能部を結合した後の試験（結合試験～安定化試験）の工数が多くなる．

本論文において評価対象とした NGN システムは大規模のシステム開発（ソフトウェア規模が数 MLine 以上で，月あたりの開発要員数が 20 人以上）であり，評価対象とした期間における設計／製造／試験工程の工数の比率は 34%～40%，10%～14%，50%～53% であった．

また，試験工程をさらに細分化し，工数を比べた場合，以下の通り，実際の試験準備と試験実施の工数が 80% であり開発工程全体の 4 割を占める事が分かる．

【試験工程における工数比率】

- 管理 (12%)
- 試験準備 (22%)
- 試験実施 (58%)
- NG 時データ収集・解析 (4%)
- NG 時対処 (5%)

このようにシステム開発において，試験項目数の多寡が開発工数に密接に関わるため，本章では試験項目数を増やす事なく開発期間の不具合摘出数を増やす手法を提案しその評価を行った．

3.3　提案手法

前述した通り，公衆網運用中の不具合発生によるサービス中断時間を少なくするためには，開発期間中に発見する不具合の「数」を増やすだけでなく，サービス継続に影響する不具合の「質」に着目し，開発期間中により多くの不具合を発見する手法が必要である．また，開発費用の増加を抑止するためにも開発工数を増加させない手法である事も必要となる．

「質」の観点から公衆網運用中のサービス継続を阻害する不具合を減らす事

も必要ではあるが，本章ではまず公衆網運用中に発生する不具合の対処毎に増加する稼働削減への効果も期待できる不具合の「数」を減少させる手法を提案し評価する．提案手法は，一つ前に先行して開発されているソフトウェアの開発時，及びそのソフトウェアが実際に運用される中での不具合発生の「数」を把握し，特定のソフト部分毎に試験項目の「数」を調整する事で全体の開発期間の延伸や開発工数の増加を抑えつつ開発時の不具合発見数を増加させ，公衆網運用中の不具合発生数を減少させる事を目標としており，以降，提案手法の具体的内容について述べる．

3.3.1　提案手法概要

通信システムに代表される，大規模で信頼性を求められるシステムの開発において，システムソフトウェアはウォーターフォールモデルに代表される幾つかの工程にて開発が行われ，開発の完了後，実際の公衆網にて使用される．また，効率的な開発を行うため大規模ソフトウェア開発においてはソースコードをいくつもの機能部として分割し，機能部間のインタフェースを定義したり，機能部毎の役割分担を明確にすることで分担可能な効率的ソフトウェア開発を実現している．

ソフトウェア開発の試験工程では，公衆網での運用中に不具合が発生しないように不具合の摘出を行うが，開発期間や開発工数には一定の限度があり，不具合の完全な枯渇のために無限に期間や開発工数を割り当てるわけにはいかない．このため，一般的なソフトウェア開発においては，開発内容や開発規模に応じて一定の開発工数及び期間が設定される．すなわち，開発規模に応じて，一定の比率で試験項目数が決まり，開発期間や開発工数が決まることになる．

試験工程の期間や開発工数を増加させずに開発完了時のソフトウェアの不具合を減らすためには，何らかの「特定部分」の試験を増加させ，別の部分の試験を少なくすることで全体としての試験数を増やさない手法が考えられる．ここで述べる「特定部分」の例としては，開発項目が複数の場合には開発項目単位，ソフトウェアが複数の機能部で構成されている場合には機能部単位，複数のメンバーで作成している場合にはメンバー単位といった手法が挙げられる．しかし，無作為に「特定部分」を選定し試験項目の数に偏りをつけ試験項目数を増減すると，試験工程終了時に不具合が残存する可能性を増加させ，公衆網

適用中の不具合発生により継続安定運用に支障をきたすことになる．

ソフトウェアの初期開発時においては，開発以前のソフトウェアの状況を図る情報が無いため，的確に特定部分を選定し，試験項目に偏りを与える事は困難であるが，段階的なソフトウェア開発の場合には，直前のソフトウェア開発時の不具合摘出傾向からソフトウェアの状況が想定でき，特定部分の選定が可能となる．

直前に開発した機能について公衆網での運用期間において不具合が発生したケースを想定した場合，公衆網運用時に発生する不具合は，開発時に発見する事が難しい不具合である可能性が高く，また，開発したソフトウェアの構造的にも発見が難しい不具合が残っている事も想定される．

このため，次のサービス機能追加開発においては，前開発で不具合が発生した機能に関連する試験項目を多くしたり，試験項目の精査を行い，関連機能についての不具合発見数を増やすための施策を行う必要がある．

3.3.2　特定部分選定手法の提案

継続的な追加機能開発が行われるシステムにおいて，一つ前の追加機能開発のソフトウェアの状況を参考とし開発を行う場合，一つ前の開発のどの部分に不具合があったのかを明確にし，今回の追加機能開発でどこをターゲットとして不具合発見数の増強を行う必要があるのかを明確にする必要がある．

前述した通り大規模なソフトウェアの場合，一般的に複数の機能部（以降 FB:Function Block と呼ぶ）で構成され，各 FB は機能単位に作られている．この FB の機能分担範囲は追加機能開発毎に大きく変更されるものでは無いため，不具合発見数の増強を図る単位としては対象とし易く，且つ，前述の特定部分の選定対象としても開発メンバや開発項目よりも変化が少ない．

この事から前述の特定部分の選定を行う場合，後続の追加機能開発単位に変動が少なく，参考とし易い条件を持つ FB を単位とし，FB によって単位開発規模あたりの試験項目数に増減の偏りを作り，システム全体としての試験数を増やさない手法とした．なお，本研究ではソースコードの 1KLine あたりの数値を単位開発規模とした．

システム全体の試験数を増やさない事で，開発期間や開発工数の増加を抑えることができるが，FB 単位の試験項目数に偏りを作ると，FB 単位の内在す

るソフトウェア不具合の偏りも発生してしまう可能性がある．このため，下記の考え方に基づいて試験項目の偏りを付ける事とする．なお，これ以降，直前（ひとつ前）に開発されたファイルを STEP1，現在開発を行っているファイルを STEP2，次の開発によって作成されるファイルを STEP3 と順に呼ぶこととする．

- STEP1 の公衆網での運用期間に不具合が発生した FB は開発時の試験が十分でなかったと判断し，STEP2 開発において単位開発規模あたりの試験項目数を増加させる．（同様に，STEP2 の公衆網での運用期間に不具合が発生した FB も同様に，開発時の試験が十分でなかったと判断し，STEP3 開発において単位開発規模あたりの試験項目数を増加させる．）
- STEP1 の公衆網での運用期間に不具合が発生しなかった，もしくは少ない／軽微である FB は STEP2 開発において単位開発規模あたりの試験項目数を削減する．（同様に，STEP2 の公衆網での運用期間に不具合が発生しなかった，もしくは少ない／軽微である FB は STEP3 開発において単位開発規模あたりの試験項目数を削減する．）

試験項目を増加させる FB が多数となる場合には対象とする FB を選定するため，複数の不具合が発生している FB の場合には不具合の混入過程に何らかの共通的類似性や傾向がある FB と考え，優先的に試験項目増加対象 FB とする．逆に，不具合が 1 件程度の場合には不具合の混入過程に類似性や傾向が見られない FB と考え，試験項目増加対象 FB としない．なお，不具合の類似性や傾向については試験項目増加時の考慮事項とした．以降，試験項目を増加させる FB を「重点監視 FB」，それ以外の FB を「非重点監視 FB」とする．（図 3.3）

3.3.3　試験項目増加／削減比率算出手法の提案

提案手法においては開発工数増を避けるため，まず初めに試験項目密度を変える事が無いように開発規模に応じた全体の試験項目数を定める．試験項目密度は試験項目数を開発規模で正規化した数値であり，単位開発規模あたりに必

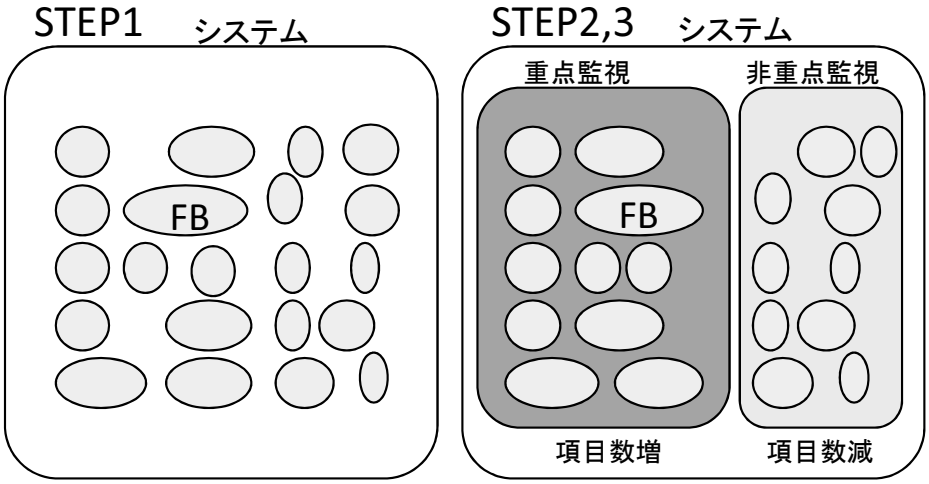


図 3.3　重点監視／非重点監視 FB の分類

要な試験項目数であるため，その増減は開発工数に大きく関連する．

各 FB の増加／削減の比率は，以下に示す通り単位開発規模あたりの試験項目数と開発期間に摘出した不具合数を FB 毎に比較することで求める．

STEP1 の公衆網での運用期間に不具合が発生しており，STEP2 での試験項目を増加対象とする FB の単位開発規模あたりの試験項目数 (STEP2) を N_g，逆に STEP1 の公衆網での運用期間に不具合が発生せず STEP2 での試験項目を減少対象とする FB の単位開発規模あたりの試験項目数 (STEP2) を N_r とする．

また，試験項目を増加対象とする FB の STEP1 の開発時における不具合摘出密度（バグ密度）を D_g，試験項目を減少対象とする FB の STEP1 開発時における不具合摘出密度（バグ密度）を D_r とすると，STEP1 では試験数が十分でなかったと判断し，試験数による不具合摘出数均一化のために STEP2 での試験項目増加対象 FB の試験項目数と減少対象 FB の試験項目数は式（3.1）で求められる値を目標値とする．

$$N_g/N_r = D_r/D_g \tag{3.1}$$

なお，不具合摘出密度 D は，一定の試験数である単位試験項目あたりに摘出した不具合数であり，試験項目数を N_{item}，不具合数を N_{bug} とすると，式（3.2）で表される．

$$D = N_{bug}/N_{item} \tag{3.2}$$

本章に示す提案手法は開発の試験工程に適用し，その結果を評価した上で次の開発の試験工程に再度提案手法を適用するといった開発毎に繰り返されるフローによって内在する不具合の発見数の向上を図る手法であり，STEP1 の結果を基に STEP2 を実施し，STEP2 の結果を基に STEP3 を実施する．

以降では，NGN 開発において具体的な N_r，N_g の値を求め，その値を目標値とし定量的に試験項目に偏りをつけた具体例を示す．また，NGN 開発において提案手法を適用した時の，公衆網適用時に残存する不具合数の結果を示す．

3.4　NGN を対象とした適用と評価

3.4.1　NGN への提案手法の適用

実際に NGN 公衆網に導入された NGN システムソフトウェアの 2 回の追加機能開発ファイル（3.3.2 節の STEP2,3）に対し提案手法を適用し評価を行った．また，提案手法の効果を測るため，提案手法を採用していない開発ファイルである STEP1 ファイルの情報と比較する．

STEP2,3 の追加機能開発を行った時点は商用開始から 3〜4 年後の開発であり，初期開発に比較し公衆網での大規模なサービス中断を生じさせる不具合発生は減っているものの，相当数の不具合が公衆網運用時に発生しており安定運用にあたっては開発時に発見する不具合の数を増やし，公衆網適用時の残存不具合数を少なくする事が不可欠であった．

(1) 重点監視 FB の選定

提案した特定部分選定手法に従い，試験項目を増加対象とする FB である重点監視 FB の選定は，1 つ前の先行して開発されたソフトウェアにおける公衆網適用時の不具合発生状況によって行う．

重点監視 FB は，3.3.2 節の特定部分選定手法に示した通り STEP1 の機能追加開発ソフトウェアを NGN 公衆網に適用した際に複数の不具合が発生した FB を対象とした．なお，本論文内で記載する NGN 公衆網で発生した不具合の数には，NGN 公衆網への提供前に運用者によるサービス動作確認検査や運用手順確認検査によって発見された不具合も含んでいる．また，重点監視 FB

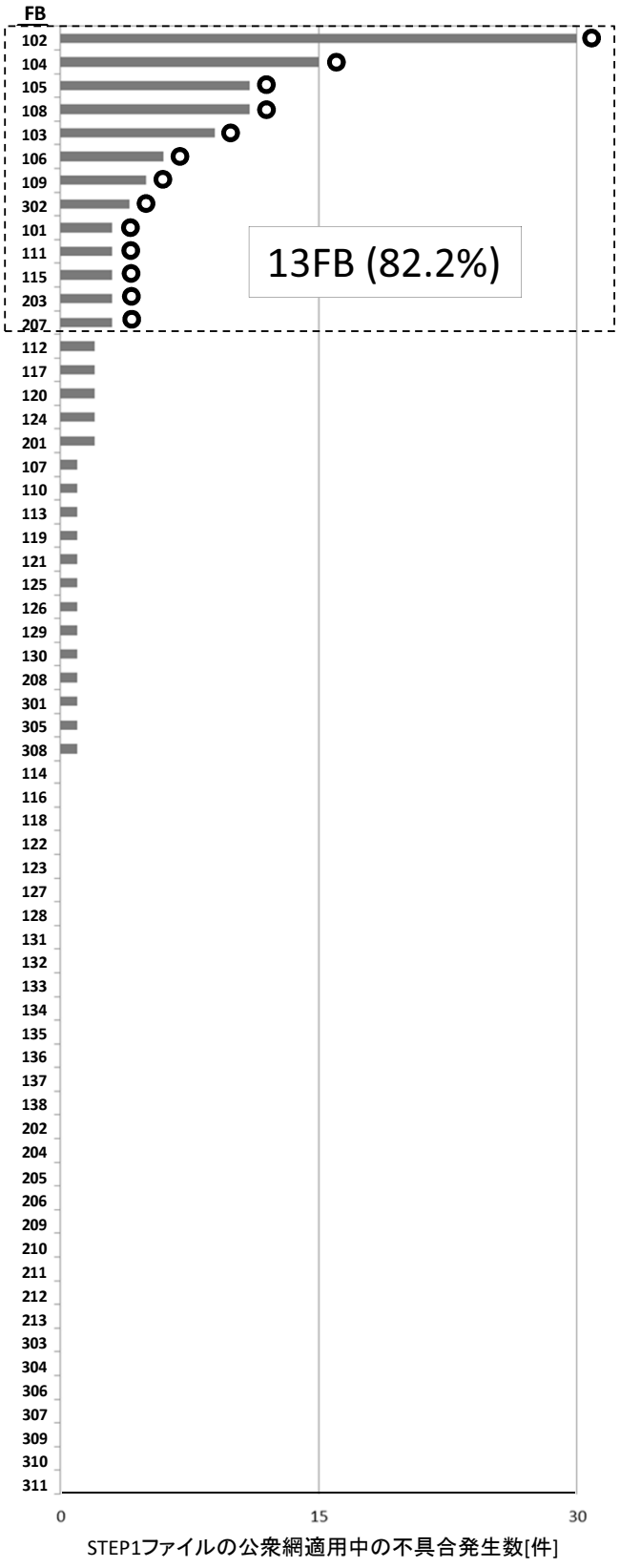


図 3.4　STEP1 の公衆網適用時の発生不具合数

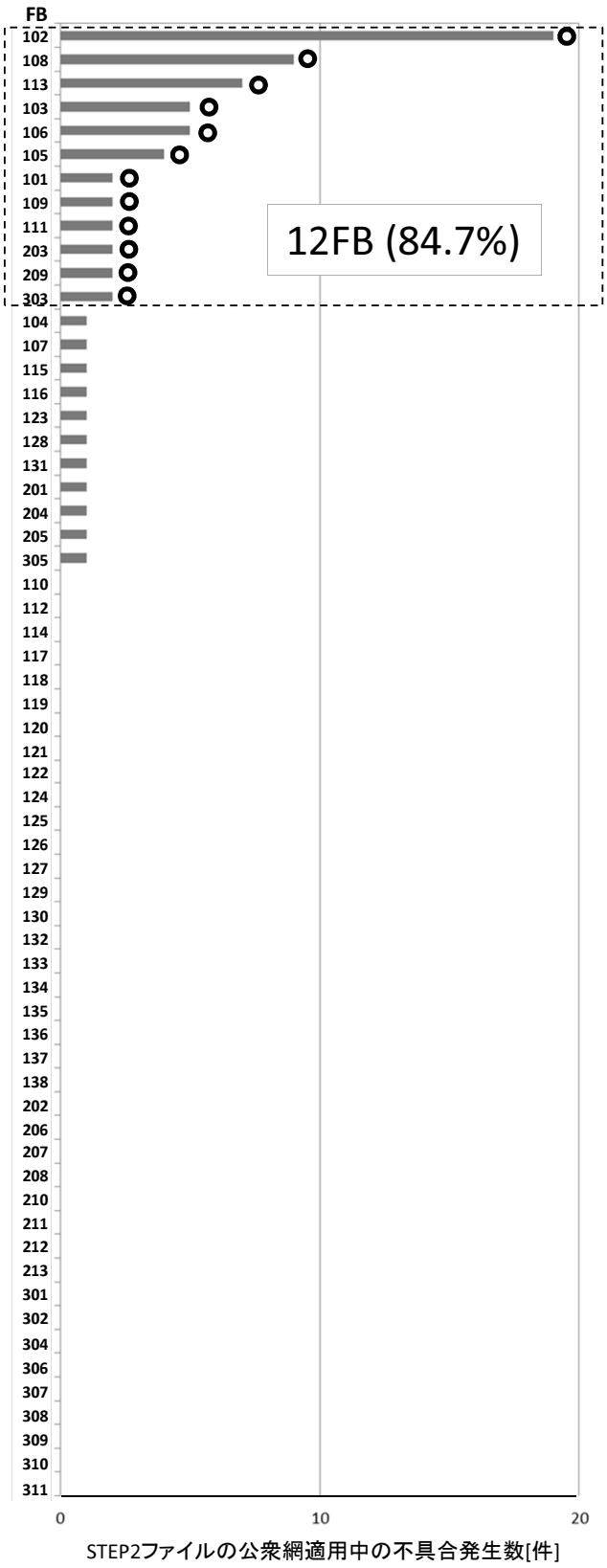


図 3.5　STEP2 の公衆網適用時の発生不具合数

の選定は試験開始前に行うが，その時点では一つ前の追加機能開発ソフトウェアは半年以上の NGN 公衆網での運用が行われている．開発システムは全国で約 200 台，1,500 万以上の加入者を収容し且つ，1 システムあたり最大 30 万程度の加入者を収容している．そのため，呼処理だけでなく不具合情報収集のための保守機能を含め半年の運用においては，システムの保有する機能及び開発した機能を網羅的に使用しており，十分ソフトウェアの状況が測れるものと判断した．

3.3.2 節に示した通り，公衆網適用中に複数の不具合が発生している FB の場合には不具合の混入過程に何らかの類似性や傾向がある FB と考え，2.2 節に示した重要不具合の公衆網適用中発生有無にかかわらず優先的に重点監視 FB とした．STEP2 においては STEP1 の公衆網運用中に 3 件以上の不具合が発生した FB を対象とし，STEP3 においては STEP2 の公衆網運用中に 2 件以上の不具合が発生した FB を対象とした．この事によって，STEP2 及び STEP3 共に公衆網運用中に発生した不具合全体の 8 割強を占める上位の FB を候補とした．またシステム全体の FB 数が 62 あり，STEP2 及び STEP3 それぞれ重点監視 FB とした数は 13FB と 12FB であり，全体の FB 数の 2 割程度に相当する．

提案手法適用の候補とした重点監視 FB について，図 3.4 および図 3.5 のグラフの上に「○」マークを付与した．縦軸は FB の番号（ID）を示し，横軸は NGN 公衆網適用中に発生したの不具合（バグ）数を示している．「○」の付与された上位の FB は STEP1 の公衆網適用中に発生した不具合の 82.2% を占め，同様に STEP2 の公衆網適用中に発生した不具合の 84.7% を占めている．

図 3.4 で「○」を付与した FB とそれ以外の FB について，STEP1 の開発期間中に摘出した単位試験項目あたりの不具合摘出数について「○」の付与されていない FB を 1 とした時の比率を表 3.1 に示す．図 3.4 で「○」を付与した公衆網適用時の不具合が多かった FB は，「○」の付与されていない FB に比べ，STEP1 開発期間における単位試験項目あたりの不具合摘出数が表 3.1 に示す様に 7 割程度と少なく，その結果，NGN 公衆網運用時に不具合を残した可能性が高い事が想定できる．なお，開発期間における各 FB の試験項目数は各 FB の開発機能を検証するために実施した試験項目の総数であるが，複数の FB に跨って作成される実施される試験項目については，開発内容によっていずれかの主たる FB の観点から試験項目が作成されるため，主たる FB の試験実施項目として扱い，3.2.2 節で示した安定化試験で行われる様なシステム全体の

表 3.1　STEP1 開発時の単位試験項目あたりの不具合数

	「○」のFB	「○」以外のFB
STEP1開発時の 不具合密度の比較	0.72	1.0

表 3.2　STEP1 公衆網適用時の重要不具合の比率

	「○」のFB（%）	「○」以外のFB（%）
STEP1 公衆網適用中の 「重要不具合」の比率	89	11

性能試験，長時間安定動作確認試験等は，すべての FB に均等に割り振る扱いとして算出した．

また，表 3.2 には STEP1 が公衆網適用中に発生した重要不具合数の比率を示す．「重要不具合」は 3.2.2 節に示した不具合管理項目のうち，『重要度』が「重要」に分類される，主要な機能の損失に繋がる不具合であり，通信不可やシステムダウンにつながる不具合等が該当する．図 3.4 で「○」を付与した公衆網適用時に不具合が多かった FB は，「○」の無い FB に比べ，重要不具合の発生比率が高い．「○」を付与した FB が占める公衆網適用時の不具合は全体の 84.7% であるが，それ以上に重要不具合が発生する比率が高くなっている．

これらの事から，「○」を付与した FB を重点監視 FB として提案手法を適用する事で，公衆網適用時の不具合発生の数を減らすと共に，サービスの継続性を阻害する重要不具合の数を減らし，システムの安定運用の継続性を高める事とした．

(2) 試験項目増加／削減比率の算出

前述した通り，NGN 公衆網適用時に不具合が多数発生した FB は開発期間中における単位試験項目あたりの不具合摘出数が少ない．このため，3.3.3 節の試験項目比率算出手順に示した様に，開発における重点監視／非重点監視毎の試験項目選定数の比率は，一つ前の STEP ファイル開発中に摘出された不具

表 3.3　STEP1 開発における試験項目数と不具合摘出数

	試験項目数		不具合数		不具合摘出密度	
	○	その他	○	その他	○	その他
STEP1	1.00	0.33	1.00	0.45	1.00	1.38

表 3.4　STEP2 開発における実施試験項目数と開発規模

	試験項目数		開発規模		開発規模あたりの重点／非重点の試験項目比率
	重点	非重点	重点	非重点	
STEP2	1.00	0.62	1.00	0.81	1.30

合数を重点監視／非重点監視に分類し，単位試験項目あたりの不具合数で比較した上で，摘出しきれていないと思われる項目数分を加味した比率にする．STEP2 の場合は重点監視／非重点監視毎の項目選定数の偏りは，STEP1 開発時の不具合摘出割合を参考とする．

表 3.3 には STEP1 での開発時の不具合摘出数を示す．表 3.3 中の「○」「その他」はそれぞれ図 3.4 で「○」を付与した FB とそれ以外を示しており，「○」を付与した FB の数値を 1 として正規化した．

STEP1 における試験項目数の割合は「○」を付与した FB に対しその他の FB は 0.33 であり，不具合数については「○」の FB に対しその他の FB は 0.45 であった．このことから式（3.1）の D_r ／ D_g は 1.38 となり，STEP2 では，公衆網適用時に不具合が多く発生した「○」の FB の試験数を全体平均より増やす目標値を 1.38 倍とした．

実際の開発における試験項目作成においては，複数の FB に跨る試験項目やシステム全体で共通的に実施される試験項目作成が必要であるため，FB 単位の試験項目作成だけでなくシステム横断的に試験項目作成を実施する．本提案においては，FB 単位の試験項目作成において STEP2,3 の試験項目策定時に「○」を付与した重点監視と判断した FB に関しては実施する試験項目も多くなるため，多くの試験項目作成を目的として開発メンバによる試験項目作成工数を増やすが，システム開発全体の工数を増やさないために，逆に重点監視 FB 以外は相対的に試験項目作成工数が減る事となる．さらに試験項目作成後，

表 3.5　STEP2 開発における試験項目数と不具合摘出数

	試験項目数		不具合数		不具合摘出密度	
	○	その他	○	その他	○	その他
STEP2	1.00	0.73	1.00	0.80	1.00	1.10

表 3.6　STEP3 開発における実施試験項目数と開発規模

	試験項目数		開発規模		開発規模あたりの重点／非重点の試験項目比率
	重点	非重点	重点	非重点	
STEP3	1.00	1.54	1.00	2.13	1.38

FB 単位に実施する試験項目を選定する際に，重点監視 FB と非重点監視 FB の実施する試験項目数に偏りをつけることで，システム開発全体の試験実施工数を増やさない様にするため，開発規模で正規化した試験数は重点監視 FB の試験項目は非重点監視 FB より多くなる．

表 3.4 には STEP2 で実際に行った単位開発規模あたりの試験項目数の重点監視 FB と非重点監視 FB の比率を示す．STEP2 においては 1.30 倍と目標値に近い試験項目数で試験を行った．

同様に STEP2 の開発時の不具合摘出数（表 3.5）を基に，STEP3 での目標とする試験項目数を表 3.6 に示す．実際の STEP3 開発においては規模あたりの重点／非重点監視 FB の試験項目比率が目標よりも多くなった．

