


	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	
--	----------------------------------	---

IPv4 アドレス枯渇対応アプリケーション チェックリスト

α 3 版

2010 年 7 月 20 日

	INTEC Systems Institute, Inc.	1/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	

改版履歴

日付	版番号	作成者	改定内容
2009/10/28	α	インテック・ネットコア	α 版作成
2010/1/29	α 2	インテック・ネットコア	アプリケーションタイプによる影響、事例の追加など
2010/7/20	α 3	インテックシステム研究所	第 5 章の API、言語情報更新

目次

1. はじめに.....	4
2. IPV4 アドレス枯渇について.....	5
2.1. IPV4 アドレス枯渇とは.....	5
2.2. ネットワーク事業者の3つの対応.....	7
2.3. 枯渇後に想定されるネットワーク構成と通信の種類.....	9
2.4. アプリケーションへの影響の可能性.....	10
2.5. アプリケーションのタイプによる影響度について.....	12
3. アプリケーションチェックリスト.....	13
3.1. (I) 動作環境のチェック.....	13
3.2. (II) アドレス変換による影響.....	14
3.3. (III) プロトコル変換による影響.....	15
3.4. (IV) IPv4 依存部分のチェック.....	16
3.5. (V) プログラム言語におけるチェック.....	18
3.6. (VI) その他のチェック.....	19
4. WEB系アプリケーションにおける注意点.....	20
5. 各プログラミング言語における整理.....	22
5.1. C/C++/C#.....	22
5.2. JAVA.....	22
5.3. PERL.....	23
5.4. RUBY.....	23
6. 関連システムのIPV6サポート状況.....	24
7. 本ドキュメントに関して.....	25
7.1. 非営利目的での利用について.....	25
7.2. 内容の充実に関する貢献についてのお願い.....	26
8. チェックリスト一覧表.....	27

1. はじめに

2011年にはインターネット上のIPアドレスの在庫が枯渇すると予想されている。枯渇後には従来はIPv4の単一なネットワークだったものが、IPv6や二重NAT、トランスレータなどが複雑に相互接続した、いわゆる「まだらなネットワーク」になってしまう可能性が高い。

一方、アプリケーション開発者はインターネット=IPv4という前提でネットワークアプリケーションを開発していると考えられ、そういう前提で開発したアプリケーションが、枯渇後の「まだらネット」で動作する保証はない。

本ドキュメントは、ネットワークアプリケーションが枯渇後の「まだらネット」でもきちんと動作するように、アプリケーション書法上の注意点をチェックリストとしてまとめたものである。新規に書くアプリケーションはもちろんのこと、既存のアプリケーションも、2011年のネットワーク事業者の対応に間に合わせる形で、本リストに従ってチェックし、修正されることをお勧めしたい。1999年には2000年問題(Y2K)としてソフトウェア業界こぞってアプリケーションのチェックを行ったが、あれほど手間はかからないにせよ、アドレス枯渇問題も同様の活動が必要になると考えている。

その趣旨を鑑み、本ドキュメントはオープンドキュメントとして自社利用等は自由とすると同時に、なるべく多くの方から追加・訂正をいただきたいと考えている。詳細な権利関係については7章に記した。

2. IPv4 アドレス枯渇について

2.1. IPv4 アドレス枯渇とは

インターネット上の住所や電話番号に相当するものが IP アドレスである。インターネット上のすべての機器にはすべてユニークな IP アドレスが割り当てられており、通信の始点と終点を表すのに IP アドレスを指定する。より正確に言えば、IP アドレスには 2 種類ある。電話番号の内線番号に相当するプライベートアドレスと、インターネット上で世界唯一のアドレスが割り当てられるグローバルアドレスがある。本ドキュメントでは特に断らない限り、IP アドレスと言えばグローバルアドレスを指すこととする。現在のインターネットで用いられている IP アドレスのバージョンは 4 であり、これを IPv4 (IP version 4) と呼ぶ。IPv4 アドレスは 32 ビット、約 43 億個の番号を表現できる。

この IPv4 の規格はインターネット標準化団体 IETF (<http://www.ietf.org/>)により 1980 年に制定され、30 年近く用いられてきた。2010 年 1 月現在、約 9%が在庫として残っており、最近の割り当て傾向を分析すると、あと 2 年程度で在庫がなくなるのは確実である。オーストラリアの研究者ジェフ・ヒューストン氏によると、世界的な在庫は 2011 年になると予想されている (図 1)。また日本の唯一の IP アドレス割り当て機関である日本ネットワークインフォメーションセンタ (JPNIC: <http://www.nic.ad.jp/>) も独自の予測を立て、ほぼ同様の時期の枯渇予測を立てている。

