管理番号＃１３－０７

平成１６年度

研究開発成果報告書

ユビキタスコンピューティング環境を実現する基盤ネットワークプロトコルの研究開発

委託先：㈱横須賀テレコムリサーチパーク

平成１７年５月

情報通信研究機構

平成１６年度　研究開発成果報告書

「ユビキタスコンピューティング環境を実現する
基盤ネットワークプロトコルの研究開発」

目　次

1　研究開発課題の背景

20 世紀後半より，情報通信技術・IT（Information Technology）の急速な進展と広範な普及によって，我々の社会は大きく変革し，いわゆる情報社会へと突入した．我が国も情報通信技術に関しては世界を牽引した数少ない国の一つして自負するに十分であり，多くの研究開発がなされてきた．現在も次世代携帯電話を始めとして，世界に貢献する成果を輩出している．

1.1 ユビキタスコンピューティング

1990 年代からの情報通信網基盤の爆発的発展は，ネットワークの大容量化と接続機器（コンピュータ）の高性能化によって，より高度なユーザサービスを実現してきた．それと同時に，近年の我が国では，これとは異なる情報通信網の急速な発展も起きている．それは，従来は情報処理能力や通信能力を持たなかった，身の回りに存在する無数の小さな「モノ」に対して，計算力と通信力を与える方向への爆発的な拡大である．こうした身の回りのあらゆるものをインテリジェント化することで，高いユーザサービスを実現する情報通信のパラダイムは，ユビキタスコンピューティング （Ubiquitous Computing）や「どこでもコンピュータ環境」と呼ばれている．このパラダイムは，1980 年代後半に日米で同時に提唱され，その後ポスト PC 時代の情報通信技術のパラダイムとして，専門家の間で広く受け入れられている．
このユビキタスコンピューティングこそが今後の日本型の IT 技術開発のパラダイムとして有望なものであり，本研究開発課題はこのパラダイムの実現に対して正面から取り組むものである．

1.2 我が国の産業構造との関係

情報通信分野は極めて広範で深いため，単一の国や会社，組織ですべてをカバーすることは，もはや不可能である．世界全体でみた情報通信分野は，様々な国の様々な組織がそれぞれに得意な部分を分担し，世界規模で協調と競争をしながら発展していくのが健全な姿である．
現在，すでにインターネットや PC の分野の技術的主導権は米国が握っている．実際，パーソナルコンピューティング分野は，心臓部である CPU や OS といった根幹技術を"Wintel"という造語が示すように，米国の特定ベンダーの独占状態にあり，我が国の研究開発は壊滅状態である．しかし，情報通信分野で十分に大規模な収益が見込める分野は，インターネットや PC の分野だけではない．特に，ポスト PC 時代の主力産業と考えられる情報家電，ネットワーク機能をもった電子機器に目を向ければ，ITRON（Industrial TRON: The Real-time Operating system Nucleus）を始めとして，我が国の独自技術が世界的に強い競争力を維持し続けている．我が国の産業構造からみて得意な部分とは，小さく緻密な機器を生産するところにある．こういった視点からも，効率的な研究開発投資を考えると，むしろ，我が国が最も得意な分野を更に発展させることを目指すべきである．こうした技術は今だ世界の先端を走っており，この点を活かした新しい産業を創造し，世界をリードする分野を積極的に開拓すれば，当該分野の世界的イニシアチブを獲得することも可能である．

1.3 緻密でクリーンな IT 技術開発への要求

21 世紀を迎え，光ファイバー網を使ったインターネット等が目指している，より速く・より大容量・より広帯域を追求する情報通信技術の重要性は高いものの，現在それに加えて更に，次のようなより緻密でクリーンな情報通信技術への要求も高まっている．第一に，豊富な容量・帯域・速度をもった IT 基盤上のデジタル情報の流れを，人間や社会の意志に基づいて確実に制御できること．権利がある人にだけに情報のアクセスを許可し，権利の無い人の不正な盗聴を防ぐこと，また，不正な情報複製ができないようにするといったことが，確実に行えることが重要である．従って，これには，広い意味での，暗号技術や認証技術などが含まれる．第二に，省電力をはじめとして，資源を浪費せず，環境への悪影響を最小限にとどめる，クリーンな情報通信技術．こうした考え方は，カームコンピューティング（Calm Computing）とも言われる．

1.4 セキュアコンピューティング

2001 年は，我が国でもサイバーテロが行なわれた．また，Nimda，CodeRed，Circum といったインターネットを介して大規模に感染するコンピュータウィルスやワームも発生した．既に米国では我が国以上にサイバーテロが行なわれ，情報社会を脅かしている．今後もこうした危険性は確実に拡大するだろう．現在の情報通信インフラで解決することが最も要求されている課題は，こうした攻撃に強い，セキュアな情報通信基盤である．しかもそれが，一般素人でも簡単に扱うことができなければならない．イ

ンターネットの当初の設計方針には，インターネットを通じて通信する者同士が信頼できないようなこと，また，これほどまでに素人ユーザの割合が多くなることは，想定されていなかった．現在これに対して抜本的対策を施さない限り，かつての公害問題のように，将来の情報社会に禍根を残すことになりかねない．

2　研究開発体の全体計画

2.1　研究開発課題の概要

本申請によるプロジェクトの研究開発課題は、我々の身の回りの、あらゆるものにマイクロコンピュータと通信機能を組み込み、それらが互いに情報を交換しながら協調動作を行い、人間生活をより高度にサポートする、ユビキタスコンピューティング環境を構築するための、次世代通信の基盤プロトコルおよびそのシステムの確立である。

1. ユビキタスコンピューティング環境が目指す最終目標

ユビキタスコンピューティング環境とは、身の回りのあらゆるものにマイクロコンピュータと通信機能を組み込み、それらが互いに情報を交換しながら協調動作を行い、人間生活をより高度にサポートする環境のことである。今まで、こうした環境を使った IT の多様な「夢」が語られており、その「夢」を実現することが本研究プロジェクトの最終目標である。

例えば、家庭では、家に設置された温度センサーが常に外気温と室内温度を監視しており、居住者が部屋の温度を下げようとした時、もしも外気温が室温より低ければ窓を開ける。しかし、部屋でピアノを弾き、外部に騒音が漏れたら、窓を閉めて自動的に空調が入る。また、自家用車で帰宅するときに、自動車のナビゲーションシステムから、到着時刻に合わせて自宅の風呂の湯を沸かすこともできる。

こうした環境を実現するためには、膨大なコンピュータを身の回りのあらゆるものに埋め込み、人間自身も常にコンピュータを携帯し、それらが互いにネットワークで接続され、情報交換しながら協調処理を行うメカニズムが必要とされる。我々は、この埋め込まれたり、携帯されたりするコンピュータを**「インテリジェントオブジェクト」**とよび、これらを接続する通信メカニズムのことを**「ユビキタスネットワーキング」**と呼ぶ。

本研究では生活空間を構成する大量のインテリジェントオブジェクトからなるネットワークを想定している。このネットワークと他の既存のネットワークとの違いは、まずネットワークにつながるノードの数が桁違いに多いことである。一人当たり数十から数百のプロセッサがある高密度のユビキタスコンピューティング環境のなかから、通信すべき適切なコンピュータを指定するためにはどうすればよいか。更にそれが何百、何千もの人が活動するビルや都市、最終的には世界までつながった時に、このユビキタスネットワーキングがどのように展開されるべきか、といったことが重要な課題となっている。従って、本研究における中心課題は、この「ユビキタスネットワーキング」の根幹となる基本方式を明らかにし、更にそれを動作させるシステムを構築することである。

これらの多くのインテリジェントオブジェクトを協調させるためには、調停動作の実現がポイントである。例えば、一億個のコンピュータがネットワークにつながった場合、全部のデータを手に入れそれに基づいて中央で方針を決定するという、中央集権的な手法で全部の動作を最適化することはもはや不可能ではないかと考えている。そのためには何らかの分散的な最適化方式を考案する必要がある。

生活の場におけるインテリジェントオブジェクトは個々のエンドユーザの都合でネットワークに突然追加されたり、はずされたり、次の日には別の箇所につながったり、ということが起こる。そのような**「アドホック（ad hoc)」**性をもったネットワークを実現しなければならない。その際に一般の人でも扱え、面倒なオペレーションが不要な**「エフォートレス（effortless)」**な性質を持つ必要がある。ユビキタスコンピューティング環境において、無数のコンピュータをちりばめた時に重要なことは、その上で、これらのコンピュータ群が 24 時間 365 日正常動作するように運用できることである。そのためには、ユビキタスコンピューティング環境を構成するシステムと社会との親和性、運用技術に対する研究も重要となる。

2. 技術課題の概要

本研究では、このユビキタスコンピューティング環境の基盤技術となる通信プロトコル（**ユビキタス**

ネットワーキングプロトコル）や、それを用いた通信網基盤の構築技術の研究開発を行う。そのために以下の研究開発項目を実施する。

1. リアルタイム通信プロトコル
2. セキュアネットワーキング、セキュアコンピューティング
3. コンパクト性
4. エフォートレスオペレーション、エフォートレスマネジメント
5. ユーザとの親和性
6. 省リソース
7. 既存通信網との親和性
8. 高度な協調・調停動作による人間生活の支援機能の実現

（1）リアルタイム通信プロトコル

ユビキタスコンピューティング環境を実現するための基本プロトコルには、人間の振る舞いや生活・社会を構成するあらゆる事象に追従して応答できるための、（ソフト）リアルタイム性が必要である。特に、身の回りのインテリジェントな機器（アクチュエーター）を制御する部分には、より強いリアルタイム性が要求される。

（2）セキュアネットワーキング、セキュアコンピューティング

先に述べたユビキタスコンピューティング環境による夢、例えば、未来の ITS（Intelligent Transportation System）のイメージとして、自動車のナビゲーションシステムから、到着時刻に合わせて自宅の風呂の湯を沸かすといったシナリオが描かれてきた。実際に、このシナリオを実現するためには、悪意ある他者が自宅の風呂を操作することを防げなければならない。更に近年は、サイバーアタックやクラッキング、サイバーテロを想定した対策も求められている。ユビキタスコンピューティング環境を、ネットワーク経由のアタックから守るためには、セキュアな通信プロトコル、セキュアな通信システムの開発が不可欠である。ネットワーク基盤の遍在化(ユビキタス化)が急速に進んでいる現在、セキュリティーを向上させる技術開発は急務である。

（3）コンパクト性

ユビキタスコンピューティング環境では、非常に膨大な数の小さな機器にコンピュータや通信機能が埋め込まれる。従って、各ノードが小さくコンパクトであること重要である。計算機能や通信機能の遍在性（ユビキタス性）を高めるためには、各ノードがコンパクト化できることが不可欠であり、本研究課題における目標としては、SOC (System-On-Chip)上で実現できる程度にまで最適化を図る。

（4）エフォートレス（Effortless）

ユビキタスコンピューティングのためのネットワークプロトコルを実現する上で重要なことは、コンピュータに詳しくない一般利用者でさえも、自身が所有する機器の安全性、蓄積情報や通信の秘匿性等を手軽に確保できることである。現在でも多くのセキュアプロトコルがあるが、そのほとんどが、認証や暗号のための鍵の管理や認証局（CA 局）からの証明書の取得といった、専門知識を要する運用作業を伴うため、普通の人が手軽に使えるものになっていない。そこで、本研究課題では、**耐タンパ性**を有するハードウェアを利用することによって、より手軽で簡便なセキュア通信プロトコルを開発する。このプロトコルによって、簡単に使えるパッケージ化された強固で安定した汎用セキュリティー基盤が実現され、コンピュータに詳しくない普通の人でも、セキュアな通信基盤の恩恵に浴することができる。

（5）ユーザとの親和性

ユビキタスコンピューティングは、人間の身の回りに計算機能や通信機能を埋め込むことで、人間生活をサポートする。そこで、重要な観点の一つは、どのように人間をサポートしていくかということである。ユビキタスコンピューティング環境においてこうした人間との境界部分、つまりヒューマンインタフェースをつかさどる分野は、**強化現実環境**(Augmented Reality)とか、**複合現実環境**(Mixed Reality)といった分野となる。本研究でも人間にとって、より自然でかつ利便性の高いサポートの手法の研究開発を進める。

（6）省リソース

ユビキタスコンピューティング環境では、ノード数が従来の分散環境やインターネット環境にもまし

て膨大な数になることから、一つのノードが使う電力量など、消費する各種リソースが小さくなるような Calm Computing 技術を確立する。従来の情報通信技術は、性能や利便性を最大化するための研究開発に重きがおかれており、省資源を最大化するといった基準に則った研究開発は比重が低かった。本研究課題で実現するプロトコルやシステムでは、こうした省電力・省資源を実現する。

（7）既存通信網との親和性

現在、IP による世界的ネットワークが構築されている。その他にも既存の通信網には、携帯電話網、通常の加入電話網をはじめとして、多様なものが存在する。これらは、性能、サービス、保守の容易性といった点で利害得失があり、これらが単一のプロトコルに統合されるとは考えづらい。ユビキタスコンピューティング環境を構成する通信プロトコルに関しても同様に、様々な利害得失をもった多様なプロトコルが混在する環境を前提とし、互いに補完的な関係になることが理想的である。

そこで、本研究開発課題では、こうした既存の多様なプロトコルとの相互接続によって、柔軟な情報交換や制御を行うメカニズムを確立する。具体的には、現在の IPv4・IPv6 に基づくインターネット、携帯電話網上に構築されている情報通信網基盤と、ユビキタスネットワーキングプロトコルとを相互接続する、ゲートウェイ技術を確立する。それは単に、ネットワーク層やトランスポート層によるゲートウェイだけではなく、セッション層やアプリケーション層にいたる、ほぼ全ての層において接続する必要があり、そのための統合的なゲートウェイ技術を構築する。

（8）高度な協調・調停動作による人間生活の支援機能の実現

ユビキタスコンピューティング環境では、単に多くのノードがあるだけでなく、それらが「協調」「調停」動作をすることで、人間生活を高度に支援する。例えば、部屋の温度が上ったら、窓を自動的に開け、もしも部屋の中でピアノをひきはじめ騒音が発生したら、窓を自動的に閉めて空調を入れるといった動作である。こうした協調・調停作業が、あちこちに埋め込まれた膨大な数のコンピュータの間で実施できなければならない。そのための通信システム、交換情報形式、超機能分散型ソフトウェアのためのプログラミングシステムなどの研究開発を行う。

3. 研究実施計画の基本方針（概要）

ここでは、2.1、2.2 で示した本研究の目標と課題を達成する実施計画の方針の概要を示す。

（1）システム単位のサブプロジェクト構成

本研究開発課題における技術的ボトルネックは、いかに大量の小さいノードを、特定の目的に沿って協調動作させるかという、システム統合技術にあると考えている。そこで、要素技術毎に細分化したサブテーマわけをした場合、最後に統合化して相互接続環境を構築するのが困難になるため、次の通り「機器」による切り分けを中心としたサブテーマわけを行う。

1. セキュアハードウェア
2. 通信システム
3. ユーザ側のエンドノードシステム（モバイル端末／情報家電／PDA 等）
4. サーバ側のエンドノードシステム（認証サーバ／アプリケーションサーバ等）
5. システム統合技術
6. 超機能分散システム指向の開発環境

（2）システム工学的検証の重視

ユビキタスコンピューティング環境は、単に情報通信のメカニズムだけを研究開発するだけでは成功しない。ユビキタスコンピューティング環境は、人間・社会生活に無数に埋め込まれ、社会活動や生活を支援するものであるため、技術的に優れていても、例えば騒音や発熱の程度によっては人間生活にはなじまない。公共の場に設置するものは、多少の悪戯や荒い扱いにも耐えなければならない。膨大な数のノードが現実社会の中で容易にかつ安全に運用できるのか、無数に埋め込まれたインテリジェントオブジェクトのメンテナンスは可能なのか、ユビキタスコンピューティング環境のセキュリティーは厳密に運用できるのか、といった諸問題がある。

そこで、本研究開発課題では、こうした社会や生活と、ユビキタスコンピューティング環境の間の親和性を重視し、本研究開発成果を実用に耐えるシステムとして完成させるために、システム工学的見地からその検証を行う。検証項目としては、①システムの信頼性の検証、②運用評価、③ユーザビリティ、④スケールファクターのシミュレーション、⑤環境アセスメントを計画している。

4. 研究実施計画の詳細（システム部分）

システム面の研究のサブテーマは、「2.3 （１）**システム単位のサブプロジェクト構成**」の部分で述べたとおり、機器種類毎に切り分けを中心とする。サブテーマとしては、以下を計画している。

【サブテーマ１】
セキュアコンピューティングの基盤となるセキュアハードウェア
【サブテーマ２】
基盤通信システムの研究開発
【サブテーマ３】
ユーザノードシステムの研究開発
【サブテーマ４】
サーバノードシステムの研究開発
【サブテーマ５】
ユビキタスコンピューティング環境を構成するシステム統合技術の研究開発
【サブテーマ６】
超機能分散システム指向の開発環境の研究開発

【サブテーマ１】セキュアコンピューティングの基盤となるセキュアハードウェア

通信のセキュリティーは一般的にはソフトウェアだけで確保することはできない。何らかのハードウェアによる情報保護が不可欠である。従来型の情報システムでは、機材が設置されている建物や部屋に対する入退館管理等による物理的な保護が前提であった。ユビキタスコンピューティング環境では、公共の場に露出して設置されたものも対象であり、ユーザが常に携帯し頻繁に紛失や盗難が起きるものまで含まれる。しかも、前述したようにセキュリティーを確保するために、ユーザが暗号・認証の仕組みを理解することが必須であってはならない。

そこで、本研究では、LSI 自体に不正アクセスが加えられないように加工を施した、いわゆる**「耐タンパー性（Tamper Resistance)」**を有するハードウェアを、ユビキタスコンピューティング向けチップとして新規に開発し、それをユビキタスコンピューティング環境のセキュアシステムの基盤パーツとする。本研究では、ユビキタスコンピューティング環境を構成する耐タンパーチップとして、以下の 2 種類を研究開発する。

1. **コンタクトレスチップ**
   ユビキタスコンピューティング環境と、ISO14443 規格に基づくコンタクトレス通信上で、ユビキタスネットワークプロトコルをサポートする。
2. **デュアルチップ**
   ユビキタスコンピューティング環境と通信する、ISO14443 上のコンタクトレス通信チャンネルと、機器組み込み用のセキュアな内部チャンネルの 2 つを有する、デュアル型の耐タンパーチップである。

従来から、IC カードなどに用いられている LSI チップも似た性質を持っているが、それらと比べて場合においても以下の新規性をもたせる。

1. 分散環境ノードとして動作
2. 公開鍵暗号技術を基盤とした暗号／認証系のサポート（PKI にも対応）
3. 高度な省電力機能

【サブテーマ２】基盤通信システムの研究開発

(2-1) 基盤プロトコル概要

基盤通信システムのサブテーマでは、ユビキタスコンピューティング環境を構成する通信システム全般を扱い、本研究の核であるユビキタスネットワークプロトコルの研究、そのプロトコルスタックの開発、ルータをはじめとした各種ネットワーキング装置を含む。ユビキタスコンピューティングシステムにおいては、その目的に最適化したネットワークアーキテクチャを導入する。今回の研究対象となるユビキタスネットワークのアーキテクチャと機能は以下の特徴をもつ。

(2-2) データリンク層

データリンク層は既存の方式を用いる。例えば、2.5GHz および 5GHz 帯の無線 LAN、Bluetooth、

ISO 14443、PHS、第3世代の移動体通信ネットワークなどである。これらの方式としては、端末と端末が直接通信する**アドホックモード**の通信形態と、基地局およびバックボーンネットワークを介した通信形態の双方を開発する。

(2-3) ネットワーク層

ネットワーク層はユビキタスネットワークに適した新しい方式を考案して実現する。通信形態は、ユニキャストとマルチキャストをサポートし、それぞれ帯域保証や優先制御を行う**リアルタイム通信**と、ベストエフォート通信を扱う。帯域保証等を行うリアルタイム通信の場合は、そのためのシグナリングを提供する。

ここでは、Layer 2 ARP (Address Resolution Protocol) が必要である。ユビキタスネットワーキングでは、IPで使用されるARPとは異なり、実世界上の位置や社会的なセマンティックスなどに基づいたノード指定に対応する（**デバイス・ノードルックアップ機能**）。

ネットワーク構成やノード間の帯域などのルーティング情報を交換するルーティングプロトコルを実現する。このプロトコルは、ユニキャストのためのプロトコルと、マルチキャストのためのプロトコルの双方、またダイナミックな変化に追従できる柔軟性が必要となる。

更に、ネットワークにおける輻輳や障害等を通知するための**OAM (Operation Administration and Maintenance)情報交換プロトコル**を備える。このプロトコルは、ネットワーク経路上の故障に加え、リアルタイム通信のための輻輳の検知やマルチキャストにおける障害情報の転送などを、統合的にサポートする。

(2-4) トランスポート層

トランスポート層のプロトコルとしては、コネクション型とコネクションレス型のプロトコルを用意する必要がある。双方のプロトコルとも、遅延変動や誤り率などのサービス品質に関して、アプリケーションが要求する品質を最小限の機能で実現するサービス品質機能を有する。コネクション型のプロトコルはユニキャストを対象とし、コネクションレス型のプロトコルはユニキャストとマルチキャストの双方を対象とする。また、通信相手の指定については、アドレスを指定する方式のほかに、要求条件を指定する方式についてもサポートする。確認応答を有し、信頼性の高い通信を可能とするマルチキャスト用のプロトコルも用意する。

(2-5) セッション層

セキュリティーや認証のための機能を提供する。相互認証と同時に鍵交換を行い、セッション中は暗号通信が行なわれる**セキュアセッション**機能、またロールバック機能を備えた**トランザクションセッション**機能、リアルタイム応答が要求される場合の**ライトウェイトセッション**機能を備える。

(2-6) アプリケーション層

アプリケーション層は、各種応用に対応するプロトコルを用意する。その他に、通信のサポートのために各種応用から共通に使用されるプロトコルを用意する必要がある。1つは、サービスルックアップ機能で、サービス名からそれを提供するノードの物理アドレスを検索するなど、各種のネットワーク構成情報の検索プロトコルを構築する（**サービスルックアッププロトコル**）。また、ネットワーク機器の状態監視や構成変更などを遠隔で行う**ネットワーク管理用プロトコル**も備える。

【サブテーマ３】ユーザノードシステムの研究開発

ユビキタスコンピューティング環境を構成するノードの中で、ユーザと直接接することが想定される機器類の研究開発である。これをここでは**ユーザノード**と呼ぶが、想定されるユーザノードには、ユーザが携帯する**移動ノード**と、生活環境に設置される**固定ノード**がある。双方とも、「**（２）基盤通信システムの研究開発**」のサブテーマで開発された、ユビキタスネットワーキングプロトコルを搭載する。移動か固定かに応じて、利用可能な計算・通信資源や、物理通信路の環境、物理的な大きさ、それに基づくユーザインタフェース等が異なるために、それぞれ固有の実現技術が必要とされる。本サブテーマでは、こうした条件に適合した様々なユーザノードの構成方法・機構を中心に研究開発を進める。

(3-1) 移動ノード

移動ノードや、常にユーザが携帯してユーザとユビキタスコンピューティング環境との間のインタフェースの役割を担う端末である。具体的には、PDA (Personal Digital Assistant)、携帯電話

スマートカードなどが想定される。固定ノードと比べた場合、移動ノードに関する研究課題として以下がある。

- 物理通信路は基本的に無線通信であり、ユーザの移動を考慮すると通信品質は安定しない、
- 紛失・盗難が起こる可能性が高いため、それに備えたセキュリティーメカニズムを備えること、
- 物理サイズが小さいため、それに適した洗練されたユーザインタフェース、
- 計算資源も限られているため、コンパクト性が求められる。
- バッテリー量の制約も大きいため、徹底した省電力機構。

移動ノードでは、こうした条件をクリアした中で、ユビキタスネットワーキングプロトコルの実現技術を確立する。ここでは、カスタム LSI を作成することを念頭に置く。

(3-2) 固定ノード

固定ノードは、ユビキタスコンピューティング環境に設置され、人間の振る舞いや、環境の変化に応じて柔軟かつ緻密に動作するインテリジェントオブジェクトである。具体的には、住宅内にある電子機器類、オフィスにあるコピー機やファックス、シュレッターといった機器、また公共の場に設置された券売機、自動販売機、チケットゲートといった機器を想定している。移動ノードと比べた場合の固定ノードに関する研究開発項目の特徴は、以下の通りである。

- 物理的認証をうけない不特定多数によって利用されることが前提となり、そのための認証機能、セキュリティー機能が必要とされる。
- 特に公共の場に置かれた機器は、ネットワーク的にも公共的なセグメント上に置かれることになり、その場合にセキュア通信の確保
- ユーザノードであると同時に、サーバ的な機能の提供も求められる。
- 回線や電源の状況は良い環境にある。

固定ノードでは、こうした特質を前提とした、ユビキタスネットワーキングプロトコルの実現技術を確立する。

【サブテーマ４】サーバノードシステムの研究開発

サーバノードは、ユビキタスネットワーキングプロトコルを実現し、特にユーザノードを対象としてネットワークサービスと機能を提供するノードである。具体的には、①セキュアセッションのための認証局、登録局、②分散トランザクションを支えるトランザクションサーバ、③サービスルックアップやデバイスルックアップのためのネットワーク環境サーバ、④ネットワーク管理のための管理サーバ、⑤各アプリケーションに依存したアプリケーションサーバなどが想定される。

従来の電子商取引分野の研究開発の経験により、サーバの運用者側の不正行為も問題とされており、サーバ側も耐タンパー性を持ったハードウェア構成が必要である。かえって移動型のユーザノードのように小型機器の方が、対タンパー性を実現することが容易であり、サーバノードの場合は、大きなノードにおいての耐タンパーハードウェアの実現技術が課題である。

また、ユビキタスコンピューティング環境においては、サーバにおいても、セキュア性を保ちながら、リアルタイム性を実現できるに十分な高応答性能を実現する必要がある。そこで、本研究開発課題では、サーバをセキュアに性能向上させるための、セキュアクラスタリング、暗号・認証処理機能を持ったデータキャッシュ・プロセスマイグレーションのメカニズムを確立する。特に、動的な情報更新が可能な高応答性を達成できるディレクトリサーバを実現する。

【サブテーマ５】ユビキタスコンピューティング環境を構成するシステム統合技術の研究開発

ユビキタスコンピューティング環境は、様々なプロトコルを用いる様々な膨大な機器がヘテロジニアスに結合したシステムであり、それを統合し協調動作させるための技術を確立する。その統合技術を要素技術に分割すると、以下の項目が挙げられる。①通信環境を意識する必要のない確実なコネクティビティおよびサービスの実現、②ネットワーク運用状況、サービス実行状況に応じた、最適なリソース配分によるコンパクトネットワーク／サービス実行環境の構築、③通信環境に最適化したサービス実行メカニズム、④ネットワークへのノードの簡単な装着／脱着と移動時のサービス継続の開発。

上記の課題を実現するときに、本研究開発課題で開発するプロトコルと既存の IP や電話網を使った通信プロトコルの相互運用をスムーズに行うため、次の基本技術の開発を目指す。①アドレスを陽に指定しないアドレス解決およびルーティングメカニズム、②プロトコル変換メカニズム、③複数の異なるネットワークをまたがったときの品質制御メカニズム、④端末と連携したサービス実行メカニズム。

【サブテーマ６】超機能分散システム指向の開発環境の研究開発

ユビキタスコンピューティング環境を構築する上での技術的な課題として、システム開発効率がある。ユビキタスコンピューティング環境は、他の分散環境と比べると、膨大なノード数に特徴がある。現在の計算機科学では、ここまで分散化された多数のノードを協調動作させるソフトウェアを効率よく開発する手法が存在しない。しかも、各ノードはリアルタイムプログラミングとセキュリティーという、単独でも困難なプログラミングを施さなければならない。従って、ユビキタスコンピューティング環境のソフトウェアの開発環境は重要であり、本研究の成否を左右する大きな課題である。

現在計画している開発環境研究は、以下の 3 段階で進める。

1. ユビキタスコンピューティング標準開発環境
2. ユビキタスネットワーキングプロトコルを扱うミドルウェア
3. ユビキタスコンピューティング環境における情報処理モデルの確立

(6-1) ユビキタスコンピューティング標準開発環境

ユビキタスコンピューティング環境は膨大な数のノード数になる。まずハードウェアやオペレーティングシステムといった基盤ソフトウェア部分に対する標準化が必要である。膨大なノード数をまちまちの開発環境で構築しては、ソフトウェアの再利用性の観点から効率よくプロジェクトを運営できない。そこで、まず本プロジェクトの標準ハードウェア、標準基盤ソフトウェアの仕様を決め、その上でのユビキタスコンピューティング環境やユビキタスネットワーキングプロトコルのソフトウェアの再利用性、移植性の高い環境を構築する。

(6-2) ユビキタスネットワーキングプロトコルを扱うミドルウェア

ユビキタスネットワーキングのアプリケーション層のプログラミングを支援するためのミドルウェアを構築する。現在は、こうした分散環境のプログラミングに適したオブジェクト指向言語である Java を用い、このクラスライブラリとしてミドルウェアを構築する。また、ユビキタスコンピューティングに適したコンパクトでセキュアな Java VM（Java Virtual Machine）の実現も行う。

(6-3) ユビキタスコンピューティング環境における情報処理モデルの確立

ユビキタスコンピューティング環境を実現する上で最も重要な分散協調動作を実現するソフトウェアの構築手法を研究する部分である。このテーマにおいても、次の２つのサブテーマを計画している。