最終的に実施した，STEP1 と比較した STEP2,3 の試験項目比率，及び試験項目数，試験項目密度を表 3.7 に示す．STEP1 を 1 とした時，STEP2,3 は STEP1 に比べ開発規模は半分程度であるため，試験項目数も少ないが，単位開発規模あたりの試験項目数（試験項目密度）には大きな変化が無い．重点監視 FB の試験項目は増加させたが，その分，非重点監視 FB の試験項目を減らしたため，全体の試験項目密度は STEP2,3 共に STEP1 に比べて 4% 程度の増加であり，この事は，STEP1 と比べ STEP2,3 においても試験工程における開発工数増がほぼ無かったことを示している．

次節では表 3.7 に示した試験項目数で開発した STEP2,3 の不具合発見状況を評価する．

表 3.7　各開発の試験項目密度比較

	開発規模	試験項目数	試験項目密度
STEP1	1.00	1.00	1.00
STEP2	0.55	0.57	1.04
STEP3	0.36	0.38	1.04

3.4.2　提案手法適用後の評価

NGN 公衆網での実績がある 3 回の追加機能開発ファイルのうち，提案手法を適用していない先行する一つ目の開発（STEP1) と，提案手法を取り入れた後続する二つ目の追加機能開発（STEP2,3）において，ファイルの開発期間，及び，NGN 公衆網で運用されていた期間全体での不具合発見数の評価比較を行った．STEP1,2,3 の開発規模は母体の 0.5 割〜1 割程度であり，開発内容はシステムの保有する呼処理機能や保守機能と全般的に開発が存在しているため比較対象と考えられるが，STEP1,2,3 は開発内容が異なるため，比較にあたっては正確な評価が困難である．このため本提案の効果を測るにあたり，各開発の FB 単位の開発規模の STEP 間類似傾向を調べ，各開発（STEP1,2,3）を比較する場合に考慮すべき事項の有無を把握する事とした．

本提案は FB 単位に開発規模に応じた試験項目の増減を行う手法であるため，各開発における各 FB 開発規模を順に並べ各開発規模の 50%，70%，90% で比較した時，50%，70%，90% 毎に占める FB 数のうち STEP1,2 間，及び 2,3 間で合致した FB 数の割合を合致率と定義し比較した．なお，比較する各 FB 開発規模はソフトウェア開発における修正／新規／削除 Line 数によって求められる値であり，開発されるプログラムのアルゴリズムの類似性等は考慮していない．STEP1 と STEP2 間の FB 単位での全体の 50%〜90% までの開発規模を占める FB の合致率，及び STEP2 と STEP3 間の合致率を表 3.8 に示す，開発規模で上位を占める FB の傾向はおよそ 7 割程度が一致しており，比較対象とした場合に評価結果について傾向が測れるものと考えた．

表 3.8　各開発の FB 単位の規模による合致率

	STEP1と2の合致率 (%)	STEP1と2の合致率 (%)
開発規模の50%	75.0	80.0
開発規模の80%	77.8	69.2
開発規模の90%	66.7	73.7

(1) 評価を行う指標値

提案手法を評価するため指標値として，“開発期間内で摘出した不具合”と“NGN 公衆網運用中に発生した不具合”の双方の全ての不具合に関する FB 毎の不具合数や対処までの期間，重要度，摘出工程を BTS に蓄積されたデータからとりまとめた．下記に，評価に利用した項目を示す．

【ソフトウェア全体，及び FB 単位】

- Line 数
- 不具合数
- 不具合（バグ）対処期間
- 発見工程
- 重要度

また，以下のメトリクスについてはソースコード静的解析ツールである Source Monitor[38] を利用し数値化を行った．

【ソフトウェア全体，及び FB 単位】

- 平均複雑度
- 平均深度

以降はこれらの情報に基づき，システム全体，重点監視／非重点監視 FB 単位に分類し提案手法の有効性の有無を評価した結果を示す．

表 3.9　各 STEP における規模と不具合数比較

	開発規模	開発時 発見不具合	公衆網適用中 発生不具合
STEP1	1.00	1.00	1.00
STEP2	0.55	1.10	0.96
STEP3	0.36	1.47	0.87

(2) 開発規模と不具合摘出数

本章で提案する手法は開発期間における発見不具合数を増やすことで，公衆網適用時の不具合発生を減らす事を目的としている．提案手法はシステム内の個々の FB に対して行われるものであるが，本項ではシステム全体として提案手法の効果の有無を把握するため，提案手法を適用していない STEP1 と提案手法を適用した STEP2,3 における，開発期間における発見不具合数と公衆網適用時の発生不具合数を比較した．表 3.9 に開発における不具合摘出数，NGN 公衆網適用中の不具合発生数について STEP1 を 1 とした時の STEP2,3 との比較を行った結果を示す．なお，STEP2,3 の不具合数については STEP1 を基準とし，開発規模で正規化している．

表 3.7 に示した通り，STEP1,2,3 はそれぞれで開発規模が異なるが，開発規模に応じて試験項目数の目標を立てるため，開発規模の比率と試験項目数の比率はほぼ同じであり，規模で正規化した試験項目密度は同等であった．

表 3.9 に示すように，提案方式を取り入れた STEP2,3 については，開発中に摘出する不具合数（開発規模で正規化）は増加傾向にあり，NGN 公衆網適用中の不具合発生数（開発規模で正規化）は減少傾向にあった．開発中に摘出する不具合数が増加した事を考えると，開発を重ねることによるソフトウェア品質の向上や，開発規模の減少に伴う公衆網適用中の不具合発生数の減少だけでは無く，提案手法の効果があったものと考えられる．また，開発時発見不具合及び公衆網適用時発生不具合の重要度について図 3.6 と図 3.7 に示す．これらの図も同様に各開発の規模で正規化している．STEP1 に比較し開発時発見不具合及び公衆網適用時発生不具合に占める重要不具合の比率も STEP2,3 では

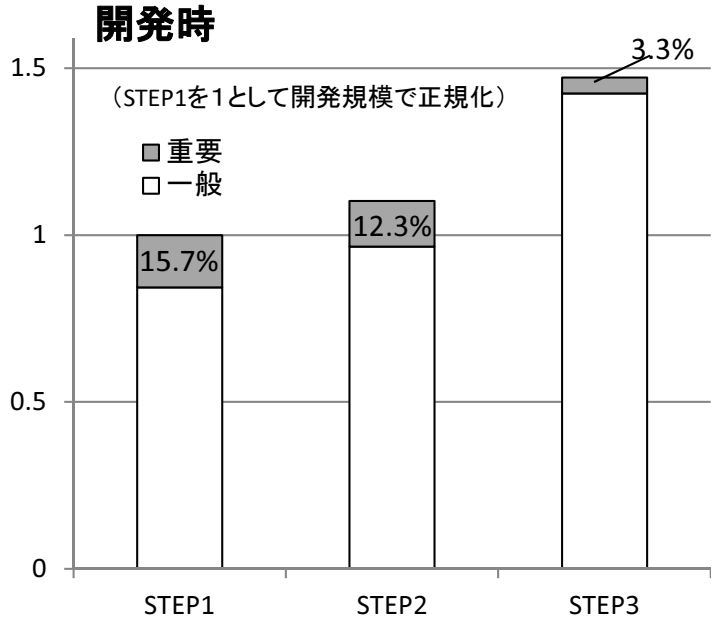


図 3.6　各 STEP 開発における不具合数と重要度

表 3.10　各 STEP 開発及び公衆網適用時の重要不具合の比率

	開発時（%）	公衆網適用時（%）
STEP1	15.7	51.4
STEP2	12.3	20.3
STEP3	3.3	4.4

下がっている．

図 3.6 及び図 3.7 の重要不具合の占める比率について表 3.10 にまとめる．開発中に発見する重要不具合の減少が STEP1 と STEP2，STEP2 と STEP3 の比較でそれぞれ 3.4 ポイントと 9 ポイント，公衆網適用時の重要不具合減少が STEP1 と STEP2，STEP2 と STEP3 の比較でそれぞれ 31.4 ポイントと 15.9 ポイントであった．また，STEP1,2,3 において発見した重要不具合全体に占める公衆網適用時に発生した重要不具合の割合はそれぞれ 32.4%，17.2%，10.5% と減少傾向にあった．開発を重ねる毎に重要不具合の比率は減るものの，公衆網適用時の重要不具合減少比率幅が多く，開発中により多くの不具合を発見する提案手法によって，公衆網適用時の発生を事前に防ぎ，同時に公衆網適用時の重要不具合発生抑止への効果も想定される．

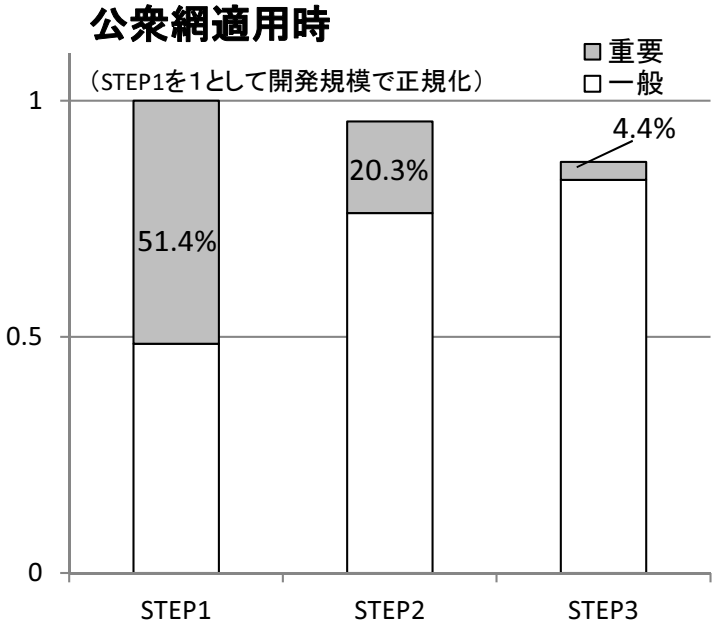


図 3.7　各 STEP の公衆網適用時の不具合数と重要度

(3) 重点監視／非重点監視 FB 別評価

ここでは提案手法である重点監視 FB を設定した STEP2 および STEP3 の開発工程内開発における不具合摘出数効果について，重点監視 FB と非重点監視 FB での比較を行う．重点監視 FB ／非重点監視 FB 毎に，開発工程で発見された不具合を分類することで，重点監視 FB を設けることの効果について評価を行った．

重点監視，非重点監視 FB はそれぞれ FB 毎の試験項目数が異なるため，STEP2,3 において，それぞれの開発規模あたり及び単位試験項目数あたりの不具合摘出数で正規化し評価を行った．正規化した重点監視，非重点監視 FB の不具合摘出率を表 3.11 に示す．「重点／非重点比率」の列はそれぞれ，単位開発規模と試験項目数あたりの非重点監視 FB の不具合数に対する重点監視 FB の不具合数の比率である．

表 3.11 に示す様に，開発規模で正規化した不具合数では重点監視 FB は非重点監視 FB と比較し STEP2,3 ともに 1.2 倍程度の不具合摘出数であった．

表 3.11　各 STEP 開発における不具合数比率

	不具合数（開発規模で正規化）		
	重点	非重点	重点／非重点比率
STEP2	1.00	0.82	1.21
STEP3	1.00	0.82	1.22

(4) 重点監視／非重点監視 FB 単位評価

ここでは更に STEP1,2,3 ファイルにおける個々の FB 単位に開発時に摘出された不具合と NGN 公衆網運用中に表面化した不具合のそれぞれの傾向について評価を進めた．

各 FB 毎の不具合発生の期待値を評価するに当たっては，不具合発生の期待値を導き出す係数が必要となる．

一般的には，開発規模とその開発における不具合数は比例関係にあり，開発の基準となる『開発規模 (V_d)』が求められれば，摘出される不具合数『N』が想定できる．開発の基準となる開発規模の尺度をソースコードのライン数と係数で表す SLOC/LOC 手法においてはソースコードに対する新規追加／修正／削除規模，及び，母体のソースコード規模で構成される式 (3.3) で表される．

$$V_d = (\alpha_1 V_r + \alpha_2 V_n + \alpha_3 V_e) + \gamma\ V_m \tag{3.3}$$

V_r:ソースコードの修正規模

V_n:ソースコードの新規規模

V_e:ソースコードの削除規模

V_m:ソースコードの母体規模

(α_1，α_2，α_3，γ は実績から求められる係数）

ソースコードに対する変更の全体量は追加／修正／削除をまとめたソースコード変更規模（V_c）で表すことができ，前述した開発規模 (V_d) に比例して存在する不具合数 (N) をそれぞれ式 (3.4)，(3.5) とすると，

$$\alpha\ V_c = \alpha_1 V_r + \alpha_2 V_n + \alpha_3 V_e \tag{3.4}$$

$$V_d = \beta\ N \tag{3.5}$$

（α，β は実績から求められる係数）

式 (3.3) に式 (3.4)，(3.5) を代入することにより，期待される不具合数 N は式 (3.6) で表される．

$$\beta\ N = \alpha\ V_c + \gamma\ V_m \tag{3.6}$$

$A = \alpha / \gamma$，$B = \beta / \gamma$ と置くと，ソースコードの変更規模と不具合数との関係は式 (3.7) に示すように，A および B を係数とする式で表される．

$$BN = AV_c + V_m \tag{3.7}$$

実績から求められる A および B の平均的な値はこれまでの数回の機能追加開発のソースコード変更規模と不具合摘出数の結果をもとに求めることが可能である．

なお，i 番目の追加開発ファイル STEPi と k 番目の追加開発ファイル STEPk $(i > k)$ に関する不具合数 N や変更規模 V_c に関する A, B の関係式は，i,k 番目の値を代入した式 (3.7) から求められ，式 (3.8)，(3.9) で表される．

$$A = \frac{N_i V_{mk} - N_k V_{mi}}{N_k V_{ci} - N_i V_{ck}} \tag{3.8}$$

$$B = \frac{V_{ci} V_{mk} - V_{ck} V_{mi}}{N_k V_{ci} - N_i V_{ck}} \tag{3.9}$$

提案方式を採用して重点監視 FB を設けた STEP2,3 開発のデータから A，B の値を導出し，A，B の値を基に STEP1 での開発工程内，NGN 公衆網運用中での発生が期待される不具合数を算出した．

期待すべき不具合数と STEP1 における実際の各 FB 単位の不具合摘出との乖離数を比較すると共に，重点監視 FB と非重点監視 FB で比較を行った．

STEP2,3 にて両方とも重点監視となった FB，及びそれ以外の FB と分類し，STEP2,3 の開発規模，不具合数のそれぞれの合計値より，式 (3.8)，(3.9) の A,B

表 3.12　STEP2,3 によるバグ期待値と STEP1 バグ数との乖離数

	期待値との乖離数 [N_{dif}]	
		規模で正規化
STEP2,3両方が重点監視FB	82	6.21
上記以外	13	0.74

の値を求め，STEP1 の開発規模を当てはめる事で，STEP1 で期待される不具合数を算出した．

実際の STEP1 での摘出した不具合数 (N_1) と STEP2,3 の値を式 (3.10) に代入した結果から導き出される不具合期待値 (N_{ex}) からの乖離数 (N_{dif}) は式 (3.10) で表され，その結果を表 3.12 に示す．

$$N_{dif} = N_{ex} - N_1 \tag{3.10}$$

表 3.12 に示す様に，STEP2,3 の両方で重点監視とした FB については不具合をより多く摘出できている結果が反映されているため期待値が高く，STEP1 で実際に摘出した不具合との乖離数が 82 件と大きくなっている．逆に，STEP2,3 で重点監視 FB としなかった FB については，STEP1 での対応する FB と比べ不具合数の乖離が 13 件と小さい．該当する FB の母体規模で正規化した場合でも，それぞれ 6.21 と 0.74 となり，およそ 8 倍の差があった．

個々の FB 単位の評価結果を図 3.8 に示す．各 FB の規模を横軸，各 FB 毎の乖離数を縦軸にとり，STEP2,3 の両方で重点監視となった FB は「●」，それ以外の FB を「×」で示す．グラフの縦軸は式 (3.10) に示す乖離数 (N_{dif}) を表しており，STEP2,3 から導き出される期待値から，STEP1 での実際の不具合数を引いた数字を示しているので，正の方向に数が大きければ期待値の方が大きく，STEP2,3 においてより多くの不具合発見ができていると考えられる．

バグ期待値から乖離した値の正負が各 FB のソフト構造の違いであるか否かを把握するため，各 FB 毎にソフト構造を示す目安となるメトリクスデータを計測した．メトリクス測定ツールとしては，前述した通り，Source Monitor を利用し，その結果を表 3.13 と表 3.14 に示す．

表 3.13 は STEP2,3 で重点監視 FB か否か，表 3.14 は図 3.8 で乖離数 N_{dif}

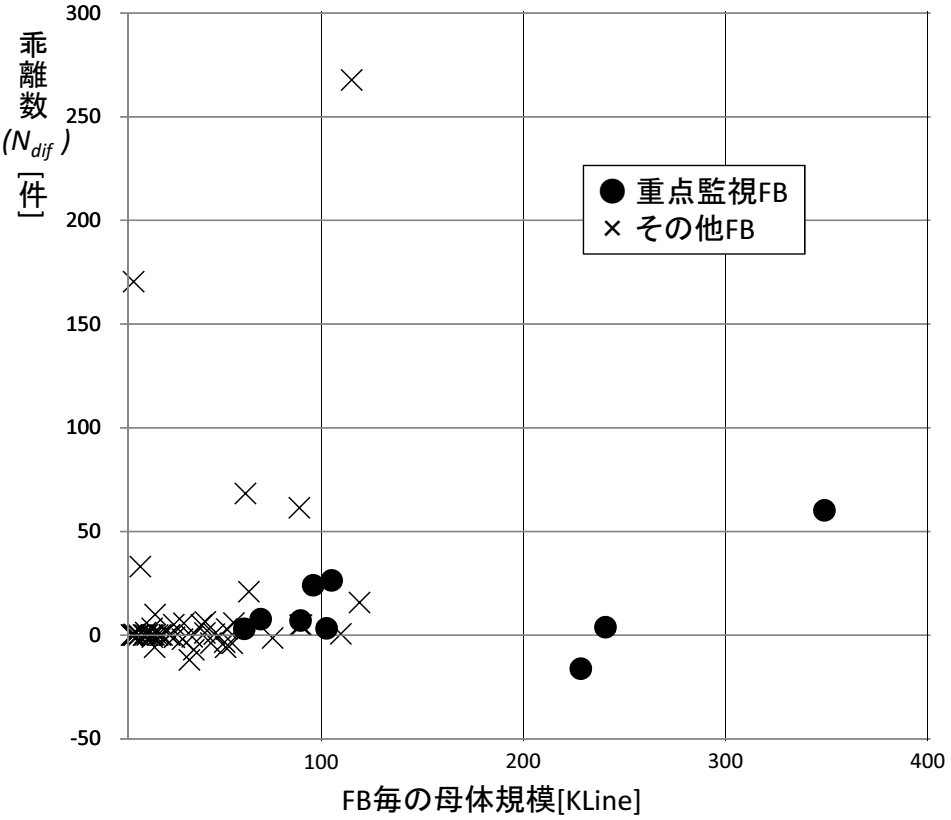


図 3.8　FB 毎の STEP2,3 によるバグ期待値と STEP1 バグ数との乖離数

表 3.13　重点／非重点監視 FB 毎のメトリクス比較

	平均深度	平均複雑度
重点監視	1.5	7.1
非重点監視	1.6	9.2
全体	1.6	8.9

が正となったか否かによって分類し，FB 単位の『平均深度』や『平均複雑度』を測定した．なおメトリクスである『平均深度』や『平均複雑度』の定義については表 3.15 に示す．

表 3.13，表 3.14 に示す様に重点／非重点，及び乖離数 N_{dif} の正負の FB によって分類し値を取得したが，平均深度，平均複雑度に大きな差は無かった．

表 3.14　STEP1 バグ数と期待値の差が正／負の FB のメトリクス比較

	平均深度	平均複雑度
乖離数N_{dif}が正	1.5	8.1
乖離数N_{dif}が負	2.4	13.6
全体	1.6	8.9

表 3.15　各メトリック値の意味

平均深度	サブルーチンコール、関数呼び出し等入れ子構造によるネストの深さの平均値
平均複雑度	サイクロマチック複雑度（完全分岐網羅[C1]）の平均値

(5) 不具合発見と解決までの時間の評価

開発及び NGN 公衆網でのサービス中を含む全行程で不具合発見から不具合解決までの時間をそれぞれの STEP で比較する事で，提案方式の効果の評価を行った．不具合発見から解決までの時間は，不具合内容の解析から改修までの時間であり，開発工数やシステムの運用工数に大きく関係する．

表 3.16 に STEP1 での開発工程及び公衆網運用期間での不具合発生から解決までの時間の平均値を示す．開発期間の結合試験時に発生した不具合の解析から対処までにかかる時間を 1 とした時の比率により表す．システム全体として結合試験及び安定化試験と工程が後になるに従い，不具合対処までの時間が長延化する．特に公衆網でのサービス中に発生する不具合対処にかかる時間が長延化するのは，開発期間に発見できなかった発生頻度が稀な不具合や複雑な条件の重ね合わせにより発生する不具合のためと考えられる．また，運用中であるため，不具合内容を解析する情報が取りにくい事にも起因している．

STEP1,2,3 での公衆網運用時に発生する不具合の総対処時間 (規模で正規化) について図 3.9 に，不具合一件あたりの平均解決時間の比 (STEP1 を 1 とする) を表 3.17 に示す．提案方式を適用した STEP2,3 においては図 3.7 に示す様に公衆網運用時に発生する不具合は減少傾向にあったが，図 3.9 の通り公衆網運用時に発生した不具合の総対処時間は STEP2 においては不具合数の減少傾向

表 3.16　STEP1 での不具合解決までの平均時間

	開発期間		公衆網適用時
	結合試験	安定化試験	
平均解決時間全体	1.00	2.19	5.12
一般	0.97	1.99	5.11
重要	1.51	2.34	5.12

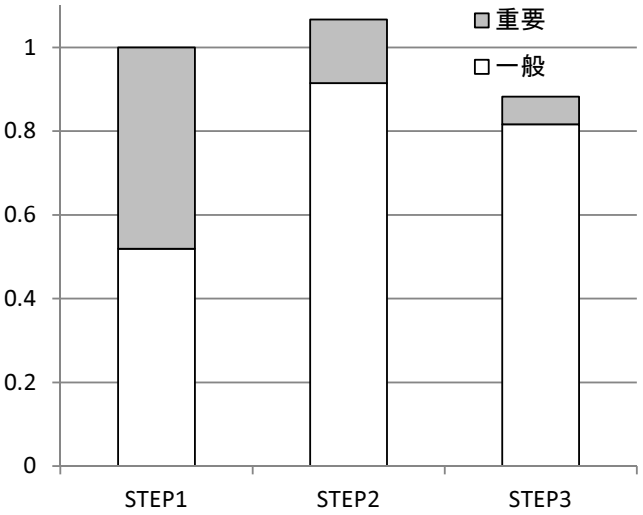


図 3.9　各 STEP での不具合解決までの総対処時間

表 3.17　各 STEP での公衆網運用時の不具合解決までの平均時間

	公衆網運用時発生不具合の 平均解決時間
STEP1	1.00
STEP2	1.12
STEP3	1.01

程の効果を見る事ができなかった．これは，表 3.17 に示す様に公衆網運用時に発生する 1 つの不具合あたりの対処時間が STEP2,3 では長延化したことによる．

3.5　考察

公衆網運用中に発生する不具合対処毎に増加する稼働削減への効果も期待できる不具合の「数」に着目し，一つ前に先行して開発されているソフトウェアの開発時，及びそのソフトウェアが実際に運用される中での不具合発生数を把握し，FB 毎の試験項目の「数」に偏りを設ける事で全体の開発期間の延伸や開発工数の増加を抑えつつ開発期間中の不具合発見数を増加させる手法を提案した．

3.4.1 項に示した通り，STEP1 の公衆網運用中の不具合が多い FB については表 3.1 に記述した通り，他の FB に比較し開発中に十分不具合を摘出しきれていない事が想定された．このため，STEP1 の開発期間中の単位試験項目あたりの不具合摘出数を出し，その数値を FB 単位に分類して計算した値を目標として STEP2 の試験項目の作成を行った．STEP3 についても同様である．

また，試験項目数については表 3.3 及び表 3.5 に示した通り，非重点監視 FB に対し重点監視 FB の比率は STEP2,3 それぞれ 1.38 と 1.10 を目標とすることとしたが，実際の開発においては STEP2 で 1.30，STEP3 で 1.38 となった．これは，STEP3 において非重点監視 FB の部分の開発規模が重点監視 FB の部分の開発規模に比較し倍以上大きく，各重点監視 FB 毎に増加させた項目数の積み上げによって全体的な規模比率の実施数を目標を上回るものになってしまったためであった．ただし，規模で正規化した STEP1,2,3 の試験項目密度に大きな差は無く，試験項目増加に伴う開発工数の増加にはなっていない．

3.4.2 項の表 3.9 に示した通り，提案手法を適用した STEP2,3 と STEP1 の開発工程において発見された不具合数，および，開発工程内で不具合を発見できず公衆網適用中に発見された不具合数を規模で正規化し比較すると，開発期間内は発見不具合数が増加し，公衆網適用中の不具合発生の比率は減少している．開発中に摘出する不具合数が増加した事を考慮すると，開発を重ねることによるソフトウェア品質の向上や，開発規模の減少に伴う公衆網適用中の不具合発生数の減少だけでは無く，提案した FB 毎の試験項目密度に差を設ける事でより開発期間内に不具合を多く発見する事に効果があったことが覗える．

また，公衆網適用時の重要な不具合の発生についても，表 3.10 にまとめた通り減少傾向にあるが，開発中を含めたシステム全体として重要不具合も減少

しており，STEP 開発を重ねる事によるシステムの全体的な重要不具合の減少も想定される．システム全体の傾向としては，開発中に発見する重要不具合の減少が STEP1 と 2，STEP2 と 3 の比較でそれぞれ 3.4 ポイント，9 ポイント，公衆網適用時の重要不具合減少が STEP1 と 2，STEP2 と 3 の比較でそれぞれ 31.4 ポイント，15.9 ポイントと公衆網適用時の重要不具合減少比率幅が大きい．また，STEP1,2,3 において発見した重要不具合全体に占める公衆網適用時に発生した重要不具合の割合はそれぞれ 32.4%，17.2%，10.5% と減少傾向にあった．開発中に重要不具合を発見する数は減っているものの，開発中により多くの不具合を発見する提案手法が公衆網適用時の重要な不具合発生抑止の観点からも，開発を重ねる事による重要不具合の減少以外の効果としてあったものと考えられる．