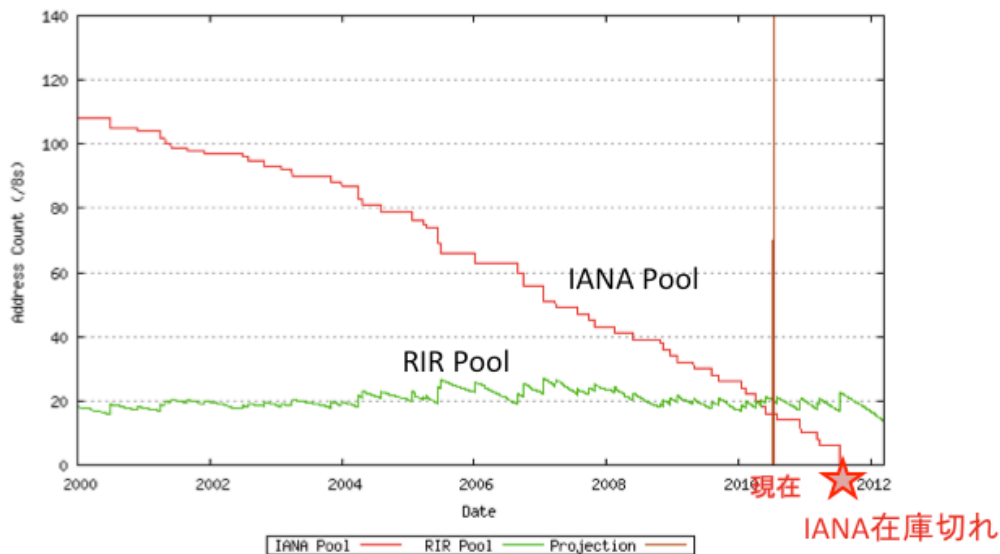


図 1 ジェフ・ヒューストン氏によるアドレス枯渇予測

さて、アドレスが足りなくなるとどうなるだろうか。

アドレス枯渇は石油などの資源枯渇とは違って、枯渇後に現在稼働しているネットワークが止まることはない。しかし、日々拡張し続けているインターネットを引き続き、拡張させていくことができなくなる。例えば、インターネットサービスプロバイダ (ISP) などのネットワーク事業者は新たな顧客に割り当てるアドレスや、新しく設置するインターネットサーバーに割り当てるアドレスが入手できなくなる。

このような理由から、ネットワーク事業者ではアドレス枯渇は、ビジネスの継続性が危ぶまれる、いわゆるリスク管理の問題としてとらえられている。図 2 に、業界 19 団体により結成されている IPv4 アドレス枯渇対応タスクフォース (<http://www.kokatsu.jp>) で発表したアクションプランを示す。大手事業者や先進的事業者は、2009 年 10 月現在、ほぼこの線表にしたがって準備を行っており、2011 年 4 月には IPv4 枯渇対応サービスを提供開始予定である。これらの新サービスは、まずは新規契約ユーザから適用されると考えられ、2011 年以降、枯渇対応サービスユーザは毎年約 200-300 万契約ずつ増えると予測される。

アクションプラン：ネットワーク関連(ISP)

ネットワーク関連プレーヤー(ISP)におけるアクションプラン (基本形)

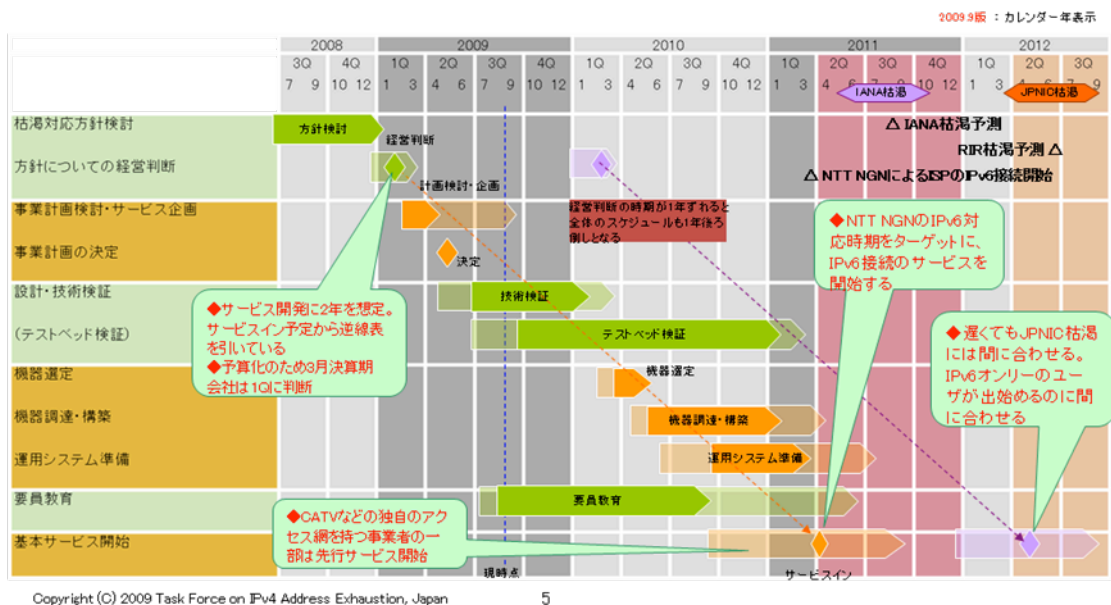


図 2 ネットワーク事業者 (ISP) のアクションプラン

2.2. ネットワーク事業者の3つの対応

ネットワーク事業者では次のように大まかに3つの枯渇対応策がある

① アドレスの回収・移転による再利用

すでに割り当て済みのアドレスの中には十分に有効活用ができていないものがあると言われている。それを再利用しようという考え方である。

実際、JPNIC などでも不要になったアドレスを再割り当てのために回収する施策を数年来実施している。また、現在のポリシーでは他者へのアドレスの移転は禁止されているが、近年の議論では移転を認めようという動きになってきている。このようにネットワークの事業者にとっては、他者からアドレスを移転してもらって、枯渇対応を行うという方法がありうる。