1. 現実世界の記述方式
2. 超機能分散環境に適したプログラミングモデル（協調動作の記述）

ア．現実世界の記述方式

ユビキタスコンピューティング環境では、現実世界にコンテクストを取得し、協調動作の結果として、何らかの作用を現実世界にフィードバックする。こうした処理をコンピュータが扱うためには、現実世界をデジタル情報で表現し、それをノード間で交換できるための標準形式を構築する必要がある。しかもユビキタスコンピューティングが扱う事象は、単にオフィス空間といったものだけでなく、人間社会生活のあらゆる場面に及ぶため、まさに、現実世界あのあらゆる事象の標準デジタル表現形式を研究開発する。

イ．超機能分散システムのプログラミング技法

従来は、ネットワーク接続された複数のノード間における分散処理は、オブジェクトベースでモデル化したソフトウェア開発が主流になっている。しかしユビキタスコンピューティング環境のようにノード数が膨大である場合、扱うオブジェクト数も膨大になるため、その間の協調動作をノードオブジェクトレベルの peer-to-peer の協調関係をベースとした動作でプログラミングしては、抽象度が低すぎるということが問題となっている。従って、本研究では、ユビキタスコンピューティング環境全体に対して「計算場（Computing Field)」と呼ばれる仮想的なプログラミング抽象を提供し、各ノードとこの計算場の間の協調動作によってプログラミングする。

5. 研究実施計画の詳細（システム工学的検証）

ユビキタスコンピューティング環境と人間社会・生活の間の親和性を重視し、本研究開発成果を実用

に耐えるシステムとして完成させるために、システム工学的見地から、以下の検証を行う。

【サブテーマ７】
ユビキタスネットワーキングシステムのシステム工学的検証

(7-1) 信頼性の検証
本研究開発課題で構築されたシステム（以下、本システム）を、実際のユビキタスコンピューティング環境と同様の設置状況において運用し、そのシステム信頼性を検証する。ユビキタスコンピューティング環境は、生活のあらゆる面を支援するタイプのシステムであるため、誤作動などは致命的であり、この点に関する検証を行う。

(7-2) 運用評価
実際に本システムに想定される技術レベルの人員が、実験的に作られた本システムの環境を、一定期間オペレーションすることによって、①統合的視点によるセキュリティー強度の検証、②運用やメンテナンスの容易性を評価する。

(7-3) ユーザビリティ評価
本システムが利用者に提供するサービスのユーザインフェース手法について検証、評価する。特に、情報通信に関する技術に明るくない一般ユーザに対するユーザビリティ、また、身体障害者や子供、老人を含めたあらゆる人に対して使えるシステムになっているかという、ユニバーサルデザインの視点による評価を重要視する。

(7-4) スケールファクターのシミュレーション
本研究開発プロジェクトはあくまでも研究段階のものであるため、本研究成果が実際に世の中に大規模に普及した場合、どのような問題が起こっていくかを、本研究における実験のサンプルデータを使ったシミュレーションによって検証する。

(7-5) 環境アセスメント
本システムを生活環境に埋め込んだ場合の、放熱、騒音、電磁波などの影響を計測し、人体や他の機器、環境に対する影響を調査する。その結果をシステムの省資源部分にフィードバックしていく。

2.2 研究開発目標

2.2.1 最終目標（平成１８年３月末）

■全体を包括する最終目標（概要）

（１） 人間の振る舞いや生活･社会を構成する事象に追従して応答するのに十分なリアルタイム性を持つ。この観点からソフトリアルタイム性で十分であるものの、正常に通信処理が行なわれた場合には、PAN、LAN 環境では 1ms 以内、WAN 経由では 0.1s 以内の誤差を目標とする。
（２） 公開鍵暗号と PKI をベースとした暗号、認証のメカニズムを有し、社会のインフラを支えるユビキタス環境にふさわしい安全性と信頼性を実現できること。
（３） 情報家電やインターネットアプライアンスといった比較的乏しい計算機環境の上でも効率よく動作するように、実行性能がよくかつ規模が小さいシステムになっていること。基盤プロトコル全体で、200～300KB 程度の規模を狙う。
（４） システムを管理するための労力が小さいこと。具体的には、ユビキタスコンピューティング環境を構成する機器が設置されたら、たとえ停電等が起きても、無設定で復旧し、基本的に機器が故障するまで、メンテナンスする必要がない。
（５） 非専門家でも扱える簡便さを有すること。例えば、暗号・認証機構を知らない人でも、セキュアネットワーキングサービスを利用できること。
（６） 省リソース対応した回路技術を確立する。特に、省電力機能による電磁ノイズ発生の問題等を解決する。
（７） IP 網、デジタル方式の携帯電話網、PHS 網、固定加入電話網、ADSL 網といった、既存通信

網との間のインターオペラビリティ―機能を有すること。
（8） ユビキタスコンピューティング環境における典型的な協調･調停動作を複数実現することに成功し、その処理コードを分散透明な高い抽象度で記述できること。

■サブテーマ別の最終目標（詳細）

ア．基盤通信システムの研究開発

（1） ユビキタスネットワーキングプロトコルのセッション層部分までの基本プロトコルの仕様を開発し、その正当性、有効性を検証する。
（2） 上記のプロトコルを実現し、評価を行う。
（3） ユビキタスネットワーキングの物理層・データリンク層を担う、以下のプロトコル・ネットワークシステムの上で動作させるためのスタブ部分の仕様を開発すること。
(3-1) Bluetooth
(3-2) ISO 14443
(3-3) IEEE 802.11
(3-4) ISO 7816
(3-5) 無線系電話プロトコル
（4） 既存のインターネット網である IPv4、IPv6 網との間で相互運用と情報交換を可能にするゲートウェイ技術およびシステムを開発する。
（5） 基本機能として、認証機能、暗号機能を有すること。
（6） 認証・暗号に用いる鍵は、本研究のサブテーマ**「カ.」**で開発したセキュアハードウェアに格納する。
（7） ソフトウェア規模は、200KB～300KB のバイナリサイズを想定する。

イ．ユビキタスコンピューティング環境を構成するシステム統合技術の研究開発

（1） IP 通信網や電話網などの既存の通信網との相互接続性を検証する。

ウ．超機能分散システム指向の開発環境の研究開発（ハードウェア部分）

（1） ユビキタスコンピューティング環境の構築の用いる標準開発プラットフォームとしてのハードウェアを開発する。
（2） その上で、サブテーマ**「エ.」**で開発した標準 OS が動作する。
（3） サブテーマ**「カ.」**で開発したデュアル型のセキュアチップを搭載している。
（4） サブテーマ**「ア.（3）」**で挙げた各種 LAN、PAN のプロトコルを搭載可能である。
（5） 音声 CODEC を備える。
（6） グラフィックチップを備える。
（7） CPU 性能として、300MHz 以上の動作周波数を持つこと、また主記憶として 32MB 以上を格納すること。ハードディスクを搭載可能な回路インタフェースを備える。

エ．超機能分散システム指向の開発環境の研究開発（ソフトウェア部分）

（1） 本研究サブテーマ**「ア.」**で開発するユビキタスネットワーキングプロトコルを標準機能で組み込み、それを本研究全体の標準プラットフォームとして利用する、標準リアルタイム OS を開発する。
(1-1) マルチタスク機能と、豊富なタスク間通信・同期機能を提供することができる。
(1-2) 省電力機能を有する。
(1-3) 本研究のサブテーマ**「カ.」**で開発したセキュアチップとの通信機能を有する。

（2） 本研究サブテーマ**「ア.」**で開発するユビキタスネットワーキングプロトコルに対して高抽象度のプログラミングインタフェースを提供するためのミドルウェアを開発する。
(2-1) オブジェクト指向言語 Java のクラスライブラリとして構成されている。
(2-2) ユビキタスコンピューティングに適する、軽くソフトウェア規模の小さい Java Virtual Machine を開発する。

（3） 現実世界記述標準形式に関する研究開発
(3-1) ユビキタスコンピューティング環境が取り扱う実世界の各種環境情報情報の標準デ

ジタル表現形式を策定する。

(3-2) ユビキタスコンピューティング環境が扱うあらゆるパラメータの表現を目指すため、その規模を例えると、「理科年表」のようなものになると考えている。

(3-3) 開発された標準記述形式は、ユビキタスネットワーキングプロトコルのプレゼンテーション層標準の一部として、全機器において使う。

(3-4) 表現の枠組みとしては、文字列形式である XML（eXtensible Markup Language）とバイナリ形式である TAD（TRON Application Databus）形式を構築する。特に後者は表現情報を効率よく表現できるための圧縮表現形式として、計算機資源が乏しいノードで利用する。

（4） 超機能分散プログラミングモデルに関する研究開発

(4-1) ユビキタスコンピューティング環境中に存在する莫大なノードの協調動作を高い抽象度でプログラミングできるプログラミングモデルおよび、そのプログラミング環境の開発を行う。

(4-2) ノード数は、数十から数万までを取り扱うことができ、ノードの分散性、動作の並列性をエンカプスレーションすることができる。

(4-3) あるユビキタスコンピューティング環境で動作していたソフトウェアをそのまま同じ機能をもった、他のユビキタスコンピューティング環境上でも稼動する移植性を有する。

オ．ユビキタスネットワーキングシステムのシステム工学的検証

（1） 本研究開発課題で構築したユビキタスコンピューティング環境の、一年程度の試験を行い、その期間の運用に耐えること。

（2） 現実社会における仕組みの中で運用しても、十分なセキュリティー強度、運用の容易性が達成できること。

（3） 情報通信分野の素人である一般ユーザでも十分本システムを使いこなし､ユビキタスコンピューティング環境の機能の恩恵を受けられること。

（4） ユーザインタフェース部分には、ユニバーサルデザインが施されていること。

（5） 本研究開発課題で作成した実験レベルのシステムの運用データに基づき､それを都市レベルに拡大して普及させた場合の各種スケールファクターが確かめられたこと。

（6） 本システムを社会・生活の場に持ち込んでも、ユーザに不快感を与えたり、社会活動に悪影響を与えたりしないこと。

■主に再委託する予定のサブテーマ

以下、中心部分を再委託する予定のサブテーマの最終目標について記す。

カ．セキュアコンピューティングの基盤となるセキュアハードウェア

（1） コンタクトレス（無線）チャンネルのみを有するコンタクトレスチップと、コンタクトレス（無線）チャンネルとコンタクト（有線）チャンネルの双方を有するデュアルチップを開発する。

（2） コンタクトレス通信チャンネルの物理層･データリンク層のプロトコルは､ISO14443 Type-C 規格を満たす。

（3） コンタクト通信チャンネルの物理層・データリンク層のプロトコルは、ISO 7816 規格を満たす。

（4） 本課題で開発したユビキタスネットワーキングプロトコルで通信する機能を備える。

（5） PKI を使った公開鍵暗号技術に基いた暗号機能・認証機能を備える。

（6） 共通鍵暗号技術に基いた、実行効率のよい暗号機能・認証機能を備える。

（7） 耐タンパー性を有しており、悪意あるユーザからの不正操作から格納情報が守られる。

（8） ユビキタスコンピューティング環境を構成するノードに組み込むことで､そのノードの通信の安全性を向上できる。

（9） CPU には 16bit 以上のワード幅を持ち､RAM: 10KB､ROM: 64KB､EEPROM: 64KB 以上を有する。

キ．ユーザノードシステムの研究開発

（1） ユーザノードとは、ユビキタスコンピューティング環境の中で、利用者が直接接するユーザインタフェースをもった機器である。移動ノード、固定ノードとして、それぞれ複数種類のインテグレーションされたユーザノードを開発する。
（2） ユーザノードは、最終的にはサブテーマ**「ウ.」**で開発した標準ハードウェアを用い、サブテーマ**「エ.」**で開発した標準 OS、ミドルウェアなどを利用して開発する。
（3） サブテーマ**「ア.」**で開発したユビキタスネットワーキングプロトコルを実現する。

ク．サーバノードシステムの研究開発

（1） サーバノードとは、ユビキタスコンピューティング環境を裏で支える基盤サーバ群を含む。
（2）サーバノードは、以下の機能を提供する。
- CA 局や鍵配布サーバを含む PKI（公開鍵インフラストラクチャ）機能
- 電子マネーや電子チケットの決済機能
- 価値情報の発行機能
- デジタルコンテンツの発行機能

（3） サーバノードも悪意ある攻撃から守るためにハードウェアに一定の耐タンパー性を持たせる。

2.2.2 中間目標（平成１６年３月末）

すでに中間目標達成時期や経過し、中間評価も経ているため、ここでは文書の簡略化のために記述を省略する。

2.3　研究開発の年度別計画

（金額は非公表）

研究開発項目	13 年度	14 年度	15 年度	16 年度	17 年度	計	備考
①セキュアハードウェア							
1-1. 全体アーキテクチャ	→						
1-2. コンタクトレス・耐タンパーチップ		→	→	－	－		一部再委託（日立製作所、等）
1-3. デュアル・耐タンパーチップ		－	－	→	→		一部再委託（三菱電機、等）
②基盤通信システム							
2-1. 基盤プロトコル全体アーキテクチャ設計・検証	→	→	→	→	→		一部再委託（KDDI 研究所，富士通、等）
2-2. 基盤プロトコル（データリンク層 IF）	－	→	→	→	→		
2-3. 基盤プロトコル（ネットワーク層）	－	→	→	－	－		
2-4. 基盤プロトコル（トランスポート層）	－	→	－	－	－		
2-5. 基盤プロトコル（セッション層～アプリケーション層）	→	→	→	→	－		一部再委託(東芝、NTT データ、等)
③ユーザノード							
3-1. 移動ノードシステム	－	→	→	→	→		一部再委託（日本電気、等）
3-2. 固定ノードシステム	－	－	→	→	→		一部再委託（東芝、等）
3-3. 強化現実型ヒューマンインタフェース	→	→	→	→	－		

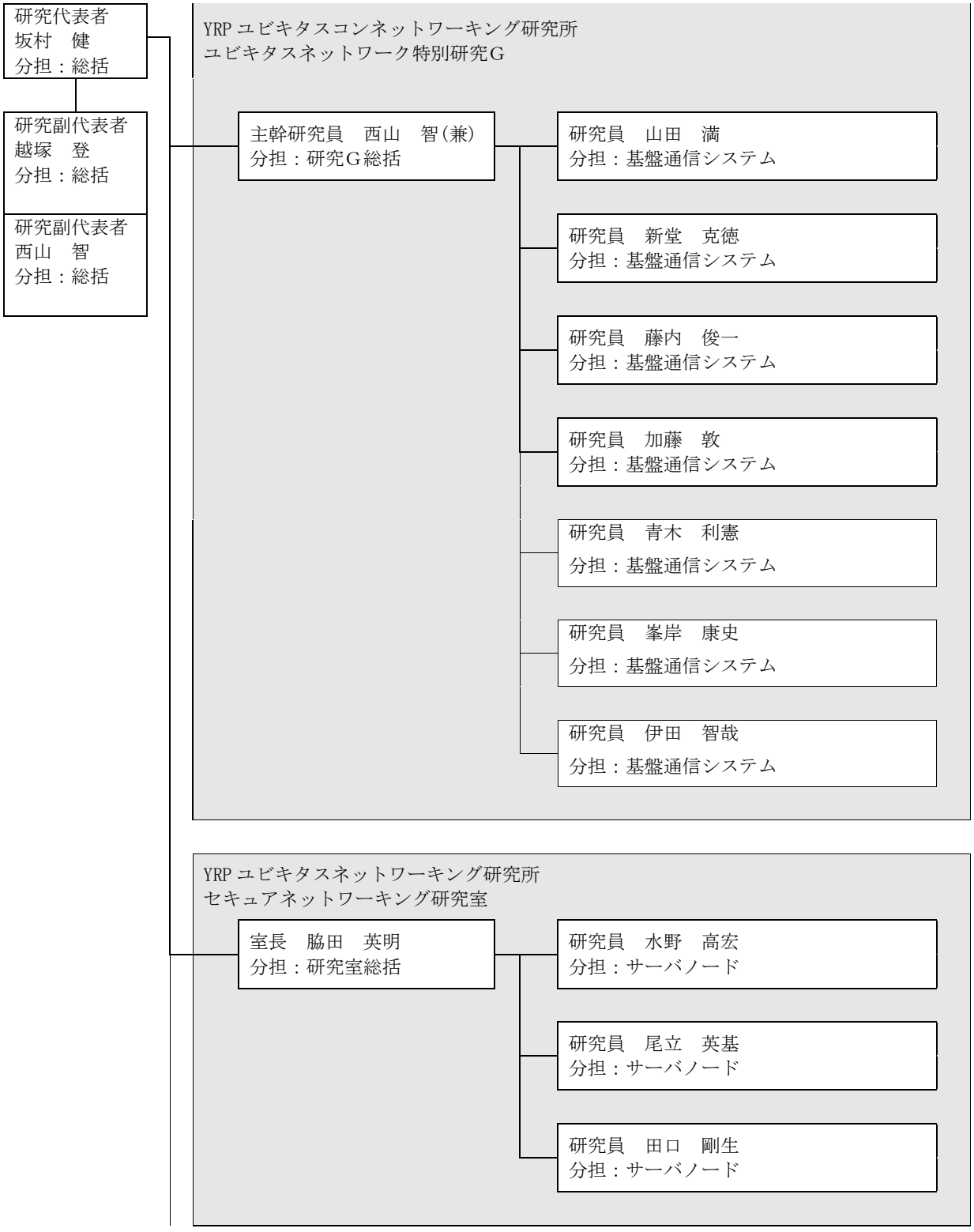
研究開発項目	13 年度	14 年度	15 年度	16 年度	17 年度	計	備考
④サーバノード							
4-1. 暗号認証通信基盤用サーバ	－						
4-2. セキュア分散化サーバ	－			－	－		一部再委託（三菱電機、等）
4-3. コマース処理用サーバノード							
4-4. デジタルエンティティ発行ノード	－	－					一部再委託（NTT データ、等）
⑤システム統合化							
5-1. ユーザノード・サーバノード統合技術	－						
5-2. 対電話網ゲートウェイ研究開発	－	－		－	－		一部再委託（富士通、等）
5-3. 対 IP 網ゲートウェイ研究開発	－	－	－				一部再委託（富士通、等）
⑥超機能分散システム指向開発環境							
6-1. 標準開発ハードウェア研究					－		
6-2. 標準開発基本ソフトウェア研究							
6-3. 標準開発ミドルウェア研究							
6-4. 現実世界記述標準形式							
6-5. 超機能分散プログラミングモデル							

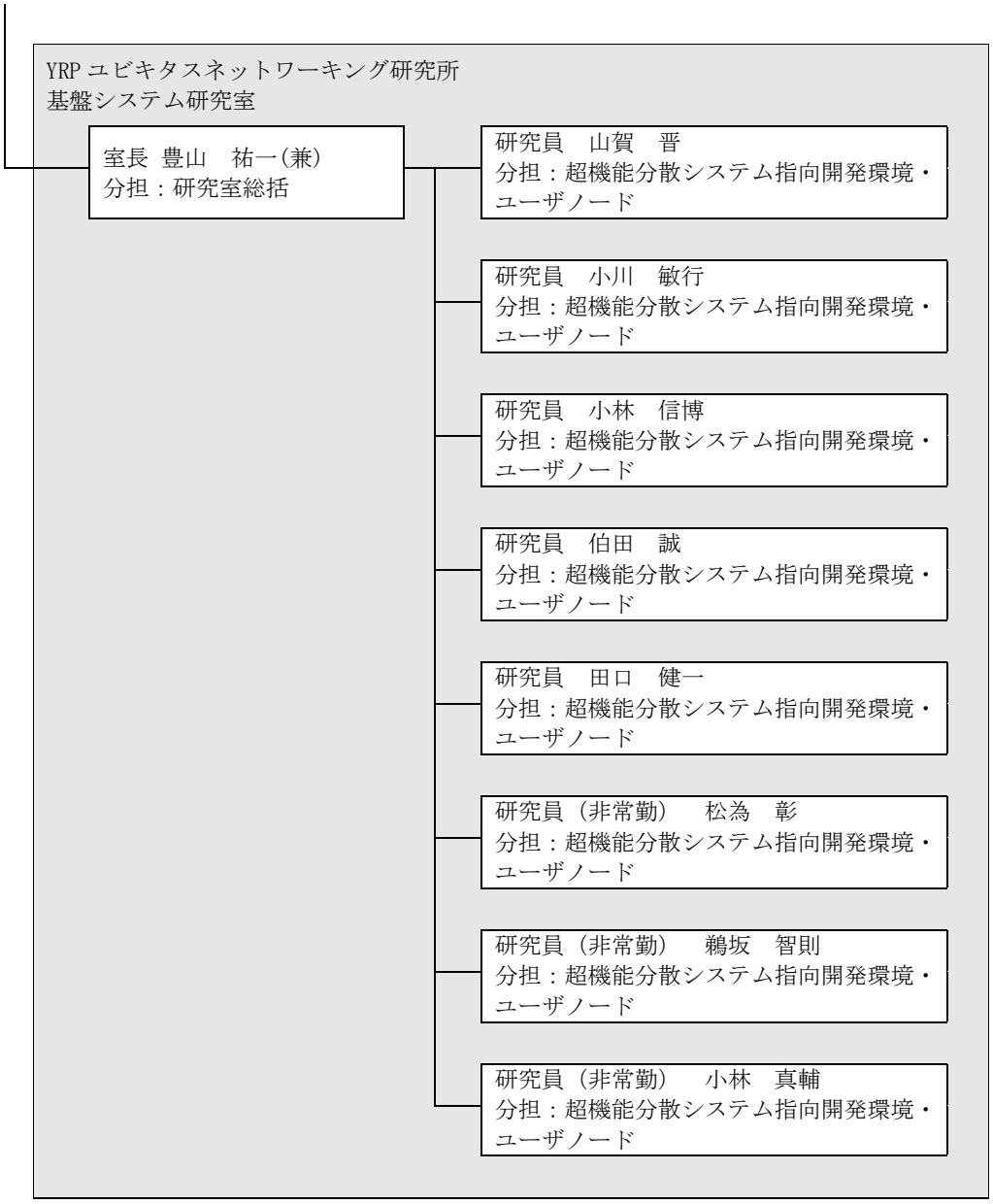
研究開発項目	13 年度	14 年度	15 年度	16 年度	17 年度	計	備考
⑦システム工学的検証	－	→	→	→	→		
間接経費	→	→	→	→	→		
合　　計							

注）１　経費は研究開発項目毎に消費税を含めた額で計上．また，間接経費は直接経費の３０％を上限として計上（消費税を含む.）.
　　２　備考欄に再委託先機関名を記載
　　３　年度の欄は研究開発期間の当初年度から記載.

3　研究開発体制

3.1　研究開発実施体制




YRP ユビキタスネットワーキング研究所
基盤システム研究室
室長 豊山　祐一(兼)
分担：研究室総括
研究員　山賀　晋
分担：超機能分散システム指向開発環境・
ユーザノード
研究員　小川　敏行
分担：超機能分散システム指向開発環境・
ユーザノード
研究員　小林　信博
分担：超機能分散システム指向開発環境・
ユーザノード
研究員　伯田　誠
分担：超機能分散システム指向開発環境・
ユーザノード
研究員　田口　健一
分担：超機能分散システム指向開発環境・
ユーザノード
研究員（非常勤）　松為　彰
分担：超機能分散システム指向開発環境・
ユーザノード
研究員（非常勤）　鵜坂　智則
分担：超機能分散システム指向開発環境・
ユーザノード
研究員（非常勤）　小林　真輔
分担：超機能分散システム指向開発環境・
ユーザノード

4　研究開発実施状況

4.1　ハイエンド型のセキュアチップの研究開発 z

本年度は 32bit のワード幅を持つより高性能な CPU を搭載したチップに関する研究を実施した．昨年度までの 16 ビット型チップと比べ，機能強化及び性能改善のための検討と，チップの耐タンパ性強化を実施した．具体的に，将来の大規模な展開に備えて，CA 階層化に対応した証明書を利用するチップ強化機能に対して，シミュレーションによる機能検証および性能評価を実施した．

4.1.1　32bit のワード幅を持つより高性能な CPU

半導体技術の一般的な進歩に基づき,セキュアチップに 32 ビット CPU コアを使うことが可能になりつつある．32 ビット CPU コアは一般の演算速度の向上，また高速化されたコプロセッサ機能など，採用の魅力は多い．様々な機能強化可能性の検討，性能改善の検討にとりくんだ．

4.1.2　機能強化及び性能改善のための検討

4.1.2.1　アプリケーションプロファイルの導入

32 ビット化することによる機能強化の改善要望はいろいろなアプリケーション開発の場からよせられた．しかし，ハードウェアリソースの制限の厳しいセキュアチップですべての要望に同時にこたえることは不可能である．32 ビット CPU コアで命令コードを格納する EEPROM 領域が増えたといっても限度がある．また RAM 領域は以前どおりに少ない．

そこで，アプリケーションにこたえるための関連した機能を取り外しできるようにして，特定のアプリケーション向きの機能をとりいれたセキュアチップという考えが登場した．一般にこのような関連機能をプロファイルとよぶ．今後はこういったプロファイル単位での機能向上を考えていくアプローチを取ることになろう．

4.1.2.2　コプロセッサの効率よい利用

PKI に頻繁に登場する多倍長数の指数乗演算に使うコプロセッサの効率良い利用はいうまでもなく重要である．

コプロセッサの利用形態を詳しくしらべたところ，多倍長数が 128 バイト程度と長いために，コプロセッサにデータをロードする時間が計算ステップに占める時間中で無視できないことが分かった．これまで実装したチップにおいてコプロッセサーに被演算数をロードする手順などを検討することで一回の PKI 計算を行うコプロセッサの計算時間を短縮できた．

なお，チップ外部からみた場合の PKI の計算の時間は以下のようなものになる．

認証に必要な全体時間 ＝ 必要な命令・応答パケットの入出力に必要な通信時間＋内部の計算時間

今回短くできたのは内部の計算時間の部分であり，現状ではパケットの入出力時間もかなりの割合となっているために外部からみた場合に劇的な改善とはなっていない．

4.1.2.3　チップの耐タンパ性強化

セキュアチップにおいては攻撃に対する耐タンパ性の向上は重要であり，本年度はひとつの対策がおこなわれたので報告する．

(1) 電力消費量のモニターによる鍵推測攻撃

ここ数年の間，電力消費量の時間変化をモニターして暗号鍵を推測しようという攻撃手法が明らかになった．チップ内部の消費電力の変動は，接触してアクセスできれば電力を供給するソケットにつながるプリント基板配線を通じて電力信号を観察することで観察できる．直接配線をモニターできなくても，チップあるいは IC カードの特定部分から漏れてくる電波信号成分をモニターすることで推測できることも報告されている．

昨年までに試験実装したチップにもこの問題があるかもしれないということで，推測を難しくす

るための改善を行った.

(2) 消費電力のランダム化対策

規則正しい計算を行なうということが推測を行うもとになっていて，その規則正しい計算に応じて消費電力が短時間に変動をしていることが原因であるので，不規則的な電力消費を行なうようにすればよい．PKI 演算そのものは高速に行ないたいので，それを変更することは好ましくない．そこで，不規則な電力消費を行なう別の演算を行ない，そちらの信号を外部に重ねて検出させることで電力消費変動を乱雑なものとして，外部からの鍵の推測を難しいものとする．

上記の変更（だけではないが）などにより，消費電力のモニターによる鍵の推測は事実上不可能な程度に消費電力の変動がランダムになった.

4.1.3 将来の大規模な展開に備えての機能検討

現状のセキュアチップの PKI における証明書は単一の CA 局が発行する単一 CA ドメインでの利用が前提となっている．これに対して，一般の PKI の利用場面では CA ドメインが階層型に登場する階層型ドメインも利用されている．この場合には上位の CA が自分の秘密鍵で，(複数ある) 下位の CA が真正であることを，下位の CA の公開鍵を含む証明書に署名をつけることで，信頼のチェインを実現する．大元のルート CA から，階層末端の CA にいたるまで信頼できる署名をつなげることで，末端の CA を信頼できると示すことができるようになっている.

この場合ルート CA がひとつあるようなモデル，すなわちすべての CA がこの階層のどこかにあるものとするモデルと，単一階層のドメインが複数あった場合と同様に独立した階層 CA ツリーが複数独立して存在することを許すモデルがありえる.

4.1.3.1 CA 階層化に対応した証明書を利用するチップ強化機能

現在実装したチップの PKI の証明書は単一階層の CA ドメイン利用を前提としている．単一階層の CA ドメインの利用は，社員証，学生証，特定の店舗・チェインストア内部などでの利用には，一般に広くつかわれており，そういった応用には十分である.

図 1-1 単一階層 CA ドメイン

一方，複数の行政単位，とくに複数の国にまたがるような大規模な展開が考えられる場合に，単一の CA 局が，その応用地域すべての証明書の発行の事務，管理を行うのは非現実的であり，このような場合には，CA を複数設立して，それぞれの地域の証明書の発行管理事務を行うのが通常である．そして，同一アプリケーションサービスを共通して行なうためには，それぞれの証明書をもつクライアント間で PKI にもとづく認証が行なえる必要がある．セキュアチップでも，将来の広範囲地域での展開にそなえて階層 CA における認証の対応を検討した．ここで，採用した階層モデルは，すべての CA の大元となるルート CA が存在して，そこからの署名を順次たどることで末端の CA の真正が判断できるというものである.

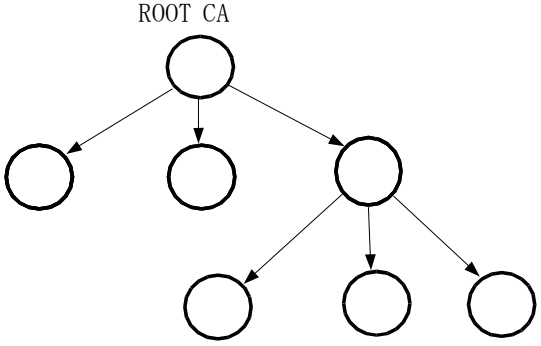


図 1-2 階層化 CA ツリー

チップのレベルでは通信相手から提示された証明書の真正を上のような階層 CA ツリーの枠組みの中で判断して，証明書に入っている公開鍵を元にした PKI 認証を行い，それ以降の手順は，公開鍵に対応する正しい秘密鍵を持っているということで相手の確認が行なえるという意味で相手を信頼できるものとみなす.