ただし，FB 単位に見た場合，STEP1 の公衆網適用中に発見された重要不具合があった FB のうち STEP2 で重点監視 FB にならず，STEP2 の公衆網適用中に重要不具合が発生した FB が 1 件あった．本提案では公衆網適用中に発生した不具合の重要不具合の有無によって重点監視 FB を決めるのでは無く，発生不具合数によって重点監視 FB を選定しており，重要不具合の有無を考慮した場合の影響についても，開発を通して明らかにしていく必要があると考える．

表 3.3 及び表 3.5 に示した通り，実際の STEP2,3 の試験においては非重点監視 FB に比べ重点監視 FB の項目数を 1.30 倍と 1.38 倍にしたが，3.4.2 項の表 3.11 に示した結果から，STEP2,3 における開発期間内発見不具合は非重点監視 FB に比べ重点監視 FB は 1.21 倍と 1.22 倍となり，試験項目の増分と同等レベルにはならなかった．最適な項目数の偏りの大きさは引き続き，比率を変えた開発を通して明らかにしていく必要がある．

3.4.2 項では FB 単位の不具合発見について評価を行った．表 3.12 に示す様に全体的な傾向として重点監視 FB については多くの不具合発見に繋がっており，非重点監視 FB についても多くの FB については不具合発見が少なくなる事は無かった．しかしながら，図 3.8 の一部に見られる様に，STEP2,3 の方が不具合発見が少ない FB もあった．図 3.8 の負の範囲を拡大し図 3.10 に示す．図 3.10 にあるように，不具合発見数が期待より少ない FB は相対的に試験項目を少なくした非重点監視 FB に多い．表 3.13 で，負領域にある FB とそうでない FB について評価した所，平均深度や平均複雑度に若干の差はあるものの，

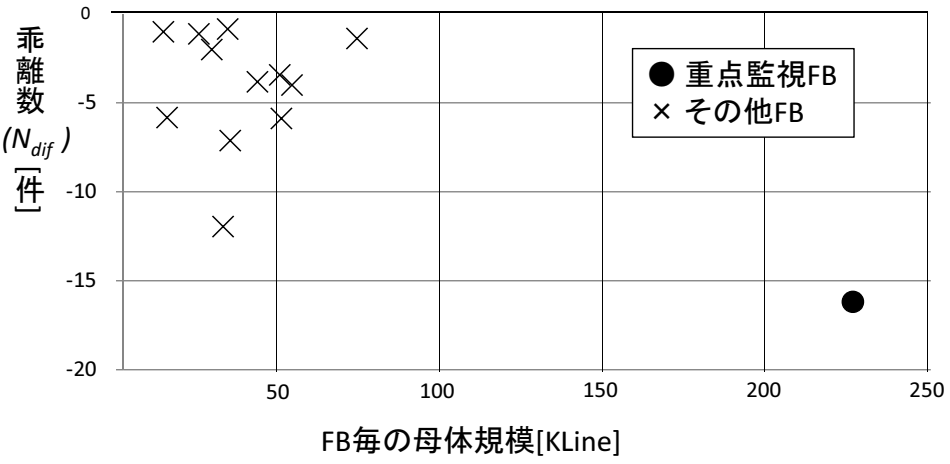


図 3.10　FB 毎の STEP2,3 によるバグ期待値と STEP1 バグ数との乖離（負領域）

構造を変更しなくてはならない程の大きな違いがなかった．FB 単位に別の評価軸での調査が必要と考えられるため，引き続きソフトウェアの記述の詳細な調査を進めたい．

公衆網適用時に発生した不具合の解決までの時間については，3.4.2 項の表 3.9 に示す様に大きな変化は見られなかった．公衆網適用中に発生した不具合の解析データ収集作業時間の変化が無い状況においては，公衆網運用時まで残存する不具合の数を少なくすることがシステムの安定運用に寄与すると考えられる．

3.6　まとめ

高信頼，高品質を求められる大規模ソフトウェアの開発において，安定した公衆網運用の確保を行いつつ，コスト増を抑制する開発を実現するため，適用施策の定量的評価が必要とされている．本章では，公衆網運用中に発生する不具合対処毎に増加する稼働削減への効果も期待できる不具合の「数」に着目し，一つ前に先行して開発されているソフトウェアの開発時，及びそのソフトウェアが実際に運用される中での不具合発生数を把握し，FB 毎の試験項目の「数」に偏りを設ける事で全体の開発期間の延伸や開発工数の増加を抑えつつ開発期間中の不具合発見数を増加させる手法を提案した．また，NGN を例に手法の評価を定量的に行った．評価にあたっては，3 つの機能追加のための開発ファイルにおける公衆網適用時の不具合発生数と発生不具合の重要度につい

て比較評価した．

開発中および公衆網適用時の結果から，開発手法の一つとして，重点監視 FB または非重点監視 FB を設け試験項目数に偏りをつける事で開発時の不具合をより多く摘出し，公衆網適用時に残存する不具合を減少させる事ができる可能性を見いだせた．また，FB 毎の試験項目数比率の目標値の求め方を提案し，その目標値と実際に不具合が摘出できた結果の乖離を示すことで，今後の方針を示す事が出来た．

複数のシステム開発時の開発規模や不具合発生数を FB 単位に詳細に管理する事によって期待される不具合数を導き出し FB 単位の詳細な評価が可能であったことを示すと共に，不具合を詳細に管理する事の重要性を示す事が出来た．

とりまとめた結果からは提案手法による重要不具合の削減や不具合改修までの期間の短縮に大きな効果は見る事が出来なかったが，公衆網適用時の不具合改修において必要な期間を明確にすることで，安定した公衆網運用の確保に必要な要素を明らかにする事ができた．

第 4 章

高品質なソフトウェア開発のための試験項目作成手法と評価

本章では，開発のための要求仕様書と試験項目を教師データとし，機械学習により自動的に試験項目を抽出する手法について提案する．

高いソフトウェア品質を継続的に維持するためには，試験工程において確実に不具合を発見する手法が重要であり，適切な試験項目を作成できるかどうかは作業者のスキルやノウハウに大きく依存している．

このスキルやノウハウの偏りによる不均質な試験項目作成を避けるため，機械学習により試験項目を自動的に作成する手法を提案した．また，本アプローチに関しキャリアネットワークである NGN のシステム開発に適用した結果を報告する．

4.1　はじめに

世界的な IoT/M2M サービスの拡大に伴うブロードバンド通信の拡大は，多くの通信キャリアに対し激しいサービス低廉化競争を余儀なくさせている [39, 40]．従来型の電話サービスをインターネットサービス網と重畳させるキャリアネットワークにおいても同様の低廉化が必要であるが，電話サービスに関しては信頼性や安全性の保証をしなくてはならないため，開発コストや維持コストが高止まる傾向にある．

通信インフラのバックボーンを支える通信設備は，ハードウェアのコモディティ化や仮想化等により，劇的なローコスト化が進んだが，NFV (Network

Function Virtualization) や SDN(Software Defined Network) を利用したソフトウェアは品質面や安全性を保証しなくてはならないため，ソフトウェア開発プロセスにおける各種品質向上施策が開発コストの高止まりを継続させる原因となっている．これらの課題に対応するためには，開発工程の自動化推進が必須となっており，開発の各工程で導入される施策を定式化し，且つ自動化するための検討が必要となっている．

ソフトウェア開発におけるプロセスの定式化，自動化検討においては開発の一連の工程に渡り進める必要があるが，この解決に向け，ソフトウェア開発プロセス自動化の 1 アプローチとして，本章では，試験項目作成作業者のスキルやノウハウに依存しない均質な試験項目を，基本設計工程の生産物である，「要求仕様書，BI/BD 書」から自動的に，安定化試験において検証すべき試験項目を作成する手法について提案し，その精度向上を目指した取り組みを行う．

固定ブロードバンド通信を支える NGN(Next GenerationNetwork) [41]. は，統合された通信基盤でインターネットサービスと従来型の電話サービス（PSTN）を提供しているが，従来型の電話による通信サービスに関しては信頼性や安全性の保証，社会的な依存性を求められ，結果的に NGN システムの開発は高信頼，高品質のため，開発時に多くの品質向上プロセスを経る事となる．

高いソフトウェア品質を継続的に維持し続けるためには，試験工程において確実に不具合を発見する手法の性能にかかっている．不具合を発見するためには仕様を正確に理解し，検証すべき試験項目を網羅的に且つ正確に実施しなくてはならないが，検証すべき試験項目が正しく選定されなかったり，網羅的に検証が行われない場合は，不具合を内包したままシステムを提供しサービスの継続性を損ねてしまう事がある．

高品質，高信頼を要求される通信システム開発においては，ウォーターフォールモデル（図 4.1 参照）による開発を進めている [42, 43].

実際の開発においては各工程の実施内容やアウトプットを詳細に定め，工程ごとに品質の判定を行う事で，手戻りの少ない高品質なソフトウェア開発を目指している．検証すべき試験項目の作成は，上流の各工程の生産物である BI/BD 書や，FD/DD 書を基に作成されるが，適切な試験項目を作成できるかどうかは作業者のスキルやノウハウに大きく依存している．このスキルやノウハウの偏りによる不均質な試験項目作成を避けるため，試験項目作成基準を定めるたり試験項目レビュー等に大きく時間を割いており，この稼働がソフト開

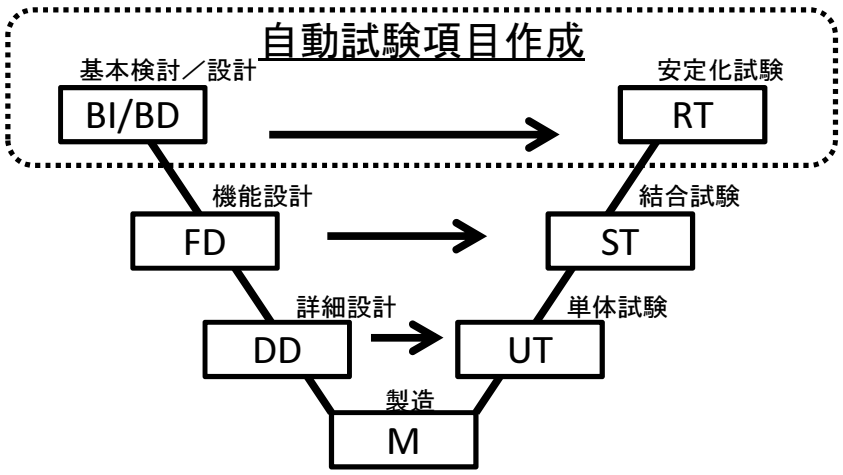


図 4.1　ソフト開発モデル (ウォーターフォールモデル) における自動試験項目作成

発のコスト増につながってしまっている.

本章では，試験項目作成作業者のスキルやノウハウに依存しない均質な試験項目をローコストで作成するため，試験項目作成作業者のスキルやノウハウに依存しない均質な試験項目を，基本設計工程の生産物である，「要求仕様書，BI/BD 書」から自動的に，安定化試験において検証すべき試験項目を作成する手法について提案する.（図 4.1 の自動試験項目作成部分）

これまでの研究においては，開発者が要件仕様の記述スタイルを機械が扱える形式で構造化し，この構造化された要求仕様を使用して試験項目を自動生成していた． しかし，自然言語文書から試験項目を生成する方法については言及されていない.

我々は，自動試験項目生成のトライアルシステムを構築した． このトライアルシステムを使用して自動的に生成された試験項目が高度に熟練したエンジニアによって作成された試験項目と同じであるかどうかを確認することによって，提案手法の有効性を評価する． 提案手法の有効性が確認できれば，高度な技術を持つエンジニアを雇用するコストが削減できることになる.

4.2　提案手法

前述した通り，構造化されていない自然言語で記述されている要求仕様書からの自動構造化による試験項目の自動生成手法を示す.

試験項目の自動生成の手順は以下の通りである（図 4.3）.

【パート 1】

(1)　要求仕様書に対し，試験項目に相当する部分にタグを付与し，タグ付けドキュメントを作成する.

【パート 2】

(2)　タグ付けドキュメントを教師データとして蓄積し，機械学習機に学習させる.

(3)　学習結果をタグの付与されていない要求仕様書に適用しタグを自動的に付与する.

(4)　タグ付けドキュメントから試験項目を自動的に作成する.

本章では，STEP2 の機械学習機への学習方法について以降の節で機械学習の性能向上を目指した手法の提案を行う.

4.2.1　要求仕様書に対するタグ付与手法

試験項目自動作成のため，構造化されていない自然言語記述の要求仕様書に対し，安定化試験項目を作成するためのタグを付与する．付与されたタグによって試験項目を作成することを目標とするため，要求仕様書に付与したタグについては，そのタグで括られる文章を集めることによって試験項目を作成することができなくてはならない．このため，タグの種類は試験項目の作成を意識した構成とする必要がある.

システムに対する試験項目は一般的に，システムに与える入力情報 (input) と，その時のシステムの状態 (condition)，そして，その入力がされた時のシステムからの出力 (output) で構成される．このため，最低限のタグの情報として，『input/condition/output』を定義した.

さらに，入力およびそれに伴う出力条件が複合的であり，詳細な条件が付与される場合を考慮し，入力条件情報 (input condition) と出力条件情報 (output condition) を定義した．また，通信システムにおいては複数のシステムで通信網が構成されており，要求仕様書においても複数のシステム（セッション制御システム／付加サービス制御システム 等）に同一の入力（SIP 等のセッションの接続信号 等）が行われ，別の出力となる記述があるため，試験対象となるシステム情報 (agent) を表すタグを定義した.

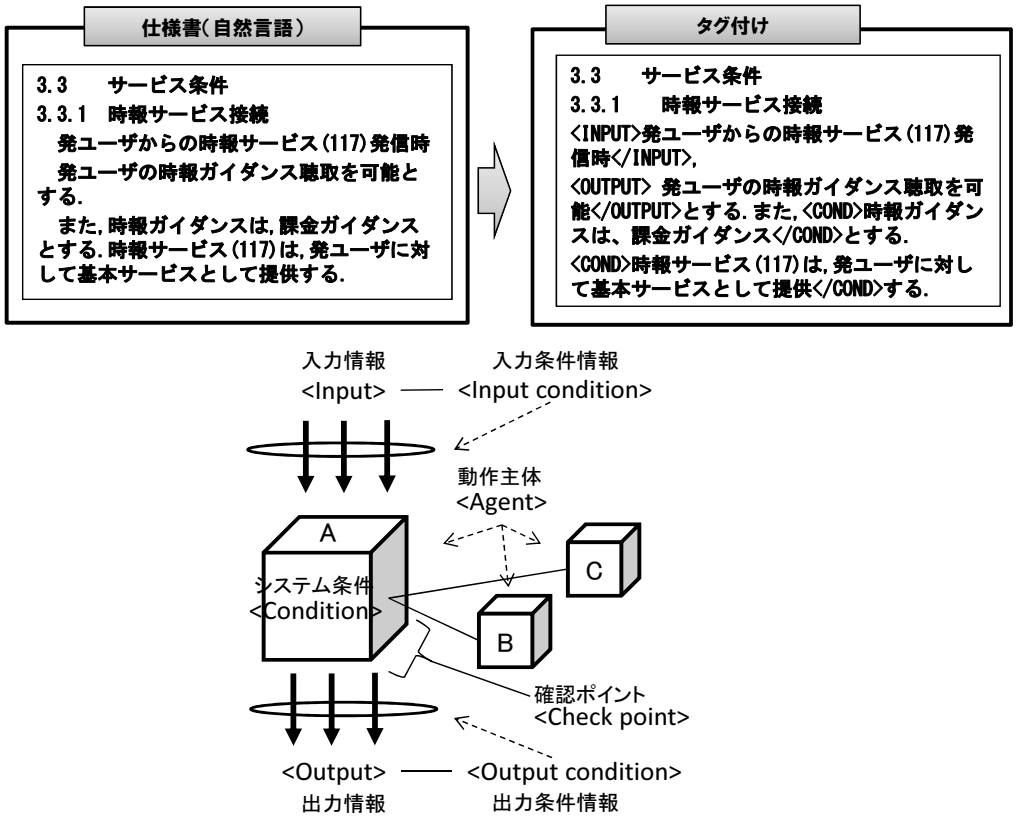


図 4.2　要求仕様書構造化のための 7 つのタグ

その他，要求仕様書にはシステム内動作の詳細が記載されているケースもあり，これはすなわち，検証実施時の確認すべきポイントであり，このシステム内動作のチェック条件 (check point) である事を示すタグを定義した．これらのタグの関係性を図 4.2 に示す.

4.2.2　機械学習機を用いた自動タグ付け

要求仕様から試験項目を抽出するためのトライアルシステムを構築した．図 4.3 の【パート 2】は，機械学習によって新しい要件仕様を自動的に構造化する手順を示している.

【パート 2】の『(2) タグ付けドキュメントを教師データとして蓄積し，機械学習機に学習させる』ために，まず教師データを作成する必要がある．要求仕様書の記述は日本語であるが，日本語は中国語，タイ語と同様，単語の間にわかち書きされていない，スペースを入れずに書かれる言語である．この様な言語において教師データを作成するためには，まず，文書を形態素解析ツールで

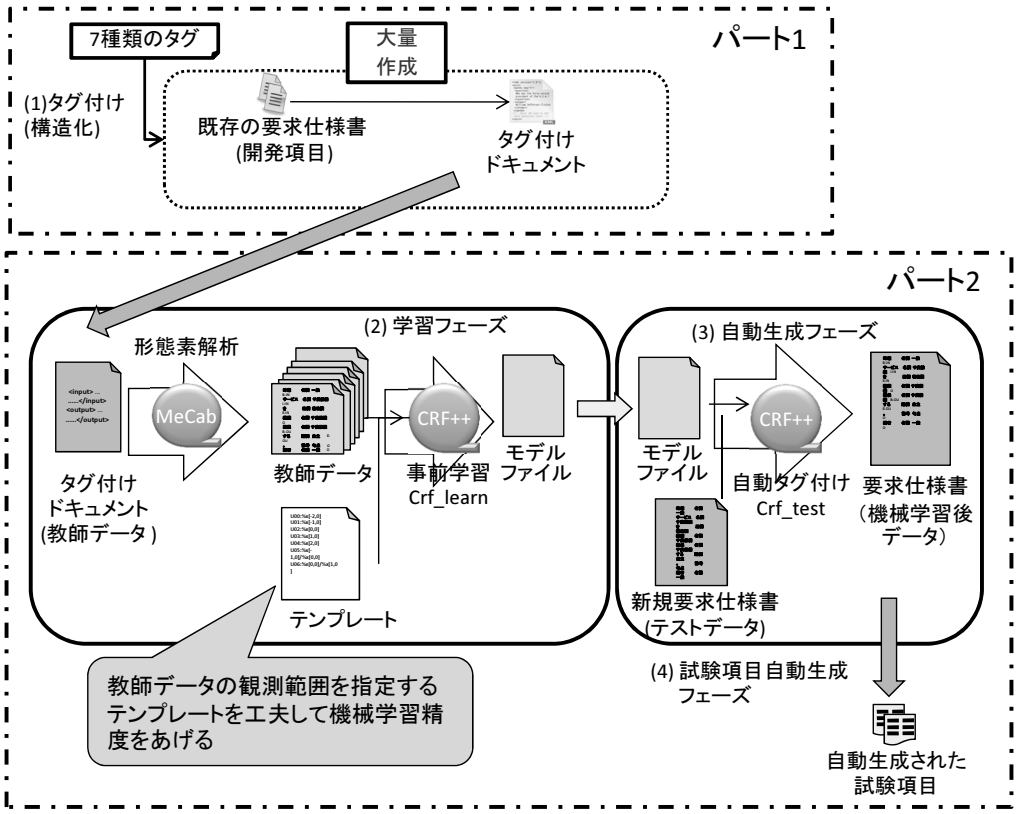


図 4.3　試験項目自動作成手順

ある MeCab[44] 等を使用して，機械学習（図 4.3）のために単語に分割する必要がある．MeCab は，品詞，品詞種別，および各単語の詳細な分類を表示する形態素解析器であるため，タグ付き構造化要件仕様から抽出された単語単位に単語カテゴリ情報も付与されている．

また分けた単語単位に，単語カテゴリ情報に加えて要求仕様書に付与していたタグを元にタグ情報を付与していく．たとえば，要求仕様書に記載されたタグ情報である「IN（入力)」,「OU（出力)」等を単語情報に付加し，タグに囲まれていない単語については，その他 (Other) を示す（O）を付加する．このようにして，単語の持つ属性の種類を含む大量の訓練データを生成し，これを教師データとして機械学習ソフトウェアがそのデータを学習できるようにする．

機械学習を行う際には教師データに加えテンプレート（素性関数を形式化したファイル）が必要となる．テンプレートは，機械学習ソフトウェア CRF++[45] での計算に利用され，教師データと共に事前学習結果であるモデルファイルとして出力される．CRF++ はオープンソースであり，シーケン

シャルデータをセグメント化/ラベリングするための条件付きランダムフィールド（CRF）のシンプルでカスタマイズ可能な機械学習器である．CRF++ は，名前付きエンティティ認識，情報抽出，テキストチャンクなど，さまざまなニューロ言語プログラミング（NLP）タスクに適用されている．

前述した通り，教師データにより学習する際，事前に学習の特徴を示すテンプレートを指定する必要がある．テンプレートとは素性関数のデザインを形式化したファイルであり，学習の際に使用される要素範囲を示している．テンプレートでは，学習データ内の現在のトークンの位置に対し，観察する情報の位置を指定する．

高品質を要求される商用の大規模通信システム開発で使用するため，要求仕様書の試験範囲を示す自動タグ付けはより正確さが求められる．テンプレートファイルを最適化し教師データに含まれる情報の関連性をより明確化することは，機械学習の精度を向上させることに繋がり，また，より正確なタグ付けに繋がる．

テンプレートファイルの各行は，入力データのトークンを指定するために使用される特別なマクロ %x [row，col] を示している．

「row」は現在のフォーカスされたトークンからの相対位置を指定し，「col」は列の絶対位置を指定する．テンプレートの観察されたトークンの相対的な位置の一例を図 4.4 に示す．

図 4.4 のテンプレートは，3 つの観測する情報を，以下の通り単語の現在のフォーカスされたトークンから得られる相対位置で示している．

- 「U00:%x[-2,0]/%x[-1,0/%x[0,0]」の記述は単語の ”0，-1，-2”（Trigram）の相対位置を指定
- 「U01:%x[0,1]/%x[1,1]」は品詞の”0，1”（Bigram）の相対位置を指定
- 「U02:%x[-2,2]」は品詞種別の”-2”（Unigram）の相対位置を指定

次に，形態素解析によって事前処理された新規の要件仕様書を，モデルファイルを利用した機械学習器（CRF++）によって計算することで，【パート 2】の『(3) 学習結果をタグの付与されていない要求仕様書に適用しタグを自動的に付与』を行う．

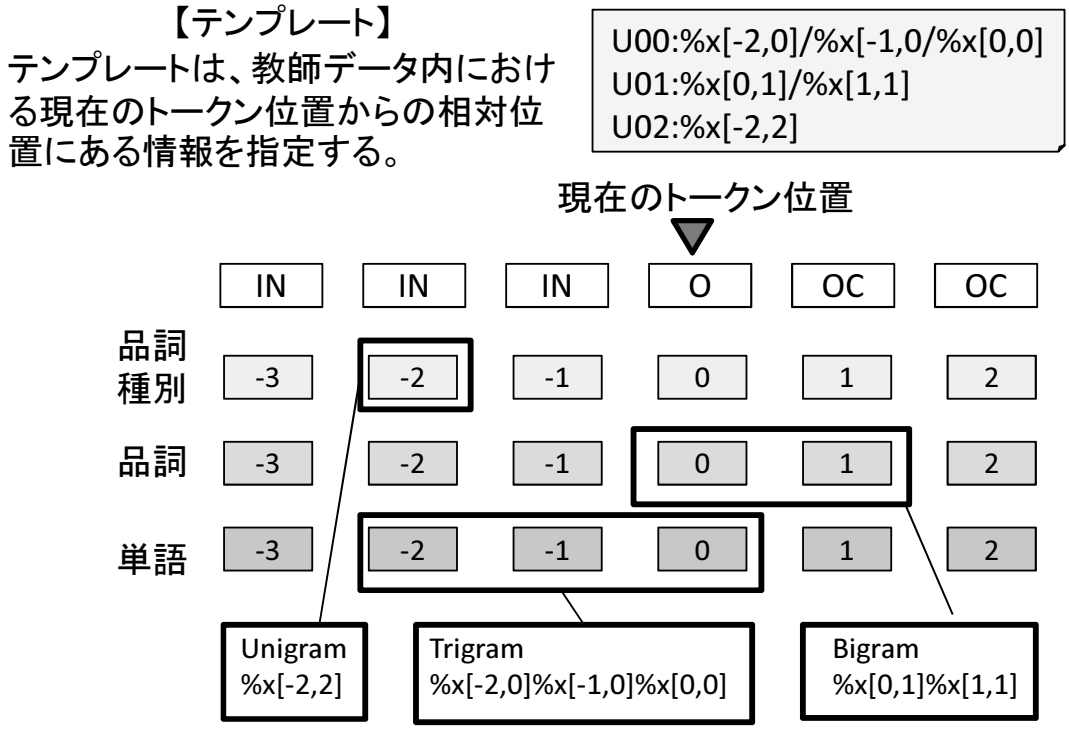


図 4.4　機械学習器 CRF++ のためのテンプレート

4.2.3　タグ付けチェックと試験項目抽出

【パート 2】の『(4) タグ付けドキュメントから試験項目を自動的に作成』を行う際に，タグ付けされた要求仕様書の試験項目としての結果が正しいかどうかをチェックするために，タグ付けされた要求仕様書から試験項目部分を抽出する方法を開発した．

Excel VBA（Visual Basic for Application）を使用して，タグを抽出する機能を作成した．スペルミスなどのタグ付けエラーによって正しい結果が導出されない事象に対しエラーを修正する必要があったため，エラーを検出する機構を設けた．試験項目抽出方法は，以下の (1)，(2)，(3) の 3 つのフェーズがある．