ただし、JPNIC によれば、このように再利用自体は必要な施策であるが、それにより枯渇問題が解消するほど十分な量は出回らないという見積もりがある。このため、この施策は単独で何とかなるものではなく、他の施策と組み合わせて実施すべきものである。

② プライベートアドレスの活用

現在の ISP の主要な方式では、グローバルアドレスを家庭単位に1つないし複数個を割り当てている。家庭内には複数の PC や情報家電などが存在する場合があるが、それらはプライベートアドレスが割り当てられ、プライベートアドレスをグローバルアドレスにマッピングするために NAT (Network Address Translation) という機能がホームゲートウェイに設置されている (図3左)。

方式②は、ISP の中に巨大な NAT 装置 (LSN: Large Scale NAT) を置き、その装置の配下の複数の加入者をプライベートアドレスで収容するというものである (図3右)。結果として、複数の家庭でひとつのグローバルアドレスを共有することになる。また、ユーザからみると、インターネットサーバーに到達するためには二重の NAT を通過する必要がある。

本方式にはいくつかの技術的な難点がある。まず、NAT という技術の制限で1アドレスに同時接続できるセッションが2バイト分つまり65000強ということがあげられる。最近のネットワークアプリケーションには1アプリケーションで数十から数百のセッションを同時に張るようなものも多く、収容制限・収容効率の点で問題が起こる。アプリケーションからすると、収容制限を超えていた場合、新しいセッションを張れないという動作になる。また、アプリケーションによっては、ユーザをアドレスによって識別しているようなものがある。例えばユーザ=アドレスを前提としたログの管理アプリケーションは意図したような動作は行えない。また、悪質なユーザの追跡が難

しくなり、セキュリティ的な問題もある。

このような問題があるため、本方式も次の③IPv6 が十分普及する前には、必要不可欠な対策と考えられているが、これだけでも枯渇問題は中長期にわたっては乗り切れない。

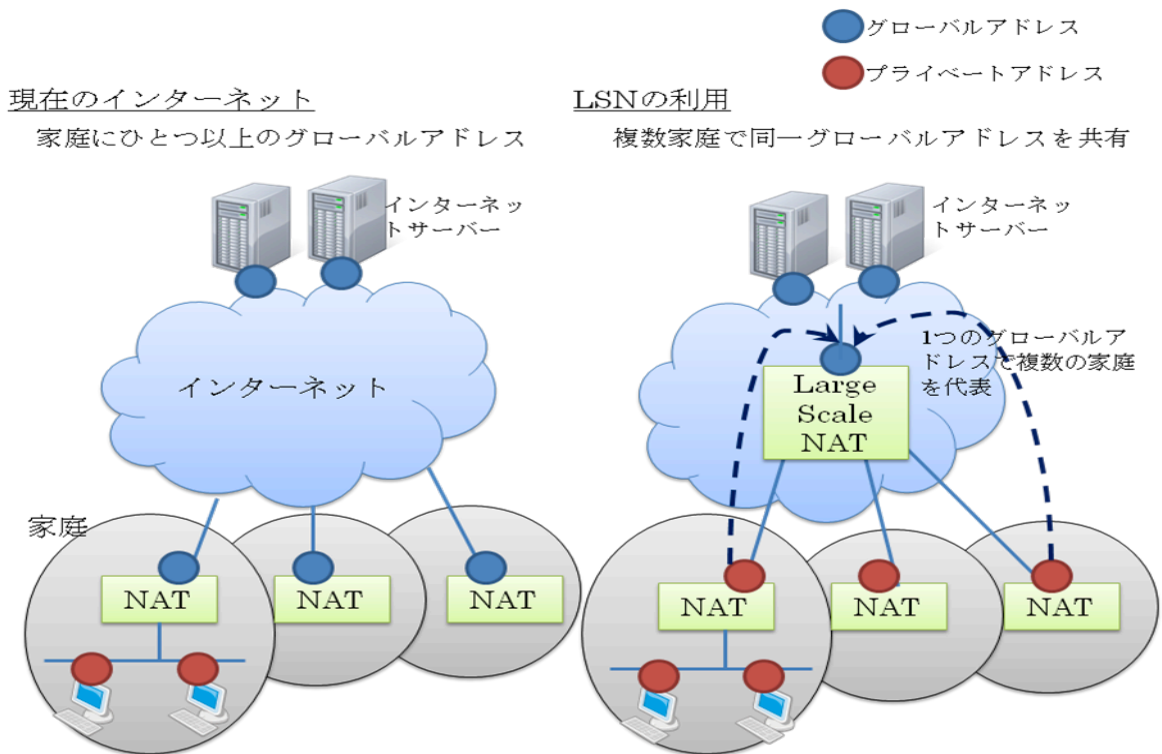


図 3 LSN のイメージ図

③ IPv6 の導入

3 つ目の対策が IPv6 の導入である。IPv6 は正式には IP version 6 と呼ばれ、1995 年に基本仕様が標準化された、IPv4 の次のバージョンの規格である。アドレスのビット幅は 128 であり、IPv4 とは天文学的な違いの空間を表現することができる。このため、中長期的には IPv6 が唯一、ほぼ永続的な解となると考えられている。