4.1.3.2 チップの命令変更：CA の発行した証明書の検証方法

階層化された CA ツリーにおける末端 CA の署名した証明書の真正性は，順次階層ツリーでおこなう必要がある．しかし，階層 CA ツリーを使った認証では，ツリーの深さに比例してネットワーク通信と計算が必要となり，RAM 容量が少ないセキュアチップではすべてを行うことは以下の点で難しい.

- 上記の処理に必要な命令，データのやり取りをチップの I/O で行うことによる処理速度の問題.
- また PKI 計算をすべてチップ内部で行うことによる処理速度の問題
- 処理に必要な RAM 領域の問題.

そこで，階層 CA ツリーを採用する場合のアーキテクチャの一環としてチップにおける証明書の検証は，外部の信頼できる「検証サーバ」にまかせ，その戻す「真偽」応答パケットの偽造などが行なえないようにすることで信頼するアプローチをとった.

証明書の検証は外部に一任することで，いったん証明書の検証がすむと，その証明書に入っていた公開鍵を利用した PKI 認証通信をおこなうことはこれまでと同じである.

4.1.4 将来の課題

4.1.4.1 アプリケーションプロファイルの多様化

32 ビット CPU コアの利用の可能性とともに，応用に対応するプロファイルはいろいろ案が提示されつつあるが，ハードウェアリソース制限が厳しいセキュアチップではすべてに対応することは到底不可能であり，これからも厳選して対応する必要がある.

現在，生体認証プロファイルなどが提案されつつある.

4.1.4.2 階層化 CA ツリー対応：政策面がより重要となる

セキュアチップにおいても階層化 CA ツリーにおける認証の枠組みが可能であることを，性能面でも検証できた.

しかしながら，CA が階層化されることで，単一ドメイン内部で閉じていないシステム全体における運営管理，セキュリティ管理が重要になる．末端の CA の選択，末端の CA にどれだけの権限をあたえるかなどのポリシー的な問題の検討が重要となるだろう．ある末端 CA ドメインの証明書を保持するセキュアチップが，階層化 CA ツリーの「遠方」にある別の CA ドメインの証明書を保持するセキュアチップをどこまで「信用する」べきかは，純粋に技術の観点だけからは論ずることができない．ポリシー議論と実際の運営形態を考慮した上で，「信用しない」という運用者が出てもおかしくない．すなわち「価値判断」が正面に出てしまうために議論が難しくなるが，避けては通れない問題である.

なお階層 CA ツリーにおける認証をすこしでも高速にするために，「近接する」ドメインでの認証だけは内部ですませるなどの特定応用における効率化は検討に値すると思われる．その場合に，チップ内部の実装の複雑さ，設計が間違っていないかの検証と実装の正しさの検証の容易さと，得られる高速化のメリットを十分に評価する必要がある.

4.2 ソフトウェア無線

4.2.1 はじめに

4.2.1.1 背景

RFID は，内部に固有の ID を持ち，読み取り器から非接触でその ID を読むことが出来るデバイスである．リーダライタにより離れた場所から自動的に ID を取得することができるので，物流管理などの応用分野での採用が進んでいる.

RFID は小型かつ安価であり，非接触で情報のやり取りが可能であることから，ユビキタス社会ではこ

れを環境に配置してモノや場所を認識し，コンテクストアウェアネスを実現することが期待されている．

RFID の標準化として，従来より密着型(ISO/IEC 10536)，近接型(ISO/IEC 14443)，近傍型(ISO/IEC 15693)のように主に想定されるアプリケーションに必要とされる通信距離により標準化が行われてきた．これはカードや電子タグに向けたアプリケーションからの標準化であるが，これを RFID のエアプロトコルの標準化を目的に制定された標準規格が ISO18000 である．ISO18000 規格は，ISO18000-1 を総則として，周波数によって ISO18000-2～ISO18000-7 に分類される．

表 2-1　ISO18000 規格の周波数による分類

	キャリア周波数
ISO18000-2	135kHz以下
ISO18000-3	13.56MHz
ISO18000-4	2.45GHz
ISO18000-5	5.8GHz
ISO18000-6	860～960MHz
ISO18000-7	433MHz

近接無線通信に様々な周波数が用いられているのは，それぞれの周波数に特徴があるからである．例えば 135kHz の長波帯を用いた近接無線通信は，金属や水の影響を受けにくいが，転送速度を上げにくい．2.45GHz 帯は水分の影響を受けやすいがタグの小型化が可能など，それぞれの周波数に長所と短所がある．それぞれのアプリケーションには適した周波数やプロトコルがあり，一つのプロトコルに統一することは現実的ではなく，多くのプロトコルが並立することは避けられない．

このような状況のなかで，様々な周波数やプロトコルの RFID が配置された環境下でどうやってこれらの RFID を統一的に自動認識するかが問題になる．現在では，ユーザが予め対象となる RFID のプロトコルを知っておき，専用のリーダライタを使い分けることで対応している．しかし，これはプロトコルの数だけリーダライタを用意する必要があり，非常に煩雑である．

そこで，プロトコルの違いを意識することなく，シームレスにあらゆる RFID をはじめとする近接無線通信デバイスにアクセスするための技術開発が求められている．一つのハードウェアで複数の無線通信プロトコルを実現させるための技術が，ソフトウェア無線技術である．

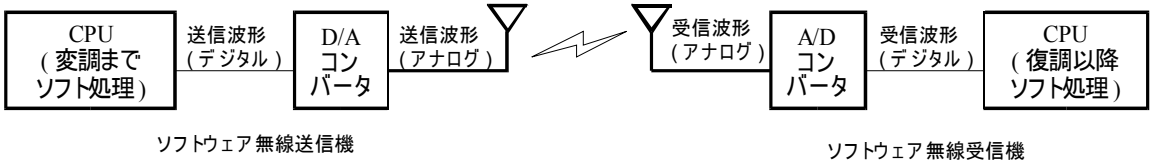


図2-1　ソフトウェア無線の基本構成

ソフトウェア無線は，一つのハードウェアで様々なプロトコルに対応するばかりでなく，将来新しいプロトコルをソフトウェアの変更のみで対応できる利点がある．

本報告では，別のプロトコルを持った複数の近接通信デバイスにアクセスするためのソフトウェア無線技術と，これを応用したマルチプロトコル近接無線通信装置（マルチプロトコルリーダライタ）の開発経過について報告する．

4.2.1.2　H16 年度の研究の概要

平成16年度の課題は，ソフトウェア無線技術を用いた近接無線通信装置の方式研究と基本設計，そのための評価実験を実施することである．

本年度は，まず小型・低電力化に適したソフトウェア無線方式として，データリンク層の機能をFPGA等の論理回路上のハードウェアブロックで構成し，ソフトウェアでスケーリング，選択するソフトウェア無線技術を提案した．その評価のためISO18000-4の近接無線通信規格のデータリンク層プロトコルをFPGA上に実装した近接無線通信装置を試作した．さらに，ISO18092規格(eTRONカード)，ISO15693規格，ミューチップ，ISO18000-4規格の4種のプロトコルに対応した近接無線装置の基本設計を行った．

4.2.2　マルチプロトコルリーダライタの概要

4.2.2.1　従来技術

一般的な RFID 近接無線通信機の物理層のハードウェア構成を図 2-2 に示す．大きく分けて，パケットの構成，解釈を行う制御回路，符号化，復号化，変・復調を行う変復調回路，発振器や二値化回路からなる RF 回路から構成される．一般的には，制御回路，変復調回路はデジタル(論理)回路，RF 回路はアナログ回路から構成される．プロトコルはハードウェアで固定され，複数のプロトコルに対応するのは不可能である．

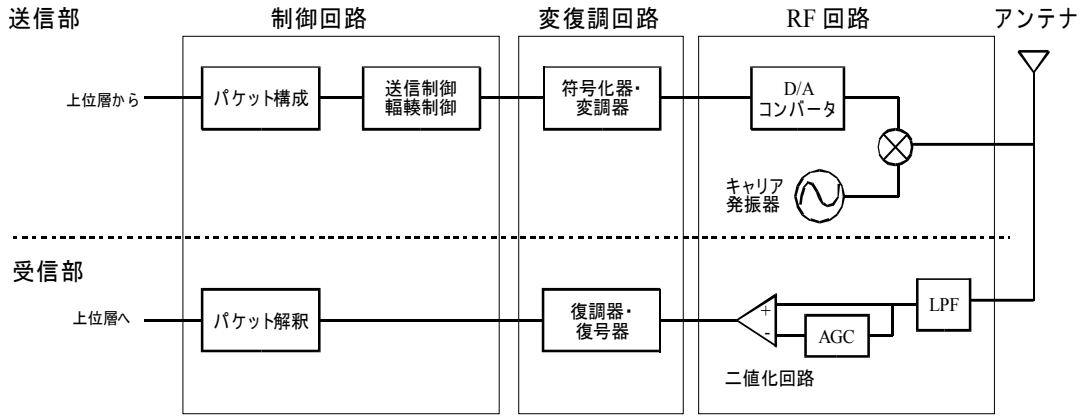


図 2-2　従来の RFID 近接無線通信機の物理層ハードウェア

本構成をソフトウェア無線技術を使ってマルチプロトコル化することを考える．RF 回路は，複数のプロトコルで汎用的に使用可能になるように，送信のための発振器と検波用のローパスフィルタ(LPF)のみの構成とし，A/D，D/A 変換器を通しデジタル化する．すべてソフトウェアで処理するソフトウェア無線の基本的アイディアに従えば，デジタル信号は直接 CPU や DSP などのプロセッサに入力し，変復調回路を含め上位層はすべてソフトウェアで処理する．この構成のソフトウェア無線技術を用いた近接無線通信技術には，米国 ThngMagic 社の Mercury がある．ThingMagic 社 Mercury4 の構成を図 2-3 に示す．

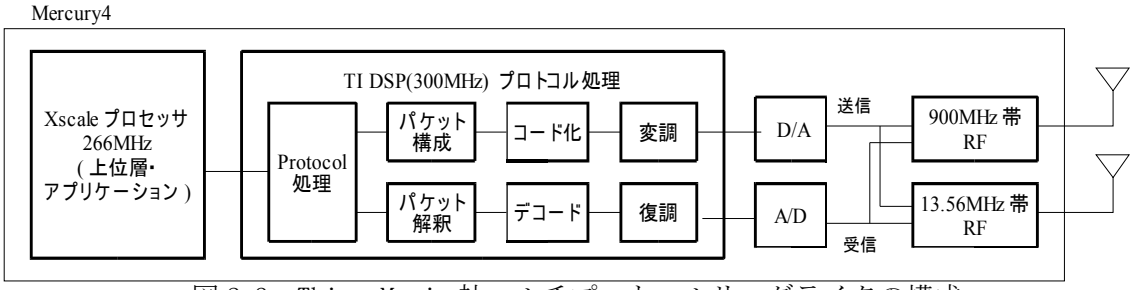


図 2-3　Thing Magic 社マルチプロトコルリーダライタの構成

Mercury4 は，米国で広く用いられている UHF(900MHz 帯域)と HF(13.56MHz)の RF 回路を持ち，ソフトウェア無線技術により複数の近接無線通信プロトコルに対応するマルチプロトコルリーダライタである．UHF 帯域では EPC Class0，EPC Class 1，EPC Rewritable Class 0，ISO18000-6B に対応し，HF 帯域では ISO15693 のそれぞれのプロトコルに対応する．ホスト CPU に Intel 製 Xscale プロセッサ(IXP4xx 266MHz)，プロトコル処理部分を Texas Instruments 製の DSP(Ti55xx 300MHz)を使用しており，装置の消費電力は最大 24W にも及ぶ．ソフトウェア無線の部分に専用の DSP を搭載しているのは，特に受信時にデータを抽出するためのフィルタ処理に高い信号処理性能が必要なためである．

このように，全てのプロトコル処理をソフトウェアで行おうとすると，高い処理性能の複数のプロセッサ構成とならざるを得ず，携帯機器に搭載するときに求められる小型，低電力の条件に合致しない．

4.2.2.2　低電力化に適したソフトウェア無線技術

一般に，ソフトウェア無線技術は通信を行うための専用回路の規模が縮小できる代わりに，高性能のプロセッサが必要になる．これはシステムの大型化と消費電力の増大をもたらすため携帯端末に搭載することが困難である．

そこで，プロトコルに依存する部分の機能ブロックを，スケーリング可能な論理ブロックとして予め

持っておき，これをソフトウェアで適宜組み合わせてプロトコルを構成する方式を提案した．本方式の近接無線通信装置の概念図を図 2-4 に示す．

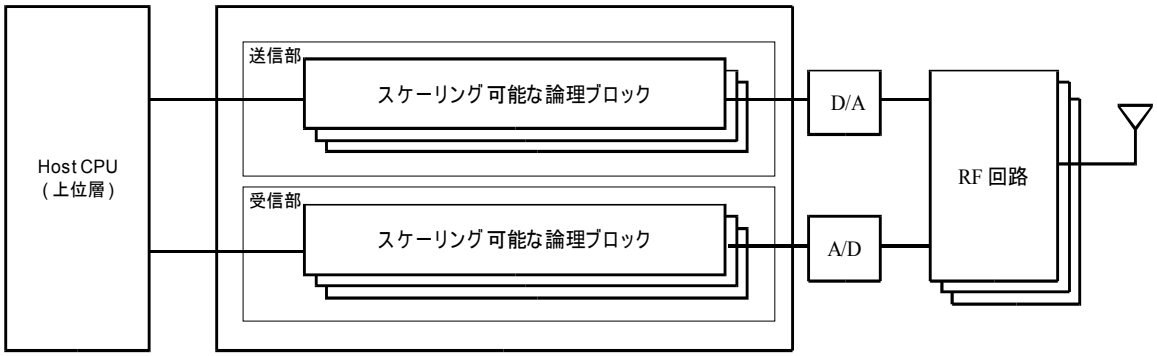


図 2-4　ハードウェアブロック選択式マルチプロトコルリーダライタの概念図

本方式では，パケットの構成・解釈，符号・複合化器といったプロトコルに依存する部分のハードウェアを予め持っておき，これをソフトウェアで適宜組み合わせてプロトコルを構成する．この方式は，ハードウェアの増加を最小限に抑えながら，複数のプロトコルに対応できる方法である．さらに，全てのプロトコル処理をソフトウェアで行う方式と比較して，高性能なプロセッサを不要にすることから，低消費電力化にも適した方法である．

4.2.3　FPGA による ISO18000-4 対応近接無線通信装置の試作

4.2.3.1　近接無線通信装置の概要

ハードウェアブロックの組み合わせによる近接無線通信回路を実証するため,FPGA により ISO18000-4 のプロトコルに対応した近接無線通信装置を試作した．

本試作の目的は，プロトコル制御部及びホスト CPU とのバス I/F をスケーラブル可能なハードウェアにすることにより，マルチプロトコル化した際の回路規模と消費電力を評価することである．ISO18000-4 近接無線通信装置の仕様を表 2-2 に示す．

表 2-2　ISO18000-4 近接無線通信装置の仕様

項目		仕様
ホスト (T-Engine)	CPU	ルネサステクノロジ SH7727
	CPU動作周波数	96MHz
	フラッシュメモリ	8MB
	SDRAM	32MB
	OS	T-kernel
バスI/F部	拡張バス周波数	48MHz
	メモリエリア	レジスタ ： エリア5 DMA ： エリア3
	バスプロトコル	レジスタアクセス ： SRAM (1soft wait) DMA ： SDRAM
	DMA	SH7727外部DMAリクエスト端子によるDMA 送受信データをCPUを介さずに自動転送可能
プロトコル制御部	通信プロトコル	ISO18000-4 mode 1
	転送レート	30～40kbps
RF部	キャリア周波数	2.45GHz (2.4000～2.4835GHz)
	チャネル帯域	0.5MHz (最大)
	変調方式	ASK/OOK
	符号化方式	Manchester (応答器→タグ) FM0 (タグ→応答器)
	送信電力	300mW

4.2.3.2　設計結果

FPGA による ISO18000-4 プロトコル対応近接無線通信装置の FPGA 設計結果を表 2-3 に，外観を図 2-5 に示す．

表 2-3　FPGA 設計結果

記述言語	Verilog RTL
記述量	約12000行（370kB） (wire宣言，コメント含む)
使用Logic Element数	10117 / 20060（50%）
使用端子数	295 / 301
動作速度	88.41MHz (ディレイのワースト値 11.311ns)
論理部(FPGA)消費電力	380mW

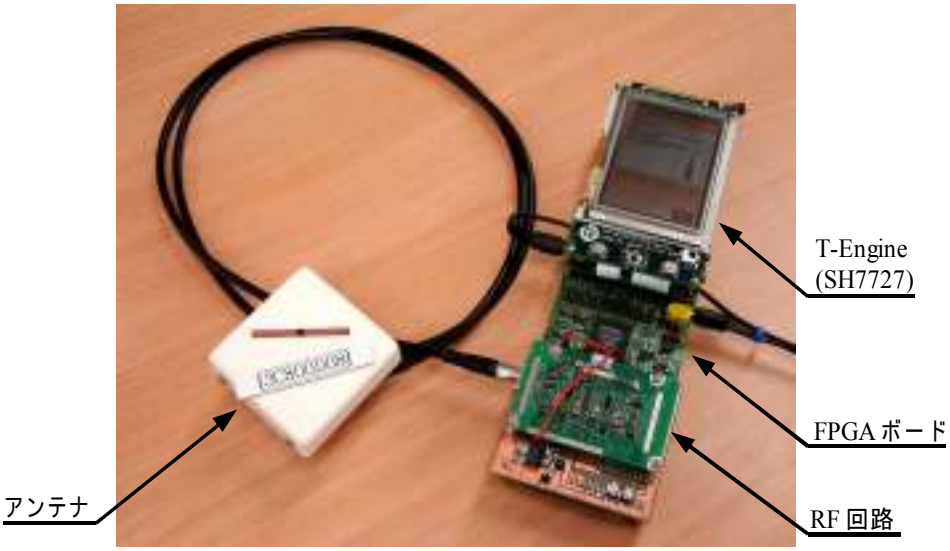


図 2-5　FPGA を使用した ISO18000-4 対応近接無線通信装置外観

設計結果では，FPGA の使用率は 50%であり，これをマルチプロトコルに拡張した場合でも実用的な論理量になることが確認された．さらに，FPGA 部分の実測消費電力は，380mW であり，携帯端末に搭載が可能である水準である．ASIC 化により更なる低消費電力化が見込まれる．

4.2.4　まとめ

ユビキタス社会では，非接触で情報のやり取りが可能である近接無線通信デバイス(RFID)を環境に配置してモノや場所を認識し，コンテクストアウェアネスを実現することが期待されている．この際，通信に適した周波数やプロトコルはアプリケーションに依存して様々で，プロトコルの違いをリーダライタ側で吸収し，様々な RFID を統一的に扱うリーダライタの技術開発が求められている．

様々な無線通信プロトコルにソフトウェアの変更のみで対応できる技術に，ソフトウェア無線技術があるが，プロトコル処理に高性能なプロセッサが必要となり，小型・低消費電力化に適さない．そこで，携帯端末に搭載するための小型・低消費電力に志向したソフトウェア無線技術として，物理層，データリンク層の機能を FPGA 等の論理回路上のハードウェアブロックで構成し，ソフトウェアでスケーリング，選択するソフトウェア無線技術を提案した．FPGA 上に ISO18000-4 の近接無線通信プロトコルを実装し，提案手法の回路規模と消費電力を評価した．回路規模は 10 万ゲート規模で一般的な FPGA や ASIC に実装可能であり，380mW の低消費電力を実現した．更に，提案手法を実装したマルチプロトコルリーダライタの基本設計を完了した．これは， eTRON カード(ISO18092 規格)，ISO15693 規格，ミューチップ，ISO18000-4 規格の 4 種のプロトコルに対応した近接無線通信装置である．H17 年度は，設計したマルチプロトコルリーダライタの実証システムを製作し，本方式の有効性を示す．

4.3 シームレス通信

近年，網間を移動しても端末に付与されたホームアドレスで通信できる，モバイル IP[1]が注目されている．モバイル IP では，ホームネットワークに存在するホームエージェントあるいはネットワーク上の通信相手が，端末宛の IP パケットを網が動的に端末に付与するアドレス(気付アドレス)に転送することで，移動先においても端末によるホームアドレスを使った IP 通信を可能としている．

しかしながら，モバイル IP では端末は移動の度にホームエージェント（あるいは通信相手）に気付アドレスの変更を登録する必要があり（これをハンドオーバ処理と呼ぶ），この処理に時間がかかるためその間通信が途切れるという問題があった．このため，ハンドオーバを高速化する方式が提案されている[2][3]が，移動先の予測が必要である，網内のルータが全て高速化方式に対応している，など利用条件が限定されていた．移動先の予測が不要な高速ハンドオーバ方式も IETF で検討されていた[4][5]が，網内のルータが全て高速化方式に対応していることは必要であった．そこで，本論文では，移動先の予測が不要で，かつ高速化方式に対応しない既存のルータが混在する環境でも動作する，高速なハンドオーバ方式について述べる．

4.3.1 モバイル IP プロトコル

4.3.1.1 モバイル IP

モバイル IP は，IP レベルでの移動に伴う位置の透過性を実現するプロトコルである．モバイル IP では，移動端末(MN) は本来属するホームネットワーク(HN) でのアドレスから離れ，別なネットワーク(FN)に接続した場合でも HN でのアドレス(ホームアドレス: Haddr) を使い通信の継続が可能である．これは，HN 上にホームエージェント(HA)が存在し，ホームアドレス宛の IP パケットを，MN が存在する網の特定の IP アドレス（気付けアドレス:CoA）宛に転送することで実現される．

モバイル IP にはモバイル IPv4(IPv4 用)とモバイル IPv6(IPv6 用)の 2 種類があり，動作が異なる．モバイル IPv4 では，移動先のネットワークにフォーリンエージェント(FA)が存在し，FA の IP アドレスが CoA となる．MN 宛 IP パケットは HA と FA との間に作られたトンネルを使って転送される．このため，移動先に FA が存在していることが前提となる．なお，移動先網での IP アドレスに余裕がある場合，移動先網で MN 毎に CoA を割り当てることで，MN が FA の代わりにトンネルの端点となることも可能である．

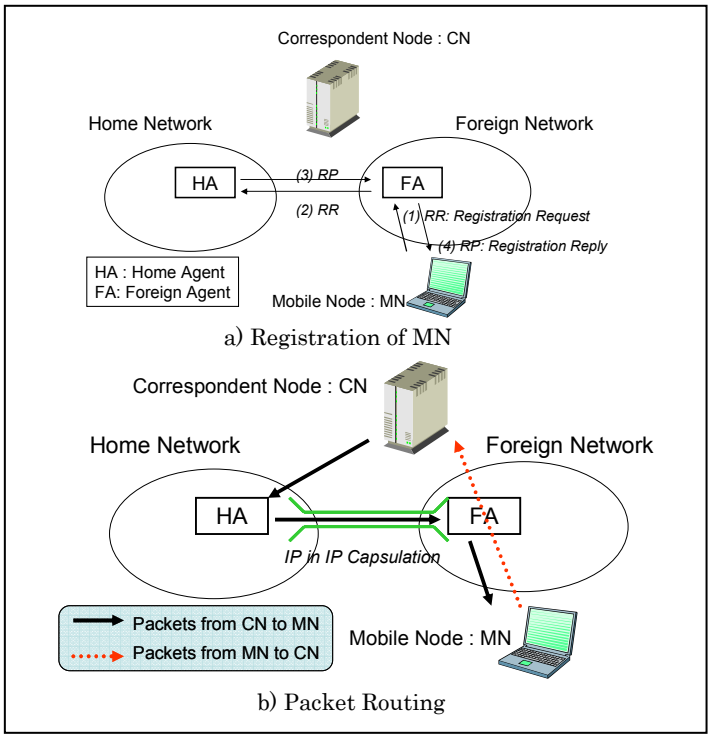


図3-1　モバイルIPv4の概要

一方 IPv6 はアドレス空間に余裕があるため，移動先で MN 毎に CoA を割り当てることが容易である．このためモバイル IPv6 では FA は用いられず，移動先で MN が割り当てられた IP アドレスが CoA となり，

Haddr 宛のパケットは IPv6 のルーティング機構により HA から CoA 宛に転送される．また，通信相手(CN)に対しても CoA を通知し，HA を経由しない通信を行なうことも可能となっている(経路最適化: Route Optimization)．

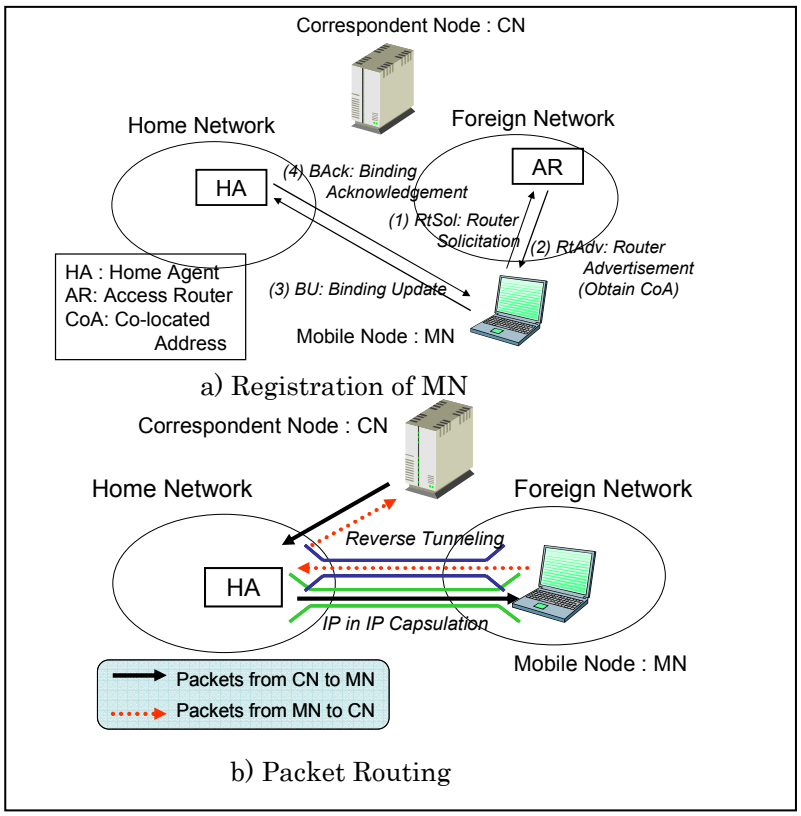


図3-2　モバイルIPv6の概要

本論文ではモバイル IPv6 をベースとした高速ハンドオーバ方式を提案する．このため以降では特に断りが無い限り，モバイル IPv6 を単にモバイル IP と呼ぶ．

4.3.2　従来のハンドオーバ高速化方式

モバイル IP は，MN が網を移動した際にホームアドレス宛の IP パケットの転送先を変更する処理（ここではハンドオーバ処理と呼ぶ）が必要となる．ハンドオーバ処理は以下のような手順からなる．

(i)　網の移動検出
(ii)　通信リンクの確立
(iii) 移動先での CoA の取得
(iv)　HA（および CN）への移動登録処理

これらの手順からなるハンドオーバ処理が完了するまでの間(無線 LAN を使用した場合での数秒間程度)については MN 宛の IP パケットは元のネットワークに転送され，MN が受け取ることはできない．利用者からは通信が行えない時間となる．そこでハンドオーバ処理時間を短くする方式がいくつか提案されている．なお上記のうち(ii)は通信リンク由来の処理時間であり短縮は難しい．但し，(i)と(ii)を並行して実施することで見かけ上短縮している場合はある．

4.3.3　高速ハンドオーバ方式の提案

4.3.3.1　解決すべき要件

前節で述べたように，従来の方式は，移動先のルータがその方式に対応していることが条件となる．そこで，本論文では，ハンドオーバによる移動先のネットワークを予測できない状況で，かつ通常のルータが混在したネットワーク上での高速ハンドオーバ方式を実現する．

4.3.3.2　解決法

ハンドオーバ処理のうち(i)の処理時間を短縮するために，MN が常にレイヤ 2 のリンク状態を監視しリンクの切断・再確立を検出し，ハンドオーバの契機とする．通常のネットワーク環境でも動作するためにネットワーク側には特にリンク状態検出等の機能を要求しない．

従来方式と同様にハンドオーバ前のネットワークのルータ(PAR) とハンドオーバ後のルータ(NAR)の間で双方向トンネルを設けることで，(iii)(iv)の処理の間にも PAR 経由での通信を可能とする．本方式では予め移動先がわからなくてもよいよう[5]と同様に MN からのハンドオーバ要求を契機として NAR から PAR にトンネルを設定する．

MN がリンク確立時に FMIP で定義された FBU(Fast Binding Update)メッセージをネットワークに対して送信するが，これはモビリティオプションで表現されるため，通常のルータでは無視される．このため，通常のルータが必ず反応するルータ要請(RtSol: Router Solicitation) に独自オプションとしてこのハンドオーバ要求を追加する．この場合，このオプションを解さない通常ルータは提案方式のオプションを含まないルータ通知(RtAdv: Router Advertisement)を返送するため，MN はすぐにルータが対応するか否かを判別できる．

トンネリングに必要な機能の一部を MN や HA が持ち，PAR や NAR が通常のルータであった場合，トンネルの終端処理を MN や HA が代替する．これにより通常のルータが混在する環境でも利用可能とする．