(1) タグを抽出する．
(2) タグ付けエラーを検出する．
(3) タグ付けエラーを訂正する．

図 4.5 はこの処理の流れを示す．

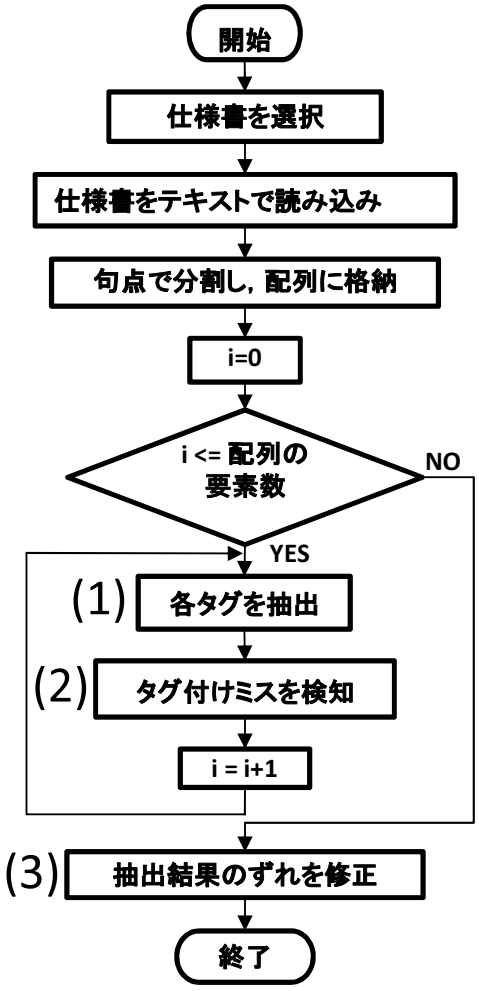


図 4.5　構造化要求仕様書のチェックと試験項目の自動抽出フロー

4.3　評価

自動タグ付けにより，すべてのタグ付き文書から試験項目が正しく抽出されたかどうかを，20 年以上の開発に携わる専門家によってタグを付与した要求仕様書と自動付与されたタグと比較して評価した（比較 1）．また，自動タグ付けによって生成された試験項目と実際の開発の試験項目を比較した（比較 2）．この試験項目比較において，正しい試験項目の割合（正しいと判断された試験項目の数/すべての正しい試験項目の数）を計算した．

最終的には全てのタグについて正確な予測を行う事で試験項目の自動作成を目指すため，比較 1 にて単語単位のタグの正誤で一致率を求めるが，完全な予

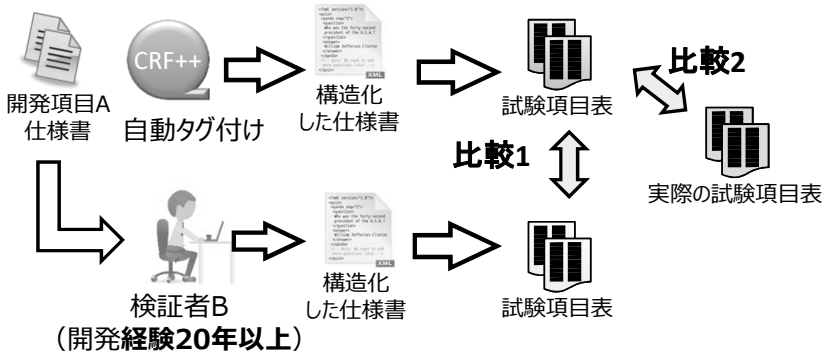


図 4.6　試験項目の正答比較評価プロセス

測ができていない状態においても機械学習による予測が試験項目となる記述を指しているか否かを比較 2 にて確認する．

提案手法により試験項目が自動作成できれば 3.2.3 に示した様に試験工数の 2 割を占める試験項目作成工数の削減が期待できる事と，項目作成における熟練者の技術が不要となる効果が期待できる．

4.3.1　試験項目自動抽出におけるテンプレート変更の効果（比較 1）

機械学習に用いるテンプレートを変更する事で教師データにおける品詞と品詞種別の観測範囲を徐々に広げ，事前機械学習と自動タグ付けを行った．使用したテンプレートの例を図 4.7，図 4.8 に示す．

図 4.9 は，テンプレートの変更により教師データの品詞と品詞種別の観測範囲を徐々に拡大した場合の，正しく付与されたタグ数とタグが付与されない部分の正答数のグラフである．

図 4.9 に示されているように，テンプレートの変更により教師データの品詞と品詞種別の観測範囲を徐々に拡大した場合，正しく付与されたタグ数とタグが付与されない部分の正答数が増加する傾向にあることを発見した．

作成した教師データセットは 35,307 ワード．新しい要件仕様書のタグ付与前のデータセットは約 3,000 ワードであった．提案手法の有効性は，トライアルシステムを用い，テンプレートを変更して自動的に生成された試験項目を，前述の 20 年以上の開発に携わる専門家が「実際の試験項目を参考に作成したタグ」と「機械学習／予測によって作成されたタグ」が単語単位で同一である

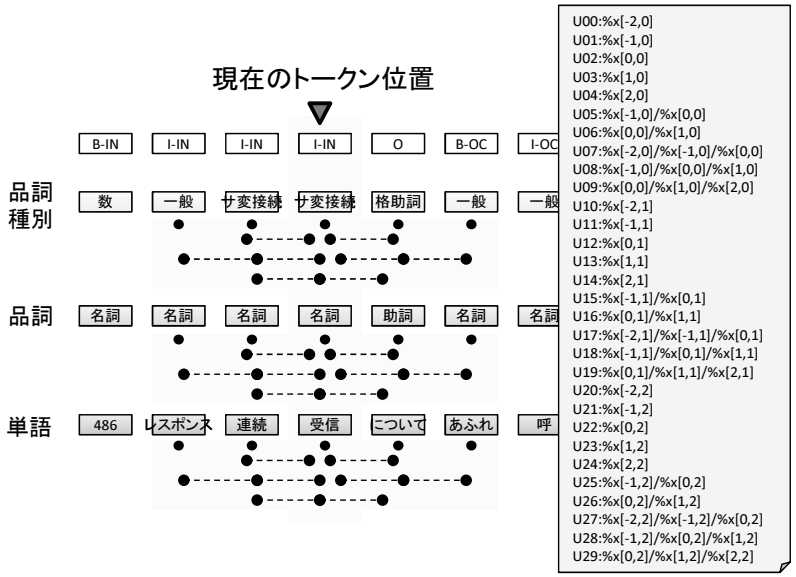


図 4.7　CRF++ ベースの基本テンプレート (テンプレート番号 3)

かどうかを比較することで評価した.

表 4.1 には，CRF++ がベースとして提供する基本テンプレートと最良評価結果のテンプレートの両方について，自動で付与したタグ数，正しいタグ数，適合率，再現率を示す.

適合率とは，全ての自動タグ付与数に占める正しい自動タグ付与数の割合（正しく付与されたタグの数／全ての自動タグ付与の数）を表す．再現率は，すべての正しいタグ数に占める自動タグ付与数の割合（正しく付与されたタグの数／すべての正しいタグの数）を表す.

CRF ++ がベースとして提供する基本テンプレートを使用した場合の適合率は（915/1501 =）60.1% であり，最良の評価を得たテンプレートであるテンプレート番号 11（図 4.8）の場合は（1163/1501 =）77.5% であった.

これらの結果は，テンプレートの適正化により正しくタグ付与ができる精度が向上したことを示している.

4.3.2　要求仕様からの試験項目抽出の有効性（比較 2）

過去の開発の要求仕様書から自動的に抽出された試験項目と実際の開発の試験項目を比較することにより，要件仕様書からの試験項目抽出の有効性を評価した実際の試験項目と，予測によってタグ付与された要件仕様書との比較は

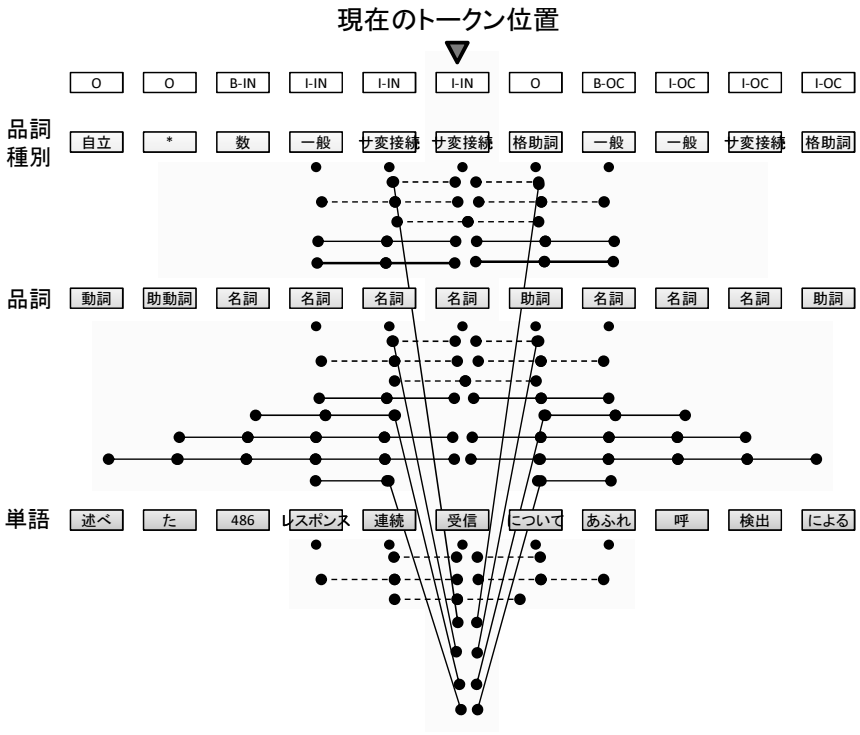


図 4.8　最良の評価結果のテンプレート (テンプレート番号 11)

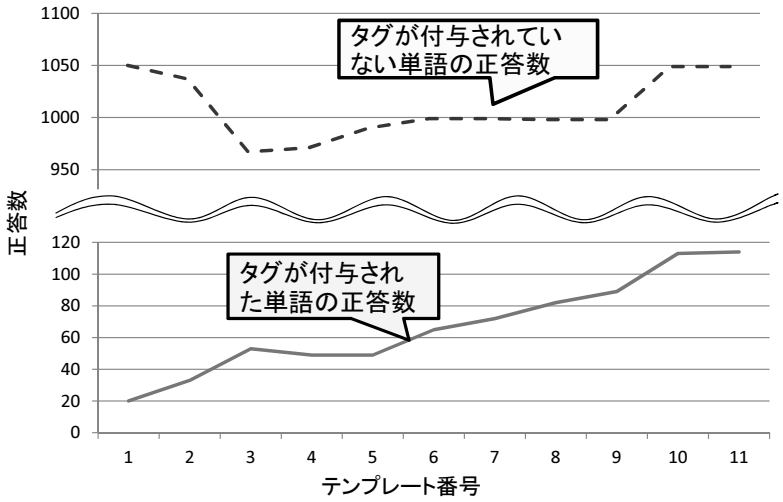


図 4.9　正答付与タグ数とタグが付与されない部分の正答数の変化

20 年以上の開発に携わる専門家が行った．比較 2 では試験項目（文章）単位での比較となるため，タグ内が部分一致の場合でも試験項目を指しているとした．評価結果を図 4.10 に示す．

タグ付けドキュメントから作成した試験項目は 65 項目で実際の項目と完全

表 4.1　テンプレート 3,11 の付与タグ数，正答タグ数，適合率，再現率

CRF++ ベースの基本テンプレート (テンプレート番号 3)

	Input	Output	Input condition	Output condition	Agent	Condition	No tag
自動タグ付与数	38	74	24	281	0	0	1084
正解タグ数	7	42	23	37	0	0	806
適合率	18.4	56.8	95.8	13.2	0.0	0.0	74.4
再現率	8.0	43.3	21.3	26.4	0.0	0.0	76.4

最良の評価結果のテンプレート (テンプレート番号 11)

	Input	Output	Input condition	Output condition	Agent	Condition	No tag
自動タグ付与数	18	16	51	109	1	0	1306
正解タグ数	6	13	42	52	1	0	1049
適合率	33.3	81.3	82.4	47.7	100.0	0.0	80.3
再現率	6.8	13.4	38.9	37.1	50.0	0.0	99.4

に一致した項目は 22 項目であった．完全一致した試験項目の特徴としては，新規追加機能の正常項目であり，残りの 43 項目は一致した項目の枝葉となる項目 (サービス加入条件やシステム条件が異なる）であった．ただし，試験項目には明記されていないが，安定化試験実施時に設定される加入者の条件や背景の呼の条件等を考慮すると，枝葉となる 43 項目の内容も含まれるため予測された項目は全て実際の試験項目と一致していたと考えられる．

逆に，実際に行った試験項目の 55 項目のうち，タグ付けドキュメントから作成した試験項目と一致しなかった差分は 33 項目あり，この差分は一般的に要求仕様書の段階では記載されることが少ない，準正常や運用手順，非機能要件等であった．

実際に行われたこれらの準正常等の試験項目の作成はスキル保有者のノウハウによるもので，『要求仕様書中には記載されていないが経験上暗黙値として試験すべき項目』として挙げたものであり，要求仕様書のタグ付け手法だけでは網羅できない．

要求仕様書にタグを付与する事で，安定化試験項目の抽出が可能となる事がわかった．ただし，正常項目のみで，およそ 40～50% 程度のカバー範囲とな

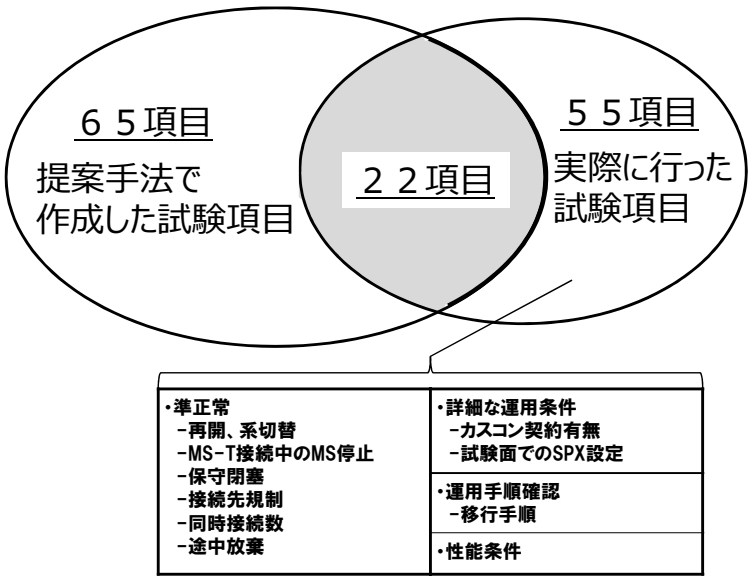


図 4.10　要件仕様書から抽出された試験項目と実際の開発時の試験項目との比較

る．それ以外は準正常項目等の要求仕様書に記載されていない要素であるが，要求仕様書にタグを自動的に付与でき，構造化できれば，反意語に置き換えたり，非機能要件を埋め込むことも可能と考える．

4.4　まとめ

本章では，試験項目作成作業者のスキルやノウハウに依存しない均質な試験項目をローコストで作成するための試験項目作成自動化について提案し，その評価をおこなった．

提案手法によって，自然言語で書かれた非構造化要求仕様書から，システム開発時の試験工程のための試験項目を自動的に生成する事が可能となった．

高度な技能を有する開発者によって，要求仕様書の試験項目に対応する記載箇所にタグが付与される．このタグ付きデータを用いて機械学習を行った．その後，機械学習の結果を，タグ付けされていない新しい要件仕様書に対し，試験項目を指し示すタグを自動的に付加することを実現した．

トライアルシステムを構築し，提案手法の有効性を評価した．実験結果は，機械学習を用いた自動試験項目生成の実現可能性を示した． また，CRF++ のテンプレートによって定義される学習範囲（素性）を，要求仕様書の記述傾向に最適化することでタグ付与の精度が向上することを明らかにした．

第 5 章

大規模通信システムを利用した P2P ネットワーク制御手法と評価

高いソフトウェア品質を維持しつつ，次世代ネットワーク (NGN) は SIP ベースのセッションによる帯域幅と品質 (損失，遅延，揺らぎ) を保証した通信機能を継続的に提供している．加えて，物理 IP ネットワークのセッションを制御するインタフェースを公開しており，NGN では SIP ベースのセッションを切換る Application Network Interface (ANI) 機能がある．P2P 通信サービスにおいては，エンドユーザの各端末がアプリケーションレベルのクライアントでありサーバであり中継ルータでもあるため，トラヒック集中が偏在する可能性がある．輻輳した端末がボトルネックとなりサービス品質が低下しないよう，P2P ネットワークの中継ルートを効率化させる P2P ネットワークトポロジー制御方法が必要であり，ANI 機能として提供されるネットワークのセッション制御機能が利用できる．本章ではユーザ端末アプリケーションの負担が少なく P2P ネットワークトポロジーを効率のよい構成へ変える方式を提案する．接続切換機能についてはいくつか方式が考えられ，これらの方式の特性を比較分析する．各方式の比較結果をシミュレーションによって検証し，実装ソフトウェアアーキテクチャのフィージビリティを試作検証によって確認した結果を報告する．

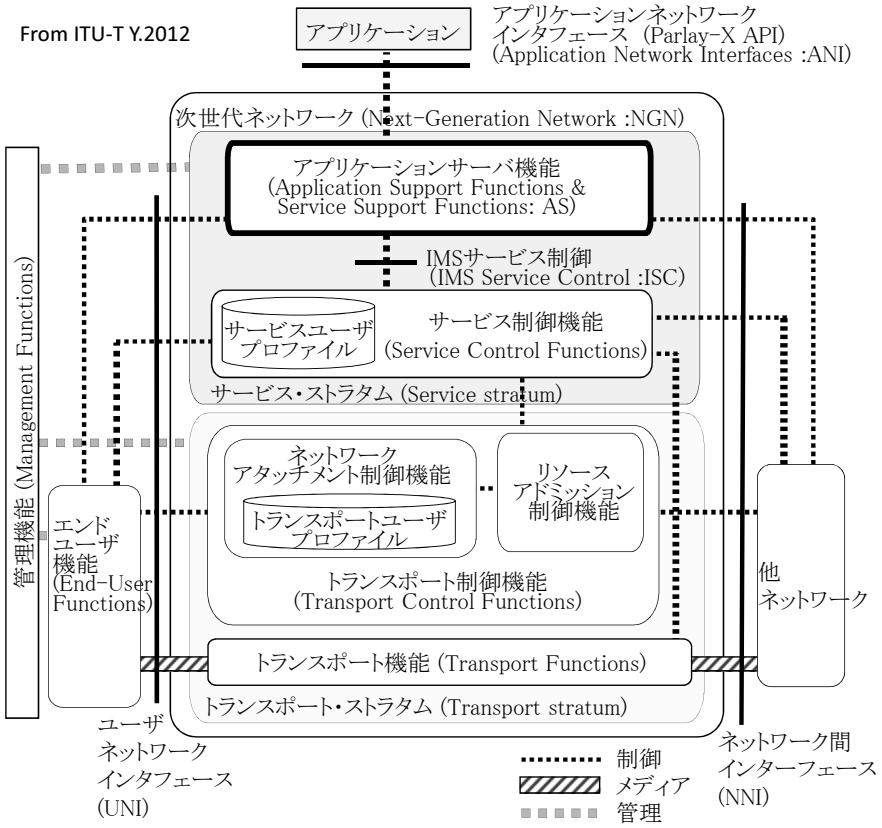


図 5.1　NGN のアーキテクチャ (ITU-T Y.2012)

5.1　背景と目的

通信事業者が提供している IP マルチメディアサブシステム (IMS)[46] に準拠した次世代ネットワーク (NGN)[47] は Session Initiation Protocol(SIP)[48] ベースのセッションによる帯域幅と品質 (損失，遅延，揺らぎ) を保証した通信機能を提供する (図 5.1)．しかし，IP 電話サービス以外にセッションを活用したサービスは広がっておらず，より多くのユーザに利用される多様なサービスが提供されるよう望まれている．

また，物理 IP ネットワークのセッションを制御するインタフェースが積極的に公開されてきている．例えば，各国の主要通信事業者のインタフェース提供 [4] に加え，現在は，ネットワークセッション制御の事業者として Twilio[5] などがでてきており，このセッション制御インタフェースの本格的な利用が注目されてきている．NGN にも SIP ベースのセッションを切換る Application

Network　Interface (ANI) 機能がある (例えば, Parlay-X Web Service API [6]). SIP サーバは端末とインタラクションする SIP プロトコルとセッション接続制御する ANI の相互連携を実現している．これを活用すれば，より魅力的なサービスアプリケーションをサービス事業者が容易に開発できると考えられる．

帯域幅と品質が保障される NGN のセッション制御機能は，継続的に安定してデータが流れることが望ましいデータストリーム型の通信サービスに有益であろうと考えられる．映像等のデータストリーム型の通信はインターネットの主要トラヒックであり今後もトラヒックが増大していくと予想されており [1]，より多くのユーザに利用されるサービスである．

ネットワークリソースの観点からみると，データストリーム転送は，データストリーム配信サーバとエンドユーザ端末間の 1 対 N のセッション接続によりデータストリーム配信を行うと，データストリーム配信サーバ側のデータ転送負荷の集中が懸念される． NGN の SIP による帯域幅と品質が保障されるセッション制御機能を利用しつつ IP マルチキャストのような方法に Session Description Protocol (SDP) でマルチセッション [7] を設定することが考えられる．しかし，実際の NGN では，帯域幅と品質を保証するセッションはユニキャスト通信機能のみを提供し，複数地点への再配信等の高度な機能は User Network Interface (UNI) で接続されるユーザ設備にて行っている．従って，多数のユーザ設備端末がデータストリームの中継点となりバケツリレー的に再度配信を端末間のセッションで行う形態がデータ転送負荷の集中を避けられ望ましいといえる．これは，ユーザ端末間のアプリケーションレベルのネットワーク，Peer-to-Peer(P2P) ネットワークを NGN のセッションを用いて構成することといえる．P2P ネットワークを適用するにあたっては，更なるトラヒック削減効果を目的とし PIN-SM，PIM-SSM を実装した IP マルチキャストを導入する事が効果的である [49]．さらに，P2P サービスを容易に構築できるプラットフォームを導入する事で簡便にアプリケーションの開発が可能となるばかりでなく [50]，ピア間の通信が活性化する事も実証実験で明らかとなっている [51, 52].

本稿では，NGN を利用した通信品質保証の SIP ベースのユニキャストセッションで端末間をつなぎ構成された P2P ネットワークで，様々な P2P サービスに利用できるプラットフォームソフトウェアの確立を目指す．

P2P ネットワークはエンドユーザの各端末がアプリケーションレベルのクライアントでありサーバであり中継ルータでもあり，トラヒック集中が偏在する可能性がある．また，各端末は Windows Update 等のピア内ほか APL 処理負荷が要因で予想できない輻輳状態に陥る可能性もある．輻輳した端末がボトルネックとなりサービス品質が低下しないよう，P2P ネットワークの中継ルートを効率化させる P2P ネットワークトポロジー制御方法を提案する．

各端末の P2P 基盤ソフトウェアが常時内部処理をチェック (CPU 使用率とアプリケーションレベルの通信バッファキュー長，等) し，輻輳発生が予見される (CPU 使用率が 100% 近くなる，等) 場合に中継ルートの切換処理を起動する．SIP ベースのセッション制御でつながった中継ルートを切換る場合，各端末の P2P ネットワークトポロジー制御機能が通信し接続切換処理を行うことが考えられる．この処理を行なう際，輻輳を予見し中継ルート切換処理の起動はできたが，接続先端末と連動した切換処理中に輻輳より切換処理が完了できず切換失敗となることが懸念される．端末間の中継ルートが物理ネットワークのセッションであることを鑑みると，輻輳しそうな端末が輻輳を予見して起動した中継ルート切換処理をネットワークプロバイダが提供するセッション制御機能によって行えば，端末が輻輳に陥っても中継ルート切換は完了できることが期待される．

本章では，P2P ネットワークのトポロジーを構成する端末間のリンクを切換る際に，ANI 機能として提供されるネットワークのセッション制御機能を利用し，ユーザ端末アプリケーションの負担が少なく P2P ネットワークトポロジーを効率のよい構成へ変える方式を提案する．接続切換機能についてはいくつか方式が考えられ，これらの方式の特性を比較分析する．各方式の比較結果をシミュレーションによって検証し，実装ソフトウェアアーキテクチャのフィージビリティを試作検証によって確認する．

本章の構成を示す．第 5.2 節に検討対象とした物理ネットワークと P2P ネットワークを紹介する．第 5.3 節では提案する処理方式を，第 5.4 節では方式の特性比較を，第 5.5 節ではソフトウェアアーキテクチャを示す．第 5.6 節では，シミュレーション実験により各方式の特性を検証した結果，第 5.7 節では試作により実装ソフトウェアのフィージビリティを検証した結果を示す．第 5.8 節では，まとめと今後の方向性を示す．

5.2　NGN と P2P ネットワーク

5.2.1　次世代ネットワーク (NGN)

NGN は，主に Service Stratum と Transport Stratum からなる (図 5.1)．Service Stratum は SIP サーバ機能が SIP ベースの端末への論理的な接続であるセッションを接続する 2 端末にそれぞれ 1 本ずつ張り Back to Back UA(B2BUA) 方式で 2 セッションをつなぐ．ANI を提供する SIP サーバは Application Server(AS) と呼ばれるが，本論文内では解りやすく SIP サーバと呼ぶ．Transport stratum は端末間の映像やセンサデータストリーム等のデータストリームを流すメディアを張り，端末間のトランスペアレントな転送を行う．メディアはセッションの接続制御によって接続される．各端末とのメディアを終端し複数のメディア間を中継し，端末に替わってメディアの接続を切換ることができるメディアブリッジ機能も具備できる．

NGN のトポロジーは，市単位などの地域の端末を収容するエッジルータを県単位などの中継ルータや地方ブロック単位などの上位中継ルータで接続するツリー構造である．SIP サーバやメディアブリッジなどは地方ブロック単位に配置され上位中継ルータを介して接続される．

5.2.2　P2P ネットワークモデルとピア接続

P2P ネットワークは，物理 IP ネットワークにつながった端末間が相互接続してトポロジーを構成する物理リソースにオーバーレイするネットワークである．P2P ネットワークトポロジーは各ピアの自律分散的な相互作用で自己組織化することにより構築される．無効なコピーパケットの発生を防ぐため，ツリー型 (単連結で閉路を持たない無向グラフ) のトポロジーを前提とする．これは，データストリーム配信の場合は中心にデータストリーム配信サーバがあり，各ピアがツリー状に接続する構成となる．中継転送する位置にあるピアを中継ピアと呼び，中継転送することがない P2P ネットワークの末端に位置するピアをリーフピアと呼ぶ．転送ルートのデータストリームサーバ中心への方向を上流とよび，末端への方向を下流と呼ぶ．