しかし、IPv6 は残念なことに IPv4 に対し、上位互換性がない。IPv6 は 2011 年以降、徐々に導入されていく予定だが、IPv6 部分が IPv4 部分に接続するためにはトランスレータなどの装置が別途必要となる。トランスレータ自体にも技術的な限界が存在する。その意味で、インターネット全体が IPv6 化してはじめて本格的に枯渇問題が解決すると言える。

上記述べたように 3 つの解決策は一長一短であり、単独で有効なものはない。したがっ

ネットワーク事業者はこの3つを並行し、使い分けていこう。典型的には、2011年以降のISPサービスは、上記の②と③を組み合わせたものとなる。すなわち、それらのユーザはIPv6およびIPv4プライベートアドレス/二重NATの両方のサービス（デュアルスタックサービス）を同時に受けることになる。一方では2011年以前に加入している既存ユーザの大半はIPv4グローバルアドレスサービスを受けており、これらの既存ユーザが当面あえてIPv6の接続に切り替えるインセンティブはない。またごく少数であるがIPv6だけのユーザの存在も否定することができない。

2.3. 枯渇後に想定されるネットワーク構成と通信の種類

前項で述べたように、枯渇前は基本的にはIPv4だけで構成されていたインターネットが、枯渇後は多様なユーザ環境をもつ、いわゆる「まだらなネットワーク」になってしまう可能性が高い。またこれは1~2年ぐらいの短期間ではなく、5年以上の長期にわたってその形態が続くと予想がもつばらである。この様子を図4にあらわした。

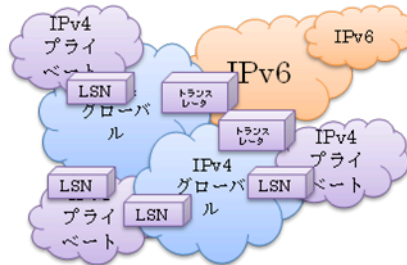
2.1節で述べたように枯渇対応新サービスは、年間200万程度増えていくとみられており、これが正しいとすれば、枯渇3年後の2014年の日本のブロードユーザの20%はIPv4プライベート/IPv6の環境にいると予測でき、決して全体からみて無視できるような割合ではない。

枯渇前のインターネット



単一のIPv4グローバルネットがフラットに相互接続

枯渇後のインターネット=まだらネット



IPv4グローバル、IPv4プライベート、IPv6が混在し、それをつなぐためにトランスレータやLSNを設置

LSN=Large Scale NAT(プロバイダ内部に置くアドレス変換装置)

図4 枯渇前と枯渇後のインターネット

これを通信のパターンとしてみると全部で下記の6通りとなる。

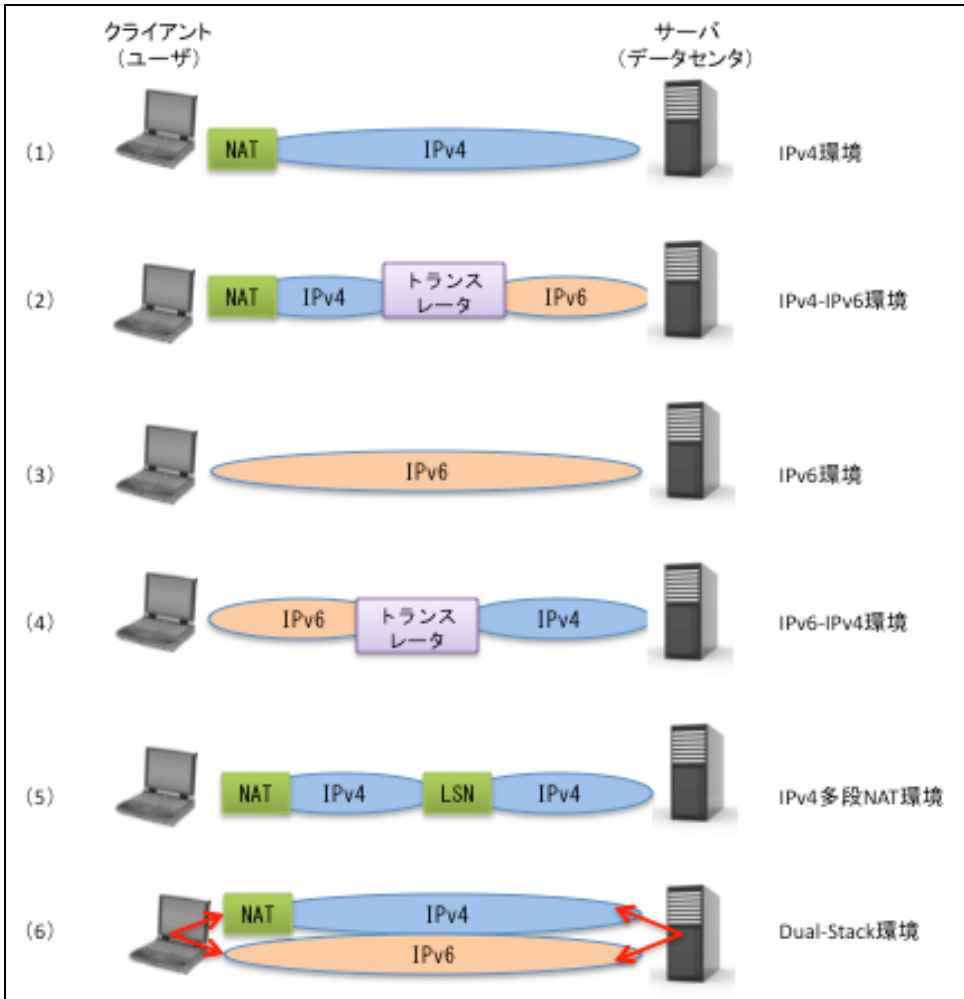


図 5 IPv4 枯渇期の通信パターン

2.4. アプリケーションへの影響の可能性

ネットワークのバージョンとアプリケーションの関係を考えてみるに、そもそもレイヤ構造的に言えば、相互に独立しているべきものである。ネットワークのバージョンが変わったとしても、アプリケーションはそのまま動作するのが理想的である。しかし、本当にそう理想的に動作するであろうか。この点が本ドキュメントの核心である。