通信の信頼性を向上させるため，アクセスネットワークのルータ(AR)や HA，MN にオプションとしてパケットのバッファリング機能を持たせる．

なお，(ii)については，本方式では特に何も行わない．また (iii)については，IPv6 のステートレスアドレスを使用しアドレス重複検出を省略することで処理時間を短縮した．

4.3.3.3　高速ハンドオーバ動作の概要

ネットワークとして図に示す構成を想定する．また MN は常にレイヤ 2 のリンク状態を監視しリンクの切断・再確立を検出可能とする．

MN はリンクの切断状態からリンク確立を検出することで，網を移動した可能性があると判断し，高速ハンドオーバ処理を起動する．以下の手順となる．

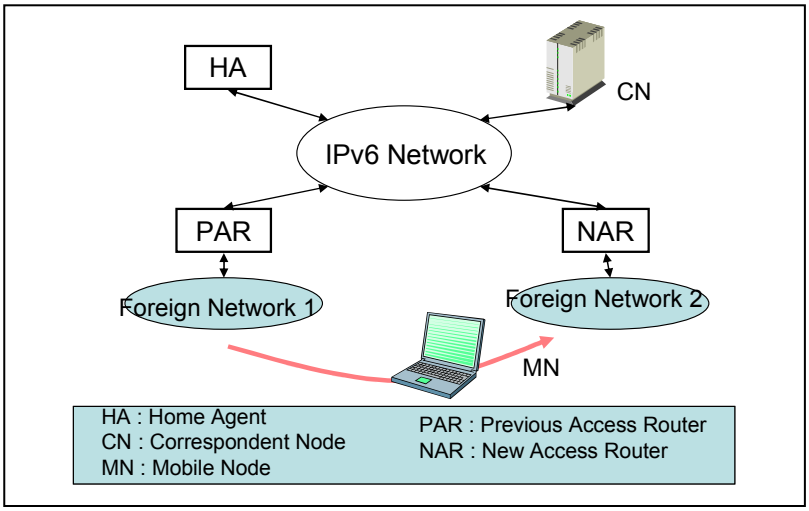


図3-3　IPv6ネットワークの例

(1) まず MN は通常の IPv6 と同様に RtSol を送信して CoA 取得などを試みる．この際に MN は提案方式の適用を示すオプションを RtSol に付加して送信する．なお，タイミングによっては RtSol 送信前にルータが定期的に送信する RtAdv を受信する場合があるが，この場合，通知を無視して改めて RtSol メッセージを送信するようにする．

(2) MN は NAR からの RtAdv 中の提案方式のオプションの有無で NAR が対応ルータか判別できる．MN の移動元，移動先ネットワークがホームネットワークであるか否かによって，また PAR，NAR の各ルータが提案方式に対応したルータ(PARp，NARp)か通常のルータ(PARc，NARc)かにより 1 に示す 8 通りの場合がある．

(2-a) FN 上の対応ルータ間の移動:

高速ハンドオーバと同様に PARp と NARp 間でトンネルを設定するが，この際 NARp から PARp へトンネル設定を行なう．

(2-b) 対応ルータから通常ルータへの移動:
MN 自身が CoA 取得後に PARp へトンネルを設定する．その後，PARp のバッファからパケットを受け取り，加えて BU 完了まで PCoA を使用した通信を再開する．また，HA への BU メッセージで HA にバッファリングの開始を指示し，MN のハンドオーバ完了後は HA がバッファリングを行なう．

(2-c) 通常ルータから対応ルータへの移動:
NARp から HA へトンネルを設定し，同様にして NARp は HA のバッファから受取り MN へ送り，加えて BU 完了までの間 PCoA を使用した通信を再開する．そして，HA でのバッファリングから NARp でのバッファリングに切り替える．

(2-d) 通常ルータ間の移動:
(2-b)とほぼ同様の手順となるが，MN からのトンネルの設定先が PARp ではなく HA となる．

4.3.4 実装と評価

提案方式を FreeBSD 上の KAME および T-Kernel 上の KASAGO という 2 つの IPv6 モバイル IP プロトコルソフトウェアを改造して実装し，評価を行った．評価用のネットワークを次図に示す．HA，CN および 2 台の AR とルータを接続した．評価用のトラフィックとして CN から HA，AR あるいはルータに接続している MN に対して UDP パケット(Port:10000 番)を 100 ミリ秒ごとに送信し，MN がその応答を返すようにした．通常ルータとしてはルータから右に出ているアクセスネットワークを使用した．アクセスネットワークへの接続には IEEE802.11b 規格の無線 LAN と 100Base-TX の有線 LAN のいずれかを使えるようにした．ハンドオーバ時間の測定は，CN からの送信 UDP パケットに対する応答の欠落数で測定した．このため，この評価での誤差は送信間隔の 100 ミリ秒である．また AR と MN 間を流れるパケットを Ethereal により捕捉し計測を行った．なお，ハンドオーバの契機となる網間の MN の移動は，有線 LAN の場合は，手動での物理的な LAN ケーブルの繋ぎ替えで，また無線 LAN の場合は無線 LAN アクセスポイントの電源断により実現した．各項目は 10 回の試行を行い平均や分散を測定した．さらに，MN 側の IP プロトコルソフトウェアを無改造の KASAGO に入れ替えることで，通常のモバイル IP 環境とした．

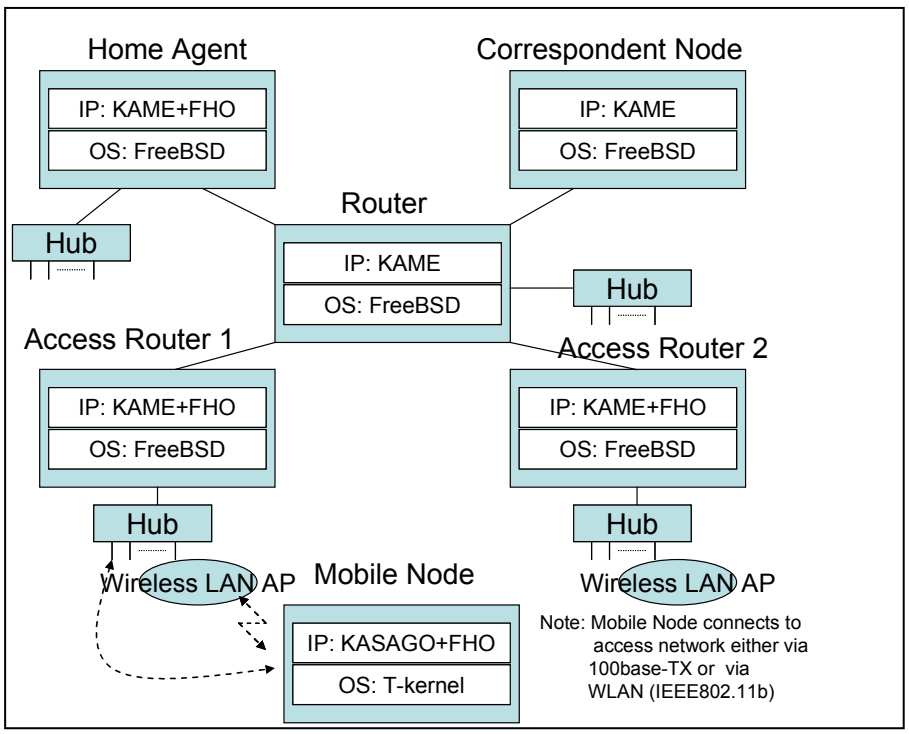


図 3-4 評価システムの構成

次表に有線 LAN をアクセスに用いた場合のハンドオーバ時間を示す．従来のモバイル IP では，ハンドオーバに平均 5.8 秒必要としており，また所要時間のばらつきも大きい．なお，この必要時間には手

動での LAN ケーブル繋ぎ変え時間（平均で 1.5 秒）を含んでいる．一方，提案方式では，対応ルータ間，ホーム・対応ルータ間，対応ルータ・非対応ルータ間など全ての組み合わせで，同様に手動でのケーブル繋ぎ変え時間を含み約 1.9 秒でハンドオーバ処理が完了した．ハンドオーバ時間の変動も殆ど見られなかった．手動での LAN ケーブル繋ぎ変え時間 1.5 秒を除くと，実質上 400 ミリ秒程度で切り替わっているといえる．従って提案方式では通常のモバイル IP と比較してハンドオーバ時間が約 1/3 に減少しており，提案方式の優位性が実証できた．また，FMIP を解さない通常のルータに対しても高速なハンドオーバが実現できていることが確認できた．

表 3-1　有線 LAN 上でのハンドオーバ時間

		平均値(秒)	分散	最大値(秒)	最小値(秒)
提案方式	対応ルータ間	1.9	0.001	1.9	1.8
	ホーム→対応ルータ	1.9	0.002	2.0	1.9
	対応ルータ→ホーム	2.0	0.003	2.0	1.9
	対応ルータ→通常ルータ	2.0	0.003	2.0	1.9
	通常ルータ→通常ルータ	2.0	0.003	2.1	1.9
従来のモバイル IP	ルータ間	5.8	2.2	8.1	4.2

次表に無線 LAN 上でのハンドオーバ時間を示す．提案方式の対応ルータ間と従来方式の比較のみを示した．従来のモバイル IP ではハンドオーバに平均 6.3 秒かかるのに対し，提案方式では 2.9 秒と約３秒短くなっている．

表3-2　無線LAN上でのハンドオーバ時間

		平均値(秒)	分散	最大値(秒)	最小値(秒)
提案方式	対応ルータ間	2.9	1.6	5.4	1.9
従来のモバイル IP	ルータ間	6.3	2.3	8.51	4.4

4.3.5　まとめ

既存の通常ルータが混在する環境においてモバイル IP ハンドオーバ処理を高速に行なう方式を提案した．従来検討されている方式と異なり，移動先の予測が不要であり，かつ本方式に対応しないルータが混在しても高速なハンドオーバ処理が行える２つの特徴を兼ね備えている．またバッファリングを行なうことでパケットロスを抑えることも可能である．

FreeBSD 上の KAME および T-kernel 上の KASAGO という２つの IPv6 モバイル IP プロトコルソフトウェア上に実装を行い，評価を行った．その結果，本方式に対応したルータ間を MN が移動した場合，通信リンクの確立検出後 400 ミリ秒程度でハンドオーバが完了し，従来のモバイル IP で約 4 秒から 6 秒程度は必要としていたハンドオーバ時間を大幅に短縮できた．また提案方式に対応しない通常のルータが混在した環境でも高速なハンドオーバが実現できていることが確認できた．

参考文献

[1] Johnson, D., Perkins, C. and Arkko, J.: Mobility Support in IPv6, IETF RFC3775, (2004).

[2] Koodli, R. (Ed.): Fast Handovers for Mobile IPv6, IETF Internet Draft, draft-ietf-mipshop-fast-mipv6-03.txt, (2004).

[3] Soliman, H., Catelluccia, C., El Malki, K. and Bellier, L.: Hierarchical Mobile IPv6 mobility management (HMIPv6), IETF Internet Draft, draft-ietf-mipshop-hmipv6-04.txt, (2004).

[4] Kempf, J., Calhoun, P., Dommety, G., Thalanany, S., Singh, A., McCann, P. J. and Hiller, T.: Bidirectional Edge Tunnel Handover for IPv6, ETF Internet Draft, draft-kempf-beth-ipv6-02.txt, (2001).

[5] Jung, H. Y., Min, J. H., Kang, H. G. and Lee, C. Y.: Fast Handoff with Chain Tunneling for Mobile IPv6, IETF Internet Draft, draft-jung-mobileip-fastho-chain-00.txt, (2002).

4.4 ユビキタス情報提供・制御用プロトコル

4.4.1 今年度の取り組み範囲

本年度は，ユビキタス情報提供・制御用プロトコル（以下，「本プロトコル」と呼ぶ）を実装した機器を数十台製造しセンサーアクチュエータネットワークシステムを構築し，機能評価・システム評価・性能評価を行う．

4.4.2 評価システム構成

本プロトコルの評価を行うにあたっては，数種類のハードウェアを試作開発してそのハードウェア上に適宜ソフトウェアを実装した．図 4-1 に，評価システムのシステム構成を示す．

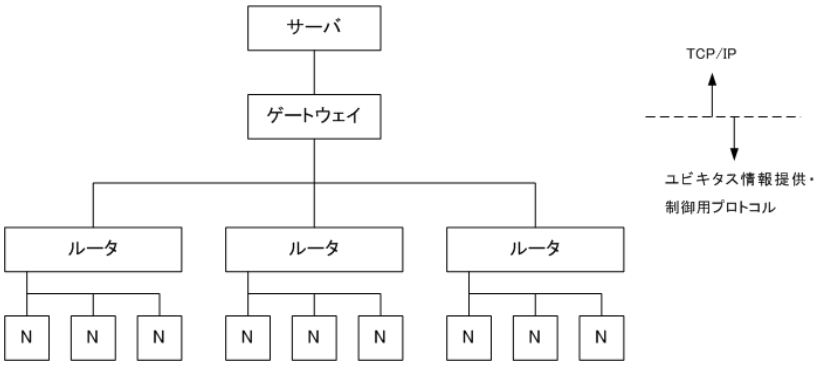


図4-1　評価システム構成概要

評価システムは，サーバ，ゲートウェイ，ルータ，および小型ネットワークノードNで構成される．以下に，各構成要素の概要について説明する．

4.4.2.1 小型ネットワークノードN

小型ネットワークノードNは，本プロトコルを使ったシステムにおける中心的な構成要素である．小型ネットワークノードは，n-Engine アーキテクチャをベースとして開発を行った．n-Engine アーキテクチャとは，T-Engine フォーラムと呼ばれる企画推進団体が定義するアーキテクチャである．図 4-2 に写真を，図 4-3 に主要機能ブロック図を示す．

図 4-2　小型ネットワークノード

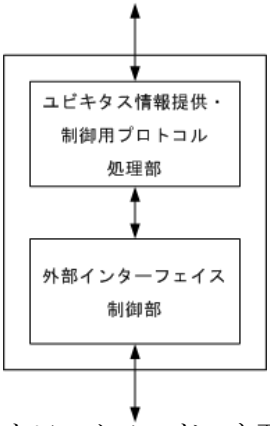


図 4-3 小型ネットワークノードの主要機能ブロック図

各種外部インタフェースから収集した情報を「ユビキタス情報提供・制御用プロトコル処理部」からネットワークに流したり，逆に「ユビキタス情報提供・制御用プロトコル制御部」から受信したデータをもとに各種外部インタフェースに接続される機器を制御する．

4.4.2.2 ルータ

ルータは，本プロトコルを使ったシステムにおいてネットワークリソースを効率的に利用するために必要不可欠な構成要素である．ルータという呼称についてであるが，本ルータは本プロトコルを処理す

る装置であり，TCP/IP パケットをルーティンするわけではない．図 4-4 に写真を，図 4-5 に主要機能ブロック図を示す．

図 4-4　ルータ

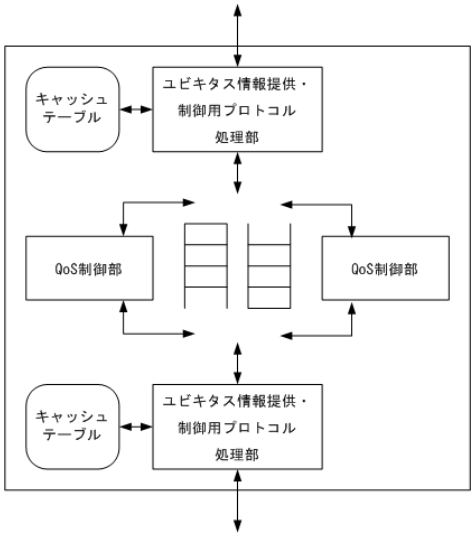


図 4-5　ルータ主要機能ブロック図

ルータのハードウェアは基本的に小型ネットワークノードのデバイスを使用している．以下に，ルータの機能的な特徴を示す．

- ネットワークを流れるパケットを監視する
- 配下の小型ネットワークノードの情報をキャッシュする
- 必要に応じてルーティング・フィルタリング処理を行う
- セキュリティ層を使ったセキュアなルーティング
- ネットワーク中のトラフィックを制御する（QoS機能）
- 配下ドメインのネットワーク制御を行う

4.4.2.3　ゲートウェイ

ゲートウェイは，本プロトコルと既存のネットワーク（TCP/IP）との相互接続を行って効率的に保守運用を行うために必要不可欠な構成要素である．ゲートウェイはハードウェアはルータとまったく同じもの（図 4-4）を使っており，ディップスイッチの切り替えによってソフトウェア的にルータとゲートウェイの機能を切り替えられるようになっている．ゲートウェイにおけるプロトコルスタックの関係は，図 4-6 のようになる．

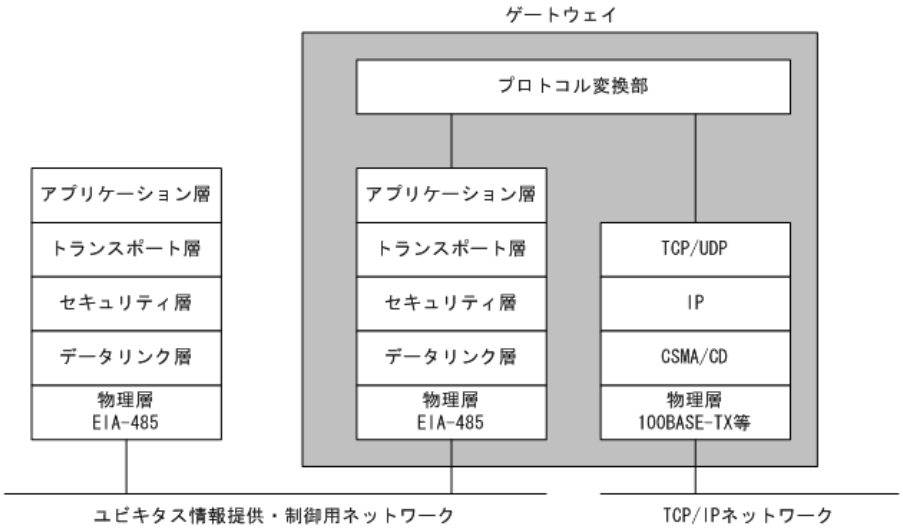


図 4-6　プロトコルスタック関係図

4.4.2.4　サーバ

サーバはゲートウェイを介して本プロトコルのネットワークと接続し，保守運用を行うために必要不

可欠な構成要素である．サーバには一般的なLinuxPCを使用し，構成管理や障害管理用のRDBMSおよびそのユーザインタフェースとしてwebサーバ(apache)とwebブラウザを搭載する．

4.4.3 評価内容

機能評価は，主に図4-7のような構成で実施した．

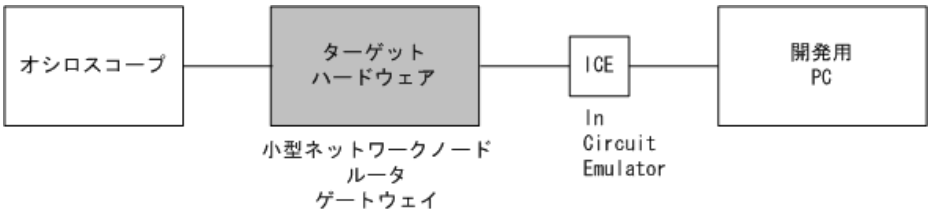


図4-7　機能評価時の構成

今回のシステムでは物理層のケーブルおよびコネクタからプロトコルスタック一式（ハードウェア・ソフトウェア），周辺装置（ルータ・ゲートウェイ，サーバ）の開発となったため，機能評価内容としては非常に多岐に渡った．

評価を効率よく実施するため，評価順序は下記の通りとした．

- LSIレジスタアクセス（データリンク層）
- LSIデバイスドライバ（データリンク層）
- 通信ケーブル，コネクタ（物理層）
- データリンク層
- プロトコルスタック上位層（セキュリティ層，トランスポート層，アプリケーション層）
- ゲートウェイ（ハードウェア・ソフトウェア），サーバ
- ルータ（ハードウェア・ソフトウェア）

4.4.4 課題

- コネクタの不具合
  今回は上記の小型ネットワークノード用に通信用のコネクタを新規に製造した．要件としては小型でデイジーチェイン接続が可能なものとした．基板側のコネクタは特に問題なかったが，ケーブル側のコネクタは構造が比較的複雑になったため，圧接不良が多発して評価を進められない期間があった．コネクタハウジング内の構造を見直して確実にケーブルを圧接できるようにする必要がある．
- 通信ケーブルの入手性
  コネクタを新規設計したことにともなって，通信ケーブルも新規設計となった．このため必要なときにすぐケーブルを入手することができず入手性が悪くなってしまった．
- LSI不具合
  LSIの致命的な不具合が２件ある．１つはパケットのCRCの値がプリアンブルの値と同じになった場合にネットワークが再構築されてしまうというもの．１つは，受信したパケットのドメインIDが自分と違う値であっても，ノードIDが自分と同じ値の場合にネットワークが再構築されてしまうというもの．現状ソフトウェアにて救済できているが，恒久対処にはならないため，LSIの改版が必要．
- システム評価・性能評価未完
  当初はシステム評価と性能評価も終える予定であったが，上記課題があったために機能評価に時間がかかり，終えることができなかった．来年度も継続して評価を行う予定である．

4.4.5 まとめ

本年度は，ユビキタス情報提供・制御用プロトコルを実装した機器を数十台製造しセンサーアクチュエータネットワークシステムを構築し，機能評価を行った．評価の結果，ハードウェア・ソフトウェアともにいくつかの不具合があったが，最終的には本プロトコルが具備すべき機能は全て実現できていることを確認した．

4.5 セキュアチップを用いた VPN 構築

4.5.1 はじめに

近年，情報のセキュリティが重要視される中，インターネットに代表される公開通信網の上に暗号技術により保護された仮想的な専用通信路を確立する VPN（Virtual Private Network）技術が普及しつつある．その主な通信ノードは，パーソナルコンピュータや専用ルータ，エンタプライズサーバというような豊富な計算資源を具備した機器である．

一方で，現在急速に研究開発が進められているユビキタスコンピューティング環境では，生活空間のあらゆるところに大小機能さまざまな組み込み機器が浸透し，それらがさまざまな通信網を介して生活に関わる情報を相互にやりとりする．このようなさまざまな組み込み機器間のデータ通信にも，上記 VPN のようなセキュリティ保護が必要となる．

我々は，秘密共有に必要な演算処理や鍵情報管理の一切を担い，対象機器に外部から秘密共有機能を提供するセキュアチップ（図 5-1）を研究，開発した．またこのチップを駆使し，組み込みノード間で VPN を構築する上で最も基本的かつ必須な機能である，ノード間での共有秘密暗号通信路を確立するためのプロトコルの研究，実装を行った．

図 5-1　セキュアチップ

4.5.2 セキュアチップを用いた組み込み機器間での暗号通信路構築手順

セキュアチップは Diffie-Hellman 鍵共有アルゴリズムを実現するために必要な以下の VPN 共有鍵構築支援機能を提供する．

1. コミット値演算機能：秘密疑似乱数を内部生成し，その乱数からべき乗剰余値（以下「コミット値」と呼ぶ）を演算し，その結果を返す機能
2. 公開鍵読み出し機能：チップが保有する公開鍵証明書を読み出す機能
3. 公開鍵検証機能：相手ノードから入手した公開鍵証明書を検証する機能
4. 共有鍵演算機能：相手ノードから入手したコミット値を受け取り，このコミット値と 1.で生成した秘密疑似乱数と 3.で入手した相手ノードの公開鍵から共有機密鍵を演算する機能

いま，組み込み機器 P，Q がそれぞれセキュアチップ A，B を接続しているものとする．また，組み込み機器 P とセキュアチップ A，および組み込み機器 Q とセキュアチップ B の間は，peer-to-peer で価値情報を安全に伝搬するための通信規約として研究している eTP（entity Transfer Protocol）に基づく通信を行っているものとする．このとき，組み込み機器 P，Q はセキュアチップ A，B が提供する VPN 共有鍵構築支援を用いて，以下のように暗号通信路を構築する（図 5-2）．

1. 組み込み機器 P はセキュアチップ A に対して，コミット値演算要求を発行する．セキュアチップ A は，コミット値演算機能を用いて秘密疑似乱数 a とコミット値 A を生成する．セキュアチップ A は，秘密疑似乱数 a を保持しコミット値 A を組み込み機器 P に返す．
2. 組み込み機器 P はセキュアチップ A に対して，公開鍵証明書読出要求を発行する．セキュアチップ A は，公開鍵読み出し機能を用いて自身が保有する公開鍵証明書 A を組み込み機器 P に返す．
3. 組み込み機器 Q とセキュアチップ B の間でも，手順 1，2，3 に相当するインタラクションを行う．
4. 組み込み機器 P はセキュアチップ A から入手したコミット値 A と公開鍵証明書 A を組み込み機器 Q に送付する．同様に，組み込み機器 Q はセキュアチップ B から入手したコミット値 B と公開鍵証明書 B を組み込み機器 P に送付する．
5. 組み込み機器 P は，組み込み機器 Q から入手した公開鍵証明書 B をセキュアチップ A に与え，公開鍵検証要求を発行する．セキュアチップ A は，組み込み機器 P から入手した公開鍵証明書 B を

検証し，検証成功／失敗のいずれかを組み込み機器 P に返す．検証に成功した場合，セキュアチップ A は公開鍵証明書 B に格納された公開鍵 B を保持する．

6. 組み込み機器 P は，組み込み機器 Q から入手したコミット値 B をセキュアチップ A に与え，秘密共有鍵演算要求を発行する．セキュアチップ A は，組み込み機器 P から入手したコミット値 B と，これまでの過程で保持している自身の秘密疑似乱数 a，公開鍵 B から秘密共有鍵 K を演算し，これを組み込み機器 P に返す．
7. 組み込み機器 Q とセキュアチップ B の間でも，手順 6，7 に相当するインタラクションを行う．

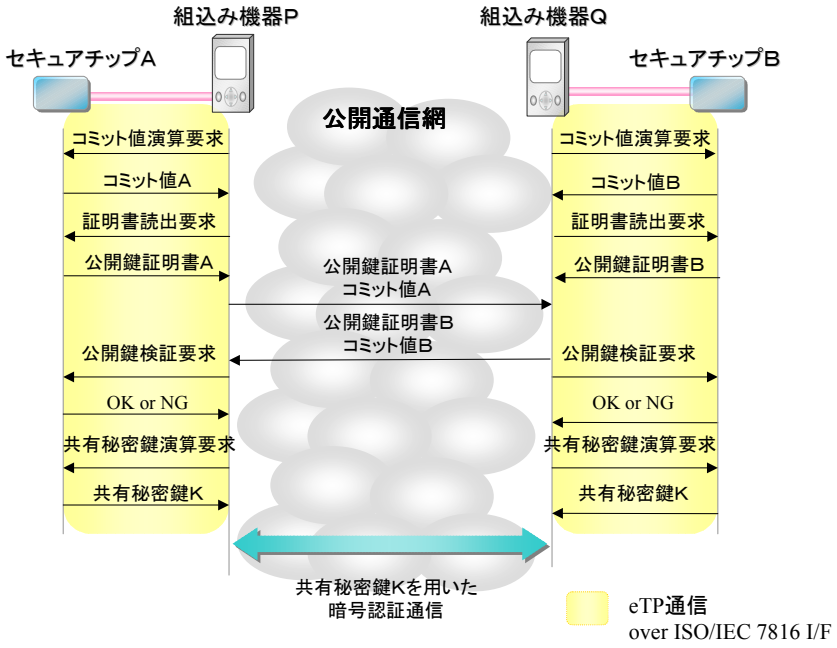


図 5-2　セキュアチップを利用した組み込み機器間での暗号通信路構築

4.5.3　組み込み機器間での暗号通信路ソケット

4.5.2 節に述べたセキュアチップを利用した組み込み機器間での暗号通信路構築手順を検証するため，暗号通信路ソケットライブラリを実装した．暗号通信路ソケットライブラリは，通常のソケットライブラリを拡張した構成になっており，通常のソケット通信のそれと同等のインタフェースを提供する．

4.5.3.1　暗号通信路ソケットのフェーズとメッセージフォーマット

暗号通信路ソケットは，鍵交換・暗号通信路構築フェーズと)暗号化メッセージ通信フェーズの 2 つの状態を持つ．

鍵交換・暗号通信路構築フェーズ，暗号化メッセージ通信路フェーズのいずれのフェーズでも，暗号通信路ソケットは，送受信メッセージを 4 オクテットからなる長さフィールドと MAC フィールドでカプセル化している．長さフィールドは，メッセージ本体のオクテットは数を示し，暗号化メッセージ通信フェーズにおいては暗号化前のメッセージ長を示す．長さフィールドが負の値である場合，これは認証並びにセキュアチップとの通信においてエラーが発生したことを示す．MAC フィールドは MAC アルゴリズムに基づく MAC 値を格納するフィールドである．本実装では鍵交換・暗号通信路構築フェーズの MAC アルゴリズムとして CRC16 を，暗号化メッセージ通信フェーズの MAC アルゴリズムとして HMAC with SHA-1 を採用した．

4.5.3.2　評価と今後の課題

実装した暗号化通信路ソケットライブラリを用いて，通常のソケット通信プログラムとほぼ同等の API を用いて，セキュアチップが接続された組み込み機器間で暗号通信路を構築し，暗号通信が正常に行えることを確認した．これによって，4.5.2 節に述べたセキュアチップを利用した組み込み機器間での暗号通信路構築手順が正常に動作することが立証された．

今回の実装では，暗号化通信路を構築する際に，組み込み機器間で毎回コミット値や公開鍵証明書を交換している．一方で，組み込み機器間の接続の場合，パーソナルコンピュータや専用ルータ，エンタ

プライズサーバなどの間の接続に比べ，接続先が既知である場合が多い．このため，接続先の公開鍵証明書をあらかじめ配布しておくことで，暗号通信路構築時の公開鍵証明書交換を省略することが可能である．