各ピアが P2P ネットワークに参加するとき，まず入り口として Well-known

の端末 (ピア) に接続し中継できるピアを発見し，一つのピアに接続する．他ピアから接続要求を受けたピアは，許容する最大接続数 (分岐数) $P(P \geq 2)$ まで接続し，すでに最大接続数まで接続していれば接続要求を拒否する．P2P ネットワークは各端末の安定性が保障されてはいないため，できるだけ中心からリーフピアまでのホップ数 (次数)(平均値を r と呼ぶ) は小さいほうが転送ルートはより安定する．従って，ピア参加時には最大接続数に達していない中継ピアで一番中心に近い中継ピアに接続する．最上流の接続候補ピアが複数ある場合は，物理ネットワーク負荷に適応した接続先選定方法により接続先ピアを選択する [53]．接続候補ピアに暫定接続しリクエストを送り，レスポンスにて Round Trip Time(RTT) を測定するとともに端末性能などの統計情報を受信する．これらの情報に基づいて，物理 IP ネットワーク上の近傍にあり，より性能に余裕のあるピアに接続要求する．接続要求をうけたピアは許容する接続数を超えていないかチェックし，超えていなければ接続する．接続を拒否された場合，接続先次点候補ピアに接続要求する．

ピアの離脱時は，P2P ネットワークが分断されないよう，離脱するピアに接続していた下流のピアが P2P ネットワークに参加する動作を行う．離脱したピアの一段下流の各子ピアが切り離され，その各ピアが自身の下流ピア群（孫ピア）との接続を維持したまま，改めて P2P ネットワークに参加する．P2P ネットワークの初期構築時と同様に，最大接続数に達していない中継ピアで最上流のピアに接続し，最上流のピアが複数ある場合は，物理ネットワーク上の近傍性などの観点から適したピアを選択する．

5.3　P2P ネットワークトポロジー制御機能

トラヒック変動による端末負荷上昇により輻輳発生が予見された場合，P2P ネットワークのトポロジーを構成する隣接端末間の接続 (P2P 接続) を切換る方式を提案する．

5.3.1　P2P ネットワークトポロジー制御方式

様々な P2P サービスに利用できることを目指し，特定アプリケーションの機能ではない P2P 接続切換により，汎用的な端末輻輳のボトルネック回避を狙う．P2P 接続を切換ることにより，不要な中継転送メッセージ数を低減する

ことができる．本制御手順を図 5.2 に示す．

各中継端末は中継転送の送信元と送信先別メッセージ密度を常時測定して統計情報を記録している．端末 B が輻輳発生を予見した時, 端末 B は統計情報を評価する．図 5.2 に示すように，端末 A と端末 C 間の中継転送のトラヒックが 100，端末 A と端末 B 間のトラヒックが 10，端末 B と端末 C 間のトラヒックが 20 とすると，端末 B は端末 A と端末 C 間の中継転送するための負荷が重いことがわかる．この場合，端末 B は中継転送負荷を抑制するため端末 A と端末 C を直接接続するよう P2P ネットワークトポロジーを切換る．

端末 A と端末 B 間のトラヒックより端末 B と端末 C 間のトラヒックが大きいため端末 B と端末 C 間のリンクを残し端末 A と端末 B 間のリンクを端末 A と端末 C 間にリンクへ切換る．つまり，切換により接続リンク数が減るピアがピア B，切換により接続リンク数が増えるピアをピア C とする．ピア C がピア B の上流ピアでピア A が下流の場合は，ピア A が 1 ホップ中心に近くなる．一方，ピア C がピア B の下流でピア A が上流の場合は，ピア C が 1 ホップ中心に近くなりピア B が 1 ホップ中心から遠くなる．もし，ピア A とピア C ともにピア B の下流の場合は，ピア A が 1 ホップ中心から遠くなる．ただし，切換により接続リンク数が増えるピア C が既に接続数の上限に達している場合は，本切換は行わない．

5.3.2　P2P 接続切換方式

5.3.1 節の P2P ネットワークトポロジー制御方式を実現する処理方式には，NGN の機能を利用する程度により 3 方式考えられる．

あらかじめ，5.2.2 節に示した方法で P2P ネットワークが構築されている．各ピアが P2P ネットワークに参加するとき，NGN では参加する端末が接続先の端末へ SIP の Invite 信号により確立されるセッションとメディアで接続する．

方式 1:端末連携方式

従来は，端末間相互連携動作のみによる P2P ネットワークトポロジーの切換処理を行っている． NGN の SIP によるセッション制御は，転送電話のように通話中端末が他端末に通話中呼をつなぎかえる指示を端末間で行う REFER メソッド方式 (RFC 3515[54]) を利用する．

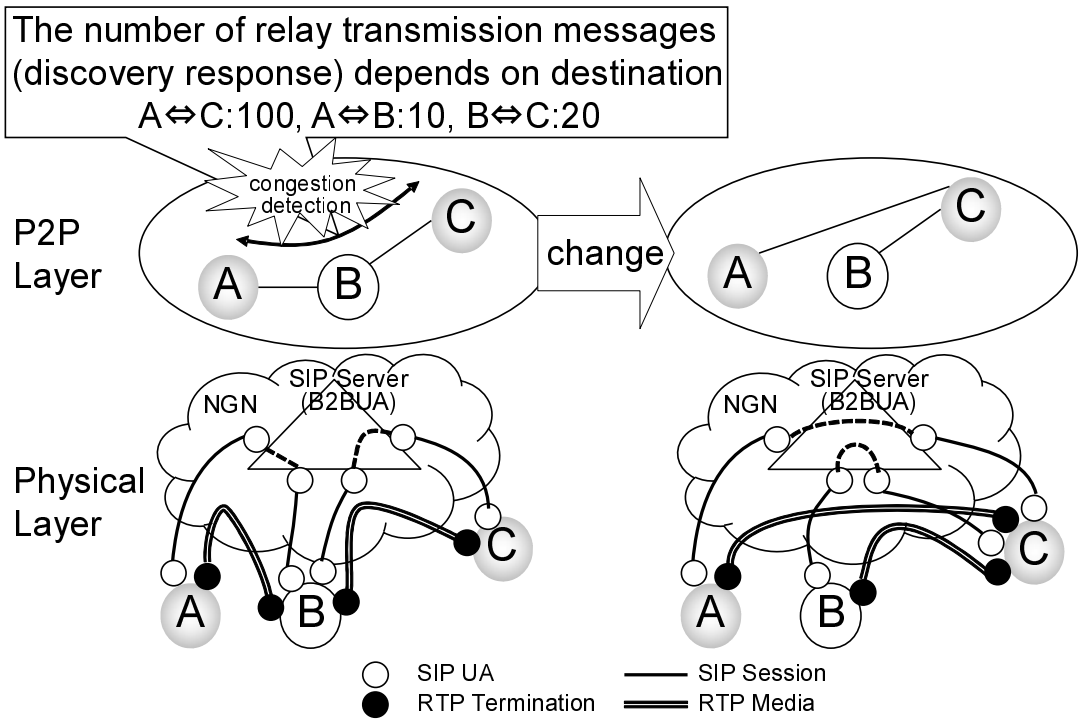


図 5.2　P2P ネットワークトポロジー制御

図 5.2 に示した P2P トポロジー切換動作により端末 A,B 間の接続を A,C 間に切換る際，従来方式である端末連携方式 (方式 1) の動作シーケンスと，端末及びネットワークの使用リソースを図 5.3 に示す．

端末 B が輻輳を予見 (CPU 使用率が 100% 近くなる，等) し，P2P ネットワークにおける端末 A,B 間の接続に代わって A,C 間を接続するよう処理を起動する．

(1) 端末 B は端末 C に SIP の Refer 信号を送信する．
(2) 端末 C は SIP の Invite 信号を SP サーバへ送信する．
(3) SIP サーバは端末 A に送信し物理 IP ネットワークである NGN のセッションを張る
(4) メディアを確立する．

これにより，P2P ネットワーク上の端末 A,C 間が接続される．このとき，切換動作は端末 B が司り，端末 B は接続処理状況の通知信号を受け取り，処理状況を監視するとともに異常時に再接続等を指示する信号を適宜送る必要がある．

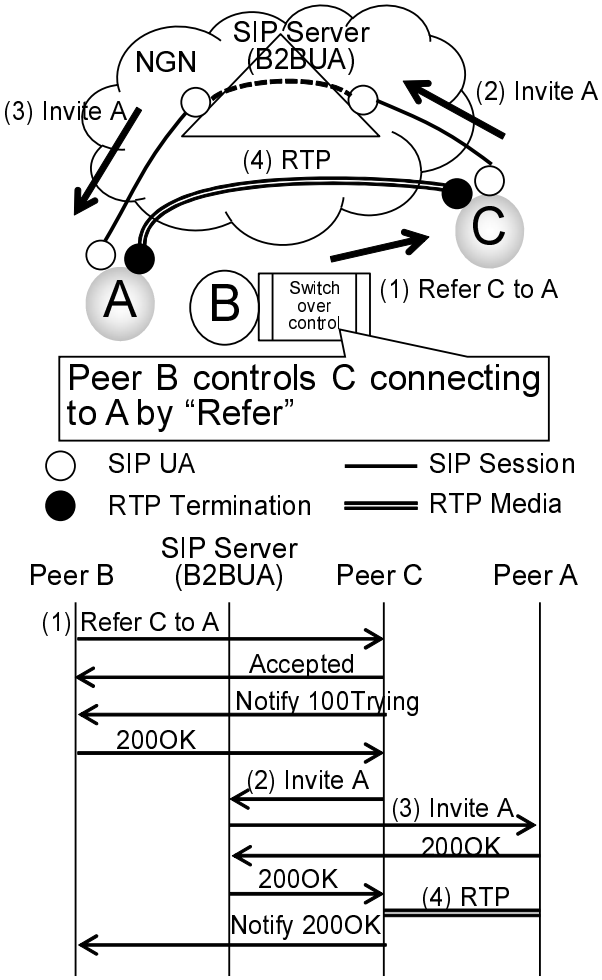


図 5.3　方式 1:端末連携方式

端末が輻輳を予見し中継ルート切換処理の起動した後，方式 1 は切換処理を端末間が通信し連携する処理なため，切換処理中に輻輳状態に陥り切換失敗となる懸念がある．

方式 2:セッション制御 ANI 利用方式

ネットワークが提供するセッション切換機能を利用し輻輳しそうな端末の切換処理負荷が軽い方式を提案する．

図 5.1 における ANI を介して Service stratum の SIP サーバによるセッションを切換ることにより端末間に IP ルータの Transport stratum を介して接続されるメディアを切換る．SIP によるネットワーク起動のセッション制御信号方式は，3rd Party Call Control(3PCC) 方式 (RFC 3725[55]) を利用する．

図 5.3 と同様に，端末 A,B 間の接続を A,C 間に切換る際のセッション制御 ANI 利用方式 (方式 2) の動作シーケンスと，端末及びネットワークの使用リソースを図 5.4 に示す．

端末 B が輻輳を予見し，P2P ネットワークにおける端末 A,B 間の接続に代わって A,C 間を接続するよう処理を起動する．

(1) 端末 B は NGN が提供するセッション制御インタフェースの Parlay-X “Make Call Session”をコールする．
(2) NGN 内の SIP サーバが SIP の Invite 信号を端末 C へ接続先をブランク (bh) にして送信する．
(3) 応答信号で得た端末 C のアドレスを設定した Invite 信号を端末 A に送信し，応答で得た端末 A のアドレスを設定した Update 信号を端末 C へ送る．端末 A と端末 C へ SIP 信号を送信し両端末にそれぞれ SIP セッションを張る．SIP の B2BUA 方式を用いて 2 本のセッションを SIP サーバが繋ぐ．
(4) メディアは端末 A と C 間で直接接続させる．

これにより，P2P ネットワーク上の端末 A,C 間が接続される．このとき，切換動作は NGN の SIP サーバが司り，処理状況を監視するとともに異常時に再接続等を指示する信号を適宜送る．

本方式では，セッションの切換処理にネットワーク機能を利用し輻輳しそうな端末の処理負荷を軽減できるが，メディアの接続を切換るため接続先端末はセッション切換処理が必要である．

方式 3:セッションとメディアの制御 ANI 利用方式

NGN では，各端末とのメディアを終端しメディア間を中継し接続を切換るメディアブリッジ機能も具備されている．これを利用した各端末の接続先切換処理の負荷がより軽い方式を提案する．

図 5.1 における Service stratum の SIP サーバによるセッション切換のみならず，Transport stratum にメディアブリッジを配備し，ネットワーク内のメディア切換機能も考える．メディアブリッジを利用した接続形態を図 5.5 に示す． P2P ネットワークにおける端末 A,B 間，B,C 間の接続は，NGN 内の SIP サーバが各端末との SIP セッションを終端しメディアブリッジが各端末とのメ

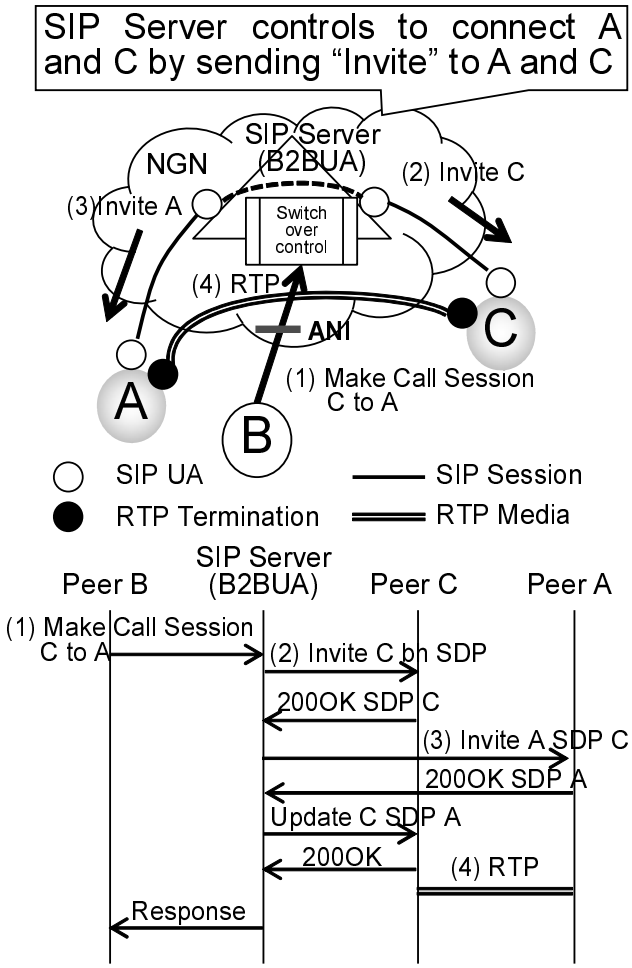


図 5.4　方式 2:セッション制御 ANI 利用方式

ディアを終端する．SIP セッションは B2BUA で SIP サーバが中継しメディアはメディアブリッジが中継し，端末間を接続する．端末間の接続を切換る際は SIP サーバとメディアブリッジにて接続が切換られる．

ANI を介して，セッション切換制御機能とメディアブリッジを利用する P2P ネットワークトポロジー制御方式も提案する (方式 3)．方式 3 の動作シーケンスと，端末及びネットワークの使用リソースを図 5.6 に示す．

端末 B が輻輳を予見し，P2P ネットワークにおける端末 A,B 間の接続に代わって A,C 間を接続するよう処理を起動する．

(1) 端末 B は NGN が提供するセッション制御インタフェースの Parlay-X “Make Call Session” をコールする．端末 C はメディアブリッジまでのセッションとメディアを 1 本増やす処理が必要である．一方，端末 A

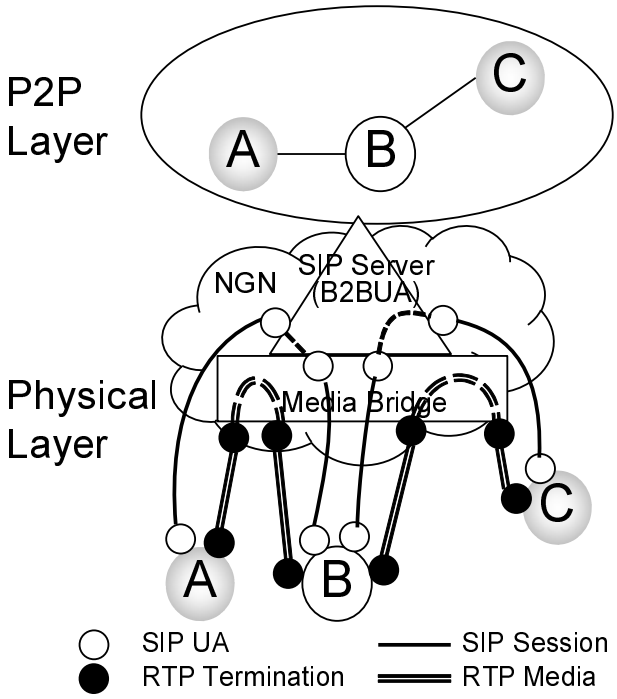


図 5.5　メディアブリッジによる接続形態

はメディアブリッジで終端されるセッションとメディアを変更しなくても端末 B への通信から端末 C への通信にメディアブリッジにて接続を切換られる．

(2) NGN 内の SIP サーバが SIP の Invite 信号を端末 C へ送信し SIP セッションを張る．

(3) メディアブリッジと端末 C 間のメディアを確立する．

(1)，(3) これにより，P2P ネットワーク上の端末 A,C 間が接続される．このとき，切換動作は NGN の SIP サーバが司り，処理状況を監視するとともに異常時に再接続等を指示する信号を適宜送る．

5.3.3　各方式の特徴

第 5.3.2 節から第 5.3.2 節に示した 3 方式の特徴と要考慮点を表 5.1 にまとめる．特徴として，利用する NGN 機能とそのねらい，P2P 接続切換の主体，メディアの終端点，から整理する．

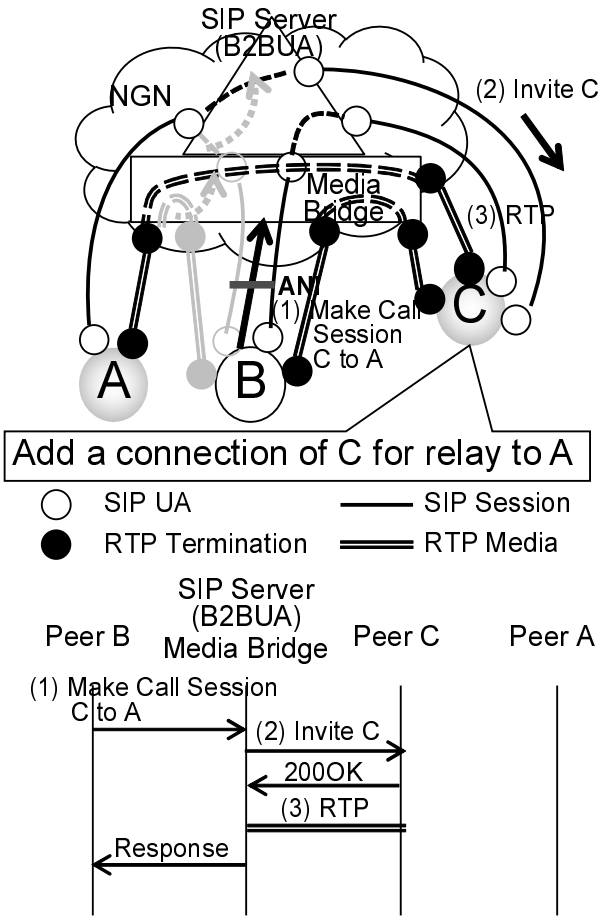


図 5.6　方式 3:セッションとメディアの制御 ANI 利用方式

5.4　各方式の特性比較

各方式の特性を比較する．CPU の計算処理量やメモリ等の記憶容量のリソース消費について，「評価の観点 1」各方式の接続切換を制御する信号処理，「評価の観点 2」メディアを通るデータストリーム転送処理，そして，「評価の観点 3」切換失敗やストリームの遅延など起こすユーザビリティ，にて比較を行う．

5.4.1　計算処理量

切換発生回数に依存して各方式が消費する CPU の計算処理量を比較し特性を明らかにする．「評価の観点 1」接続切換を制御する信号処理では，中継端末

表 5.1　各方式の特徴

特徴	方式 1	方式 2	方式 3
利用する NGN 機能とそのねらい	SIP による接続機能	SIP の接続機能に加え，ANI によるセッション切換機能を利用し輻輳端末の負荷軽減	SIP の接続機能，ANI のセッション切換機能に加え，メディアブリッジによる接続切換機能を利用し接続切換負荷軽減
P2P 接続切換の主体	輻輳発生端末	SIP サーバ	SIP サーバ
メディアの終端点	端末間	端末間	端末とメディアブリッジ間
要考慮点	端末が輻輳状態に陥り切換失敗となる可能性	方式 1 と比較した端末負荷軽減効果の確認	方式 1 と比較した端末負荷軽減効果の確認，メディアブリッジの接続数限界の影響

が輻輳発生を予見した時に起動される切換処理時に利用される端末と SIP サーバの信号送受信処理 CPU 処理量があげられる．その際，切換処理を行う各ピアはメディアを通るデータストリーム転送処理も行っており「評価の観点 2」データストリーム送受信処理量も含めた比較を行う．

- 物理ネットワークにオーバーレイする端末間の接続である P2P ネットワークのトポロジーはデータストリームサーバを中心に接続数 (分岐数) が $P(P \geq 2)$, 中心からリーフピアまでのホップ数 (次数) の最大値 (半径) が r のバランスドツリー (平衡木) (図 5.7)[53] とする．
- 1 信号の送受信に必要な計算処理量は信号種別によらずほぼ同様で tr と

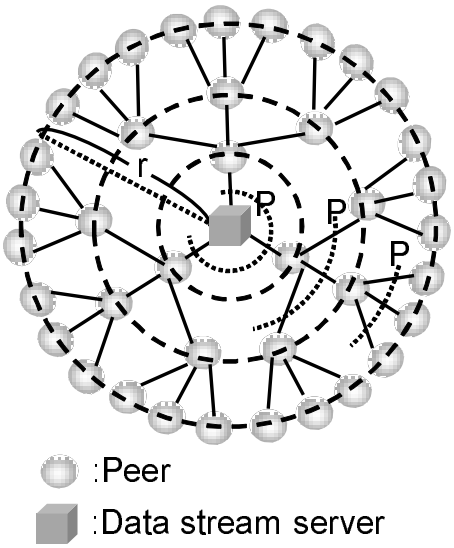


図 5.7　P2P ネットワークモデル（バランスドツリー）

する．

- データストリームの送受信による処理量はデータストリームのデータ速度によるため，今回は 1 信号の送受信処理量 tr の L_{md} 倍，データストリームの送信もしくは受信による処理量を $L_{md}tr$(データストリームの中継には $2L_{md}tr$) とする．

リーフピア数は末端の円周に位置するピアの数であり，式 (5.1) で表される．

$$PP^{r-1} = P^r \tag{5.1}$$

中継ピア数は，初項 P，公比 P で初項から $r-1$ 項までの等比級数であり，式 (5.2) で表される．

$$\sum_{i=1}^{r-1} PP^{i-1} = \frac{P(1-P^{r-1})}{1-P} \tag{5.2}$$

(1) データストリーム送受信処理量

「評価の観点 2」データストリーム送受信処理量は，切換方式のかかわらず想定することができる．データストリームの転送ルートを図 5.8 に示し，1 転送ルートにデータストリームが発生する確率を β とする．

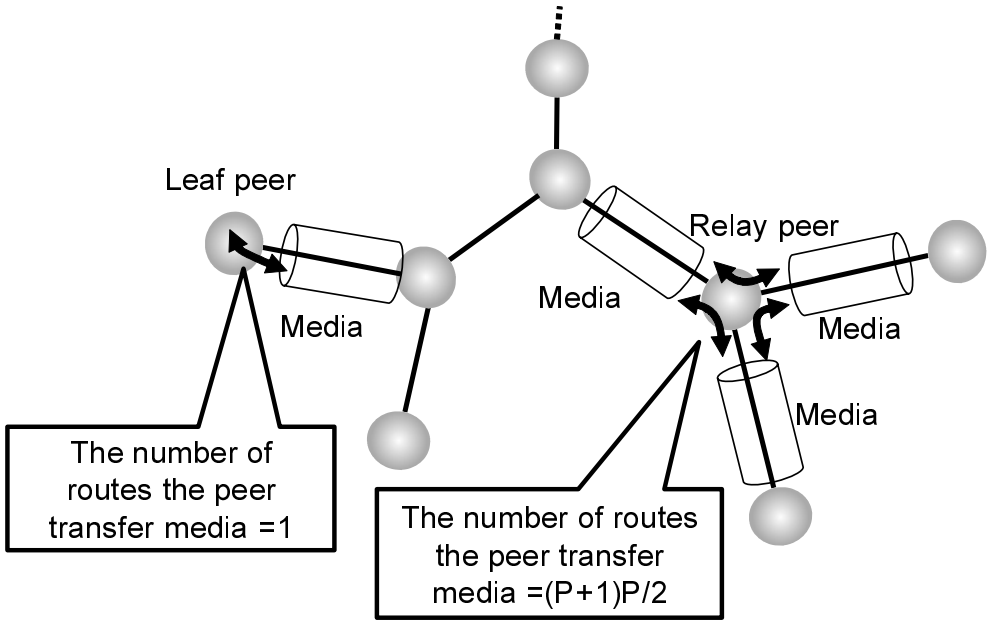


図 5.8　メディア送受信処理を行う転送ルート

各中継ピアがデータストリーム送受信処理を行う転送ルート数は，$P+1$ 個の接続しているピア 2 個の組み合わせ $\frac{(P+1)P}{2}$ 本の転送ルートがある．1 中継ピアがデータストリームの中継を行う確率は式 (5.3) で表される．

$$\frac{\beta(P+1)P}{2} \tag{5.3}$$

1 中継ピアにかかるデータストリームの中継処理量は，式 (5.3) とデータストリームの送信と受信による処理量 $2L_{md}tr$ との積 $\beta(P+1)PL_{md}tr$ で表される．各中継ピアがデータストリームの送信ピアまたは受信ピアとして転送を行う転送ルートは $P+1$ 本あり，1 転送ルートにデータストリームが発生する確率 β なので，1 中継ピアがデータストリームの送信ピアまたは受信ピアとして転送を行う確率は $\beta(P+1)$ である．1 中継ピアにかかるデータストリームの送受信処理量は，これとデータストリームの送信または受信による処理量との積 $\beta(P+1)L_{md}tr$ で表される．従って，1 中継ピアにかかるデータストリーム転送の処理量は式 (5.4) で表される．

$$\beta(P+1)^2L_{md}tr \tag{5.4}$$

一方，リーフピアは受信のみ行われ中継転送がないため，1 リーフピアがメディア送受信処理を行う転送ルートは 1 つである (図 5.8)．データストリーム

の発生確率は $\beta \times 1 = \beta$ である．1 リーフピアには式 (5.5) のデータストリーム送受信処理量がかかっていると表される．

$$\beta L_{md} tr \tag{5.5}$$

(2) 切換発生回数

中継ピア B が輻輳を検出すると，P2P 中継ルートを切換る．1 転送ルートに確率 β で発生するデータストリームが中継ピア輻輳等により切換が発生する確率を γ とすると，1 ピアに発生する単位時間あたりの切換発生回数は式 (5.3) と γ の積である式 (5.6) で表される．

$$\frac{\beta\gamma(P+1)P}{2} \tag{5.6}$$

P2P ネットワーク全体での切換発生回数は，中継ピア数の式 (5.2) 倍であり，式 (5.7) で表される．

$$\frac{\beta\gamma(P+1)P^2(1-P^{r-1})}{2(1-P)} \tag{5.7}$$

(3) 各方式の 1 切換における処理量

方式 1,2,3 を方式 i ($i = 1, 2, 3$) とし，各方式の 1 切換における処理量を算出する．各方式の 1 切換処理量は概ね信号の送受信処理量であり，1 信号の送受信処理量 tr と 1 切換処理での信号送受信回数の積で表される．