少なくとも、今までに開発されたネットワーク上のアプリケーションの多くは、ごく最近のものを除いては、IPv4 で動作することを念頭に開発されたものであろう。プログラムのバグのいくつかは、動作環境の想定が異なったため起こるのだそうだが、まさにインターネット=IPv4 という前提で作られたプログラムがあるとするなら、それはそのまま IPv6 や「まだらなネット」上で動作する保証はない。

アプリケーションへの影響可能性としては、大まかに LSN に起因するものと IPv6 に起因するものの 2 種類があると考えられる。

[1] LSN に起因するもの

- ① UPnP が通らない
二重 NAT になると UPnP が通らなくなる。
- ② P2P アプリが動かない
例えばマッチングサーバからアドレスを貰って直接通信するようなゲームアプリケーションなどは動作しない。
- ③ セッション数の限界
LSN のポート数制限にひっかかるとセッションが張れなくなる。アプリケーション利用者からは画面が不完全に表示されるように見えたり、表示に時間がかかるようになる。
- ④ ISP ユーザの単位がアプリ側で認識できないことによる管理等の問題
広範囲のユーザがひとつの IP アドレスで代表されるため、IP アドレスの特定を前提としたアプリケーションやソリューションは意図通り動作しない。一部の EC サイト、広告バナーなどでアドレスをもとに解析、サービス提示している場合には問題が生じる。また、メール送信で POP 認証を通過したユーザの IP アドレスに送信許可を与え、SMTP 利用を可能にする POP before SMTP を利用しているような場合も問題が残る。

[2] IPv6 やトランスレータに起因するもの

- ① IPv4 アドレスが直書きしてあるプログラム
IPv4 アドレスがプログラムに埋め込まれているとうまく動作しない。
 - IPv4 アドレスの埋め込み（サーバアドレスなど）を行っていないか？
 - サーバアドレスの直書き URL はないか？
 - 設定 GUI 等で IPv4 アドレス入力を想定していないか？
 - プログラム内部処理で IPv4 アドレスを想定(4 バイト変数など)
 - IP アドレスも文字列として扱われていると問題
 - C 言語などでの IPv4 依存した型や関数の利用、ソケットの設定していないか。
例えば、gethostbyname() では IPv6 は扱えない
- ② IPv4 アドレス自体をデータとして扱うプログラム
SIP、ネットワーク管理、などでは、データとして IP アドレス自身を扱っているため、IPv6 アドレスもデータとして扱うようにする必要がある。
- ③ アドレス範囲により、動作を変えるプログラム
一部に、IPv4 のアドレスの範囲により、フィルターやコンテンツ制御をしているプログラムがあり、これらの動作は意図通りにはならない

- ④ IPv4 依存関数／ライブラリの利用
各種プログラミング言語には IP アドレスを扱う API が容易されているが、アドレスファミリーを意識しない古い API を使っていると問題である。
- ⑤ 関連システムの IPv4 依存性
OS、ミドルウェア、データベースなど関連のシステムが IPv6 をサポートし、動作できる状態になっているかはきちんと確認する必要がある。

2.5. アプリケーションのタイプによる影響度について

現在のところ判明している状況では、アプリケーションへの影響度はアプリケーションのタイプによりかなり異なるようである。以下に、影響度のイメージを示す。

タイプ	対応負荷	必要な対応例	備考
Web 系 (http)		たいていの場合は既存のコンテンツはそのまま動作。 IP アドレスを利用して来訪者数カウント、ログ管理、コンテンツ制御等を行っている場合には改修が必要。	
クライアントサ ーバソフト		ミドル・DB・フレームワーク等の IPv6 対応チェックが必要。古いシステム・古い関数・古い構造体等の利用は改修が必要。アドレス直書きもチェック必要	
ゲーム・家電ソフ ト、組込系ソフト	 	http で実現されているものは問題なし ただし P2P アーキテクチャのゲーム等はそのま までは動作しない。 また UPnP も不可になる。	
IP 電話系		トランスレータ等の助けを借りないと基本的 には動作しない	Skype, メッセンジャ ーのボイスチャット 等を含む
イントラネット アプリ		イントラネットの IPv6 化の予定がなければ、 とりあえず対応の必要なし	