また，暗号化通信路構築の高速化のため，以前に構築した暗号化通信路の秘密共有鍵を用いて認証するなどの軽量プロトコルも必要であるが，一方でこれらは接続ポリシーに依存する．今後は，今回実装した暗号化通信路ソケット機能にポリシネゴシエーション機能を追加し，組み込み機器がセキュアチップを駆使し，さらに汎用的に使用できる VPN 構築機能を有するための研究・開発を進めたい．

4.6 uTAD

4.6.1 uTAD／Contents

4.6.1.1 はじめに

ユビキタス情報社会では，ネットワークで接続されたコンピュータが，人やモノ，周囲の環境の状態などのコンテクストを認識し，人や状況に合わせた最適なサービスを提供する．このような，現実世界の状況に基づいて情報処理を行うことをコンテクストアウェアネスと呼ぶが，その実現のためには，環境中に埋め込まれた，多数のセンサーや端末コンピュータ，あるいは情報を蓄積しているサーバなどのノードが情報を交換し合い，それを適切に処理する必要がある．そのため，統一された情報の表現形式が求められる．

uTAD/Contents は，さまざまなモノの情報を記述するための標準データ形式である．ここで言うモノとは，具体的な一つ一つの商品や物体だけでなく，サービスに関する情報や，物のクラスに共通のメタ情報，あるいはセンサーからの情報など，情報として区別する必要があるもの全般を含むものとする．

4.6.1.2 uTAD/Contents の利用シーン

ノード間で交換される情報のうち，センサーノードで取得された各種コンテクスト情報，機器を制御するための制御情報などは，主に数値情報として扱うことが出来る比較的単純な情報である．一方，より上位のアプリケーションで扱われる情報は，人や商品の属性情報，流通履歴など，一定の構造を持った複雑な情報を扱うことになる．どちらの場合でも，応用ごとに適したデータ表現形式があり，それを抽象化した場合，データ形式の変換や解釈のためのオーバーヘッドが発生し，効率は悪くなる．そこで，個々のアプリケーションやシステムの中だけで利用される情報に関しては，標準データ表現形式の利用は想定せず，それぞれに特化した形式で行うものとする．一方，ユビキタス情報環境において，個別に作成された様々なアプリケーションやシステムが情報を交換する場合には，膨大な数のノードが相互に関係し，また，新しいサービスやシステムが追加されたりするため，共通の標準データ表現形式を利用することが要求される．

uTAD/Contents は，このようなユビキタス情報環境の様々なシステム間で情報をやり取りするための，標準データ表現形式として設計されている．uTAD/Contents を用いた情報交換の仕組みは，以下のようになる．

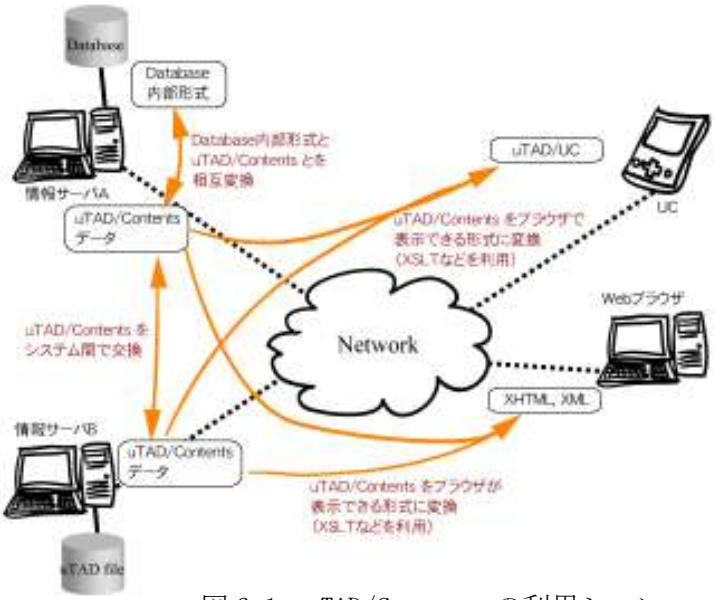


図 6-1　uTAD/Contents の利用シーン

4.6.1.3 uTAD/Contents の設計方針

標準データ表現形式の適用範囲となる情報としては，センサーノードからの数値情報のように単純な形式のものから，流通追跡などの大規模なアプリケーションで利用される複雑なデータベース形式の情報まで，様々なものが想定される．センサーノードなどの小型ノードで処理可能とするためには，単純な表現形式が要求される．一方，データベース形式などを取り扱う場合は非常に複雑な構造となり，また，常に新しいアプリケーションが現れるということを考慮すると，標準データ表現形式は，非常に単純な基本構造をベースとし，必要に応じて拡張できる形式が望ましい．

次に，複雑なアプリケーションに対しては，すでにいくつかの標準コード体系や，標準フォーマットが採用されている場合も多い．そのような標準体型を全て否定し，新たに標準データ形式を作成するのは，効率が悪い．そのため，標準データ形式としては，既存の標準コード体系や標準フォーマットとの親和性の高い形式であることが望ましい．

以上の要求を満たすため，uTAD/Contents は RDF/XML [4] を利用して記述することとした．RDF [2] とは，メタデータなどのリソース相互の関係を，特定のアプリケーションに依存しない形で記述する標準の形式で，情報をリソース（主語）とプロパティ（述語），そしてプロパティの値（オブジェクト，目的語）の三項関係で記述するデータモデルであり，RDF/XML は，それを XML の形式で表記したものである．uTAD/Contents による記述例は図 6-2 のようになる．

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'
  xmlns:utad='urn:utad:schema:utad:base:0:0:0#'
  xmlns:pc='urn:utad:schema:pc:base:0.0.0#'>

  <pc:pc rdf:about='ucode:00112233445566778899aabbccddeeff'>
    <utad:label xml:lang='ja'>デスクトップ PC FOOBAR-200 model A</utad:label>
    <utad:documentation xml:lang='ja'>デスクトップ PC 記述例 1 </utad:documentation>

    <utad:documentationSource>http://www.example.com/desktoppc/example.html</utad:documentatio
    nSource>
    <utad:prototype rdf:resource='ucode:0123456789abcdef0123456789abcdef'/>
    <pc:memory rdf:parseType='Resource'>
      <rdf:value>2<rdf:value>
      <utad:unit rdf:resource='urn:utad:units:datacapacity#GByte' />
    </pc:memory>
  </pc:pc>
</rdf:RDF>
```

図 6-2　uTAD/Contents 記述例

uTAD/Contents では，基本的なプロパティのセットを uTAD core 仕様として規定する．これには，モノの名称，包含関係，履歴情報の記述，単位付き値の表記などが含まれている．そして，さまざまなモノや情報に関しては，それぞれクラスを定義し，RDF Schema [3] によって語彙を拡張することで対応する．RDF/XML を利用して記述することで，必要な語彙を自由に拡張することが可能となっており，単純な情報記述から複雑な情報記述までを統一的に扱う，高い拡張性を持たせることが出来る．また XML による記述となるため，XHTML や SVG[6]などのような XML を利用して定義されているの既存の標準フォーマットや，RSS 1.0[5]，Dublin Core[1]などの既存の RDF による標準フォーマットを，uTAD/Contents の記述中に取り込む，あるいは逆に各フォーマットでの記述に uTAD/Contents を埋め込むことが可能であり,既存のデータフォーマット体系との親和性の高い記述形式となっている.さらに uTAD/Contents ではプロトタイプベース・オブジェクト指向を取り入れている．プロトタイプ記述を利用することで，共通部分の多いコンテンツの記述を行う場合でも，プロトタイプ参照によって共通部分の記述をまとめられるため，小さなデータサイズでの記述を行うことが出来る．

4.6.1.4 uTAD/Contents 仕様

(1) モノのクラス

uTAD/Contents では，モノに関するさまざまな情報を，プロパティとして扱う．そして，モノに付随する各プロパティに関して，その値を列挙してゆくことで，モノに関しての情報記述を行う．しかし，モノの情報記述に必要なプロパティは，モノの種類や，その情報を何ために利用するかによって，大きく異なる．たとえば「乗用車」に関しての情報であれば「乗員数」「排気量」「最大積載量」などのプロパティが必要であるが，「食品」では「賞味期限」「栄養表示」などが必要である．

このように，様々なモノに関する情報記述を，様々な目的のために行う必要がある．そのため，uTAD/Contents では，モノに関する情報記述を，モノの種類や情報の利用目的に対応した，さまざまなクラスに分類する．そしてクラスごとに必要なプロパティを定義し，それを利用して情報記述を行う．

uTAD/Contents では，uTAD/Contents で標準的に必要なプロパティを定義した uTAD core クラスを基底クラスとして，そこから階層的に様々なモノのクラスを派生させる．例えば図 6-3 では，uTAD core クラスから「食品」というクラスを派生させており，そこでは「栄養素」というプロパティが定義されている．さらに「食品」クラスから「野菜」クラスと「加工食品」クラスが，さらに「野菜」クラスから「大根」クラスと「みかん」クラスが派生しており，それぞれで，必要なプロパティが追加されている．この例では，階層的なクラス継承のみが挙げられているが，必要に応じて，多重継承を行うことも可能である．たとえば「デジタルカメラ付き携帯電話」であれば「デジタルカメラ」クラスと「携帯電話」クラスの二つのクラスを多重継承することになる．

なお，uTAD/Contents 仕様では，uTAD core クラスのプロパティに関しての定義と，クラスを派生させるための仕組みに関しては規定するが，個々のクラスに関しては，業界団体や個々の店舗ごとに規定することとする．

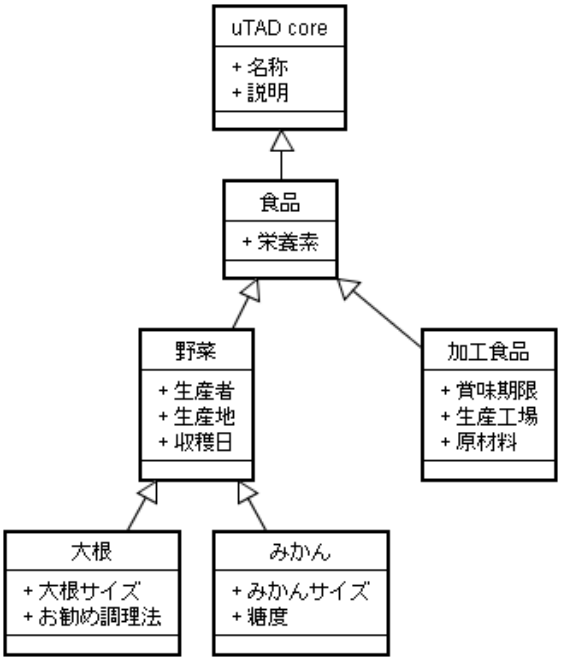


図 6-3　uTAD/Contents のクラス

(2) プロトタイプ

モノのプロパティはクラスごとに拡張できるが，ほとんどの部分が同じで，一部分だけが違うようなモノのクラスが多数作られることが考えられる．例えば，「デスクトップ PC」というクラスに対して，そのサブクラスとして，様々な機種のクラスが考えられる．また，同じ機種のクラスでも，CPU 速度やメモリ量，ハードディスク容量などの組み合わせによって，非常に多くのモデルのバリエーションが考えられる．さらに，同じモデルのクラスに所属する PC に関しての記述を考えてみると，大部分はまったく同じで，シリアル番号などのごく一部のみが異なるような uTAD/Contents 記述が大量に存在することになる．このように，「大部分がまったく同じであるが，一部のプロパティの値だけが違う」記述が大量に必要になると，情報記憶のための容量が無駄である．

そこで uTAD/Contents では，プロパティの拡張に関してはクラス継承による拡張を行うが，それ

と同時に，プロトタイプベース・オブジェクト指向を取り入れる．プロトタイプを用いることで，モノのクラスが必要以上に細分化してしまうような場合は，適当な位置でクラスの継承を止め，そこから先は，クラスにしたがって記述された他の uTAD/Contents 記述をプロトタイプとし，それをコピーして，さらに必要なプロパティを追記，あるいは書き換えすることで，新しい uTAD/Contents 記述を作ることが出来る．プロトタイプを利用することで，共通部分の記述が大きくなっても，その部分はプロトタイプの参照を行うだけで記述することが出来るようになり，データサイズを大幅に削減することが出来る．

(3) uTAD core クラス

uTAD core クラスは，全ての uTAD/Contents のクラスの基底クラスとなるもので，全ての uTAD/Contents での記述に必要な基本的なプロパティを定義している．uTAD core で規定されているプロパティは，以下のものである．

utadVersion	記述に用いられている uTAD/Contents 仕様のバージョン番号.
label	記述対象となっているものや情報の名称.
documentation	モノや情報に関する説明文.
documentationSource	モノや情報に関する説明文を，URI によって参照する場合はこのプロパティを用いる
comment	uTAD/Contents 記述に関するコメント．データを処理する際は無視される.
isPrototype	プロトタイプ記述であるかどうかの区別のためのプロパティ．isPrototype プロパティの値は XML Schema Part 2: Datatypes の Built-in datatype である boolean 型を用いて表記し，true の場合はプロトタイプ記述，false あるいは isPrototype プロパティが省略されていた場合はモノの記述であるとする.
prototype	uTAD/Contents で記述されているモノのプロトタイプ．プロトタイプを利用して記述された uTAD/Contents では，まず，プロトタイプのプロパティを読み込む．その後，個々のモノに関する uTAD/Contents 記述でのプロパティを読み込み，プロトタイプで記述されていないプロパティの場合はそのプロパティが追加され，プロトタイプで記述済みのプロパティの値は上書きされる.
components	モノや情報が階層構造を持っている場合，この uTAD/Contents で記述されているものや情報を構成する，構成要素の一覧を記述する．成分，部品，コンテナの中身などの表記に利用される．「成分」「構成部品」「箱の中身」など，異なった意味での「構成要素」が必要となる場合は，モノのクラスごとに，components のサブプロパティを定義して区別する.
isMemberOf	モノや情報が，ほかのモノや情報の構成要素である場合（components プロパティに含まれている場合）に，そのモノや情報を指定する.
currentVersion,versionLog	どちらも，uTAD/Contents のバージョン管理に用いられる.
history	ログとして蓄積される情報を記述するためのプロパティ

(4) プロパティの値の記述

プロパティの値の表記方法は特に制限しないが，XML Schema Part 2:Datatypes で定義されているデータタイプに相当する場合は，なるべくその表記方法を利用することとする.

4.6.2 uTAD／UC Browser

例えば,RFID をモノにつけてその流通を監視したり,場所につけてそこを案内させたりするためには個々のモノや場所にユニークな識別子をつけて、それらの状況（コンテクスト）を認識させる事が必要である.このコンテクストを認識するために付けられる識別子を uCode と呼び,その標準化と管理を進めている.

実世界の様々なモノ･場所には RFID などで識別するための uCode タグが埋め込まれている．この uCode タグには識別子としての uCode 以外に可能な範囲内で，モノ・場所に関する属性情報も格納している．現状ではどのようなタグでも記憶容量などの制限があり，モノ・場所のすべての情報を格納することは不可能である．そこで,格納できない属性情報はネットワーク上のデータベースに格納し,uCode をキーとして情報をユビキタス ID センターに問い合わせる事で,モノとしての情報のありかを得て，それをもとに実際の情報を獲得する.

この一連の処理の中心に位置するのが,ユビキタスコミュニケータ（以下 UC）である.UC は uCode 解決サーバと呼ばれる uCode を管理するデータベースから実世界にある膨大な ucode や情報サーバの対応を得る事ができる.またその通信において暗号通信を行なうため,悪意のある者が不正にその情報を読み出す事を防ぐことができる.このようなアーキテクチャをユビキタス ID アーキテクチャ（uID アーキテクチャ）と呼ぶ（図 6-4）.

図 6-4　ユビキタス ID 技術アーキテクチャ

uID アーキテクチャにおいて現実世界（モノ・場所）の属性を記述し,コンテンツとして情報サーバに蓄えられるのが uTAD（Ubiquitous TRON Application Databus）である.

これは uTAD ユビキタス情報配信のための標準データ形式を定め,XML/RDF をベースとした形式を採用している.この uTAD は大きく分けて uTAD/Contents と uTAD/UC Browser と呼ばれる（以下 uTAD/UC）ものに分ける事が出来る.これらの特徴は以下である.

・uTAD/Contents
　実世界のデータの詳細を記述することに重点を置き,サーバ間で記述されたコンテンツの交換・流通を行なう.
・uTAD/UC
　主としてユーザノードシステムを構成する UC において,より解釈が軽く,表現に重視を置く.

これら２つの uTAD の関係を図 6-5 に示す.メタ情報を蓄える uTAD/Contents から，表示を高速化させるため uTAD/UC Browser に変換を行い,UC に転送を行なう.UC ではその uTAD/UC を解釈し表示を行なう.

図 6-5　uTAD の関係図

4.6.2.1　言語仕様

UC の表示制御を行なうために定義したタグの一覧を表 6-１に示す.

表6-1　タグ一覧

タグ名称	内容
Ubicontents	全ての要素のルート
Contents	画面全体の属性を管理
Image	画像を管理
Sound	音声を管理

Component	タップ制御などを管理
Movie	動画を管理
Init	初期化パラメータ設定
Label	ラベル表示指定

これらのタグを階層的に記述する事により制御を実現する.また,このuTAD/UCはコンテンツサーバにおいて自動生成されることを前提とするため整形済みXML形式とする.

4.6.2.2 記述例

```
<?xml version="1.0" encoding="EUC-JP" ?>
<ubicontents version="00.A0.06">
  <contents id="" encoding="EUC-JP" description="" attribute="" title="radish" cache="false"
    functionarea="false" functionurl="" >
    <image url=" $ SCHEME://$PROJECT/common/Information-display.jpg" pos_x="0" pos_y="0" width="480"
      height="640"></image>
    <image url="$SCHEME://$PROJECT/$LANG/sub/radish_parts.jpg" pos_x="0" pos_y="400" width="480"
      height="352"></image>
    <movie url="$SCHEME://$PROJECT/$LANG/radish.mov" pos_x="0" pos_y="48" width="480" height="352"
      autostart="true"></movie>
    <component eventtype="tap" action="load" url="$SCHEME://$PROJECT/$LANG/radish1.xml" pos_x="0" pos_y="0"
      width="480" height="640" attribute="" keycode=""></component>
    <component eventtype="mediaend" action="load" url="$SCHEME://$PROJECT/$LANG/radish1.xml" pos_x="0"
      pos_y="0" width="0" height="0" attribute="" keycode=""></component>
    <component eventtype="key" action="load" url="$SCHEME://$PROJECT/$LANG/radish1.xml" pos_x="0" pos_y="0"
      width="0" height="0" attribute="" keycode="center"></component>
  </contents>
</ubicontents>
```

4.6.2.3 ファンクションエリア

ユーザは一時的にシステムが提供するファンクションエリアを他の用途に使用できる.contentsタグで使用の可否とその記述ファイルの指定を行える.この章ではファンクションエリアの指定について述べる.

(1) タグ一覧

ファンクションエリアの設定を行い,ユビキタス・コミュニケータの表示制御をするために定義されたタグの一覧を表6-2に示す.その記述方法は前項までのuTAD/UCに準拠し,使用するタグに制限が有るだけである.

表6-2 タグ一覧

タグ名称	内容
ubicontents	すべての要素のルート
contents	画面全体の属性を管理
image	画像を管理
component	タップ制御などを管理

ただし,contentsタグにおける属性に関しては設定を行ってもタイトルなど無視される項目が存在するので注意が必要である.

4.6.2.4 記述例

(1) コンテンツにおけるファンクションエリアの指定(下線部)

```
<?xml version="1.0" encoding="EUC-JP" ?>
<ubicontents version="00.A0.06">
  <contents id="" encoding="EUC-JP" description="" attribute="" title="radish" cache="false"
    functionarea="true" functionurl="file://$PROJECT/$LANG/common/back.xml" >
  (以下略)
```

(2) ファンクションエリアの指定 (file://$PROJECT/$LANG/common/back.xml)