- 輻輳発生し接続数が減るピア B の処理量を $L_{bi} tr$
- 切換で接続数が増えるピア C の処理量を $L_{ci} tr$
- 切換で接続先が変わるピア A の処理量を $L_{ai} tr$
- SIP サーバの処理量を $L_{si} tr$

各方式の処理シーケンスから 1 切換における処理量の係数が分かる (表 5.2)．方式 2 は従来の方式 1 よりサーバリソースが多く必要であるが端末リソースの負荷が軽減され，方式 3 は更に端末リソース負荷が軽減され且つ通信負荷も軽減されている．

表 5.2　各方式の 1 切換における処理量

係数	方式 1	方式 2	方式 3
L_{ai}	2	2	0
L_{bi}	5	2	2
L_{ci}	7	4	2
L_{si}	4	8	4

(4) 中継ピアの処理量

ピア B に輻輳が発生する確率が式 (5.6) であり，中継ピア B に輻輳が発生してピア B にかかる処理量は式 (5.8) で表される．

$$\frac{\beta\gamma(P+1)PL_{bi}tr}{2} \tag{5.8}$$

ピア C に輻輳が発生してピア B にかかる処理量は，(ピア B の接続数が増えるピアとしての処理量)+(ピア B が接続数は変わらず接続先がかわるピアとしての処理量) である．

(ピア B の接続数が増えるピアとしての処理量) は，ピア C 輻輳発生確率の式 (5.6) をピア C が接続しているピア数 $(P+1)$ で割って得られるピア B の接続数が増えるピアとなる確率と $L_{ci}tr$ の積であり，$\frac{\beta\gamma PL_{ci}tr}{2}$ である．

(ピア B の接続数が変わらず接続先がかわるピアとしての処理量) は，ピア C 輻輳発生確率の式 (5.6) をピア C が接続しているピア数 $(P+1)$ で割って得られるピア B の接続数が変わらず接続先がかわる確率と L_{ai} の積であり，$\frac{\beta\gamma PL_{ai}tr}{2}$ である．

ピア A が輻輳発生ピアの場合にピア B にかかる処理量は，ピア C が輻輳発生時と同様であり，他のピア B に接続している全ピアが輻輳発生時もピア C が輻輳発生時と同様である．従って，ピア B が接続しているピア $(P+1$ 個$)$ が輻輳発生時にピア B にかかる処理量は，式 (5.9) で表される．

$$\frac{\beta\gamma P(P+1)tr}{2}(L_{ci}+L_{ai}) \tag{5.9}$$

全中継ピアにかかる処理量 L_{tti} は，1 中継ピアにかかる処理量である式 (5.8)+

式 (5.9)+ 式 (5.4) を中継ピア数 (式 (5.2)) 倍した，式 (5.10) で表される．

$$L_{tti} = \frac{\beta P(P+1)tr(1-P^{r-1})}{2(1-P)} (P\gamma(L_{ai} + L_{bi} + L_{ci}) + 2(P+1)L_{md}) \tag{5.10}$$

(5) リーフピアの処理量

リーフピアに輻輳が発生しても切換は発生しないので，リーフピアが輻輳発生ピアとしてかかる処理量は 0 である．

上流ピアが輻輳発生ピアの場合にリーフピアにかかる処理量は，(リーフピアの接続数が増えるピアとしての処理量)+(リーフピアの接続数が変わらず接続先がかわるピアの処理量) である．

(リーフピアの接続数が増えるピアとしての処理量) は，ピア C 輻輳発生確率の式 (5.6) を上流ピアが接続しているピア数 $(P+1)$ で割って得られるリーフピアの接続数が増えるピアとなる確率と $L_{ci}tr$ の積であり，$\frac{\beta\gamma PL_{ci}tr}{2}$ である．

(リーフピアの接続数が変わらず接続先がかわるピアの処理量) は，ピア C 輻輳発生確率の式 (5.6) を上流ピアが接続しているピア数 $(P+1)$ で割って得られるリーフピアの接続数が変わらず接続先がかわる確率と $L_{ai}tr$ の積であり，$\frac{\beta\gamma PL_{ai}tr}{2}$ である．

リーフピアが接続しているピアは 1 個であり，それが輻輳発生時にリーフピアにかかる処理量は，式 (5.11) で表される．

$$\frac{\beta\gamma Ptr}{2}(L_{ci} + L_{ai}) \tag{5.11}$$

全リーフピアにかかる処理量 L_{lti} は，1 リーフピアにかかる処理量である式 (5.11)+ 式 (5.5) をリーフピア数 (式 (5.1)) 倍した式 (5.12) で表される．

$$L_{lti} = \frac{\beta P^r tr}{2}(\gamma P(L_{ai} + L_{ci}) + 2L_{md}) \tag{5.12}$$

(6) 全ピアの切換処理量

方式 i $(i = 2, 3)$ の P2P ネットワークの全ピアの処理量 L_{termi} は全中継ピアの処理量 L_{tti} と全リーフピアの処理量 L_{lti} の和であり，式 (5.10) と式 (5.12) の和で求められ，式 (5.13) で表される．

$$L_{termi} = L_{tti} + L_{lti} =$$

$$\frac{\beta tr\gamma}{2}(\frac{P(P+1)(1-P^{r-1})}{1-P}(P(L_{ai}+L_{bi}+L_{ci}) \\ +2(P+1)\frac{L_{md}}{\gamma})+P^{r}(P(L_{ai}+L_{ci})+\frac{2L_{md}}{\gamma})) \tag{5.13}$$

(7) 切換処理量比較結果

各方式における全ピアの処理量の比較を従来方式の方式 1 を”1”として方式 2 との比率 $\frac{L_{term2}}{L_{term1}}$，方式 3 との比率 $\frac{L_{term3}}{L_{term1}}$ で各方式を比較評価する．

「評価の観点 1」各方式の接続切換を制御する信号処理は，P2P ネットワークトポロジーパターンによる各方式の特性が異なると考えられ，P と r の組み合わせにより比較評価する．今回は最大規模でピア数が十分な数 960,799 個となる，$2 \leq P \leq 7$，$2 \leq r \leq 7$ とした．

中継ピアの輻輳発生要因がデータストリーム転送によるもののみならず Windows Update 等のピア内他 APL 処理負荷が要因となる場合もあるため，1 転送ルートにデータストリームで中継ピア輻輳等により切換が発生する確率 γ はデータストリーム転送とは関係なく一定の値と仮定する．1 転送ルートにデータストリームに中継ピア輻輳等により切換が発生する確率の違いにより各方式の特性が異なると考えられる．

データストリームの送受信処理と信号の送受信処理が別プロセッサで実行されるアーキテクチャの場合を想定し，データストリーム送受信負荷が信号処理に影響与えない場合の評価の観点 1 の比較結果を示す．$\frac{L_{term2}}{L_{term1}}$ と $\frac{L_{term3}}{L_{term1}}$ を図 5.9 に，$\frac{L_{term2}}{L_{term1}}$ を拡大したグラフを図 5.10 に，$\frac{L_{term3}}{L_{term1}}$ を拡大したグラフを図 5.11 に示す．

接続切換を制御する信号処理の計算処理量だけを見ると，図 5.9 から，既存の方式 1 より提案方式である方式 2 のほうがよく，方式 3 がさらによい．図 5.10 と図 5.11 の P に対する値から，方式 3 は P がより大きな値の場合によりよく，方式 2 は P がより小さい値の場合によりよい．r に対しては，方式 2 はより大きい値の場合によりよく，方式 3 はより小さい値の場合によりよい．

「評価の観点 2」から，データストリームの送受信による処理量が信号処理の方式に与える影響を明らかにする．

式 (5.13) は，データストリームの送受信による処理量と信号の 1 送受信処理量 tr との比率 L_{md} と，1 転送ルートにデータストリームに中継ピア輻輳等により切換が発生する確率 γ の比率である $\frac{L_{md}}{\gamma}$ に依存して変化する．従って，

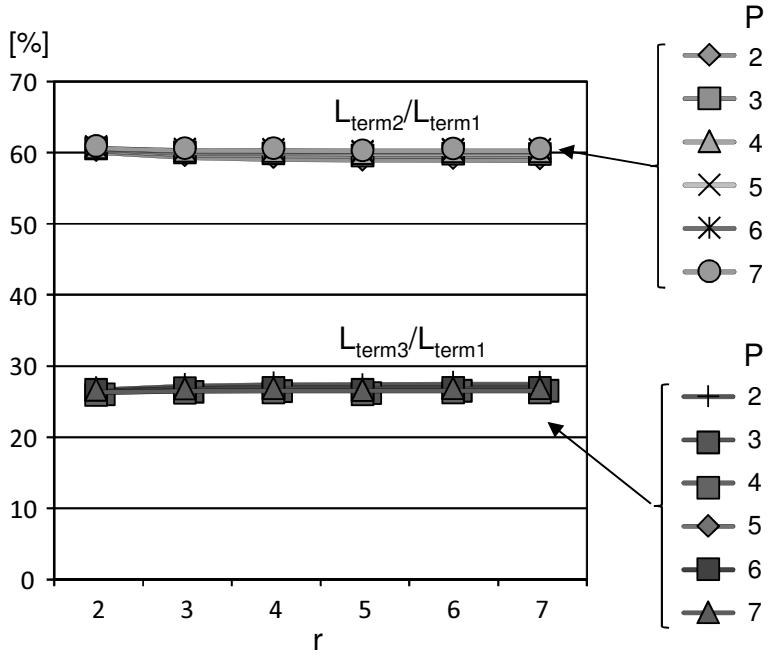


図 5.9　方式 2，3 の方式 1 との比率

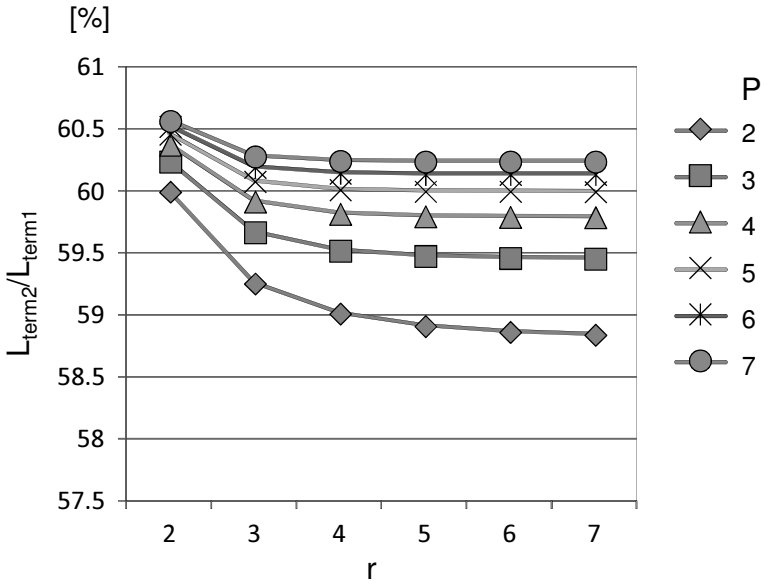
$\frac{L_{md}}{\gamma}$ により各方式を比較評価する．

今回は $\frac{L_{md}}{\gamma}$ を”0(上記，データストリームの送受信処理の影響がない場合)”,”1(例えば，$L_{md} = 1, \gamma = 1$)”,”4”,”10”,”20”,”40”,”60”,”80”の場合で算出し比較した．

$\frac{L_{md}}{\gamma}$ が”4(例えば，$L_{md} = 2, \gamma = 0.5$)”の場合の特性を，図 5.12，図 5.13，図 5.14 に示す．

図 5.12 から，データストリームの送受信による処理の影響があっても，方式 1 より方式 2 のほうがよく方式 3 がさらによい．しかし，その効果は小さくなっている．図 5.13 と図 5.14 の P に対する値から，方式 2 はデータストリームの送受信による処理の影響がない場合は P がより小さい値の場合によりよかったが，データストリームの送受信による処理の影響があると P が大きい値の場合によりよくなる．

さらにデータストリームの送受信による処理の影響の特性を明らかにするため，各 $\frac{L_{md}}{\gamma}$ の値における P と r によって異なる $\frac{L_{term2}}{L_{term1}}$ と $\frac{L_{term3}}{L_{term1}}$ の平均値を図 5.15 にドットで示す．$\frac{L_{md}}{\gamma}$ に対する傾向を理解しやすいように各ドットを結ぶ補助線も記載する．これは第 5.6 節にて行うシミュレーション実験結果と比較しやすくするためにも使用する．



r

図 5.10　方式 2 と方式 1 の比率

データストリームの送受信による処理負荷が大きくなると，方式 2，3 ともに従来方式に対する優位性は小さくなる．提案方式は，データストリームの送受信による処理負荷が小さい場合に有効である．

(8)SIP サーバの切換処理量比較

SIP サーバの処理量 L_{nwi} は，1 切換における SIP サーバの処理量 $L_{si}tr$ に P2P ネットワーク全体での切換発生回数 (式 (5.7)) をかけて求められる．また，SIP サーバはメディア処理を行わないためデータストリーム送受信処理量はかからない．従って，式 (5.14) で表される．

$$L_{nwi} = \frac{\beta\gamma(P+1)P^2(1-P^{r-1})}{2(1-P)} L_{si}tr \tag{5.14}$$

式 (5.14) より各方式の SIP サーバの処理量は $L_{si}tr$ に比例する．表 5.2 の $L_{si}tr$ の比率から，同じ P, r 値で L_{nwi} は下記比率となる．

$$L_{nw1} : L_{nw2} : L_{nw3} = 1 : 2 : 1$$

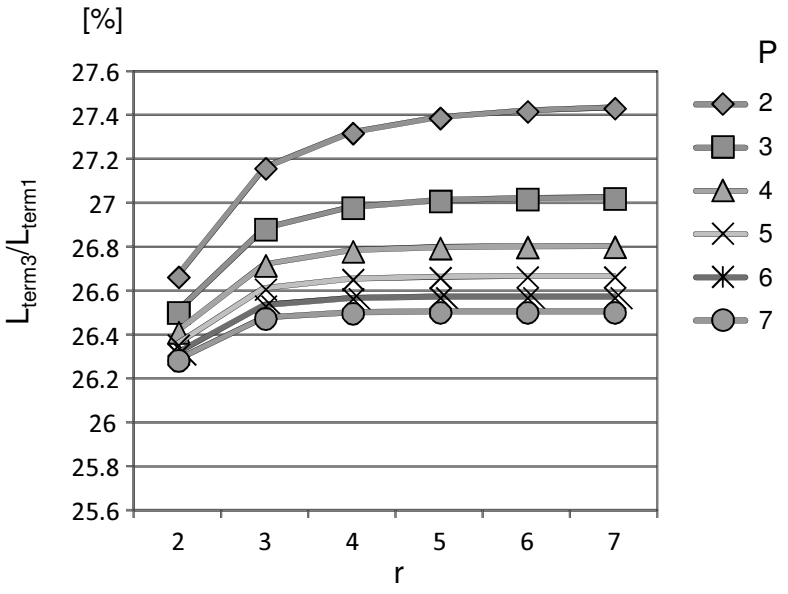


図 5.11　方式 3 と方式 1 の比率

(9) 全ピアの参加・離脱処理量比較

方式 1 と方式 2 は同じ参加・離脱の処理であり差がない．また，方式 3 はメディアブリッジを用いて SIP セッションを 2 本確立する信号処理だが，方式 1，2 も B2BUA のため，SIP セッションを 2 本確立する信号処理が必要である，従って，全ピアの参加・離脱処理量は 3 方式で差がない．

5.4.2　メモリやネットワーク帯域の静的容量

(1) メモリ等の記憶容量

各方式が消費するメモリ等の記憶容量を比較する．接続切換を制御する信号処理の視点では，SIP セッションの UA に必要な端末および SIP サーバのメモリ量があげられる (1UA のリソース量を *sipua* とする)．データストリームを通すメディアの視点では，端末およびメディアブリッジの RTP 終端点のメモリ量 (1 終端点のリソース量を *rtpterm* とする) があげられる．

1 端末間接続あたりの制御信号処理とメディアに必要なメモリリソース量は，方式 1(従来方式) と方式 2 は同じで，1 端末に *sipua* + *rtpterm*, ネットワーク設備 (SIP サーバ) に 2*sipua* である．方式 3 は図 5.5 に示すように 1 端末に *sipua* + *rtpterm*，ネットワーク設備 (SIP サーバ，メディアブリッジ) に

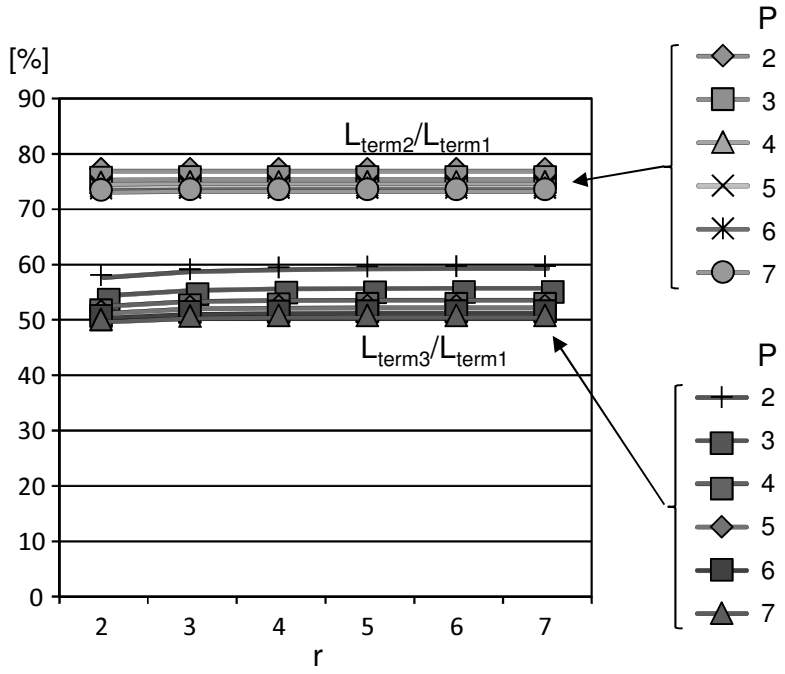


図 5.12　方式 2，3 の方式 1 との比率 ($L_{md} = 2, \gamma = 0.5$)

$2sipua + 2rtpterm$ であり，ネットワークリソースをより多く必要とする方式である．

(2) ネットワーク帯域

第 5.2.1 節に示すように，NGN は端末を収容するエッジルータを中継ルータで接続するツリー構造であり SIP サーバやメディアブリッジなどは地方ブロック単位に配置され中継ルータを介して接続される．簡単化のため，同一数の端末を収容する n 台のエッジルータが 1 台の中継ルータに接続する構成と仮定し，1RTP メディアの帯域を rtp とする．本 P2P 接続切換が継続的に実施されると端末間接続が NGN 全域へ均一に分散されると考えられる．

方式 3 では接続する両端末がエッジルータと中継ルータを介してメディアブリッジまで接続する必要がある．1 端末間接続は，端末からエッジルータまでの帯域 + エッジルータから中継ルータまでの帯域 + 中継ルータからメディアブリッジまでの帯域，図 5.5 に示すように 2 端末分必要で，$6rtp$ である．端末間接続 m 本あたり $m \times rtp \times 6$ を使用する．

方式 1(従来方式) と方式 2 は同じで，端末間接続 m 本のうち，$\frac{1}{n}$ がエッジルータ内で折り返し $m \times \frac{1}{n} \times 2 \times rtp$ を使用し，$\frac{n-1}{n}$ が中継ルータで折り返し

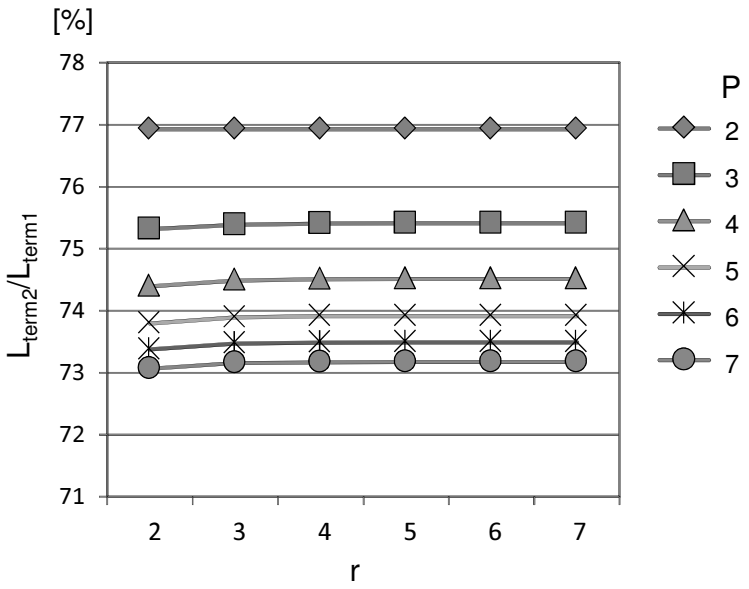


図 5.13　方式 2 と方式 1 の比率 ($L_{md} = 2, \gamma = 0.5$)

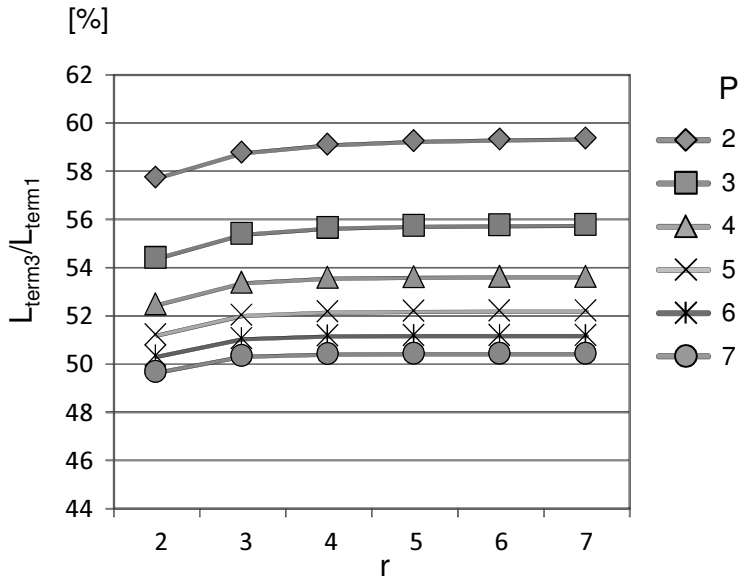


図 5.14　方式 3 と方式 1 の比率 ($L_{md} = 2, \gamma = 0.5$)

$m \times \frac{n-1}{n} \times 4 \times rtp$ を使用する．合わせて，$m \times rtp \times 2 \times (\frac{2n-1}{n})$ の帯域を使用する．

方式 3 は方式 1，2 より多くのネットワーク帯域を必要とする方式であり，方式 1，2 の $\frac{3n}{2n-1}$ 倍の帯域を使用する．

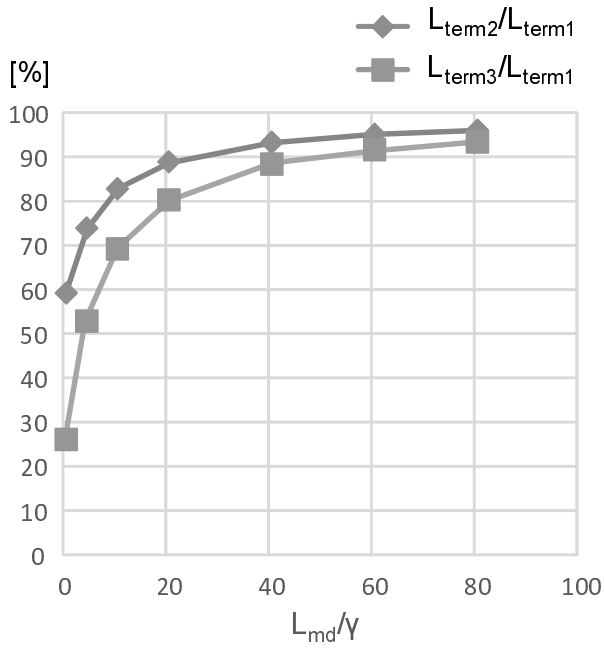


図 5.15　方式 2，3 と方式 1 との比率の平均と $\frac{L_{md}}{\gamma}$ の関係

(3) 方式 3 のメディア送受信数限界の評価

方式 3 はメディアブリッジを利用するため，同時接続メディア数に上限がある．主要な製品は固定チャネル数を収容可能であり，例えば 200 チャネル収容可能なメディアブリッジの場合は 1 メディアが 2 チャネル使うので同時接続メディアの上限が 100 となる．