3. アプリケーションチェックリスト

本章では、一般的なアプリケーションを枯渇対応ネットワークでも動作可能とする際に注意が必要となる点に関して列挙する。

以下のチェックリストにおいては、以下の記法によって各項目を記述していくこととする。

番号-番号 チェック内容

<想定動作> チェック内容が満たせないときに懸念される動作

<原因> その原因

<対処法> 問題を回避するための方策の例

<備考> 備考

<例> このチェックリストに該当するような具体例

3.1. (I) 動作環境のチェック

まずは、アプリケーションが動作する環境の確認が必要である。また、対象とするアプリケーションが正しく IPv6 に対応していた場合であっても、システムとして利用するパッケージ・フレームワークの動作に関してもチェックが必要である。

(注) アプリケーションが通信を行うネットワークについては、インターネットなど枯渇の影響を受けて「まだらなネットワーク」になる部分以外に、例えばフロントエンドサーバとバックエンドサーバをつなぐ内部ネットワーク等がある。この内部ネットワークはアプリケーションサービス運用者が特に必要としなければ IPv4 環境のまま残してもよい。したがって、本チェックリストが動作環境やソースコードのチェック対象とするのは、前者の「まだらネットワーク」上の通信に関連する部分だけである。つまり、ソースコード上の記述をチェックする際に、その通信が外部通信なのか、内部通信なのかを意識してチェックを進める必要がある。

I-1. アプリケーション自体が IP ネットワークを利用しているものかどうか？

(以下、利用しているという前提のもと、チェックを続ける)

I-2. ネットワークを利用するために、なんらかのパッケージやフレームワークを使っているか？

I-3. ミドルウェア、データベースなどの関連システムが IPv6 対応をしておき、かつ動作可能な状態になっているか？

- <想定動作> 関連システムへの通信とサーバへの通信プロトコルが異なり問題が発生する可能性がある
- <原因> 関連システムが IPv6 に対応していないため
- <対処法> IPv6 に対応したバージョンのシステムを利用する
- <備考> 主要なシステムの IPv6 対応概要については 6 章を参照

I-4. コールする他のプログラムは IPv6 に対応しているか？

- <想定動作> 自プログラムがきちんと対応していてもコールしても動作しない
- <原因> 利用するプログラムが IPv6 に対応していないため
- <対処法> IPv6 に対応したバージョンのプログラムを利用する
- <備考> 主要なプログラムの IPv6 対応概要については 6 章を参照
- <例> 付録参照

3.2. (II) アドレス変換による影響

IPv4 通信を行うアプリケーションにおいて、IPv4 アドレス枯渇期には様々な通信制限により正しく動作しない可能性がある。IPv4 アドレス枯渇期には、1 つのグローバルアドレスをより多くのユーザでシェアすることが想定され、LSN のような大規模なアドレス変換機器がネットワーク中に存在することとなる。アドレス変換ではポート番号を用いて 1 つの IP アドレスシェアを実現するため、ポート番号数 (16 ビット分 : 65536) の上限や内部のアドレスを外部から指定することが困難である。

II-1. UPnP を利用した接続手法を利用していないか？

- <想定動作> LSN 配下の二重 NAT の環境では UPnP は通らない
- <原因> LSN において UPnP 情報を伝達できないため
- <対処法> LSN において配下の NAT ルータからの UPnP 情報をリレーする仕組みを実装する等の対策が考えられるが、アプリケーションレベルでは難しい
- <備考>

II-2. NAT トラバーサル の仕組みを利用した接続手法を利用していないか？

- <想定動作> LSN 配下の二重 NAT の環境では NAT トラバーサルは通らない
- <原因> NAT トラバーサルでは NAT ルータの外側アドレスを伝え合うことで NAT の内側への通信を可能にするが、二重 NAT になると外側のアドレスが LSN 配下になり一意とならないため

<対処法> NAT ルータ自身に NAT トラバーサル機能を実装して中継することで対応が可能かもしれないが、アプリケーションレベルでは難しい

<備考> STUN(RFC3489, Simple Traversal of UDP Through NATs)サーバ等を利用した場合もこれに該当

II-3. 大量の同時セッションによる IPv4 通信を行っていないか？

<想定動作> 想定数のセッションが張れない場合がある

<原因> LSN での 1 ユーザ辺りに割り当てられるポート数制限によりセッションが拒絶されるため

<対処法> セッションを張る際にエラーが帰ってきた場合の処理で対応する
セッションによって優先度がある場合にはなんらかの対応をとる
なるべく最小限のセッションを張るようにする

<備考> 最小限のセッションを張ってもエラーになる場合は起こりうるが、その確率は低くなるはずである

II-4. サーバ側から利用ユーザを IP アドレス (IPv4) で特定する仕組みを有していないか？

<想定動作> ユーザを一意に識別できない

<原因> LSN では、複数ユーザが同じ IP アドレスを共有し、利用してしまうため

<対処法> IPv6 で実現する、そのほかのユーザ識別手段を用いるなど

<備考> アクセスログ管理などで問題となる


3.3. (III) プロトコル変換による影響

IPv6 は前述したように IPv4 との上位互換性がない。そのため、IPv4 ノードと IPv6 ノードが直接通信する場合には、トランスレータのようなプロトコル変換機器が間に必要となる。ただ、トランスレータは万能な機器ではなく、全ての通信の仲立ちを実現する場合にはアプリケーション (プロトコル) 毎にトランスレータを実装する必要がある。

一般的なプロトコル (http など) を利用するアプリケーションであっても、扱うデータ中に IP アドレスを含んだ設計であるとプロトコル変換が困難となる。さらには、IPv6 アプリケーションにおいても、IPv6 にしかない機能 (フローラベルや宛先オプションヘッダなどの拡張ヘッダ) を利用したものは、IPv4 ノードとの通信ができなくなる。