```
<?xml version="1.0" encoding="EUC-JP" ?>
<ubicontents version="00.A0.06">
  <contents id="" encoding="EUC-JP" description="" attribute="" title="" cache="false" functionarea=""
    functionurl="" >
    <image   url="file://$PROJECT/$LANG/common/funcbase.jpg"   pos_x="0"   pos_y="540"   width="480"
      height="100"></image>
    <image   url="file://$PROJECT/$LANG/common/back.jpg"   pos_x="180"   pos_y="550"   width="120"
      height="100"></image>
     <component   eventtype="tap"   action="load"   url="file://$PROJECT/$LANG/radish.xml"   pos_x="180"
      pos_y="550" width="120" height="80" attribute="" keycode=""></component>
  </contents>
</ubicontents>
```

参考文献

[1] Dublin Core Metadata Initiatives (DCMI), http://dublincore.org/.

[2] Klyne G. and Carroll J. (Editors), "Resource Description Framework (RDF): Concepts and Abstract Syntax," W3C Recommendation 10 February 2004, http://www.w3.org/TR/rdf-concepts/.

[3] Dan B. and R. V. Guha (Editors), "RDF Vocabulary Description Language 1.0: RDF Schema," W3C Recommendation 10 February 2004, http://www.w3.org/TR/rdf-schema/.

[4] Dave B. (Editor), "RDF/XML Syntax Specification (Revised)," W3C Recommendation 10 February 2004, http://www.w3.org/ TR/rdf-syntax- grammar/.

[5] [RSS] Dan Brickley, et al., "RDF Site Summary (RSS) 1.0," 2000, RSS-DEV Working Group, http://purl.org/rss/1.0/spec.

[6] "Scalable Vector Graphics (SVG) — XML Graphics for the Web," http://www.w3.org/Graphics/SVG/.

4.7 ユビキタス・コミュニケータ (UC)

4.7.1 UC 関連

4.7.1.1 ユビキタス･コミュニケータとは

ユビキタス・コミュニケータ (UC) は，ユビキタスコンピューティング環境とユーザ間をつなぐコミュニケーションツールである．具体的には，複数の通信手段を搭載した携帯情報端末であり，場所やモノに設置した IC タグ・赤外線・微弱無線マーカなどから ucode を取得し，ユーザに様々な情報を提供するものである．

最新のユビキタス・コミュニケータでは，昨年度の実験評価から得た知見を元に，コミュニケーション（ネットワーク）機能の強化を中心に，多数の機能を付加している（図 7-1）．

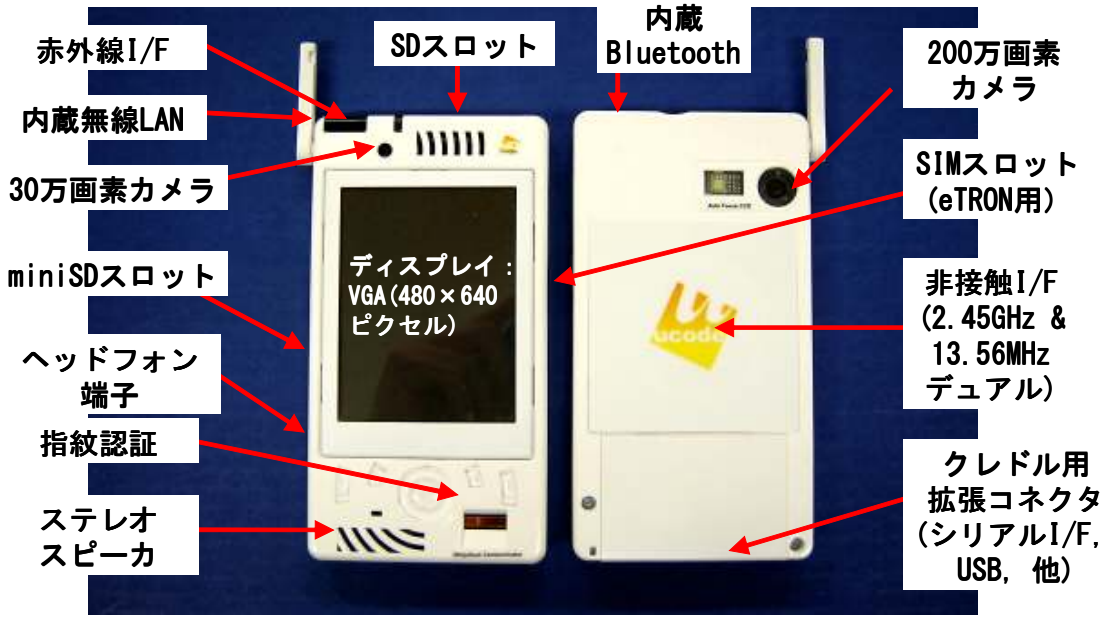


図 7-1　各部機能名称

(1) UC の主な機能

最新のユビキタス・コミュニケータでは，コミュニケーション（ネットワーク）機能の強化のため，無線 LAN など従来，カードなどでアドインしていた機能を内蔵している．また，カメラ機能なども標準実装し，高速画像処理と合わせ，様々な用途に対応することが可能である．

UC の特徴的機能について以下に挙げる．

① マルチバンド ucode タグリーダ機能

2.45GHz と 13.56MHz のマルチバンドリーダを内蔵する．これにより，1 台の UC 端末で，μチップや eTRON カードなどと通信することが可能である．

② 赤外線送受信機能

赤外線方式によるデータ送受信が可能である．これにより ucode の受信や，UC をリモコンとして使用することも可能である．

③ Bluetooth 機能

内蔵する Bluetooth により，外部接続の RFID リーダライタやヘッドセット，その他の周辺機器をワイヤレスで接続することが可能である．

④ 無線 LAN 機能

IEEE802.11b に対応する．これにより各無線アクセスポイントへの接続や UC 同士の通信が可能となる．

⑤ カメラ機能

CMOS 30 万画素(前面)と CCD 200 万画素（背面），2 種類のカメラが実装されている．

⑥ 専用 ASIC による高速な画像処理

画像処理用 ASIC（図 7-2）を搭載することにより， MPEG4 動画の再生，JPEG 画像の展開・拡縮・回転が可能である．

⑦ クレードルによる機能拡張

平時使用しない機能については，拡張クレードルの接続により使用することが可能になる．主な機能としては，USB，シリアル通信などがある．

図 7-2　画像処理 ASIC 概観

(2) UC 機能の応用

UC の各機能を利用することで，色々な用途への適応，システムへの組み込みが可能となる．以下にその応用例を挙げる．

① モノ・場所に関連した情報の提供

UC の持つ代表的なコミュニケーション機能（マルチバンド ucode タグリーダ・赤外線受信モジュールなど）を使用することにより，モノや場所に設置した ucode 情報を引き出すことが出来る．情報は音声・静止画再生だけでなく，動画再生なども可能であり，観光ガイドなどへの応用も期待できる．また外部の RFID リーダ（例えば杖型リーダ）と BlueTooth を連動させ，歩行者ナビゲーションなども行うことが可能である．

② ucode タグを応用したユーザインタフェースの実現

カード型の ucode タグを UC にかざすことで，そのカード（ucode）に対応した言語に UC の再生モードを切り替えることが出来る．モードを切り替える方法は色々考えられるが，カードタグ

を利用することで直感的に，また容易に UC を操作することが出来る．

③ カメラ機能の応用

カメラ機能を利用することで 1・2 次元バーコードから ucode を取得することが可能である．また UC の前面カメラと内蔵無線 LAN の組み合わせにより，VoIP（Voice over Internet Protocol）による電話機能なども実現することが出来る．

4.7.1.2 UC ソフトウェア

UC に搭載されているソフトウェアは以下のような構成になっている．

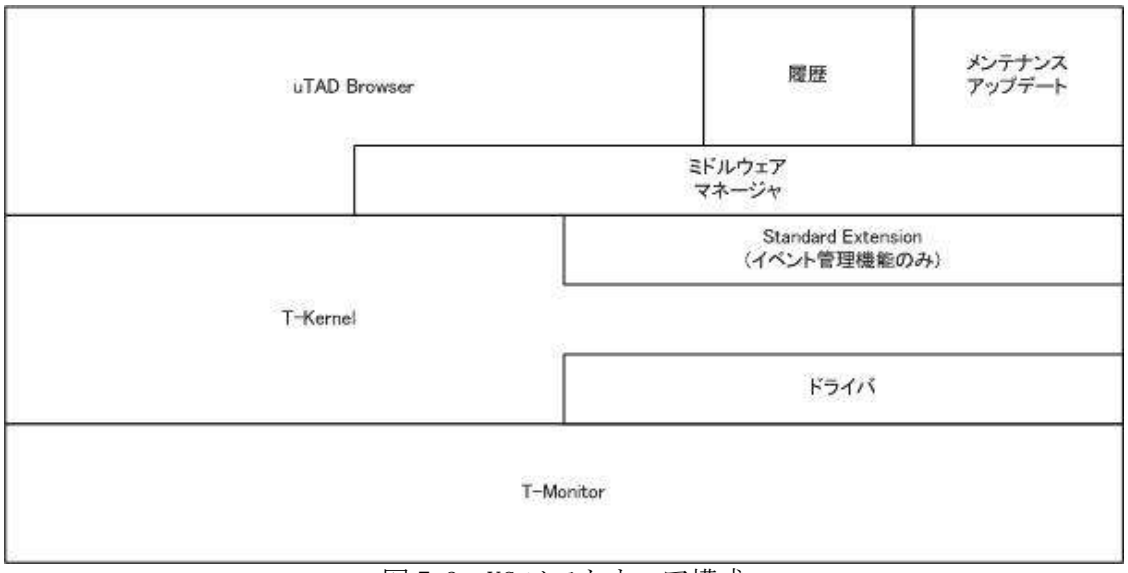


図 7-3　UC ソフトウェア構成

UC は OS として，一般公開されている T-Monitor，T-Kernel を UC 向けにカスタマイズしたものを用いており，更に T-Kernel Standard Extension のイベント管理機能を用いて動作している．その環境上で動作させるために，ドライバ，ミドルウェア，アプリケーションは T-Kernel 仕様に沿って実装されている．

4.7.1.3 ドライバ・ミドルウェア

ドライバは，T-Kernel の一部に組み込まれる．ソフトウェア階層の最も下に位置し，UC のハードウェアを制御する．

ミドルウェアは，ドライバと上位アプリケーションの中間に位置し，ドライバの持つハードウェア制御機能を上位アプリケーションが利用するための基本インタフェースを提供する．ドライバへのアクセスは，T-Kernel のシステムコールを介して行なう．

また，マネージャは，ドライバやミドルウェアの持つ機能を上位アプリケーションが簡便に利用できるようにするためのインタフェースを提供する．

4.7.1.4 アプリケーション

(1) uTAD Browser

uTAD Browser は，UC に具備された各種デバイスにより受信した ucode を元に，対応する uTAD コンテンツを取得し，表示/再生を行う機能を持つ．また，ボタンやタッチパネルなどの HMI イベントの発生を契機に，コンテンツのブラウジングや UC 本体の制御を行う機能も持つ．

(2) 履歴

履歴機能とは，過去に参照したコンテンツを再度参照するための機能である．uTAD Browser にて取得した ucode/URL は uTAD Browser にて蓄積されている．その蓄積された ucode/URL のリストを表示し，HMI から目的のものを選択することにより，いつでも過去に参照したコンテンツを再び参照することを可能にしている．

(3) メンテナンス/アップデート

メンテナンス/アップデート機能とは，UC 内のデータの更新や取得を行う機能である．主に以下のような場面で使用される．

- UC が内部に持つ設定情報，コンテンツデータの更新
- UC で蓄積された履歴情報の取得

UC はメンテナンス/アップデート機能が動作する時に無線 LAN 経由でネットワークに接続され，インターネットを介してデータの送受信を行う．

(4) 設定値管理

SD カード上の設定ファイル（A:/system/config）にて設定値を管理している．

4.7.2 UC-Phone

新規 R/W への対応や実証実験への展開を通じて，ユビキタス環境におけるユーザノードとして実用に耐えうるようにハードウェア，ソフトウェア双方のブラッシュアップを実施している．

表 7-1　ハードウェア仕様

通信方式	PHS 方式
表示部（LCD，タッチパネル）	TFT2.4”　240×320pix（QVGA）65000 色
RFID Reader/Writer	モジュール交換方式（μチップ，MB89R116/118 対応
バーコードリーダ	レーザ式，1 次元バーコード
外形寸法	143H×55W×26.6D
質量	144g（バッテリ含む）
バッテリ容量	690mAh

4.7.2.1 新規 R/W への対応

ハードウェア的には,R/W 接続に使うインタフェースを汎用化して新規 R/W の接続を簡易化している．外形寸法，シリアルコネクタ仕様にあわせた R/W であれば UC-Phone に接続可能である．

ソフトウェア的には，

- ドライバ部分に新規 R/W 向けのプロトコルを実装
- ユーザインタフェースに R/W 切り替え用の設定パラメータを追加

といった改造を実施することで対応可能である．1 度対応したプロトコルであればその後はソフトウェア変更（ROM 書き換え）せずに使用する R/W を切り替えることができる．

ハードウェア，ソフトウェアの対応が簡易化できたことで新規 R/W に対応するハードルを大きく下げることができた．今期は周波数の異なる 2 種類のタグ R/W への対応を完了している．

- μチップ
- MB89R116，MB89R118

いずれも独立した R/W ハードウェアを交換して接続することで動作させるタイプとなっている．

今期は，読み取り機器側であるユーザノードにおいて，新規 R/W 対応にかかるコストダウンに成功した．増加傾向にある各種実証実験を進める上で，早期投入可能なユーザノード端末を用意しておくことは今後も重要であると考えている．

表 7-2　対応 RFID タグ

タグ名称・種別	ベンダー	周波数
MB89R116/MB89R118	富士通	13.56MHz
μチップ	日立製作所	2.45GHz
T-Junction	凸版印刷	2.45GHz

4.7.2.2 パッケージ化

実証実験の活発化とともに，簡単に実験をスタートさせる仕組みが必要となっている．RFID タグ，UC-Phone ハードウェア，サーバノード側ソフトウェアの一部をパッケージ化し，実験キットとして提供する体制を整備した．図 7-4 に実験キットの構成を示す．

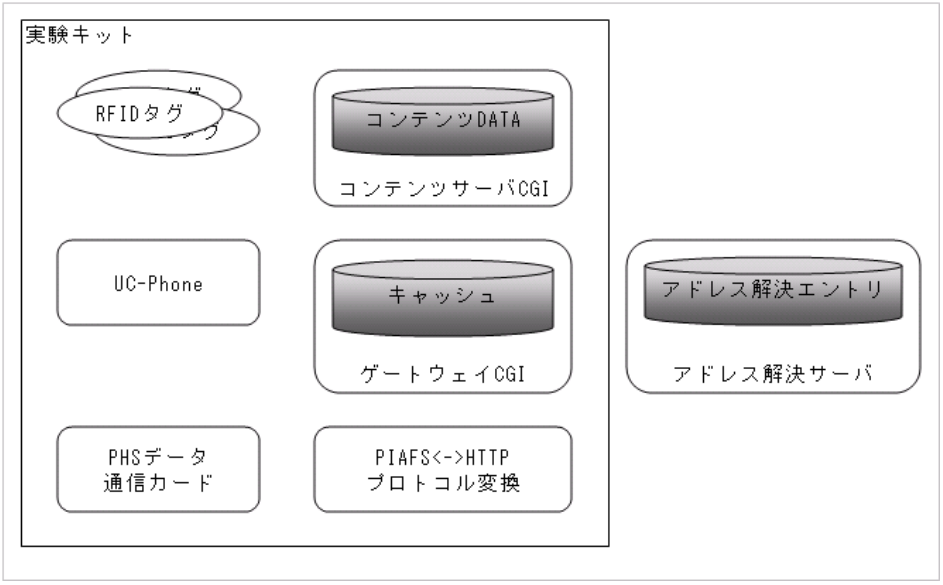


図 7-4　実験キット構成

パッケージ化した実験キットの導入により，１連のユビキタス情報配信アプリケーションとして UC-Phone を動作させることができるようになっている．この実験キットをベースにコンテンツサーバ上のアプリケーションサービスを拡張することでさまざまな応用システムに適用可能である．

4.7.2.3　その他ブラッシュアップ

(1) ユーザオペレーションの改善

実験でのフィードバックを得ながら，タグ読み取り～コンテンツ表示までのユーザオペレーションをブラッシュアップさせていった．また，用途によって複数のオペレーション・シーケンスを切り替えられるように，ソフトウェアに設定項目を追加している．こうしたエンドユーザ・ユーザビリティを考慮したユーザインタフェース見直しについて，重要な研究テーマになってきている．

(2) コンテンツ表現力・視認性の向上

昨期試作版から，高精細 LCD への置き換えを実施した．基本表示能力（表示色数，解像度）をあげておくことで「利用シーンに応じたコンテンツ表現の切り替え」を可能にするためである．これにより，従来版に比較してコンテンツ視認性が飛躍的に向上している．

(3) 省電力

新規 R/W 対応や LCD 置き換えなど，ハードウェア変更にともない使用電力が増加する傾向にある．また，用途の広がりにより設定が切り替えられたほうが良い項目が増えた．ソフトウェアにおいて，以下の項目に重点をおいたブラッシュアップを実施した．

- 省電力機能強化
- 設定項目追加（動作切り替え，タイムアウト，サーバ設定など）
- 処理効率向上

4.7.2.4　まとめ

今期は，昨期試作版で得られた知見をもとに，ハードウェア，ソフトウェア両方のブラッシュアップに注力して研究開発を進めてきた．また，他研究テーマの成果と統合を積極的に進め，サーバノードとのシステム連携部分をパッケージ化することに成功した．広がりを見せている各種実験や応用システムにすばやく対応できる仕組みが整いつつある．今後は，ユーザノード単体でのブラッシュアップだけでなく，サーバノードとの連携部分を含めたシステム全体のブラッシュアップを図ることで実用レベルに引き上げていくことが重要である．

4.8 サーバノードシステム（1）

4.8.1 アドレス解決サーバーゲートウェイ・キャッシュアーキテクチャ

4.8.1.1 取り組み概要

ユビキタスコンピューティング環境では，環境に埋め込まれた膨大な数のセンサーやアクティブタグなどのノードが，多様なネットワークを介して相互接続される．それらのノードから得られるコンテクストなどの情報を有効活用するためには，ノード情報を瞬時に検索･取得する仕組みが必要となる．そこで本研究では，ユビキタス PKI サーバ，アドレス解決サーバ，セキュア発行サーバ，ユビキタス ID サーバなどの要素技術の研究開発，及びそれらを統合したユビキタス情報サービス提供システムに関する研究開発を行う．

昨年度までの取り組みでは，膨大な量のユビキタス情報の管理･検索を実現するための基本アーキテクチャの設計，分散管理方式の設計，情報検索方式の設計，セキュリティ方式の設計を実施した．しかし，十分な検索スピードを実現するための性能や，セキュリティ機能の実証については，残存課題として解決していなかった．

そこで本年度は，昨年度までの研究成果に基づき，情報をより効率的に管理･検索するためのゲートウェイアーキテクチャの研究開発と，膨大な量の情報をセキュアに管理するための機能の実装･評価を行った．

4.8.1.2 ゲートウェイアーキテクチャ

(1) アーキテクチャ概要

膨大な数のノード同士が多様なネットワークを通じて相互接続される環境下では，ネットワークの輻輳やサーバへの負荷集中の発生が予想され，ノードの管理情報を実用的な時間で取得することが困難になる可能性がある．そこで，中間ノードとしてゲートウェイを設置し，サーバやクライアントが行うべき処理をゲートウェイが肩代わりすることにより，サーバやネットワークの負荷を緩和するためのゲートウェイアーキテクチャを導入する．

(2) 方式設計

本年度は，既存端末・ネットワーク・ソフトウェアを利用するという方針に基づき，ゲートウェイアクセスプロトコル仕様の策定，及びキャッシュ方式の設計を行った．

(a) プロトコル設計

クライアントとゲートウェイ間の通信方式として，HTTP をベースとしたプロトコルを策定した．基本アーキテクチャでは，ユビキタス ID サーバへは ID 解決プロトコルによりアクセスする必要があったが，ゲートウェイへのアクセスを HTTP ベースとすることにより，携帯電話等の HTTP を利用できる既存端末からも ID 解決が可能となった．また，イントラネットやホームネットワーク等の環境でも利用が可能である．

(b) キャッシュ方式

従来の DNS における名前解決情報や Web におけるコンテンツの取得では，検索キーと解決情報が 1 対 1 となっているため，その対をキャッシュすればよい．しかし，ユビキタス ID はその数が膨大になるため，ID とそれに関連付けられる情報は一般的に n：1 の関係になることが想定される．そこで，ID の解決情報と共に，ID のどの領域を使って解決を行ったかを示す解決マスクも取得し，ID と解決マスクを対でキャッシュすることで，n：1 のキャッシュを実現した．これにより，キャッシュ情報を最低限に抑えることが可能となり，効率的なキャッシュが可能となる．

(3) 評価

本研究では，低機能な機器がより効率的に ID 解決を行うためのゲートウェイアーキテクチャを提案した．次に，実験システムにゲートウェイを実装し，提案方式の応答性能改善に対する有効性を検証する．実験システムの構成を図 8-1 に示す．

実験システムでは，ゲートウェイにキャッシュ機能及び暗号通信機能を実装し，クライアントとゲートウェイ間は平文で通信を行う構成とした．クライアントには PHS 端末を利用した．また，コンテンツはゲートウェイ自身が保持する構成とし，コンテンツとしては約 1.5k バイトのテキストファイルを使用した．この環境下において，キャッシュがある場合と無い場合の，ID 読み出しからコンテンツ表示までの所要時間の計測を行った．結果を表 8-1 に示す．

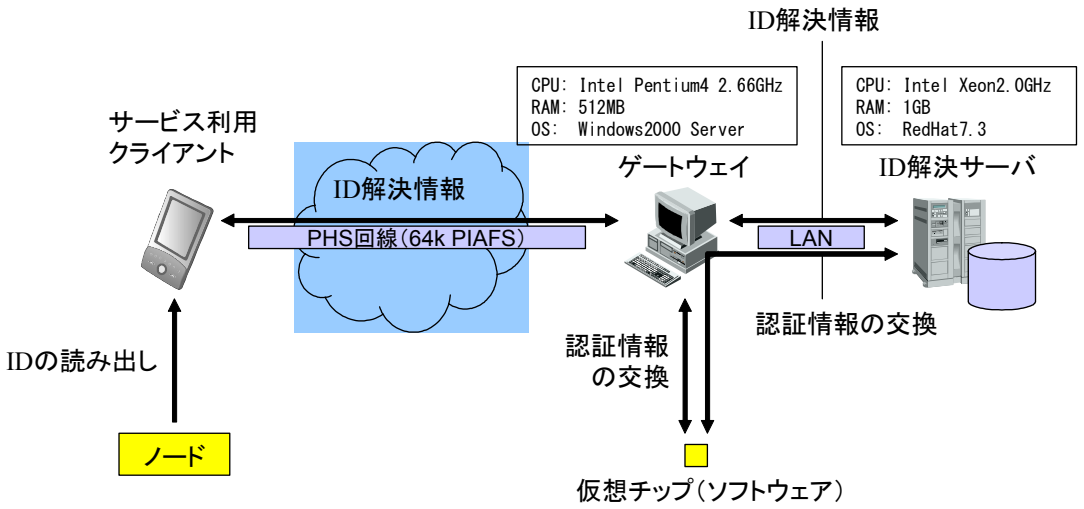


図 8-1　ゲートウェイ実験システム

表 8-1　ID 解決の所要時間

	処理主体		ｷｬｯｼｭ無し	ｷｬｯｼｭあり
	ｸﾗｲｱﾝﾄ	GW		
ID 読み出し	○		3.2	3.2
ucode の送信	○		1.2	1.2
ID 解決		○	5.4	
コンテンツ取得(GW)		○	0.1	0.1
コンテンツ取得(クライアント)	○		1.5	1.5
コンテンツ表示	○		4.1	4.1
合計			15.5	10.1

(単位：秒)

キャッシュがある場合にはユビキタス ID サーバにアクセスする必要が無いため，サーバアクセスの所要時間である 5.4 秒を短縮することができた．また，サーバアクセスの所要時間自体も，後述するクライアントが直接 ID 解決サーバと暗号通信路を構築した場合の ID 解決（表 8-2 を参照）と比較して，12.3 秒から 5.4 秒に短縮されており，PHS･ゲートウェイ間の ucode の送受信時間を考慮したとしてもゲートウェイ経由での ID 解決の方が高速であるという結果となった．なお，ID 解決以外の所要時間については，主に PHS 上での処理と PHS とゲートウェイ間の通信時間に費やされているため，端末性能やネットワーク帯域の拡充により高速化できると考えられる．

以上の結果より，ゲートウェイに適切なキャッシュを配置することで，ID の読み取りから情報の表示までの所要時間を短縮できることを確認した．また，PHS のような処理速度の遅い端末では，ゲートウェイの利用が高速化の一手段として有効であることが確認できた．

4.8.1.3　セキュリティ機能の実装と評価

(1) ID 管理検索システムへの実装

ID 管理検索システムは,膨大な数のノードから読み取られるコンテクスト情報を収集･分析するのに有用なシステムである．しかし，“人間の目に見えないほど小型であること”“モノや環境などのあらゆるモノに遍在的に組み込まれること”“無線インタフェースにより読み取りが可能であること”といったユビキタスコンピューティング環境が持つ特性に起因した，新たなセキュリティ的脅威の発生が危惧されている．これらの脅威に対抗するため，我々のこれまでの研究成果である超小型チップにも実装可能な軽量な認証暗号プロトコルを ID 管理検索システムに適用することとし，昨年度までにその適用方式の検討を行った．これをうけ，本年度は ID 管理検索の実験システムに認証暗号プロトコルの実装を行った．

実験システムの構成を図 8-2 に示す．

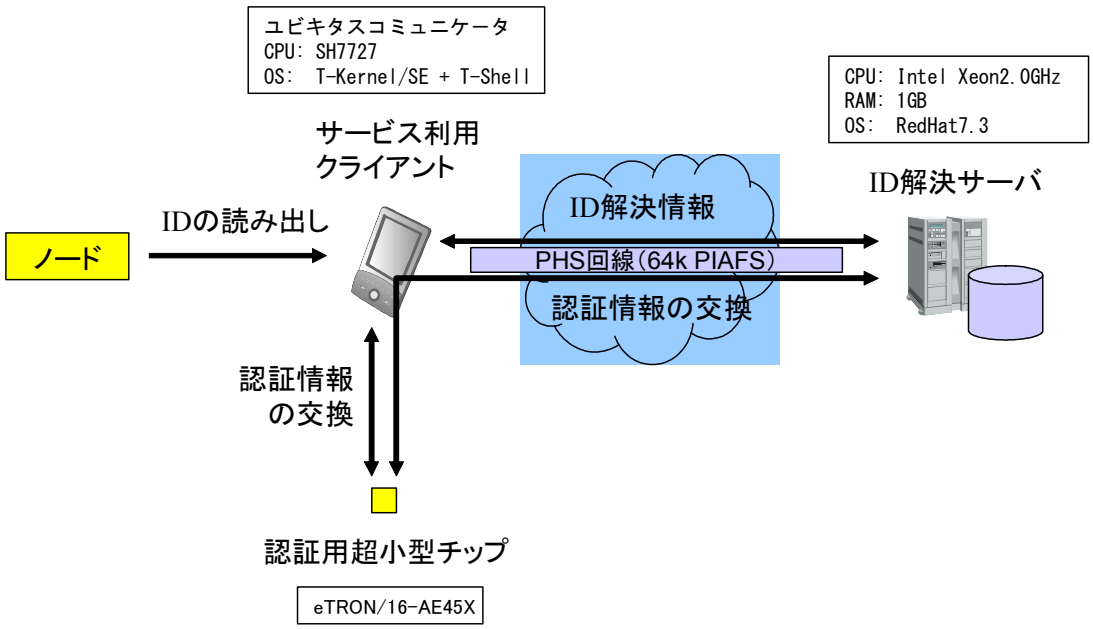


図 8-2　実験システムの構成

実験システムでは，認証用の超小型チップを利用することでクライアントの認証及びユビキタス ID サーバの認証を行う．認証用超小型チップは，クライアントとの相互認証が成功すると，ユビキタス ID サーバと相互認証･暗号鍵の交換を行い，クライアントに暗号通信路を提供する．これにより，以下のようなセキュリティ機能が実現可能となる．

- クライアント認証に基づいたサーバアクセス制御
- ID 及び管理情報の第三者への秘匿
- 不正発覚時のアクセスクライアントの同定

また，本方式では耐タンパ性の高い超小型チップを利用しているため以下の点で有利である．

- 認証情報の偽造によるなりますしが困難である．
- 認証用超小型チップを携帯することにより複数のクライアント端末による利用が可能である．

(2) 性能測定

CPU パワーが制限された認証用超小型チップを利用した認証方式では,処理速度の観点で十分なパフォーマンスを実現できるかどうかを検証する必要がある．そこで，実験システムを利用して性能評価を実施し，ボトルネックとなる処理の分析を行った．結果を表 8-2 に示す．

表 8-2　処理速度測定結果

手順	所要時間	主な処理項目
1	0.1	● 認証用超小型チップへのポーリングコマンドの実行
2	0.5	● 認証用超小型チップへのセション構築コマンドの実行
3	8.5	● 認証用超小型チップへの認証情報作成コマンドの実行 ● ユビキタス ID サーバにおけるサーバ側認証情報の作成 ● 認証用超小型チップと識別解決サーバ間の認証情報の送受信
4	2.4	● 認証情報からの暗号通信用共通鍵の生成
5	0.8	● ユビキタス ID サーバにおける ID 解決処理 ● クライアントとユビキタス ID サーバ間の ID 解決情報の送受信
合計	12.3	

（単位：秒）

実験により，ノードからの ID の取得から，情報に関する位置情報の取得まで，約 12 秒かかることがわかった．所要時間の大半は認証用超小型チップにおける公開鍵演算であり，チップの処理性能がそのまま全体の処理速度の遅延につながる結果となった．

(3) 評価

認証暗号プロトコルを実験システムに実装したことで，クライアント･サーバ間の相互認証，通信路の暗号化等の機能的な要件を実現できることを実証できた．しかし，処理速度に関しては，実験システムの性能は実用的なシステムに適用できるレベルとは言いがたく，処理速度の高速化を実現するための方策が必要であると言える．所要時間の短縮には様々なアプローチがあるが，本研究では認証暗号プロトコルの利用技術の観点からいくつかの改善策を提案する．

(a) セション管理の最適化による高速化

（ア）セションの事前確立

ユビキタス ID サーバとの暗号通信路の確立を事前にバックグラウンドで行っておくことで，要求発生時に認証に費やされる時間を短縮することができる．本方式の適用の条件として，問い合わせに行くユビキタス ID サーバが既知である必要がある．例えば，初回に問い合わせに行くサーバが固定的な所属地域を持った端末や，DHCP のように接続するネットワークにより自動的に問い合わせ ID 解決サーバアドレスを取得できる環境での適用が考えられる．

（イ）共有鍵の再利用

ユビキタス ID サーバと暗号通信路を確立したときに共有鍵を記憶しておき，次のセションでは前回の共有鍵に基づいて新しい共有鍵を再共有することで，2 回目以降の要求発生時の認証に費やされる時間を短縮することができる．クライアントではユビキタス ID サーバの識別番号と最後に使用した共有鍵をペアで記憶しておく必要があるため，比較的限られたユビキタス ID サーバにアクセスするクライアントに適用可能である．また，サーバ側にも同様のことが言える．

さらに，2 回目以降のアクセスでは共通鍵認証となるため，公開鍵認証と比べてセキュリティ的には堅牢ではない．従って，ユビキタス ID サーバを利用するサービスに求められるセキュリティレベルを見極め，共有鍵の有効期限を適切に設定する必要がある．

(b) ゲートウェイ利用による高速化

前述したゲートウェイを導入し，ゲートウェイとユビキタス ID サーバ間で公開鍵認証を行い，クライアントとゲートウェイ間は簡便な認証方式とすることで，認証に費やされる時間を短縮することができる．この場合，クライアントとゲートウェイを同一ネットワーク上に配置するなど，軽い認証方式でもセキュリティが担保できる構成とする必要がある．認証方式としては，例えば，イントラネットにおけるドメインのログイン認証や，ネットワークアドレス認証などがある．

4.8.1.4 まとめ

本年度は，ゲートウェイアーキテクチャの導入により，膨大な数のノードの情報をより高速に取得するための基本的な枠組みを実現した．この結果，限られた機能しか持たないクライアントがユビキタス ID サーバと直接通信を行う場合と比べて，応答速度を短縮できることが確認できた．さらに，PHS や携帯電話等の既存端末からのアクセスにも対応可能な構成とし，より汎用的な仕組みを構築できた．