全消費チャネル数は，中継ピアに発生するデータストリームに消費するチャネル数とリーフピアに発生するデータストリームに消費するチャネル数の和である．全中継ピア数に発生するデータストリーム数は 1 中継ピアにデータストリームが発生する確率である式 (5.3) に中継ピア数の式 (5.2) をかけた数である．中継ピアに発生するデータストリームでは 2 メディア使用し，4 チャネル使用する．全中継ピアに発生する消費チャネル数は式 (5.15) で表される．

$$\frac{2\beta(P+1)P^2(1-P^{r-1})}{(1-P)} \tag{5.15}$$

である．全リーフピア数に発生するデータストリーム数は 1 リーフピアにデータストリームが発生する確率 β にリーフピア数の式 (5.1) をかけた数であ

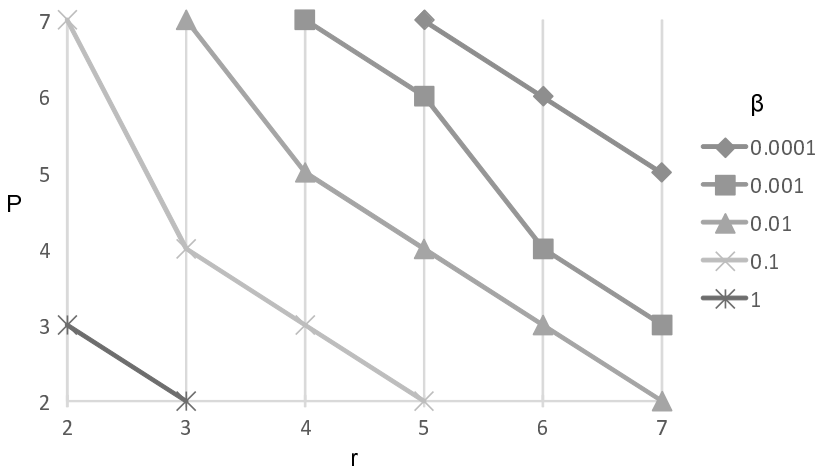


図 5.16　方式 3 のメディア送受信数限界の特性

る．リーフピアに発生するデータストリームでは 1 メディア使用し，2 チャネル使用する．全リーフピアに発生する消費チャネル数は式 (5.16) で表される．

$$4\beta P^r \tag{5.16}$$

従って，全消費チャネル数は，式 (5.15) と式 (5.16) の和であり式 (5.17) で表される．

$$2\beta P^2\left(\frac{(P+1)(1-P^{r-1})}{(1-P)} + 2P^{r-2}\right) \tag{5.17}$$

全消費チャネル数の上限を 200 とすると，式 (5.18) の制約がある．

$$2\beta P^2\left(\frac{(P+1)(1-P^{r-1})}{(1-P)} + 2P^{r-2}\right) \leq 200 \tag{5.18}$$

方式 3 のメディア送受信数限界により，P と r の上限がある．その特性を評価する．式 (5.18) に示すように，β によって上限がかわるので，P と r(, $2 \leq P \leq 7$，$2 \leq r \leq 7$) の 2 軸のグラフに，チャネル数 200 の上限を超えない点を β の値 ($0.0001 \leq \beta \leq 0.5$) を複数あげ，図 5.16 に示す．

5.4.3　切換処理失敗危険度

「評価の観点 3」切換失敗やストリームの遅延などは，限界負荷状態の中継ピアが切換処理に必要な全信号を送信・応答するために必要な時間に依存し，切換に必要な信号送受信数に比例すると考えられる．

中継ピアの処理負荷が高まり CPU 使用率が 100% 近くまで達した場合に切換るが，高負荷となった中継ピアが切換処理に必要な全信号をタイムアウト時間 (500ms 等) 内に応答できないと切換失敗が発生してしまう．限界負荷に達し切換が必要なピア B が，切換るために送受信しなければならない信号数は，表 5.2 に示すように，方式 1 : 方式 2 : 方式 3 = 5 : 2 : 2 である．従って，切換失敗する確率は従来の方式 1 に比べて提案方式の 2,3 ともに半分以下であり，切換失敗する危険度が少なく，ユーザビリティが高い方式であると言える．

また，方式 3 は図 5.5 に示すように端末間は中継ルータを介して接続されるメディアブリッジが各端末とのメディアを終端するため，方式 3 のメディアは方式 1，2 と違い 2 本のメディアを中継し距離が長くなる．一方，NGN は SIP のセッション接続した通信は帯域幅と品質（損失，遅延，揺らぎ）を保障するよう通信設備が設計されている．従って，メディアの中継や距離による遅延はユーザビリティに影響は少ない（NGN でも SIP を使わないベストエフォート型の通信はトラヒック状況により品質に影響がでる）．

5.4.4　各方式の比較結果

各観点からの比較評価結果をまとめ，各方式の特性を示す．P2P ネットワークの適用先の要件 (規模，データストリーム転送処理量，等) に従い方式 2 または 3 を選択することとなる．

- 「評価の観点 1」の接続切換を制御する信号処理の計算処理量だけを見ると，図 5.9 と図 5.12 から，既存の方式 1 より提案方式である方式 2 のほうがよく，方式 3 がさらによい．
- 「評価の観点 2」では，図 5.15 に示したように，データストリームの送受信による処理負荷が大きくなると，方式 2,3 ともに従来方式に対する優位性は小さくなる．提案方式は，データストリームの送受信による処理負荷が小さい場合に有効である．
- 「評価の観点 1」の P に対しては，図 5.10 と図 5.11 及び図 5.13 と図 5.14 から，方式 3 はより大きな値の場合によりよい．方式 2 はデータストリームの送受信による処理の影響がない場合はより小さい値の場合によりよく，データストリームの送受信による処理の影響があると大きい値の場合によりよくなる．

- 「評価の観点 1」の r に対しては，図 5.10 と図 5.11 及び図 5.13 と図 5.14 から，方式 2 はより大きい値の場合によりよく，方式 3 はより小さい値の場合によりよい．
- 第 5.4.1 節に示したように，SIP サーバの切換処理量は，逆に方式 2 は方式 1 より処理量が多い．一方で方式 3 は方式 1 と同等である．端末，SIP サーバともに計算処理量は方式 3 が小さくよい．
- 第 5.4.1 節に示したように，全ビアの参加・離脱の処理量は 3 方式で差がない．
- 第 5.4.2 節，第 5.4.2 節に示したように，メモリやネットワーク帯域の静的容量は方式 3 が多くのネットワークリソースを消費する．
- ブリッジできるメディア数の上限から，各ピアのメディア確立頻度によるが，方式 3 には第 5.4.2 節に示したように，接続できるピア数に条件があり，その範囲内に収まる小さな P2P ネットワークのみが利用できる方式である．
- 第 5.4.3 節に示したように，「評価の観点 3」の接続失敗危険度は方式 1 より方式 2，3 が半分以下に改善されると考えられる．方式 3 のメディアは方式 1，2 と違い 2 本のメディアを中継し距離が長くなり不利であるが，NGN の通信設備は通信品質を保証しておりユーザビリティの低下は少ない．

5.5　ソフトウェアアーキテクチャ

これら提案手法を実装した，端末とネットワーク側サーバ内ソフトウェアアーキテクチャについて，全体像とセッション制御 ANI 利用方式 (方式 2) 実行時のソフトウェアコンポーネント間連動動作を図 5.17 に示した．

ネットワーク内の SIP サーバでは，通常の SIP プロトコルによるセッション接続処理を受け付け，B2BUA のセッション間接続制御を行う呼処理が必要である．今回，それに加えてセッション制御 ANI を介してセッション接続処理を受け付け実行する必要がある．同じ呼処理を行う部分を Call control コンポーネントとして共通化しサービスによって異なる差分をシナリオとして選択実行可能とし，SIP，HTTP，RTP 等のプロトコル処理部分を共通部品化するソフトウェア構成とした．機能部品の共用化とサービス機能追加を簡便で個別

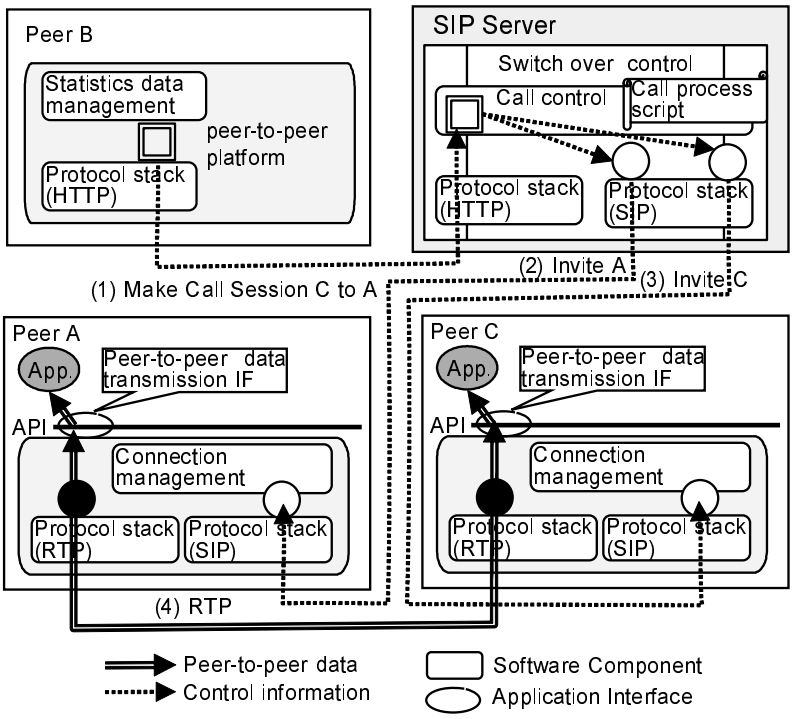


図 5.17　ソフトウェア構造

に行えることにより，ソフトウェア開発を高効率化するとともに，サービス間の処理競合による性能の劣化を回避する．

本ソフトウェア構造では，各端末間の接続・切換制御機能，及び，接続方式の選択動作はネットワーク内の Switch over control 機能のシナリオで実現できる．

各端末では，常時 peer-to-peer platform コンポーネント内の Statistics data management コンポーネントが CPU 使用率とアプリケーションレベルの通信バッファキュー長等の統計情報を収集している．CPU 端末 B が輻輳を予見した時，Statistics data management コンポーネントは peer-to-peer platform コンポーネントの P2P ネットワークにおける端末 A,B 間の接続に代わって A,C 間を接続する処理を起動する．

(1) peer-to-peer platform コンポーネントは Protocol stack (HTTP) コンポーネントを使って NGN 内 SIP Server のセッション制御インタフェースの Parlay-X “Make Call Session”をコールする．“Make Call Session”を受けた Switch over control コンポーネント内の Call control コンポーネ

ントは Call process script に記載されたシナリオに従い端末 A と端末 C へセッションを張る処理を行う．

(2) Protocol stack (SIP) コンポーネントを介して端末 A へ SIP の Invite 信号を送る．

(3) Protocol stack (SIP) コンポーネントを介して端末 C へ SIP の Invite 信号を送る．peer-to-peer platform コンポーネント内の Connection management コンポーネントが受けて両端末にそれぞれ SIP セッションを張る．この段階で SIP Server の Call control コンポーネントでは SIP の B2BUA 方式を用いて 2 本のセッションをつないでいる．

(4) SIP のセッションが Session Description Protocol (SDP) で示す RTP が Protocol stack (RTP) コンポーネントを介して端末 A,C 間に接続されメディアが確立される．

5.6　シミュレーション実験による特性検証

第 5.4 節での各方式の特性は P2P ネットワークの理想的なトポロジーでの定常状態を分析して求めたが，P2P ネットワークは各端末の動的な参加・離脱や本切換処理により動的にトポロジーが変化する (各リーフ端末の半径はそれぞれ異なる)．また，データストリームを中継するピアが輻輳しても切換で接続数が増えるピアがすでに接続数の上限まで接続していた場合，切換できない場合がある．各端末の動作状況を Excel VBA を用いてシミュレートし，解析結果と同じ特性が得られるか検証した．

5.6.1　シミュレーション動作

初期状態として，全ピア順次参加していく．より上流に位置するピアに接続する．接続受付数の上限に達していると下流に位置するピアに接続する．

(1) ピアの離脱と参加

あるピアが離脱する際は，離脱したピアの一段下流の子ピアが切り離され，子ピア自身の下流の子ピア (孫ピア) との接続を維持したまま，改めて P2P ネットワークに参加する．

P2P ネットワークに参加する際は，P2P ネットワークの初期構築と同様に，より上流に位置するピアに接続する．接続受付数の上限に達していると下流に位置するピアに接続する．

(2) 切換動作

1 転送ルートにデータストリームが発生する確率 β は，全転送ルートに対して一定で単位時間あたり 1 データストリームをランダム選択された転送ルートに発生させる．

データストリームが中継ピア輻輳等により切換を確率 γ で発生させる．ただし，輻輳が発生した転送ルートのピアで切換によって接続数が増えるピアの接続数が上限に達して切換できないケースがあり，その現象をシミュレートする．

切換により，P2P ネットワークのトポロジーはバランスドツリーではなくなり，中継ピア数とリーフピア数の比率も変わる現象をシミュレートする．

切換時に中継ピア B，A，C に発生する，テータストリーム転送負荷と，各切換方式の切換処理負荷を測定し，集計する．

5.6.2　シミュレーション条件

各端末が許容する接続数 (分岐数) が P で P2P ネットワークトポロジーがバランスドツリー (平衡木) の場合に中心からリーフピアまでのホップ数 (次数) が r で，$(2 \leq P \leq 7), (2 \leq r \leq 7)$ の範囲の端末数で実験した．

データストリームの送受信処理と信号の送受信処理が別プロセッサで実行されるアーキテクチャの場合を想定し，データストリーム送受信負荷が信号処理に影響与えない場合の「評価の観点 1」の実験を行った．

あるデータストリームが中継ピア輻輳等により切換が発生する確率 γ とデータストリームの送受信による処理量が 1 信号送受信処理量 tr の何倍か L_{md} を $\frac{L_{md}}{\gamma}$ を”0(上記，データストリームの送受信処理の影響がない場合)”,”1(例えば，$L_{md} = 1, \gamma = 1$)”,”4”,”10”,”20”,”40”,”60”,”80”の場合を測定した．これにより「評価の観点 2」データストリームの送受信による処理量が信号処理の方式に与える影響を明らかにする．許容する接続数 (分岐数)P，初期半径 r により違いはあるが，代表値として，各端末が許容する接続数 (分岐数)$P = 4$，初期半径 $r = 4$ のケースを実験した．

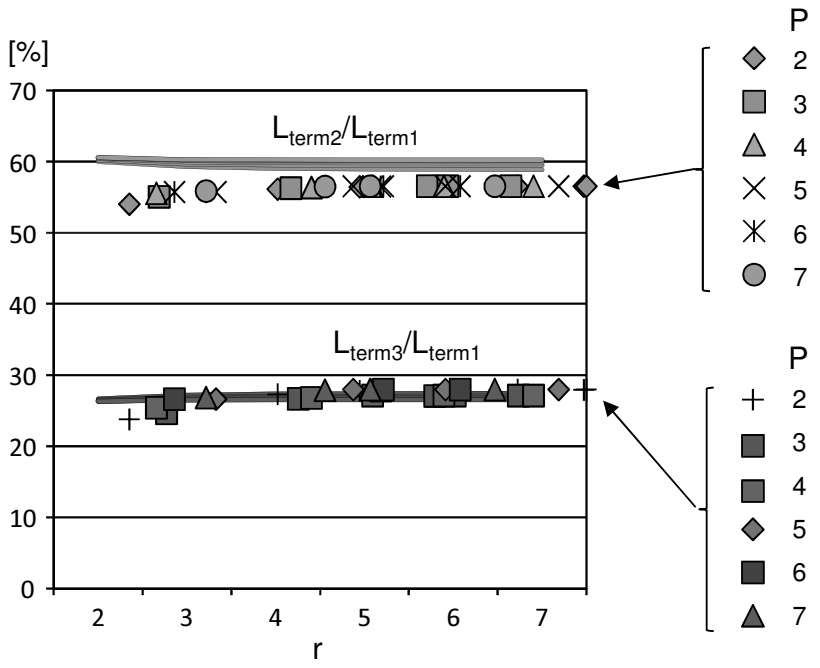


図 5.18　方式 2，3 の方式 1 との比率

P2P ネットワークトポロジーが変化の過渡状態ではなく，十分な切換動作回数を発生させて観測する．総ピア数以上の切換回数を発生させ，特にピア数が少ない場合，切換数が少なく P2P ネットワーク構造の変化が過渡的で偏りが起こる可能性があるので，ピア数が 500 未満の場合は，500 回の切換回数を発生させる．

5.6.3　シミュレーション結果と考察

(1) 評価の観点 1

P と *r* の組み合わせによる各方式における全ピアの処理量の実験結果を，図 5.9 を実線で表した上にドットで示す (図 5.18)．横軸の *r* 値は，実験によって構成された実験終了時の P2P ネットワークにおけるリーフピアまでのピア間ホップ数 (次数) を測定し，その平均値の *r* を求めプロットした．

P2P ネットワークトポロジーの偏りの影響で解析結果グラフと完全一致はしないが，変動範囲を考慮しても方式 2，3(特に方式 3) が方式 1(従来方式) より効率がよいことが確認できた．

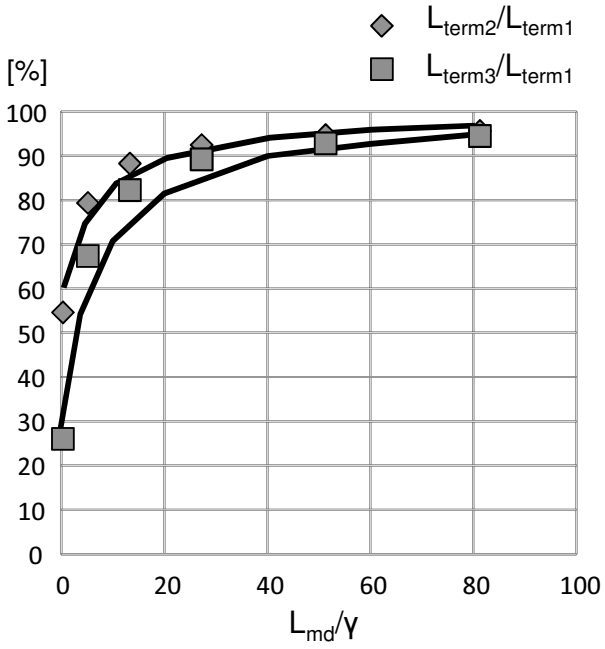


図 5.19　方式 2，3 と方式 1 との比率の平均と $\frac{L_{md}}{\gamma}$ の関係

(2) 評価の観点 2

データストリームの送受信による処理の影響の特性を確認する．各 $\frac{L_{md}}{\gamma}$ の値における，$\frac{L_{term2}}{L_{term1}}$ と $\frac{L_{term3}}{L_{term1}}$ の実験結果を図 5.19 に示す．図 5.15 に示したデータの補助線を図 5.19 に転写し，その上に実験結果値をドットで示した．実験において全データストリーム発生数と切換発生数を測定し，切換が発生する確率 γ を求め，横軸の $\frac{L_{md}}{\gamma}$ の値としてプロットした．

図 5.19 に示した通り，プロットした全実験値の解析結果データ補助線との差は 0～10% であり，解析結果に近い実験結果となった．方式 2，3 は方式 1 よりもよいが，データストリーム送受信処理量が多くなると優位性は小さくなる傾向を確認した．

(3) 評価の観点 3

シミュレーションによって，端末間の接続が NGN のトポロジー上は全エッジルータにほぼ均一に分散されることを確認した．それにより，方式 1，2 でも，ほとんどの端末間のメディアが中継ルータを介して接続される．その結果，必ず中継ルータを介してメディアを接続する方式 3 のストリームの遅延な

どのユーザビリティの不利な点が大きな差ではないことを確認した．

5.7　実装フィージビリティ検証

セッション切換，メディア切換サーバソフトウェアの実装フィージビリティを確認するために，ソフトウェア実装・性能評価を行った．

商用 IP 電話サービス用の SIP サーバソフトウェアをベースに ANI 公開機能を追加試作しラボ内実験環境で検証した．

利用を想定する商用ネットワーク環境は，NGN における SIP ベースのセッションで端末間をつなぎ構成された P2P ネットワークを対象とし，日本国内全域に広がる規模で複数の NGN を跨る接続があると想定した．接続切換の制御を行う SIP サーバは汎用 IA サーバが数台 (2 台程度) で映像等のデータストリームの送受信に耐えられる PC 相当の端末が数十万台を収容し，接続の管理と接続切換制御を可能とすることを想定した．P2P ネットワークに接続する処理，および，接続を切換る処理のトラヒックは，数十万 BHCA(1 時間当たりの最大負荷，サーバ CPU 使用率 30% 以内) を想定した．メディアブリッジは 1 台あたり数百メディアを収容可能な装置を数台 (需要による) と想定した．

ラボ内実験環境では商用 NGN と同一機種を用いてネットワークを構成し，SIP サーバ 2 台に商用で想定する端末台数とトラヒックを SIP 疑似呼装置で負荷をかけて検証した．

1SIP サーバあたり，ユーザ数 10 万，処理能力 30 万 Busy Hour Call Attempts (BHCA)，同時接続数 2 万本の性能を確認した．負荷をかけつつ 1 週間以上の連続運転検証をクリア (異常発生せず) した．このことから，実装ソフトウェアのフィージビリティが確認できたといえる．

ANI 提供機能分のメモリ消費が追加となるが，端末発呼の通常呼処理とプロトコル処理機能と呼制御機能の共用しつつ ANI 提供機能の局在化 (処理競合なし) が実現できた．ANI 提供機能追加による通常呼処理の性能劣化は見られなかった．

5.8　まとめ

P2P ネットワークの各端末の自律動作による発展性を活かしつつ端末が主導しネットワーク機能がサポートすることによるリソース使用の効率化を狙い，

P2P ネットワークトポロジー制御機能を提案した．これは P2P ネットワークのトポロジーを構成する端末間のリンクの切換機能がアプリケーションとして ANI 機能として提供されるネットワークのセッション制御機能を利用する．

物理ネットワークの端末間接続制御機能にはいくつか方式が考えられ，各方式の特性を使用するネットワーク内設備および端末リソースの観点から比較分析した．

各方式比較結果をシミュレーションによって検証し確認した．また，実装ソフトウェアアーキテクチャのフィージビリティを試作検証によって確認した．

第 6 章

おわりに

本章では，成果のまとめを述べ，今後の課題について述べる．

6.1　成果

高品質／高信頼を要求される大規模通信システムにおいて開発時の試験項目の「数」を調整する事で，開発時不具合発見数を増加させ公衆網適用時の不具合発生数を減少させる手法と，機械学習を用い不均質な試験項目作成を避け適切な試験項目作成を実現する手法について提案し効果を明らかにした．また，大規模通信システムを利用した P2P 通信サービスを実現する際に，端末やネットワークに負担が少ない P2P ネットワークトポロジー構成変更方式について提案し効果を明らかにした．

第 1 章では，本論文の背景と課題，目的を明確に示し，第 2 章では，研究対象である NGN の概要，また，NGN システムを利用した P2P ネットワークサービスにおける相互接続の概要を述べ，関連するシステム開発手法及び P2P ネットワークサービス技術の研究・開発を紹介した．

第 3 章以降に本論文の課題を解決する提案とその有効性を示した．提案内容と有効性をまとめ，以下に示す．

6.1.1　高品質なソフトウェア開発の実現

一つ目の課題である高品質なソフトウェア開発を実現する成果を以下にまとめる．

(1) 開発工数（試験工数）を増やすことなく品質の向上を実現

1) 第 3 章に，公衆網運用中に発生する不具合対処毎に増加する稼働削減への効果も期待できる不具合の「数」に着目し，開発時の試験項目の「数」を調整する事で，開発時不具合発見数を増加させ，公衆網適用時の不具合発生数を減少させる手法についてを提案した．

2) 提案方式は，試験工程で機能部毎の試験項目数を調整する事でシステム全体としての試験項目数を変えない，すなわち開発工数を増加させない試験項目選定手法である．

3) 提案方式をキャリアネットワークである NGN のシステム開発に適用し，公衆網適用時の不具合発生数を減少できたことを確認した．

4) 本システム開発技術は，大規模通信システムのみならず，一般の大規模システムにおいても高品質なソフトウェアを実現しつつ工数の増加を抑止できる可能性を持つ．

(2) 品質を維持したまま，開発工数の低減を実現

1) 第 4 章に，開発のための要求仕様書と試験項目を教師データとし，試験項目作成作業者のスキルやノウハウに依存しない均質な試験項目をローコストで作成するための試験項目作成自動化について提案した．

2) 提案方式は，これまでの開発における試験項目を参考に，試験項目箇所にマーキングされた要求仕様書を大量に作成し，マーキングされた大量のドキュメントを教師データとして学習機にて学習させた後，学習データを基にマーキングされた要求仕様書から試験項目を自動的に作成する方式である．

3) 提案手法によって，自然言語で書かれた非構造化要求仕様書から，システム開発時の試験工程のための試験項目を自動的に生成する事が可能となった．高度な技能を有する開発者によって，要求仕様書の試験項目に対応する記載箇所にタグが付与される．このタグ付きデータを用いて機械学習を行った．その後，機械学習の結果を，タグ付けされていない新しい要件仕様書に対し，試験項目を指し示すタグを自動的に付加できることを確認した．

4) 本システム開発技術は，大規模通信システムにおける均質な試験項目抽

出のみならず，一般システムの場合でも要求仕様書にタグ付けした教師データを用い学習する事により均質な試験項目抽出ができる可能性を持つ．

6.1.2　[課題 2]P2P 通信サービス提供におけるより効率のよい P2P ネットワークトポロジを構成の実現

二つ目の課題である P2P 通信サービス提供におけるより効率のよい P2P ネットワークトポロジを構成する成果を以下にまとめる．

1) 第 5 章に，P2P ネットワークの各端末の自律動作による発展性を活かしつつ端末が主導しネットワーク機能がサポートすることによるリソース使用の効率化を狙い，P2P ネットワークトポロジー制御機能を提案した．
2) 提案方式は，P2P ネットワークのトポロジーを構成する端末間のリンクの切換機能がアプリケーションとして ANI 機能として提供されるネットワークのセッション制御機能を利用する方式である．
3) 物理ネットワークの端末間接続制御機能にはいくつか方式が考えられるため，各方式の特性を使用するネットワーク内設備および端末リソースの観点から比較分析し，各方式比較結果をシミュレーションによって検証し確認した．また，実装ソフトウェアアーキテクチャのフィージビリティを試作検証によって確認した．
4) 本技術は，IoT の進展に伴うハンドヘルドを含めたデバイス相互の大容量且つ信頼性の高い通信に適用できる可能性を持つ．

6.2　今後の課題

信頼性や安全性の保証や社会的な依存性を保証しなくてはならない基礎的なインフラである大規模通信システムの開発において，コスト（開発工数）を増やすことなくソフトウェア品質を高め手法を発展させることで，将来の多様なネットワークサービスを持続的に提供できる環境の実現に貢献していく．また，P2P サービスの適用を基礎にした数多くのサービスをタイムリーに提供で

きる環境の実現に貢献していく．

(1) 高品質なソフトウェア開発に向けて：不具合の「質」への拡張

重要不具合の減少傾向から，開発を重ねる事による不具合の減少も想定されたため，提案手法の明確な定量評価に十分繋がっているとは言い難い部分も想定される．このため，今後は，他の複数の追加開発においても提案手法の評価を行い適正な値の導出やソフト構造による違いを明らかにする共に，重要不具合の定義にあるような「質」を考慮した開発時不具合発見を増やす手法を検討し，手法の自動化を進める事で効果／効率的な開発の実現をめざしていく．

(2) 高品質なソフトウェア開発に向けて：より高精度な「試験項目自動作成」への拡張

今後はさらに，より多くの教師データを作成し，機械学習へ適用させていくとともに，タグ付けドキュメントから試験項目を自動抽出する精度を高めていく．

(3)P2P サービスの拡大に向けて：動的なトポロジー制御への拡張

ネットワークトポロジー制御のため提案した 3 つの方式の比較分析から得られたメリット・デメリットを鑑み，端末，サーバ，ネットワークのリソースの状況に応じて最適な方式を動的に決定する方法を検討する．各切換方式と動的方式決定機能を実現する，端末とネットワーク側のサーバが連携した実装ソフトウェアアーキテクチャを明らかにしていく．

謝辞

本研究を進めるにあたり，終始暖かい御助言と貴重なる御指導を賜り，また，多くの御支援を頂戴いたしました九州大学大学院システム情報科学研究院 福田晃教授に深く感謝の意を表します．

また，本論文をまとめるにあたり，貴重なる御助言と御指導を賜りました九州大学大学院システム情報科学研究院 峯 恒憲准教授ならびに，久住憲嗣准教授に心より感謝の意を表します

研究の意義や進め方，研究を進める姿勢，貴重なる多くの御助言と御指導を賜りました日本大学工学部情報工学科 上田清志教授に心より感謝の意を表します．

本論文をまとめる機会を与えていただいた NTT ネットワークサービスシステム研究所 岡村浩之プロジェクトマネージャに深く感謝の意を表します．

研究着手当初から研究の意義や進め方等をお教えいただいた大阪工業大学 須永宏教授 に深く感謝の意を表します．

P2P ネットワークトポロジ制御の研究，実験に協力してくれた NTT ネットワークサービスシステム研究所 岩田哲弥氏に感謝いたします．

P2P サービスプラットフォームの研究，実験に協力してくれた NTT ネットワークサービスシステム研究所 松村裕亮氏，大石哲矢氏，NTT 東日本 酒井孝次氏に感謝いたします．

NGN のネットワークシステム開発手法を共に検討してくれた NTT 東日本 篠田隆弘氏，NTT ネットワークサービスシステム研究所 山田剛史氏に感謝いたします．

最後に，私を支えてくれる妻 有希子，子 優希，勇宏，美津希，両親，弟に感謝します．

参考文献

[1] Cisco Systems Inc., “Cisco Visual Networking Index: Forecast and Methodology, 2015-2020,” http://www.cisco.com/web/JP/solution/isp/ipngn/literature/white_paper_c11-481360.html, May 2015.