III-1. 通信データ中に IP アドレスを用いていないか？

<想定動作> ヘッダ変換型のトランスレータなどではアプリケーションが正しく動作しない

	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	
--	----------------------------------	---

- <原因> ヘッダ変換型トランスレータは、プロトコル変換時にデータの中身まで操作しないため
- <対処法> データ内でノードを指定する際に IP アドレスを用いず FQDN で指定する
- <備考>
- <例>

III-2. IPv6 アプリケーションとして IPv6 特定の機能を利用していないか？

- <想定動作> IPv6 のフローラベルなど、IPv4 に存在しない機能は正しく動作しない
- <原因> IPv6 特定の機能を IPv4 の機能に置き換える変換は不可能であるため
- <対処法> IPv4 ノードとの通信も対象とする場合には IPv6 特定の機能を含むアプリケーションとしない
- <備考> IPv4 ノードとの通信を行わないサービスであれば問題ない
- <例>

3.4. (IV) IPv4 依存部分のチェック

IP アドレスを扱う際に IPv6 アドレスを意識した実装でないと、IPv6 を扱う際にアプリケーションの改修が必要となる。IP アドレスを扱う際のチェック項目は以下の通り。

IV-1. GUI における入力項目として IPv4 だけを考慮していないか？

- <想定動作> IPv6 アドレスが入力できない
- <原因> IPv6 アドレスは 16 進数表記で最大 39 文字になり、IPv4 の 10 進数表記とは表記方法が異なるため
- <対処法> IPv6 も入力可能なように GUI を変更する
- <備考> 画面上で「□. □. □. □」と IPv4 固有の入力形式になっているものだけでなく、「□」という一つの入力形式でも入力文字数上限が 15 文字になっていたり、数字やピリオド以外の文字を禁止していたりするように IPv4 依存になっている可能性があるため留意する

IV-2. IP アドレスをデータとして扱う際に、32 ビット（4 バイト）もしくは文字列 15 文字で扱っていないか？

- <想定動作> IPv6 アドレスがデータとして扱えない
- <原因> IPv6 アドレスは 128 ビットのアドレス長を持ち、IPv4 の 32 ビットと大きく異なるため
- <対処法> データとして IPv6 でも扱えるように拡張する
- <備考> IPv6 を文字列として扱う場合には、IPv6 アドレスの表記法における省略

	INTEC Systems Institute, Inc.	16/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	

形に注意する必要がある。同一 IPv6 アドレスでも複数の表現が可能であるため、正規化を正しく行っておくことが推奨される。

<例>

IV-3. プログラムコード内に IPv4 アドレスが埋め込まれていないか？

- <想定動作> IPv6 環境では動作しない
- <原因> 通信相手を IPv4 アドレスで指定されると IPv4 限定になってしまうため
- <対処法> IP アドレスを用いず FQDN によって通信相手を指定するようにする
- <備考> サーバアドレス、ネットマスク、ループバックアドレスなどに留意する
- <例>

IV-4. アプリケーションの設定ファイルで IPv4 アドレスを指定する項目がないか？

- <想定動作> IPv6 環境では動作しない
- <原因> IPv4 アドレスを指定すると IPv4 限定の動作になってしまうため
- <対処法> IP アドレスを用いずに FQDN を指定する仕様に変更する
- <備考>
- <例>

IV-5. URL 内に IPv4 アドレスを指定されていないか？

- <想定動作> IPv6 環境では動作しない
- <原因> IPv4 による通信が強制されるため
- <対処法> IP アドレスを用いずに FQDN を指定する仕様に変更する
- <備考>
- <例>

IV-6. アドレスの範囲で動作を変える仕様を含んでいないか？

- <想定動作> IPv4 固有の範囲指定をしている場合、IPv6 環境では意図通りの動作が行われない可能性がある
- <原因> 32 ビットの IPv4 と 128 ビットの IPv6 では、アドレス長が異なるため、IPv6 の場合の範囲指定方法の考慮が必要となる
- <対処法> IPv6 での動作が可能となるよう拡張する
- <備考> IPv6 では、到達範囲を表すリンクローカルスコープ、グローバルスコープが導入されている
- <例>

3.5. (V) プログラム言語におけるチェック

アプリケーションプログラムを IPv6 対応する際には、ネットワークプログラミング部分において IP のバージョンに依存しない実装が求められる。また、各プログラミング言語において、IPv6 対応が段階的になされて来た過程から、利用するバージョンによっては古い IPv6 の仕様に基づく API も存在する。これらの点を踏まえ、プログラムで用いられている API の精査が必要である。

V-1. IPv4 しか扱えない API を利用していないか？


- <想定動作> 名前解決などにおいて IPv6 アドレスを扱えない
- <原因> IPv6 での通信を行うためには AF_INET6 を扱える必要があるため
- <対処法> IPv4 専用の API がプログラムコード中に含まれていないか調査し、IPv6 も扱える API に置き換える
- <備考> ifdef などで IPv6 対応コードを追加するだけではなく、プロトコルファミリーとは独立した実装が重要である
IPv4 依存 API の詳細は 0 章を参照