セキュアな情報管理技術の開発では，昨年度までに検討した方式を実験システムに実装することで性能上のボトルネックを明確化し，今後の性能向上に向けた改善案を示すことができた．

4.8.2 ユビキタス情報サーバ

4.8.2.1 目的

ユビキタス情報サーバシステムは，ユビキタス環境をバックエンドで支える，大量の計算資源と通信資源を有するインフラシステムである．昨年度までに研究を行ったユビキタスＰＫＩサーバや，アドレス解決サーバ，セキュア発行サーバ，ユビキタスＩＤサーバの要素技術を統合し，ユビキタス情報サービスを提供するシステムへ進化させることを目的として，サーバ群の実装アルゴリズムやシステム構成をチューニングし，性能向上をはかり，実証実験での適用性を向上したユビキタス情報サーバを開発した．

4.8.2.2 ユビキタス情報サーバの概要

昨年度はユビキタスＩＤサーバの応用モデルとして食品トレーサビリティを採用し，それを具体的に実現するユビキタス情報提供システムを開発した．

本年度は昨年度の成果を踏まえ，ユビキタスＰＫＩサーバや，アドレス解決サーバ，セキュア発行サーバ，ユビキタスＩＤサーバの要素技術を統合し，より効率的かつ安全なシステムの開発を行った．

具体的には次のような内容を実施した．

(1) 超小型チップと印刷紙媒体の併用

流通用容器や商品棚に超小型チップによる電子タグを貼付して無線通信による迅速なトレース情報の取得を可能にするとともに，無線通信が不向きな商品やより安価な商品でもコストに見合った導入ができるよう印刷紙媒体（バーコード）による光学タグも，生産・流通・店舗の各段階で併用可能にした．

(2) 流通履歴記録・商品情報閲覧用携帯情報端末

流通過程での履歴の記録や店舗での商品情報の閲覧用に，PDA サイズの携帯情報端末として【サブテーマ３】で研究開発したユビキタス・コミュニケータを適用した．ユビキタス・コミュニケータでは，【サブテーマ１】で研究開発したセキュアチップによる認証方式を採用し正当なシステム運用者のみがアクセスできるようにした．

(3) 多種類の商品情報閲覧端末への対応

販売・消費段階では，多様な消費者にあわせて KIOSK 端末，ユビキタス・コミュニケータ，PC，携帯電話といった多種類の端末から商品情報が閲覧できるようにした．

(4) ユビキタス情報サーバへのアクセスの高速化

日々大量に消費される食品のような商品でもシステムへの情報アクセスが迅速に効率よく行えるよう，前節のアドレス解決におけるキャッシュ方式でのアクセスを採用した．本システムでは情報の問い合わせ元となる KIOSK 端末やユビキタス･コミュニケータなどの各端末に予め“所属地域”（＝○○百貨店，△△ストア等）の情報を持たせ，その情報によって問い合わせるサーバを選択する方法を取った．これにより即答性要求にも対応可能となっている．

(5) システムの標準化

様々な個別のシステムが存在しても相互に情報交換ができるように，ユビキタス ID 技術としてコード体系，情報データ形式，通信手順などの方式（プロトコル）の標準化を行った．この技術では商品などを識別するための ID の解決と，トレーサビリティにおけるトレース情報を提供するサービスとの独立性が高いため，複数のトレーサビリティのシステムが共存可能となっている．

(6) システムの付加価値の強化

生産・流通段階におけるトレーサビリティ情報の記録の手間とコストの負担による普及阻害に対する対策として，生産・流通段階のプレイヤが有する情報に広告情報を付加することで付加価値を持たせ，利益を還元できるようにした．

商品情報の食品トレーサビリティの対象としては，昨年度の青果物のみから，食肉，加工品，日配品にまで拡大し，より実用に即したシステムを実現した．

4.8.2.3 システム構成

昨年度開発したシステムをユビキタス情報システムの「基幹システム」と位置付け，調整を行った．このシステムはユビキタスＩＤセンターと密にかかわり，本年度新規に開発した青果物，食肉，加工品，日配品を取り扱う各サブシステム（図 8-3）を包括する．また，レガシーシステム（バーコード）既存のシステム群をも包括する形で稼動可能である．これらシステムは，異種システムの相互運用を前提とした，連邦型システム（Federated System）として構築した．

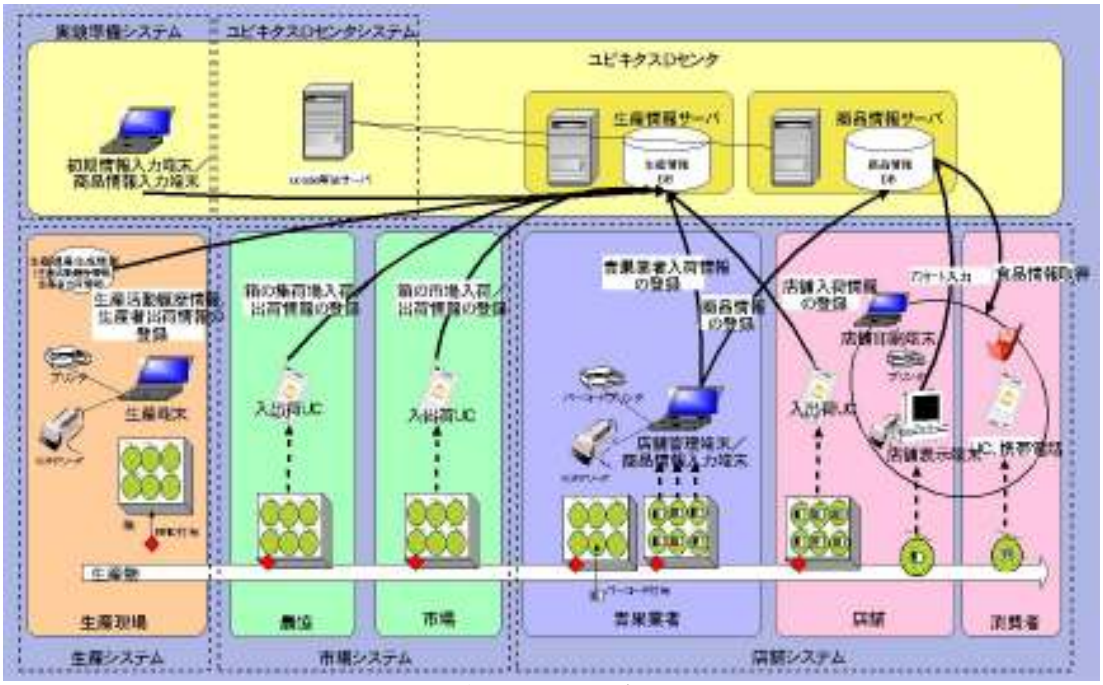


図 8-3　サブシステム（青果物）の構成

4.8.2.4　開発の成果

開発したユビキタス情報システムを活用し，青果物，食肉，加工品，日配品の生産，流通，販売，消費の全段階をカバーしたフィールド試験を実施し，有効性の検証を行った．

(1) フィールド試験の概要

表 8-3 実施スケジュール

スケジュール	内容
平成 16 年 12 月～平成 17 年 2 月	生産・加工段階での実証
平成 17 年 1 月～2 月	流通段階での実証
平成 17 年 2 月 2 日～平成 17 年 2 月 15 日 平成 17 年 2 月 24 日～平成 17 年 3 月 2 日	三越百貨店での実証（青果物，食肉，加工品） 京急ストアでの実証（加工品，日配品）

表 8-4 実施内容

段階	内容
生産段階	生産システムおよび IC チップを使用した生産情報の記録・管理
流通段階	流通システムおよび IC チップ，バーコードを使用した入出荷支援，情報の記録・管理
店舗段階	店舗システムおよび IC チップ，バーコードを使用した入出荷支援，情報の記録・管理 店舗端末による一般消費者への商品情報の開示 PC，携帯電話による一般消費者への家庭での商品情報の開示

表 8-5 試験結果

段階	期間	利用状況
生産段階（三越百貨店用青果物，食肉，加工品）	平成 17 年 12 月 ～2 月 15 日	生産者数：3 出荷数：168 個
流通段階（三越百貨店用青果物，食肉，加工品）	平成 17 年 1 月 28 日 ～2 月 15 日	流通拠点：11 拠点 配送回数：66 回 使用タグ数：超小型チップ 24 個 バーコード 150 個
店舗段階（三越百貨店）	平成 17 年 2 月 2 日 ～2 月 15 日	使用タグ数：超小型チップ 80 個 バーコード 5,442 個 商品情報総アクセス件数：2,419 件

生産段階（京急ストア用加工品，日配品）	平成 17 年 2 月 21 日 ～3 月 1 日	生産者数：4 出荷数：68 ロット
流通段階（京急ストア用加工品，日配品）	平成 17 年 2 月 21 日 ～3 月 1 日	流通拠点：5 拠点 配送回数：29 回 使用タグ数：224 個（バーコード）
店舗段階（京急ストア）	平成 17 年 2 月 24 日 ～3 月 2 日	使用タグ数：超小型チップ 47 個 バーコード 10,393 個 商品情報総アクセス件数：1,697 件

(2) フィールド試験による有効性の検証

・電子タグと印刷紙媒体の融合

超小型チップによる電子タグと印刷紙媒体（バーコード）のタグを混在して使用し，機能やコストに応じた利用方法を確認できた．

・タグの貼り付け方

異なる販売形態をもつ様々な店舗，多種類の取り扱い品目に対応できるよう品物へのタグの貼り付け方も様々なものを用意し，その有効性を検証できた．このようなノウハウが今後，様々な品物へのタグ付与に役立つと考えられる．

・多様な情報アクセス手段

販売・消費段階で多種類のデバイスから，購入後も自宅など店舗以外からでも商品情報の閲覧を可能にしたため，利用促進に非常に効果的であることがわかった．アクセス傾向の分析により，極力簡易な操作でのアクセス手段の提供が有効であるとわかった．

検索データのキャッシュ方式，および認証方式を採用した流通履歴登録機能や商品情報の閲覧機能については，実行速度・表示速度は全く問題なく，商品情報の閲覧では，アクセス件数等は昨年度実績で同等であるが，応答時間は 2～3 秒程度改善された．

・加工食品への対応による対象商品の拡大

加工食品では，原材料のトレーサビリティチェーンが加工品のトレーサビリティチェーンに結合・集約されるという多段階の構成であるため，二次加工，三次加工の商品のトレーサビリティにも対応可能である．「生産」－「流通」－「販売」－「消費」の流れを統一的に扱うことによって，対象商品を大幅に拡大できるとともに，トレーサビリティの標準化に有効である．

・他のシステムとの連携

生産・流通の一部の拠点では，そこでの導入済みのレガシーシステムとの連携を図った．既存システムの製造ロット番号や識別番号等を投入できるようにし，照合時に取り違え等の防止に有効であった．

・システムの共用

昨年度のシステムを基幹システムとし，各サブシステムで共有した．多種類の品種や異なる販売形態を持つ店舗でもシステムの共用が可能であることが実証された．

4.9 サーバノードシステム（2）

4.9.1 はじめに

本節では，ユーザノードとサーバノードシステムの連携で実現させた UC-Phone 向けサーバシステムの各種実装について述べる．

4.9.2 システム概要

ユーザノード・サーバノードシステム連携の概略を図 9-1 に示した．UC-Phone（ユーザノード）でのタグ ID 読み取りに始まり，サーバノードに ID を送信する．サーバノードは受け取った ID に関してアドレス解決を実行する．また，アドレス解決結果の URL からコンテンツ取得し，結果をユーザノードに返却する．最終的にユーザノードである UC-Phone でコンテンツを表示するまでの流れをあらわしている．

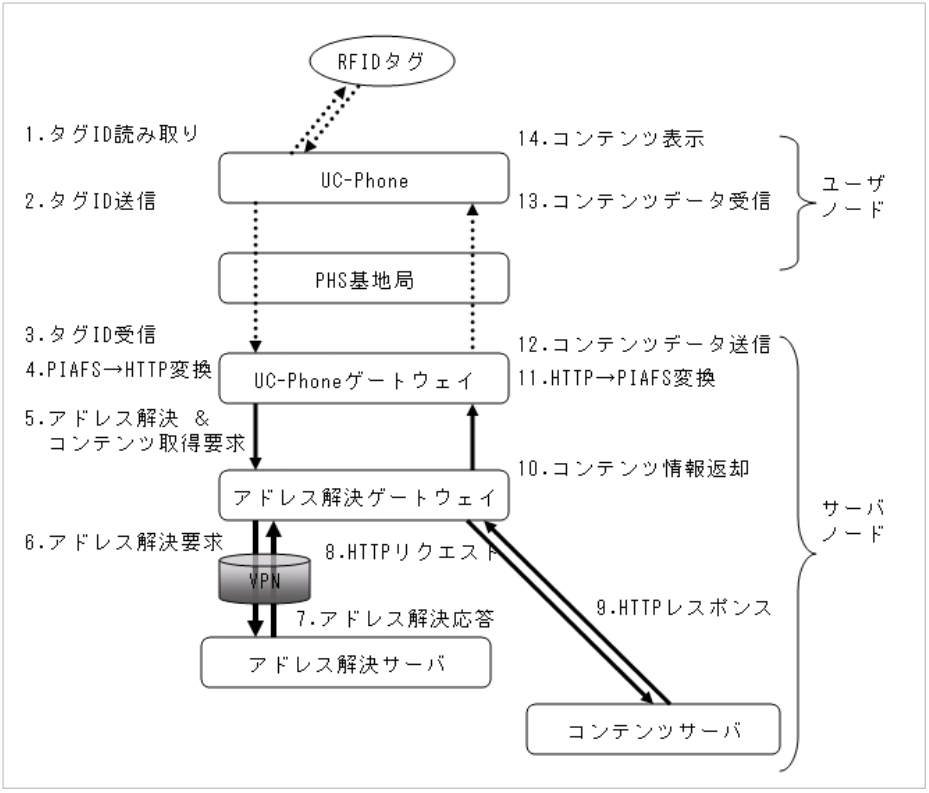


図 9-1　ユーザノード・サーバノード連携

UC-Phone 端末は PHS 網を使った音声通話と PIAFS によるデータ通信が可能となっている．この UC-Phone とサーバノードシステムを連携させるうえで，PIAFS←→HTTP プロトコル変換機能を UC-Phone ゲートウェイと呼ぶ中継ノードに実装している．各ノード間で使用するプロトコルを表 9-1 に示した．

表 9-1　使用するネットワークプロトコル

接続ノード	プロトコル
UC-Phone 端末～PHS 基地局	PIAFS
PHS 基地局～UC-Phone ゲートウェイ	PIAFS
UC-Phone ゲートウェイ～アドレス解決ゲートウェイ	HTTP
アドレス解決ゲートウェイ～アドレス解決サーバ	eTP（VPN）
アドレス解決ゲートウェイ～コンテンツサーバ	HTTP

アドレス解決ゲートウェイは，UC-Phone や携帯電話端末などユーザノードとして計算機資源が乏しい端末向けに，アドレス解決やコンテンツ取得を代行する機能を実装させたものである．また，アドレス解決ゲートウェイとアドレス解決サーバとの通信はノード間で VPN セッションを構築し，セキュアにアドレス解決情報をやりとりする仕組みを取り入れている．

4.9.3　アドレス解決ゲートウェイの実装

4.8 節にて記載したアドレス解決ゲートウェイ仕様の主だったものを HTTP ベースの CGI として実装した．HTTP リクエストを扱えるユーザノードであればクライアントの種別を問わない実装となっている．

4.9.3.1　アドレス解決サーバを使ったアドレス解決機能

アドレス解決ゲートウェイの基本機能は，接続要求元のクライアント（ユーザノード）に代わり，アドレス解決サーバに対してアドレス解決要求を発行し結果を取得することである．得られたアドレス解決情報をそのままクライアントに返却するか，解決結果の URL からコンテンツを取得したあとでクライアントに返却するか，クライアントが要求時のパラメータ指定により切り替え可能な仕様となっている．

強固なセキュリティを確保する目的でアドレス解決サーバとの通信は，VPN セッションを確立した上でアドレス解決プロトコルによるパケット送受信を行う実装とした．

4.9.3.2　アドレス解決結果のキャッシュ機能

処理性能の向上を目的として，アドレス解決ゲートウェイ上にキャッシュアルゴリズムを実装している．これは VPN セッションを構築して取得したアドレス解決結果をキャッシュとしてゲートウェイ内部に保持しておく．新規にクライアントから要求を受けたときに，ゲートウェイはまずキャッシュしたアドレス解決情報を検索する．合致するエントリが存在する場合は TTL フィールド(有効期間)を参照し，これが有効期間内であればキャッシュしたデータをアドレス解決結果として使用するものである．

用途に応じて適切な有効期間を設定しておくことで，このキャッシュ機能が働き，アドレス解決サーバと通信する場合に比べ，効率よくアドレス解決結果を取得できるようになる．

VPN セッションを構築してのアドレス解決には一定の処理時間が必要なため，実運用ではこうしたキャッシュ機能の必要性が高まる．また，接続先のゲートウェイが同一であれば他ユーザノードとキャッシュ情報を共有することができる．これはアドレス解決処理の効率化と同時に，アドレス解決サーバに対するトラフィック軽減にも有効である．

図 9-2 は倉庫(A)と事務所(B)で同じアドレス解決ゲートウェイに接続している例である．会社 X でモノが流通する場合，図中のアドレス解決ゲートウェイにて解決結果をキャッシュすることで全体的な処理効率アップにつながる．

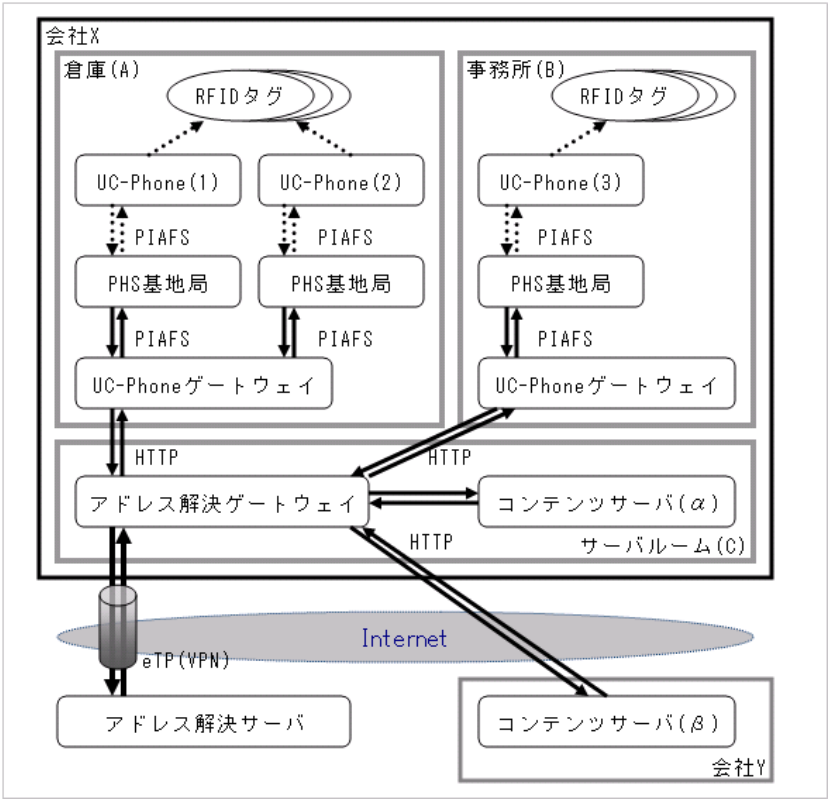


図 9-2　アドレス解決ゲートウェイを共有

4.9.3.3　コンテンツダウンロード機能

アドレス解決ゲートウェイ仕様として，コンテンツのダウンロードまで実行するモードを定義している．この場合は，アドレス解決で取得した URL からデータをダウンロードし，クライアントに返却する動作となる．今期実装では，HTTP によるダウンロード機能をサポートした．クライアントからのリクエスト受付とレスポンス返却には HTTP 拡張ヘッダを用いている．アドレス解決機能で取得したコンテンツサーバ URL に対して HTTP リクエストを発行し，コンテンツをダウンロードする．取得したコンテンツは HTTP ボディでクライアントに返却する．クライアントは，HTTP ボディと HTTP 拡張ヘッダの両方を

参照することで，コンテンツ本文だけでなく，アドレス解決結果も取得することができる．

コンテンツダウンロード機能はオプションであり，ユーザノードに対してアドレス解決結果のみを返却する動作モードも指定可能である．

4.9.4 まとめ

ユーザノードとサーバノードを連携させ，１連のユビキタス情報取得システムを実装することができた．また，基本機能の実装だけでなく，処理効率改善をテーマとしたアドレス解決ゲートウェイ仕様の実装も進めることができている．今後は，これらシステムの性能評価とブラッシュアップを継続しつつ，実用を見据えて研究を進めていくことが必要である．

4.10 CA 局

4.10.1 概要

CA 局に関しては階層化に対応した証明書発行機能，証明書の有効性を検証し応答を返す証明書検証機能を拡充し，また高速化のためにはユビキタス環境における様々な運用方式に適したサーバ構成やキャッシュアルゴリズムを組み込んだ．

4.10.2 システム構成

CA 局システムは，証明書の発行・管理などを行う CA 局サーバ（以降 CA サーバと呼ぶ），CA サーバが発行した証明書をセキュアチップに書き込むクライアントツール，証明書の検証を行う証明書検証サーバなどのコンポーネントから構成される認証システムである．システム構成を図 10-1 に示す．また，各コンポーネントの概要について以下に述べる．

(1) CA 局サーバ（CA サーバ）

CA サーバ S/W が動作する Solaris マシンを， CA サーバと呼ぶ．操作員は Linux 上の WEB ブラウザ（Netscape または Mozilla）を経由して CA サーバの CA 局にアクセスし，CA 証明書の管理や発行，eTP 証明書の発行などを行うほか，カード提供／運用者の登録情報の管理や CA サーバの設定を行うことができる．また CA サーバには失効情報サーバも存在し，CA 局が稼動すると失効情報サーバも同時に稼動する．

(2) 証明書検証サーバ

証明書検証サーバは，セキュアチップで使用する専用の公開鍵証明書の検証を，セキュアチップに代わり実施する．操作員は証明書検証サーバの稼動や停止，証明書ストアの表示や追加，失効情報サーバの設定，証明書検証サーバの署名鍵の更新を行うことができる．

(3) クライアントツール

クライアントツールは，セキュアチップに対応した証明書の発行申請や，CA サーバで発行した eTP 証明書をセキュアチップに書き込む機能を持つ．Linux マシン上で動作する．

(4) データベース

CA サーバが発行する証明書やカード提供／運用者の登録情報などはすべて，WEB サーバを経由してデータベースで管理される．

(5) WEB サーバ

WEB サーバは，CA サーバと Linux 上の WEB ブラウザとの間の情報のやりとりを管理する．

(6) Hardware Security Module（HSM）

CA 局の秘密鍵が外部に漏洩すると，第三者によって証明書を自由に作成されてしまう可能性があり，セキュリティ上問題となる．特に大規模に利用される CA 局の秘密鍵が漏洩した場合，その事実を検出することすら困難な場合もある．そこで，CA 局の秘密鍵は耐タンパ性のある Hardware Security Module（HSM）に格納する．

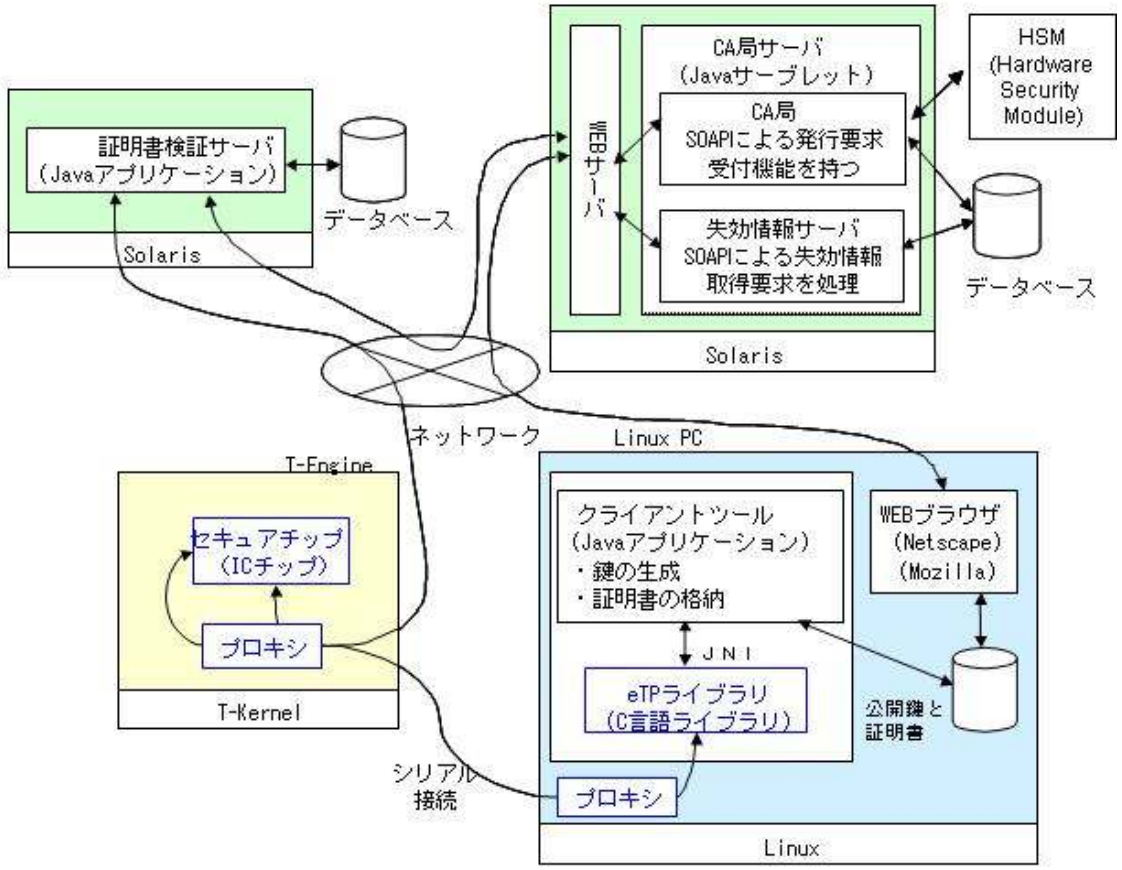


図 10-1　CA 局システム構成図

CA 局サーバと Linux PC 間は，ブラウザベースの HTTP(一部 SOAP)で通信を行い，WEB ブラウザとクライアントツールの間は，XML 形式の申請書，証明書情報データ・ファイルの受け渡しをオフラインで行う．クライアントツールとセキュアチップ間は，専用ライブラリを使用した eTP 通信を行う．そして，CA 局サーバと証明書検証サーバは，HTTP(一部 SOAP)で通信を行う．

4.10.3 階層化に対応した証明書発行機能

ユビキタスコンピューティング環境においては，我々をとりまく生活空間のすみずみまで大小機能様々な組込み機器が配置され，それらが様々なネットワークを介して情報を相互にやりとりする．そのような状況において，セキュリティを確保する為に証明書を利用するエンドエンティティの数は大幅に増大する．また，セキュリティの必要となる用途や運用のポリシーも多様性を増していくため，CA 局における証明書の発行・管理の負荷は益々高まっていく．

そこで CA 局の負荷分散ならびにスケーラビリティ確保のために，証明書の階層化と，これに対応した証明書の発行機能を研究,開発した．

昨年度までの PKI の信頼モデルにおいては，１つの CA 局が各エンドエンティティに対して証明書を発行していた．

しかし，エンドエンティティの数が増大した場合に，１つの CA 局が全てのエンドエンティティの証明書の発行・管理を行うことは，処理能力が有限であることから現実的とは言えない．

また，複数の CA 局を設置することにより，証明書の発行・管理対象となるエンドエンティティを分散させることも可能ではあるが，その場合は，異なる CA 局に属するエンドエンティティ同士の証明書の有効性を確認できないという問題が発生する．

そこで今回は，従来の信頼モデルを拡張し CA 局を階層化することとした．拡張した信頼モデルにおいては，まず CA 局がサブ CA に対して証明書を発行し，更にサブ CA がエンドエンティティに対して証明書を発行する．サブ CA の階層は多段化することが可能であり，階層化に対応したサブ CA 局の証明書には，その証明書を発行した CA 局または上の階層のサブ CA 局の署名が付加されている．

なお，エンドエンティティの証明書の有効性は，エンドエンティティの証明書から，その証明書の発

行者であるサブ CA 局の証明書，更にそのサブ CA 局の証明書の発行者である CA 局の証明書へと信頼の連鎖を遡ることで確認できる

以上のように，従来の信頼モデルを拡張し CA 局を階層化するとともに，それに対応した証明書の発行機能を開発することで，各 CA 局の発行・管理するエンドエンティティの数を分散し，負荷を低減させるとともに，ユビキタス環境に対応したスケーラビリティを確保することができた．

4.10.4 証明書検証機能

ユビキタスコンピューティング環境においては，我々をとりまく生活空間のすみずみまで大小機能様々な組込み機器が配置され，それらが様々なネットワークを介して相互に情報をやりとりする．そのような状況において，セキュリティを確保するために PKI の証明書が利用されるが，この証明書が本当に信頼できるものであるかどうか，証明書検証により正当性を確認する必要がある．証明書検証の手順は，1.証明書経路構築（Path Discovery）と 2.証明書経路検証（Path Validation）の２つに分類され，それぞれ以下の処理を行う．

(1) 証明書経路構築（Path Discovery）
　証明書の検証者は特定の CA 局を信頼点（Trust Point，Trust Anchor）として設定する．そして，検証対象である証明書を発行した CA 局を順に遡り，信頼点まで到達した時点で証明書経路構築が完了する．もしも証明書経路構築に失敗した場合は，その証明書の検証が行えないため証明書は信頼できないと判定される．
(2) 証明書経路検証（Path Validation）
　証明書経路構築に成功した場合は，各証明書の有効性の確認，ならびに証明書を発行した CA 局の署名の検証を行う．信頼点から検証対象である証明書までの各処理に成功すれば，証明書の正当性が確認できたこととなり，証明書は信頼できると判定される．

なお，ユビキタスコンピューティング環境において数多く利用される組み込み機器においては，計算資源ならびに通信速度に厳しい制約があるため，上記の証明書検証処理を実行することが大きな負担となる．そこで今回は，大量の計算資源と通信資源を有し，証明書の検証を代行する証明書検証サーバをインフラシステムに追加した．

証明書検証サーバにおける証明書検証機能の動作概略図を図 10-2 に示す．

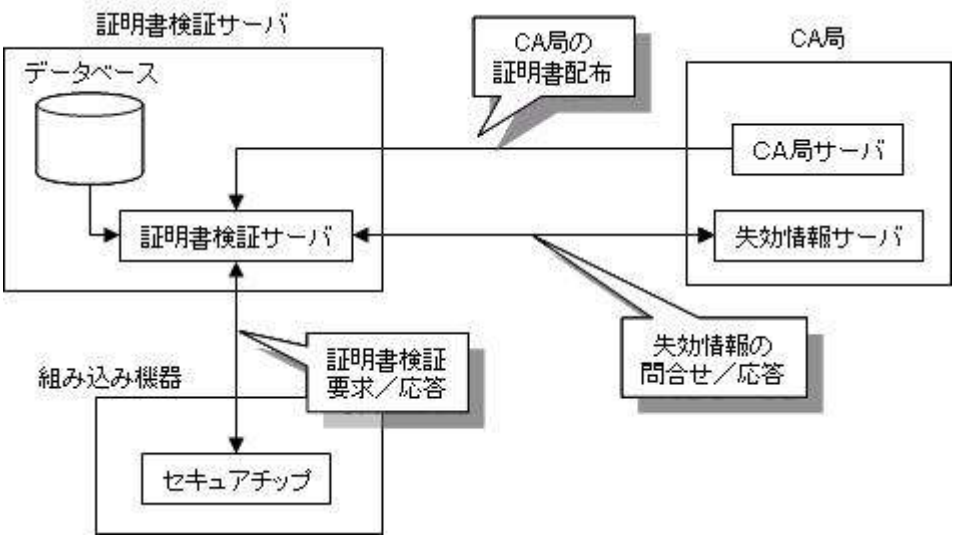


図 10-2　証明書検証機能の動作概略図

証明書検証サーバには，信頼点となる CA 局の証明書が予め配布されている．また，証明書の失効情報の問い合わせ先となる失効情報サーバに関する情報がデータデースに登録されている．

そして，組み込み機器内のセキュアチップなどからの証明書検証の要求をうけると，証明書経路構築ならびに証明書経路検証を行い，証明書の正当性を検証する．その際，証明書の失効状態は失効情報サーバに問い合わせることで確認する．そして最後に，改ざん防止のための署名を証明書の検証結果に付

加したうえで応答を返す.
以上で述べた証明書検証機能をもった証明書検証サーバを開発したことで，計算資源ならびに通信速度に厳しい制約がある組み込み機器における処理が軽減され，セキュリティの確保が容易になった.

4.10.5 ユビキタス環境に適したサーバ構成

本年度は前述のように,CA局の負荷の軽減ならびにスケーラビリティ確保のためにCA局を階層化し，それに対応した証明書の発行機能を開発した．このように階層化された CA 局による信頼モデルには，サーバである CA 局の構成によって以下のような複数のバリエーションが存在する.

(1) WEB モデル
証明書の検証者には予め複数の信頼点が与えられている．これは，自分の証明書の信頼点とは異なる信頼点を持つ証明書を，信頼できるものとして受け入れることを意味する．インターネットの WEB ブラウザなどで多用されているモデルである.
(2) ツリーモデル
ツリー構造の最上位にルート CA と呼ばれる CA 局を配置し，その下位にサブ CA と呼ばれる CA 局を複数配置する．信頼点はルート CA 1 つのみとなり，自分の証明書の信頼点と他の証明書の信頼点が共通になる.
(3) 相互認証モデル
独立した信頼関係にある CA 局同士が，お互いの証明書を相互に発行することで信頼関係を形成する．これにより，他の証明書を発行した CA 局は，自分の証明書の信頼点である CA 局からみて，サブ CA として位置づけられるようになる.
(4) ブリッジモデル
独立した信頼関係にある CA 局同士の相互認証のために，更に上位にブリッジ CA と呼ばれる CA 局を配置する．そして，CA 局とブリッジ CA がお互いの証明書を相互に発行することで信頼関係を形成する．なお，この信頼関係はブリッジ CA と 相互認証を結んだ CA 局全てに及ぶ.

以上のようなバリエーションの中から，ユビキタス環境に適したサーバ構成を検討する際に，以下の２点について特に留意した.

(1) 高速な認証
セキュリティを確保するために必要な認証がリアルタイムに実行可能となるような構成とする.
(2) スケーラビリティの確保
証明書を利用するエンドエンティティの大幅な増加に対応可能とする.

そして今回は，先に述べた 4 つのサーバ構成のバリエーションのうち(2)ツリーモデルを採用した.
これは次のような理由による．まず，相互認証モデルおよびブリッジ認証モデルでは証明書経路が複雑化するために証明書検証の際の証明書経路構築の時間が飛躍的に増大してしまい，リアルタイム性を確保することが難しくなる．また WEB モデルでは，信頼点となる CA 局が増加した場合に，無数の組み込み機器に信頼点を追加することが困難である.
これに対して，ツリーモデルの場合は，証明書経路がシンプルであるので証明書経路構築が容易に行える．また，最終的な信頼の基となる信頼点が単一であるので，信頼関係の境目を意識することなく広くシームレスな利用が可能となる.
以上で述べたように，今回はユビキタス環境に適したサーバ構成として，階層化された CA 局による信頼モデルにツリーモデルを採用し，高速な認証とスケーラビリティを両立させることができた.

4.10.6 証明書検証の高速化（キャッシュ機能）

ユビキタスコンピューティング環境におけるセキュリティを確保する為に，PKI の証明書が利用されるが，この証明書の正当性を証明書検証により確認する必要がある．そして，その確認の頻度はより高いことが予想されるため，システムには高速な処理が求められる.
証明書検証の手順は,(1)証明書経路構築 (Path Discovery) と(2)証明書経路検証 (Path Validation) の２つに分類されるが，このうち(1)の証明書経路構築に関しては，前節にて述べたユビキタス環境に適したサーバ構成により処理の高速化を図った.