[2] 河合，篠田，入江，守屋，関，福田，菊間，”大規模ソフトウェア開発における上流工程の自動化検討,”電子情報通信学会総合大会講演論文集，通信 (2)，p.13.，2016

[3] 篠田，菊間, ”大規模ネットワークシステム開発の品質維持手法に関する一考察,”電子情報通信学会総合大会講演論文集，通信 (2)，p.13，2016

[4] rogrammableWeb.com, “programmableWeb,” http://www.programmableweb.com/api/, 参照 Aug. 2016.

[5] Twilio Inc., “Twilio,” http://www.twilio.com/, 参照 Aug. 2016.

[6] TTC 仕様書, “Open Services Access (OSA); Parlay X Web Services; Part 1:Common,” http://www.ttc.or.jp/jp/document_list/free/3gpps2010/TS /TS-3GA-29.199-01(Rel9)v9.0.0.pdf, 情報通信技術委員会, July 2010.

[7] IETF, “SDP: Session Description Protocol,” https://tools.ietf.org/html/rfc4566, IETF, July 2006.

[8] G. Camarillo, M. A. Martin, “IMS 標準テキスト,” リックテレコム, Mar. 2010.

[9] ITU-T Recommendation Y.2012, “NGN アーキテクチャの概要 (Overview of the NGN architecture),” http://www.ttc.or.jp/jp/document_list/pdf/j/TR/TR-1014v1.pdf, TR-1014, TTC 技術レポート, 情報通信技術委員会, Jun 2006.

[10] NTT 東日本, “NTT 東日本フレッツサービス,” http://flets.com/

[11] NTT 西日本, “NTT 西日本フレッツサービス,” http://flets-w.com/

[12] K. Kikuma, M. Asamura, and T. Murakami. ”Congestion Control in Broadband Common channel Signalling Network,” Proc. of The 13th International Conference on Information Networking(ICOIN13),pp.11D-3.1-3.4 , Jan. 1999.

[13] 誉田，山田,”同種バグの摘出を可能とするバグ分析と 1+ ｎ施策,” プロジェクトマネジメント学会 2011 年度秋季研究発表会予稿集, 2011, pp.157-162.

[14] 平山, 他,”ソフトウェアエンジニアリング ソフトウェア品質向上への取組み,” 東芝レビュー Vol.56 No.11 Page.47-55, Nov. 2001.

[15] 坂井，他,”大規模分散処理システムのソフトウェア試験とその実践,”情報処理学会デジタルプラクティス Vol.4 No.1 Page.51-59, Jan. 2013.

[16] (独) 情報処理推進機構ソフトウェア・エンジニアリング・センター (IPA-SEC),”ソフトウェア開発データ白書 (2014-2015),” Dec. 2014.

[17] (独) 情報処理推進機構ソフトウェア・エンジニアリング・センター (IPA-SEC),”定量的品質管理とその実践的取組み,” Aug. 2012.

[18] P. Godefroid, N. Klarlund, and K. Sen.: DART: Directed Automated Random Testing. Proceedings of the 2005 ACM SIGPLAN Conference on Programing Language Design and Implementation (PLADI2005), pp.213-223 (2005)

[19] M. Khandai, A. A. Acharya, and D. P. Mohapatra.: A Survey on Test Case Generation from UML Model. International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 2 (3), pp.1164-1171 (2011)

[20] S. Masuda, N. Hosokawa., F. Iwama, T. Matsuodani, and K. Tsuda.: Semantic Analysis Technique of Logics Retrieval for Software Testing from Specification Documents. IEEE Software Testing, Verification and Validation Workshops (ICSTW), pp.1-6 (2015)

[21] S. Masuda, T. Matsuodani, and K. Tsuda.: Detecting Logical Inconsistencies by Clustering Technique in Natural Language Requirements. IEICE Transactions on Information and Systems Vol.E99-D No.9, pp.2010-2018 (2016)

[22] ISO/IEC/IEEE JTC 1/SC7: Software and system engineering - software testing - part 4 Test techniques. ISO/IEC/IEEE JTC 1/SC 7. pp.70-72 (2015)

[23] PPS, “PPStream,” http://www.ppstream.com, 参照 Aug. 2016.

[24] PPLive Inc., “PPLive,” http://www.pplive.com/en/index.html, 参照 Aug. 2016.

[25] WebRTC initiative,“WebRTC,”https://webrtc.org/, 参照 Aug. 2016.

[26] Mist Technologies,“MistCDN,”https://www.mist-t.co.jp/mistcdn, 参照 Aug. 2016.

[27] BitTorrent, “BitTorrent Live,” https://btlive.tv/, May 2016.

[28] 3GPP, “Study on IMS based peer-to-peer content distribution services,” http://www.etsi.org/deliver/etsi_tr/122900_122999/122906/13.00.00_60 /tr_122906v130000p.pdf, 3GPP TR 22.906 V13.0.0 version 13.0.0 Release 13, April 2016.

[29] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan “Chord: a scalable peer-to-peer lookup protocol for internet applications,” IEEE/ACM Transactions on Networking, Vol. 11, pp.17-32, Feb. 2003.

[30] D. A. Bryan, B. B. Lowekamp, and C. Jennings, ”SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System”, Proceedings of the 2005 International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA) ’05, June 2005.

[31] X. Jin, W.-P.K. Yiu, S.-H.G. Chan, and Y. Wang,”On maximizing tree bandwidth for topology-aware peer-to-peer streaming,”IEEE Transactions on Multimedia, vol.9, no.8, pp.1580-1592, Dec. 2007.

[32] 総務省, “情報通信白書 平成 28 年度”http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/pdf/index.html, Aug 2016.

[33] Cisco Systems Inc., “Cisco Visual Networking Index: Forecast and Methodology, 2015-2020,”http://www.cisco.com/web/JP/solution/isp/ipngn/literature

[34] 総務省,“電気通信サービスに係る内外価格差調査”http://www.soumu.go.jp/menu_news/s-news/01kiban03_02000315.html, Jul 2015.

[35] Mozilla Foundation, “Bugzilla,” http://bugzilla.org/

[36] Redmine using the Ruby on Rails framework, “Overview - Redmine,” http://www.redmine.org/

[37] 独立行政法人情報処理推進機構（ＩＰＡ）, “ソフトウェア開発データ白

書 2016 − 2017,” Oct. 2016.

[38] Campwood Software,“SourceMonitor v3.5,”http://www.campwoodsw.com/sourcemonitor.html

[39] Ministry of Internal Affairs and Communications, Japan: White Paper on Information and Communications in Japan. http://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2016/2016-index.html, Economic Research Office, ICT Strategy Policy Division, Global ICT Strategy Bureau, Ministry of Internal Affairs and Communications, Japan (2017)

[40] Cisco Systems Inc.: Cisco Visual Networking Index: Forecast and Methodology, 2015-2020. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf (2016)

[41] ITU-T Recommendation Y.2012: Functional requirements and architecture of the NGN release 1. Telecommunication Standardization Sector of ITU https://www.itu.int/rec/T-REC-Y.2012-200609-S/en (2007)

[42] B. W. Boehm: Guidelines for Verifying and Validating Software Requirements and Design Specifications. Proc. EURO IFIP 79, London, pp. 711-719 (1979)

[43] Bundesrepublok Deutschland: V-Model XT, Version 1.1.0. http://ftp.uni-kl.de/pub/v-modell-xt/Release-1.1-eng/Dokumentation/pdf/V-Modell-XT-eng-Teil1.pdf (2004)

[44] Mecab: Yet Another Part-of-Speech and Morphological Analyzer. https://github.com/jordwest/mecab-docs-en (2013)

[45] CRF++: Yet Another CRF toolkit. http://taku910.github.io/crfpp/\#source (2013)

[46] G. Camarillo, M. A. Martin, “IMS 標準テキスト,” リックテレコム, Mar. 2010.

[47] ITU-T Recommendation Y.2012,“NGN アーキテクチャの概要 (Overview of the NGN architecture),”http://www.ttc.or.jp/jp/document_list/pdf/j/TR/TR-1014v1.pdf, TR-1014, TTC 技術レポート, 情報通信技術委員会, Jun 2006.

[48] IETF, “SIP: Session Initiation Protocol,” https://tools.ietf.org/

html/rfc3261, IETF, June 2002.

[49] T. Oh-ishi, K. Sakai, K. Kikuma, and A. Kurokawa, ”Study of the Relationship between Peer-to-Peer Systems and IP Multicasting,” IEEE Communications Magagine, vol41(1), pp80-84, Jan. 2003.

[50] M. Yamada, R. Ono, K. Kikuma, and H. Sunaga, ”A study on P2P platforms for rapid application development,” The 9th Asia-Pacific Conference on Communications(APCC 2003), pp368-372, Mar. 2003.

[51] K. Kikuma, Y. Morita, and H. Sunaga, ”A Study of a P2P community on a P2P communication platform”, International Conference on Communication Technology 2003(ICCT2003),pp.153-156, Apr. 2003.

[52] H. Sunaga, K. Ueda, T. Iwata, K. Kikuma, and M. Takemoto, ”P2P applications using the semantic information oriented network,” Fourth International Conference on Peer-to-Peer Computing, pp. 272-273, Aug. 2004.

[53] 上田, 須永, 中田, 福田, ”物理ネットワーク負荷に適応した P2P ネットワークトポロジー制御,” 電子情報通信学会論文誌 B, Vol.J92-B No.11 pp.1750-1763, Nov. 2009.

[54] IETF,“SIP Refer Method,”http://www.ietf.org/rfc/rfc3515.txt, IETF, April 2003.

[55] IETF,“3PCC in SIP,”http://www.ietf.org/rfc/rfc3725.txt, IETF, April 2004.

略語表

3PCC: 3rd Party Call Control

ANI: Application Network Interface

API: Application Program Interface

AS: Application Support Functions & Service Support Functions

B2BUA: Back TO Back User Agent

BHCA: Busy Hour Call Attempts

BTS: Bug Tracking System

CDN: Content Delivery Network

CPU: Central Processing Unit

CRF: Conditional Random Field

DHT: Distributed Hash Table

FB: Function Block

FTTH: Fiber To The Home

HTTP: HyperText Transfer Protocol

ID: InDex

IEEE: The Institute of Electrical and Electronics Engineers

IETF: The Internet Engineering Task Force

IF: InterFace

IoT: Internet of Things

IMS: IP Multimedia Subsystem

M2M: Machine to Machine

MBST: Maximum Bandwidth Sum Tree

MSMT: Minimum Stress Multicast Tree

NFV: Network Functions Virtualization

IP: Internet Protocol

ITU-T: International Telecommunication Union Telecommunication Standardization Sector

LOC: lines of code

NGN: Next-Generation Network

P2P: Peer-to-Peer

P2PSIP: Peer-to-Peer Session Initiation Protocol

P2PTV: Peer-to-Peer TV

PC: Personal Computer

PSTN: Public Switched Telephone Network

RTP: Real-time Transport Protocol

RTT: Round Trip Time

SDN: Software-Defined Network

SDP: Session Description Protocol

SIP: Session Initiation Protocol

SLOC: source lines of code

UA: User Agent

UML: Unified Modeling Language

UNI: User-Network Interface

WebRTC: Web Real-Time Communication

用語

公衆交換電話網 (Public Switched Telephone Network:PSTN)

通信事業者の施設から各加入者宅まで通信回線を引き込み，回線交換方式の音声通話サービスを提供する従来の公衆回線網のこと．「公衆網」，「公衆回線網」，「公衆電話網」または「公衆電話交換網」などとも表記する．

ファイバートゥザホーム (Fiber To The Home :FTTH)

局（設備センター）から各ユーザー宅側までの間を，光ファイバーで結び超高速なブロードバンド・ネットワークを実現する光アクセス・システム．最後の“H”の部分には，局からユーザーまでの区間（H : Home，ユーザー宅まで）が示されていて，他には FTTC（C:Curb，歩道の縁まで），FTTB（B:Building，オフィス・ビルまで）などがある．局側とユーザー間の通信システムは，局側とユーザーを 1 対 1 で接続する「ポイント・ツー・ポイント」と，局側と複数のユーザー 1 対多接続する「ポイント・ツー・マルポイント」の 2 つの方式がある．

次世代ネットワーク (Next-Generation Network :NGN)

現在別々に構築されているインターネットサービス用 IP ネットワークと電話サービス用の電話ネットワークを，IP 技術を用いて QoS やセキュリティを向上させた IP 通信ネットワークとして統合し，現行の公衆ネットワークを代替する次世代 IP ネットワーク．電話だけでなくテレビ放送も IP ネットワークで統一的に提供しようとする動きも進められている．NGN が備えるべき特徴としてはエンドツーエンド QoS 保証の提供，モビリティへの対応等が挙げられる．NGN を実現するためには通信事業者やサービスプロバイダが異なるネットワークを相互接続し，すべてのユーザにサービスを提供できるシームレ

スなネットワーク環境の実現が必要である．そのためには国際的な標準化を進める必要性があり，国際電気通信連合 電気通信標準化部門 (ITU-T) 等を中心に標準化が進められている．

IP Multimedia Subsystem (IMS)

これまで固定ネットワーク通信や移動体通信，放送等で行なわれていたサービスを IP 化し，融合したマルチメディアサービス等を実現するための規格であり，その規格に沿って作られたシステム・ソリューション．音声通話とデータ通信の統合を目指すために，IP および SIP 技術を取り入れた仕様が，3rd Generation Partnership Project (3GPP) によって標準化されている．

セッション確立プロトコル (Session Initiation Protocol :SIP)

2 つ以上のクライアント間でセッションを確立するための IETF 標準の通信プロトコル．SIP のおもな用途は電話，テレビ電話やインスタント・メッセージングのような双方向のリアルタイム通信である．このようなリアルタイム通信において，基本的に通信者は対等であり，サーバとクライアントというような役割分担は存在しない．SIP においてはこれを，両者がサーバとクライアントの機能をあわせもつというかたちで表現している．すなわち，SIP は基本的には 要求-応答型 のプロトコルだが，要求者 (UAC : User Agent Client) がクライアントであり，応答者 (UAS : User Agent Server) がサーバであるが，両者がこれら両方の役割を演じる．

セッション記述プロトコル (Session Description Protocol :SDP)

ストリーミングメディアの初期化パラメータを記述する形式の一つ．SDP は，セッションの告知やセッションへの招待，他のマルチメディアセッションを開始するために必要な情報を記述することを目的としている． IETF の RFC4566.

Parlay-X

Web サービス環境で，通信サービスを利用したアプリケーション開発を行うためのオープンなアプリケーションインタフェース (API)．正式には，「Parlay-X Web Services」という．通信事業者や通信機器ベンダーが参加する

業界団体「Parlay グループ」によって規定されている．Parlay-X により，ネットワークやサーバ機種を問わず，また，通信サービスに精通していなくても容易にアプリケーションを開発することができる．NGN において ANI を提供する手法の 1 つとして利用が見込まれている．尚，Parlay グループでは，CORBA に準拠した C/C++ 用と Java 用の API として，「Parlay/OSA」も規定している．

Peer-to-Peer (P2P)

ネットワーク上で対等な関係にある端末間を相互に直接接続し，データを送受信する通信方式．また，そのような方式を用いて通信するソフトウェアやシステムの総称．データの送り手と受け手が分かれているクライアントサーバ方式等と対比される用語で，利用者間を直接つないで音声やファイルを交換するシステム等が実用化されている．一部にクライアント/サーバの思想が残ったハイブリッド P2P 型と集中管理サーバを一切必要としないピュア P2P 型がある．

P2PSIP WG

IETF において，通信エンドポイントであるピア間通信のセッション制御にピアツーピア（P2P）アーキテクチャを使用する分散型 VoIP（Voice over Internet Protocol）及びインスタントメッセージ通信アプリケーションのためのプロトコルおよびメカニズムを開発するために設立されたワーキンググループ．

P2PTV

P2P を用いたソフトウェアアプリケーションで動画をストリーミング又は，ファイルとして P2P ネットワークに配信する．サーバを用いずにコンテンツ配信が可能であるため，サーバ負荷を低減する事ができる．

アプリケーションネットワークインタフェース (Application Network Interfaces :ANI)

ITU-T で標準化されている次世代ネットワーク (Next-Generation Network :NGN) のアプリケーションサーバ機能 (Application Support Functions & Service Support Functions: AS) が第三者のユーザに提供するインタフェース．デ

ファクトの ANI として最も有名なのは Parlay-X である．

オーバレイネットワーク

あるコンピュータネットワークの上に構築された別のコンピュータネットワーク．オーバレイネットワーク上のノードは，下位ネットワークのトポロジを意識せずに通信することができる．多くの P2P ネットワークはオーバレイネットワークである．

Internet of Things(IoT)/Machine-to-Machine(M2M)

IoT は，さまざまな「モノ」をインターネットに接続して，離れた場所から操作したり，状態を確認したりする技術．幅広い分野での活用が期待されている．M2M は，機械同士が IP ネットワークを介して相互に通信することを意味ししており，リソースを確保する従来型のコネクションオリエンテッド通信と異なり，IP ネットワークを利用する事で通信コストの削減が可能となと共に，機器を追加しても通信コストの負担がほとんど増えないことメリットとしている．

WebRTC

World Wide Web Consortium (W3C) が提唱するリアルタイムコミュニケーション用の API の定義で，プラグイン無しでウェブブラウザ間のボイスチャット，ビデオチャット，ファイル共有ができる．IETF による関連プロトコルと W3C によるブラウザ対応 API の標準化が進められてきており，W3C による WebRTC のドラフトは Chrome と Firefox で特別に実装されるという形で実験的に行われている．

ウオーターフォールモデル（V モデル）

システム開発の手順を模式化したモデルの一つで，設計やプログラミングといった各段階を一つずつ順番に終わらせ，次の工程に進んでいく方式．最も古典的な開発モデルの一つで現代でも広く普及している．Ｖモデルとも呼ぶ．開発プロジェクトを時系列に，「要求定義」「外部設計（概要設計）」「内部設計（詳細設計）」「開発（プログラミング）」「テスト」「運用」などの作業工程に分け，これらの工程を一度で終わらせる計画を立て進捗管理をする．原則として前工

程が完了しないと次工程に進まない事で，前工程の成果物の品質を確保し，前工程への手戻りを最小限にするソフトウェア開発手法.

不具合管理システム (Bug Tracking System :BTS)

バグトラッキングシステムとも呼び，プロジェクトのバグを登録し，修正状況を追跡するシステム．ソフトウェアの開発においてはバグの修正が重要な作業であり，バグを漏らさず修正し，再発を防ぐには，バグの発見日時や発見者，再現方法，修正担当者，修正履歴，修正方法，重要度，テスト状況などの多くの情報を残し管理する必要があり，多数のバグを管理するためのバグ管理システムが生まれた．バグ管理システムの多くは，ウェブサーバ上で動作し，ウェブブラウザ経由でアクセスできるようになっている.

静的ソースコード解析

コンピュータのソフトウェアの解析手法の一種であり，実行ファイルを実行することなく解析を行うこと．静的コード解析はソースコードに対して行われることが多く，少数ながらオブジェクトコードに対して行う場合もある．本論文内で使用した SourceMonitor 等ツールを使用した解析を意味することが多く，人間が行う作業はインスペクション，コードレビューなどと呼ぶ.

Unified Modeling Language :UML

オブジェクト指向のソフトウェア開発において，データ構造や処理の流れなどソフトウェアに関連する様々な設計や仕様を図示するための記法を定めたもの．ソフトウェアのモデリング言語の標準としてとして広く普及している．UML 2.4.1 は ISO/IEC 19505-2:2012 として標準化されている.

形態素解析

文法的な情報の注記の無い自然言語のテキストデータ（文）から，対象言語の文法や，辞書と呼ばれる単語の品詞等の情報にもとづき，形態素（Morpheme, おおまかにいえば，言語で意味を持つ最小単位）の列に分割し，それぞれの形態素の品詞等を判別する作業．自然言語処理の分野における主要なテーマのひとつであり，機械翻訳やかな漢字変換など応用も多い．もっぱら言語学的な観点を主として言語学で研究されている文法にもとづく解析もあるが，本論文で

はコンピュータ上の自然言語処理として解析を行っている．

条件付き確率場 (Conditional random field:CRF)

無向グラフにより表現される確率的グラフィカルモデルの一つであり，識別モデル. 自然言語処理，生体情報工学，コンピュータビジョンなどの分野で連続データの解析などに利用される. 特に形態素解析，固有表現抽出，ゲノミクスに応用され，隠れマルコフモデルに関連するような問題において，代わりとしても用いられる．

本研究に関する著者の発表論文

(1) 菊間, 篠田, 上田, 福田, ”継続的安定運用を目指した大規模通信システムの開発手法と評価”, 電子情報通信学会論文誌, Vol.J101-B,No.12,pp.1065-1082,Dec. 2018.

(2) K. Kikuma, T. Yamada, K. Ueda, and A. Fukuda, ”Automatic Test Case Generation Method for Large Scale Communication Node Software”,The 6th International Conference on Emerging Internet, Data & Web Technologies,pp429-503 (EIDWT-2018)Mar., 2018

(3) 菊間, 上田, 岩田, 福田, ”物理ネットワークセッション制御機能を利用した P2P ネットワーク制御”, 電子情報通信学会論文誌 Vol.J100-B No.5 pp394-410, May. 2017

(4) K. Kikuma, M. Asamura, and T. Murakami. ”Congestion Control in Broadband Common channel Signalling Network,” Proc. of The 13th International Conference on Information Networking(ICOIN13),pp.11D-3.1-3.4 , Jan. 1999.

(5) T. Oh-ishi, K. Sakai, K. Kikuma, and A. Kurokawa, ”Study of the Relationship between Peer-to-Peer Systems and IP Multicasting,” IEEE Communications Magagine, vol41(1), pp80-84, Jan. 2003.

(6) M. Yamada, R. Ono, K. Kikuma, and H. Sunaga, ”A study on P2P platforms for rapid application development,” The 9th Asia-Pacific Conference on Communications(APCC 2003), pp368-372, Mar. 2003.

(7) K. Kikuma, Y. Morita, and H. Sunaga, ”A Study of a P2P community on a P2P communication platform”, International Conference on Communication Technology 2003(ICCT2003),pp.153-156, Apr. 2003.

(8) H. Sunaga, K. Ueda, T. Iwata, K. Kikuma, and M. Takemoto, ”P2P appli-

cations using the semantic information oriented network," Fourth International Conference on Peer-to-Peer Computing, pp. 272-273, Aug. 2004.

- 発表論文 (1),(4) は本論文の第 3 章に関する論文である.
- 発表論文 (2) は本論文の第 4 章に関する論文である.
- 発表論文 (3),(5),(6),(7),(8) は本論文の第 5 章に関する論文である.