そこで，残る(2)の証明書経路検証に関して，今回は証明書検証サーバにキャッシュ機能を追加することで高速化を図った．

既に 4.10.4 節にて述べたように，証明書検証機能を提供する証明書検証サーバは，証明書経路検証の際に証明書の失効状態を失効情報サーバに問い合わせる．ここでは，証明書検証サーバと失効情報サーバ間の通信による遅延と，失効情報サーバにより処理時間が必要となる．そこで，証明書の失効状態を，証明書検証サーバがキャッシュすることで，証明書経路検証に必要な処理時間の短縮を図ることとした．

今回開発した証明書検証サーバにおける失効情報のキャッシュ機能は，既に行った失効状態の問い合わせ結果をサーバ内部に蓄積し，その情報を次回の失効状態の問い合わせの際に再利用するものである．なお，このキャッシュ機能は，システムやサービス毎に異なると思われる運用ポリシーに対応するため，キャッシュ有効期間やキャッシュサイズなどのパラメータを変更することが可能となっている．更に，キャッシュ機能の有効／無効の切り替えや，モジュール交換によるキャッシュ機能そのものの更新も可能である．従って，今後の実証実験や実運用などの結果をもとに，より最適なキャッシュ機能へと改良を加えることが容易に可能である．

4.11　超機能分散システム指向開発環境

4.11.1 T-Kernel／Standard Extension

「T-Kernel　Extension」とは，リアルタイム OS「T-Kernel」をベースに，より高度な OS 機能を実現するための拡張プログラムである．この中でも特に標準的機能として規定する Extension が，「T-Kernel/Standard Extension」である．

「T-Kernel/Standard Extension」の主な機能としては，MMU（Memory Manegement Unit）による論理多重空間やメモリ保護機能，プロセス間同期通信機能，ファイルシステム機能などが挙げられる．

4.11.1.1 「T-Kernel / Standard Extension」の機能

「T-Kernel/Standard Extensiopn」は，「T-Kernel」の上位レイヤに存在し，「T-Kernel」機能を拡張する（図 11-1）．

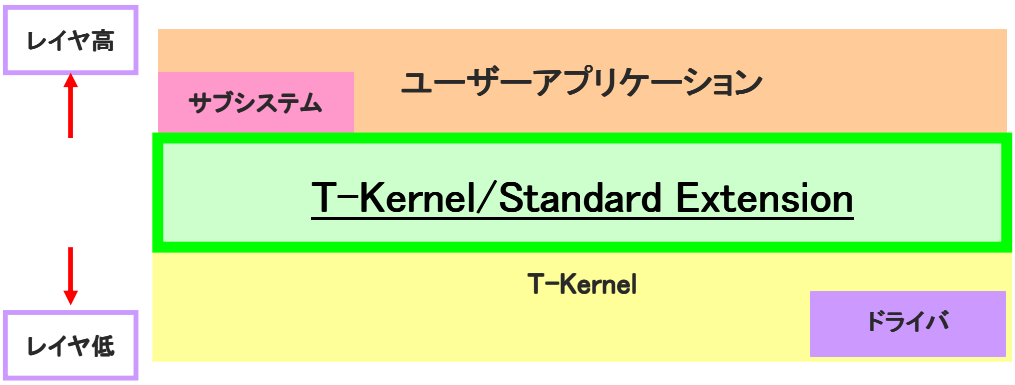


図 11-1　T-Engine アーキテクチャにおけるソフトウェア構成

(1) メモリ管理

MMUによるメモリ管理を実現する．これによりプロセス（ユーザーアプリケーションなど）を，メモリ保護された論理アドレス空間上で動作させることが可能になる．

メモリ管理機能では，データメモリ領域の管理を行う．管理されるメモリ領域は，ローカルメモリ空間，共有メモリ空間，システムメモリ空間の 3 種類あり，それぞれの空間に対して，指定した大きさのメモリブロックの割当て / 解放等の機能を提供する．

システムコールでは，ブロック単位でメモリを割り当てる機能のみを提供し，ブロックサイズは，MMU機能などに依存し，システムに最適な値が決められる．

(2) プロセス/タスク管理

シングルユーザ/マルチプロセスのシステムを実現するための機能を提供する．

プロセスは，OS が管理するプログラム単位である．簡単に述べると

プロセス = タスク + メモリ管理 + リソース管理

となる．したがって，プロセスには１つ以上のタスクが存在する．そしてこのタスクが優先度順および時分割のスケジューリングにより平行実行される．

プロセスは，システム開始時に初期化プロセスが生成され，必要に応じ，各プロセスが新しいプロセスを生成していく．そしてシステム全体としては，初期化プロセスをルートとする木構造を構成する．このとき，生成したプロセスを親プロセス，生成されたプロセスを子プロセスと呼ぶ（初期化プロセスには親プロセスはいない）．

プロセスの優先度とスケジューリング

プロセスは生成時（tkse_cre_prc）に 1～255 の優先度を与えることが出来る．このとき，各プロセスは，優先度により３つのグループに分けられる．

Ａ．絶対優先度グループ （優先度 1 ～ 127）
優先度順にスケジューリングされ，同一優先度の場合は，ラウンドロビン方式でスケジューリングされる．

Ｂ．ラウンドロビングループ１ （優先度 128 ～ 191）

Ｃ．ラウンドロビングループ２ （優先度 192 ～ 255）
グループ内でラウンドロビン方式でスケジューリングが行われる．優先度は相対的なスケジューリングの頻度を示す．

プロセスとタスクの優先度

プロセスの優先度は，プロセス生成時（tk_cre_prc コール）に指定する．このとき，

プロセスの優先度 ＝ メインタスクの優先度

となり，指定された優先度のメインタスクが生成，実行される．サブタスクの優先度については，タスク生成時（tk_cre_tsk コール）に指定することになる．

タスクのスケジューリング

スケジューリングはタスク単位に行われる．

例）以下は，タスクのスケジューリング的には同等で，どちらも同名のタスクの優先度は同じである．

・プロセス１がタスクＡ，Ｂを持ち，プロセス２がタスクＣ，Ｄを持つ．
・プロセス１がタスクＡを持ち，プロセス２がタスクＢ，Ｃ，Ｄを持つ．

T-Kernel レベルから見たスケジューリングの処理

Extension のタスク優先度は,以下の規則に従い T-Kernel のタスク優先度にマッピングされる．
（Extension におけるタスク優先度と T-Kernel におけるタスク優先度は同じではない）

・絶対優先度グループ
T-Kernel 優先度 ＝ Extension 優先度＋システム予約優先度数（定数）
タイムスライス時間 ＝ 絶対優先度グループ・タイムスライス時間（定数）

・ラウンドロビングループ n
T-Kernel 優先度＝ラウンドロビングループ n 優先度（定数）
タイムスライス時間＝ラウンドロビングループ n のベースタイム（定数）－Extension 優先度

(3) プロセス/タスク間の同期・通信機能

プロセス間の同期・通信機能として，以下の機能を提供する．

・プロセス間メッセージ機能
・グローバル名機能

また，タスク間の同期・通信機能として，以下の機能を提供する．

・セマフォ
・ミューテックス
・イベントフラグ
・メッセージバッファ
・ランデブ

(4) グローバル名管理
プロセス間で共有するデータを名前により生成，参照する機能を提供する．

(5) 標準入出力管理
標準入出力管理では，ファイル入出力に関連した基本機能を提供する． また異なる複数のファイルシステムに対応し，ファイルシステムの相違をほとんど意識することなく，統一的にファイル操作することが可能である．なお現在，対応可能なファイルシステムは，以下の通りである．
・標準ファイルシステム
T-Kernel 標準ファイルシステム
・拡張ファイルシステム
FAT ファイルシステム（FAT12, FAT16, ＦＡＴ32）
CD-ROM ファイルシステム（ISO 9660 Level 1）

(6) イベント管理
イベント管理機能は，各種デバイスからの通知を「イベント」という形で統一的に取り扱う仕組みで，インタラクティブなヒューマンインタフェースを提供する．
イベント管理の対象となるデバイスは，主としてキーボード，ポインティングデバイスであるが，ucode イベント(16 バイト長のユニークな数値データ)などを取り扱うことも可能である．

(7) その他
デバイス管理：デバイスを操作するための機能を提供する．
時間管理：時間の管理を提供する．
システム管理：システムプログラムのロード，アンロードの機能を提供する．

4.11.2 T-Dist の研究開発

4.11.2.1 はじめに

現在，組み込み機器の高機能化に伴い，組み込みソフトウェアの大規模化・複雑化が進んでいる．今後，身の回りのあらゆるモノや場所にコンピュータが組み込まれるユビキタスコンピューティング社会になると，ますます組み込みソフトウェアの需要が増大し，組み込み分野の「ソフトウェア危機」の発生が懸念されている．特に，従来の組み込みソフトウェアは，機器毎にカスタムメイドとして製造されており，ハードウェアへの最適化にはメリットがあるものの，実装や検証の効率や開発期間という点でデメリットが大きい．

この問題を解決するため，本研究では，安全にソフトウェア資産を再利用し，テスト期間の短縮を図りながら信頼性の高い組み込みソフトウェアを開発するための，ソフトウェア流通プラットフォームの構築を研究開発テーマとした．

4.11.2.2 本研究開発の目標

Ｔ－Ｄｉｓｔ（ＤｉｓｔはＤｉｓｔｒｉｂｕｔｉｏｎ＜流通＞の略）とは，4.11.1 で開発したＴ－Ｋｅｒｎｅｌ／ＳＥ上のソフトウェア流通を強力かつ安全にサポートするプラットフォームであり，本プラットフォームを利用することにより，組み込みシステムにおけるソフトウェアの再利用性を高め，開発コストの低減と開発期間の短縮を目指している．

Ｔ－Ｄｉｓｔでは，ソフトウェアが不正利用されないようにソフトウェアを暗号化し，更に改ざん防止の電子署名を付与して配信する．ソフトウェア利用者は，利用ライセンスを格納した【サブテーマ１】で開発された耐タンパチップを保持する．流通ソフトウェアを利用するためには，この利用ライセンスが必要で，このライセンスを利用してソフトウェアの暗号を復号してソフトウェアを実行する．これにより，ソフトウェアを利用者に対して安全に配送し，許可された利用者だけが実行可能となる．

本研究は，このソフトウェアの不正利用を防止しながらソフトウェアの流通促進を図るＴ－Ｄｉｓｔシステムを構築し，検証を行うことを目標としている．

4.11.2.3 成果概要

前年度に一部設計を行ったＴ－Ｄｉｓｔのパイロット実装を完了し，実用の組み込みソフトウェアを使った機能検証実験を実施し，機能評価，性能評価を実施した．

4.11.2.4 成果説明

(1) パイロット実装概要

Ｔ－Ｄｉｓｔシステムの機能評価，性能評価を実施するため，前年度に一部設計を行った仕様に基づきパイロット実装を行った．パイロット実装の全体システム構成概要を図 11-9 に示す．また，モジュール構成を図 11-10 に示す．

図に示される通り本パイロット実装は，Ｔ－Ｄｉｓｔサービス提供サーバ，ホストマシン（ＰＣ），ターゲットマシン（Ｔ－Ｅｎｇｉｎｅ），耐タンパチップの４つの要素から構成されている．

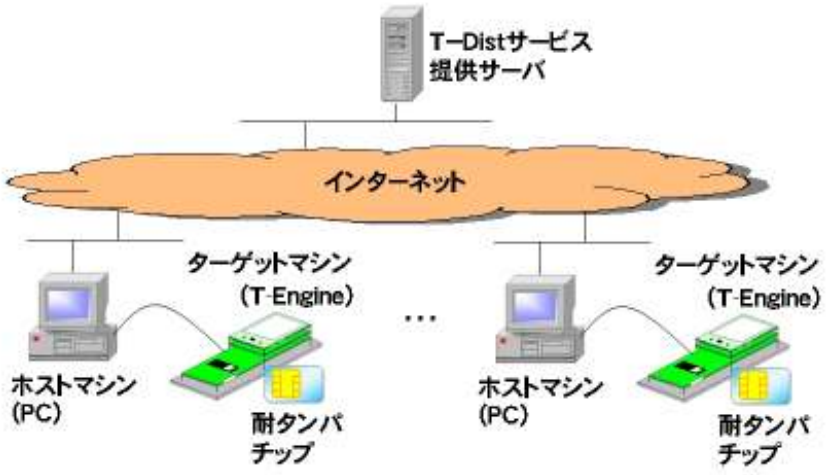


図 11-9　全体システム構成概要

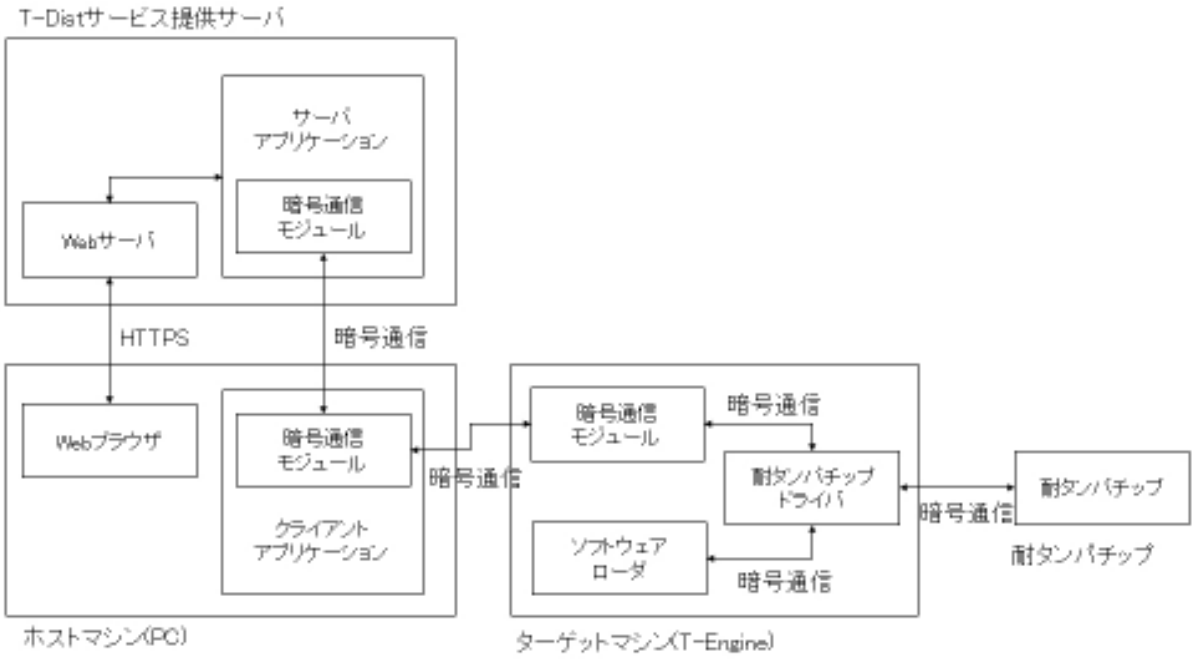


図 11-10　モジュール構成

(2) 機能評価結果

今回実装を行った各機能に対して，機能評価試験を実施した．以下に各機能の評価結果を示す．

・暗号化ソフトウェア生成機能
　指定されたフォーマット通りにソフトウェアが暗号化されるとともに署名が付与され，正しく生成されていることを確認した．
・ソフトウェアライセンス生成・発行機能
　ソフトウェアライセンスが正常に生成・発行されることを確認した．また，ソフトウェアライセンスが仕様通りの形式となっていること，本ソフトウェアライセンスを利用して暗号化ソフトウェアが実行可能な形式に復号できることを確認した．
・ソフトウェア登録・削除・検索機能
　Ｗｅｂブラウザベースでソフトウェアの登録・削除・検索が可能であることを確認した．
・ソフトウェアライセンス一覧表示機能
　耐タンパチップ内に格納されているソフトウェアライセンスが一覧表示されることを確認した．
・ソフトウェアライセンスダウンロード要求機能
　サーバより，ダウンロード可能なソフトウェアライセンスを一覧表示し，選択したソフトウェアライセンスを耐タンパチップ内に直接ダウンロードされることを確認した．
・耐タンパチップ初期化機能
　耐タンパチップが初期化され，再び利用可能であることを確認した．
・ソフトウェアフォーマットチェック機能
　内容が改ざんされている場合は実行できないようになっており，フォーマット検証，署名検証が，正しく実行されていることを確認した．
・ソフトウェアライセンスチェック機能
　仕様に則ったライセンスであれば，正しく実行されていることを確認した．また，不当なライセンスである場合や，権利が無いライセンス，ライセンスが存在しない場合等には，ソフトウェアが実行できないことを確認した．
・ソフトウェア実行機能
　非暗号化の場合と全く同一にソフトウェアが実行されることを確認した．

(3) 性能評価結果
本システムにおいて表 11-5 に示す３項目の性能評価を実施した．

表 11-5　性能評価検証項目

1	非暗号化時と T-Dist を利用した暗号化時のソフトウェア起動時間のターンアラウンドタイム
2	ソフトウェアの暗号化（発行）に要するターンアラウンドタイム
3	ソフトウェアライセンス発行のターンアラウンドタイム

なお，検証に利用したサンプルソフトウェアは以下の通りである．

・サンプルソフトウェアの仕様
　サイズが異なるサンプルソフトＡ，サンプルソフトＢを準備し，各種性能測定を行った．なお，ソフトウェアの内容に関しては，本性能検証とは関連が無いため割愛する．
　サイズは下表に示す通りである．Ｔ－Ｄｉｓｔ利用方式による暗号化・署名付与を行うことにより，ヘッダ，フッター，パディングが付加され，ソフトウェアのサイズが約 300byte 増加する．

表 11-6　サンプルソフトウェアのサイズ

ェア名	ソフトウェアサイズ	Ｔ－Ｄｉｓｔ利用方式による 暗号化時のソフトウェアサイズ
ソフトＡ	29,881 byte	30,176 byte
ソフトＢ	788,553 byte	788,848 byte

以下に性能検証結果を示す.

・検証項目１

非暗号化と暗号化時で処理時間に違いが発生した．非暗号化時は，単にソフトウェアをメモリ上に展開するだけであるため，双方等も１秒未満の短時間で処理が行われた.

一方，ソフトウェア暗号化時は耐タンパチップにアクセスしてライセンスを確認し，さらにソフトウェアを復号するため，非暗号化に比べて起動時間が長くなった.

また，サイズの違いにより，ソフトウェアの復号処理の処理時間に差が出た．これは，ソフとウェア復号処理時間がサイズに比例するためと思われる．しかしながら，組み込みシステム向けのソフトウェアのサイズは通常数キロバイト～数十キロバイト等，比較的小さいものが多いため，大きな問題はないと思われる.

表 11-7　検証項目１検証結果

ェア名	化時	T-Dist 利用方式による暗号化時 （署名検証処理）（復号処理）
ソフトＡ	0.1 秒未満	約 3.5 秒 （1 秒未満）（0.1 秒未満）
ソフトＢ	１秒未満	約 8 秒 （1 秒未満）（約 3 秒）

・検証項目２

Ｗｅｂを介して，利用者からのソフトウェアダウンロード要求を元に，ソフトウェアを暗号化・署名付与し，ダウンロード可能な状態にするまでにかかる時間である.

計測の結果，ソフトウェアのサイズによる処理性能の違いはほとんど無く，双方とも１秒未満でダウンロード可能な状態となった.

表 11-8　検証項目２検証結果

ェア名	かかるターンアラウンドタイム
ソフトＡ	１秒未満
ソフトＢ	１秒未満

・検証項目３

ソフトウェアライセンスの発行は，サーバと耐タンパチップ間で双方向認証による暗号化通信路を構築するとともに，その通信路を介してライセンスを書き込むため，サンプルソフトＡ，Ｂどちらのライセンスも発行に約 10 秒程度のターンアラウンドタイムがかかった．既存の組み込みシステムであっても，ライセンスの取得には書面やメールなどさまざまな方式があり一概に比較はできないが，双方向認証による安全な暗号化通信の上でサーバからの取得に約 10 秒かかるレベルであれば，問題無いと思われる.

5 参考資料・参考文献

5.1 研究発表・講演等一覧

① 研究発表

（査読有り）

1. M. Terada, K. Mori, K. Ishii, S. Hongo, T. Usaka, N. Koshizuka, K. Sakamura: “TENeT: An Architecture for Distributed SmartCard”, in Proceedings of the 2nd International Conference on Security in Pervasive Computing.
2. Lee Hoi Leong, Shinsuke Kobayashi, Noboru Koshizuka, Ken Sakamura: “CASIS: A Context-Aware Speech Interface System”, in Proceedings of the International Conference of Intelligent User Interfaces (IUI 2005), ACM, Jan. 2005.
3. Shinsuke Kobayashi, et. al.: “T-Air: Low Power Wireless Sensor Network Platform for Ubiquitous Computing”, in Proceedings of the 1st International Workshop on Networked Sensing Systems, June 2004.
4. 小林真輔，早川幹，越塚登，坂村健: 「T-Engine を用いた ISO18000-4 タグリーダライタのプロトタイプ設計」， 第 12 回 FPGA/PLD Design Conference ユーザープレゼンテーション論文集， pp. 79～84, 2005 年 1 月.
5. 小林真輔，越塚登，坂村健：「H. 263 デコーダを用いた『組込みソフトウェア開発プラットフォーム：T-Engine』の評価」，情報処理学会 DA シンポジウム 2004，ポスターセッション，2004 年 7 月.

（査読無し）

1. 坂村健：「ユビキタス時代の半導体技術」，応用物理，第 73 巻，pp. 1155，2004 年. (招待論文)
2. 坂村健：「ユビキタス時代のシステム技術」，映像情報メディア学会誌，Vol. 59，No. 1，pp. 27～32，2004 年. (招待論文)
3. 越塚登：「ユビキタス ID センター」，情報処理，2004 年 6 月. (招待論文)
4. 越塚登，坂村健：「ユビキタス ID 技術とその応用」，電子情報通信学会誌，Vol. 87，No. 5，2004 年 5 月，pp. 374～378. (招待論文)
5. 小林亜鈴，上向俊晃，井ノ上直己，小池淳，山田満，坂村健：「統合ＰＤＡ端末の開発（１）～端末実装」，信学総大，2005 年 3 月.
6. 松尾賢治，橋本真幸，小池淳，山田満，坂村健：「統合ＰＤＡ端末の開発（２）～顔認証アプリケーションの実装と高速化」，信学総大，2005 年 3 月.
7. 服部元，松本一則，菅谷史昭，小池淳，山田満，坂村健：「統合ＰＤＡ端末の開発（３）～携帯端末のためのＷｅｂページ自動分割」，信学総大，2005 年 3 月.
8. 石川彰夫，川田亮一，小池淳，山田満，坂村健：「統合ＰＤＡ端末の開発（４）～３次元自由視点ＶｏＤシステムの実装」，信学総大，2005 年 3 月.
9. 上向俊晃，小林亜鈴，井ノ上直己，小池淳，山田満，坂村健：「統合ＰＤＡ端末の開発（５）～エラー耐性を強化した通信放送融合型データ配信システムの実装」，信学総大，2005 年 3 月.
10. 加藤恒夫，河井恒，小池淳，山田満，坂村健：「統合ＰＤＡ端末の開発（６）～分散型音声認証システムの実装」，信学総大，2005 年 3 月.
11. 加藤 淳，藤内 俊一，諸隈 立志，坂村 健：「ＵＮＰ：ユビキタス環境下における制御系ネットワークプロトコル」，情報処理学会ユビキタスコンピューティング研究会 第 7 回研究発表会，2005 年 3 月.
12. 越塚登:「ユビキタス ID 技術とトレーサビリティ」，農業情報学会シンポジウム 2005 春 予稿集，農業情報学会，2005 年，pp. 45～52.
13. 西山智，山田満，越塚登，坂村健，「ユビキタスサービスのためのエージェントプラットフォームの提案」，第 120 回情報処理学会 DPS 研究会(IPSJ SIG-DPS)，2004 年 11 月 4～5 日.
14. 渡辺伸吾，西山智，越塚登，坂村健：「既存ルータ混在環境におけるモバイル IP ハンドオーバ高速・高信頼化」，マルチメディア，分散，協調とモバイル (DICOMO2004) シンポジウム，情報処理学会，2004 年 7 月.

② 一般講演

1. 越塚登：「ユビキタスネットワークにおける GIS の活用」，GIS フォーラム，総務省、独立行政法人情報通信研究機構，2005 年 2 月 16 日.
2. 越塚登：「ユビキタス ID センターにおける RFID への取り組み～実証実験を中心に～」，情報処理学会連続セミナー2004「IC タグ」，2004 年 12 月 17 日.
3. 越塚登：「ユビキタス ID 技術」，TRONSHOW 2005，12 月 9 日.
4. 越塚登：「トロン教育普及グループの活動」，TRONSHOW 2005，12 月 8 日.
5. 徳田，越塚，下條，竹内，山田：「パネルセッション」，ユビキタスネットワークシンポジウム 2004，11 月 30 日.
6. 越塚登：「基調講演：ユビキタスが拓く未来」，精密工学会 第 301 回講習会ユビキタスで変わる製造業：ヒット商品を産み出す組み込み OS，11 月 16 日.
7. 越塚登：“Technologies and Activity of Ubiquitous ID Center”，SmartLabel Asia 2004，11 月 11 日.
8. 越塚登：「T-Engine，ユビキタス ID 技術の最新動向」，韓国 SYSKON（System Kernel Conference），11 月 6 日，ソウル.
9. 越塚登：「ユビキタスが開く未来」，栃木県ユビキタス講演会『ユビキタス・ネットワーク社会がやってくる』(10 月 8 日)
10. 越塚登：「ユビキタスと食：情報産業としての食品産業」，ユビキタス社会と地域振興：沖縄の可能性，沖縄国際大学産業情報学部，食のトレーサビリティシステムを広げる会（9 月 25 日）
11. 越塚登：「T-Engine とユビキタス ID 技術」，第２回「ワイヤレス・センサー・ネットワーク社会に向けたナノメートル CMOS システムとその要素技術の研究」に関する先導的研究開発委員会，日本学術振興会，2004 年 9 月 7 日.
12. 越塚登：“Ubiquitous Network: An Introduction to the Activity of Ubiquitous ID Center”，第三回日中韓情報通信大臣会合ビジネスフォーラム（7 月 27 日）.
13. 越塚登：「ユビキタス ID 技術の最新動向―実用化に向けた具体的事例」，ESEC 2004，2004 年 7 月 8 日.
14. 越塚登：「T-Engine，ユビキタス ID 技術の最新動向」，Networld+Interop　東京 2004，2004 年 7 月 2 日.
15. 越塚登：「組込みリアルタイムプラットフォーム T-Engine・T-Kernel」，Networld+Interop 東京 2004，2004 年 6 月 30 日.
16. 越塚登：「ユビキタス ID の最新動向―RFID を使ったユビキタスネットワークの実現にむけて」，情報通信月間講演会（中国），2004 年 6 月 25 日.

③ 文献

（著書）

1. 坂村健，竹村健一：「すべてのモノにコンピュータを，ユビキタス社会，始まる」，太陽企画出版，2004 年.
2. 坂村健：「ユビキタス、ＴＲＯＮで出会う『どこでもコンピュータ』の時代へ」，NTT 出版，2004 年.

（一般記事）

1. 坂村健：「自律的移動支援プロジェクトから『ユビキタス国土』へ」，都市政策　季刊　第 117 号，2004 年 10 月号，pp. 20～30.
2. 坂村健：「ユビキタス・コンピューティング技術」，建築雑誌，pp. 8～9，2005 年.
3. 坂村健：「ユビキタス社会の現状・展望 ～ 産・官・学の進むべき方向 ～」，ESP (Economy Society Policy)，January 2005，特集：IT 化進展の検証・展望，pp. 42～45，2004 年.
4. 坂村健：「ユビキタス社会における産業と物流」，港湾，pp. 21～23，2005 年.
5. 越塚登：NEC IT Square，UID レポート連載 (http://www.blwisdom.com/uid/)
   (ア)第１回：ユビキタス ID センター
   (イ)第２回：組込み技術の先にあるユビキタス
   (ウ)第３回：食品トレーサビリティ実証実験（１）実験の狙いと方法

(エ)第４回：食品トレーサビリティ実証実験（２）実験結果
(オ)第５回：自律的移動支援プロジェクト
(カ)第６回：ユビキタス ID 技術（１）ユビキタス ID アーキテクチャ
(キ)第７回：ユビキタス ID 技術（２）ucode と ucode タグ、ユビキタスコミュニケーター
(ク)第８回：ユビキタスセキュリティー
(ケ)第９回：TRONSHOW2005 誰でもできるユビキタス（１）
(コ)第 10 回：TRONSHOW2005 誰でもできるユビキタス（２）
(サ)第１１回：ユビキタスとユニバーサル
(シ)第１２回：ユビキタスと循環型社会

6. 越塚登：「ユビキタスが拓く食の安全：食品トレーサビリティを可能としたユビキタス技術は食を豊かにする」， CLINICIAN， Vol. 52， No. 536, pp. 40～45.
7. D. E. カラー, H. マルダー：「世界を見守る賢いセンサー網」，日経サイエンス，第 34 巻，第 9 号，pp. 66～75.（翻訳監修）
8. 越塚登：「センサーネットワークには標準化が必要だ」，日経サイエンス，第 34 巻，第 9 号，p. 73.
9. 越塚登，峯岸康史：「ユビキタス ID センターが描く IC タグ普及のシナリオ」，無線 IC タグ導入ガイド，日経 BP，2004.
10. 越塚登：「ユビキタス ID センターの技術と活動」，RFID の開発と応用Ⅱ，シーエムシー出版，2004.
11. 越塚登：「文化と IT」，異文化，法政大学国際文化学部，2004 年 5 月号，pp. 11～15.
12. 越塚登：「ユビキタス ID 技術の詳細と適用事例」，Computer & Network, LAN, オーム社，2004 年 5 月号.