V-2. IPv6 対応した古い実装の API を使っていないか？

- <想定動作> IPv6 の通信はできるが互換性が保てない可能性がある
- <原因> IPv6 の仕様は標準化後も変化しており新しい仕様に対応できないため
- <対処法> 古い IPv6 仕様の API がプログラムコード中に含まれていないか調査し、新しい API に置き換える
- <備考> 例えばアドレスファミリーとして IPv6 の AF_INET6 が使えたとしても、C 言語における gethostbyname2(3) のように IPv6 しか扱えないような関数を利用することは移植性に欠けるなどの問題がある
古い API の詳細は 0 章を参照

V-3. IPv6 のソケットオープン時に IPv6_IPV6ONLY を指定しているか？

- <想定動作> デュアルスタック運用時に IPv4 の通信も IPv6 アプリケーションで受け取れず、しまい想定と異なる動作となる場合がある
- <原因> IPv6 のソケットを IPv6_V6ONLY オプションでオープンしないと IPv4 の通信を IPv4 射影アドレスとして扱ってしまい、IPv4 用のアプリケーションにて受信すべき通信も奪ってしまうため
- <対処法> IPv6 ソケットオープン時に IPv6_V6ONLY オプションをつける
利用するサーバ OS において IPv6_V6ONLY がデフォルトで動作する設定にする

	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	
--	----------------------------------	---

<備考>

3.6. (VI) その他のチェック

対象とするアプリケーションが正しく IPv6 に対応していた場合であっても、アプリケーションとは別のネットワーク設定により通信障害が起こる場合も想定される。ただ、アプリケーション側の対処だけで問題がすべて解決できるものではない。

VI-1. IPv4 アプリケーションで DF ビットを指定していないか？

- <想定動作> トランスレータを介した通信において正しく通信できない可能性がある
- <原因> 経路中のルータにおいて ICMPv6 がフィルタリングされている場合に、正しく最小 MTU サイズを求めること (PMTUD, Path Maximum Transmission Unit Discovery; パス最大転送ユニット ディスカバリ) ができないため
- <対処法> アプリケーションにおいて予め小さい MTU サイズを扱う (たとえば TCP の MSS を利用するなど) ことで回避できる
- <備考> 根本的な問題は経由するルータにおけるフィルタリングであり、IPv6 では ICMPv6 を全て落とす運用が問題であることを理解することが重要

	INTEC Systems Institute, Inc.	19/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	

4. Web 系アプリケーションにおける注意点

本章では特にインターネット上で大部分を占める Web 系/http 系のアプリケーションをとりあげる。前章のチェックリストはアプリケーション一般について記述しており、Web 系アプリケーションには問題にならないチェック項目も多いので、本章では少し要点を絞って記述することとする。

まず、基本的には Web 系のアプリケーションはそのままでも IPv6 環境や、LSN 環境(多重 NAT 環境)、トランスレータ環境などでも走ることが確認されている。そのようななかで特に改修が必要なケースは以下のようなケースである。

1) URI ではなく、IP アドレスを直書きしているケース

他のサーバのリソースを指定する場合に、通常は URI を指定するが、たまに IP アドレスを直書きしているケースが散見される。外部サーバの場合にはそのサーバ運用者の都合で IP アドレスが書きかえられる可能性を考えると IP アドレスを直書きすることは、無作法と考えられるが、内部のサーバを指定している場合も URI で書くべきである。

2) 訪問者数の簡易カウントとして、異なる IP アドレス数を用いているケース

LSN 配下においては、IP アドレスを契約者の識別には用いることはできなくなるため、意図とは異なった動作になる。ただし実際には IP アドレスを用いているケースはあまりなく、ログイン情報や Cookie などを用いているケースのほうが多い。

3) ログ管理などで Cookie だけでなく、IP アドレスを使っているケース

サーバログに IP アドレスを書き出していて、さらにそれをスクリプトなどで利用しているケースがたまに見かけられる。この場合、LSN 配下においては、IP アドレスを契約者の識別には用いることはできないため、意図とは異なった動作になる可能性がある。

4) 特定時間内に大量の同一 IP アドレスからのトラフィックを遮断するケース

揭示版等のセキュリティ対策として、同一 IP アドレスから短時間に大量のトラフィックがあった場合に、それをセキュリティアタックであると考え、その IP アドレスからのトラフィックを遮断するようなオペレーションがよく行われている。これは通常、ファイアウォール等の設定の問題であるが、管理アプリケーションなどでファイアウォールへの自動処理をしている場合も考えられる。

5) IP アドレスと地域情報とを紐づけているケース

IPv4 アドレスと、そのアドレスの利用地域の情報を紐づけたデータベースの提供をしている会社がある。たとえば IPv4 のアドレスが xx.xx.xx.xx ならば、それはどの都道府県、どの市町村にあるかを確率とともに示したようなデータベースである。

アクセスユーザの地域情報を考慮したデフォルト画面やデフォルト選択肢などは、ウェブページの直帰率に影響があるとのことで、さまざまところで利用されている。このほか、地域に適した広告バナーの表示、特定の地域からのアクセスだけを許す／許さない制御、アクセスユーザの地理的分布などのマーケットデータ入手、などにもこの情報は用いられている。

このような応用では、今後その動作の正確性が失われてくる可能性がある。

6) セッションを同時に大量にはるような場合

LSN においてポート数の制限により、一部のセッションが張れなくなる場合がある。ユーザからはアプリケーションが動作しない、あるいは表示が不完全のように見えるが、セッションを短く解放していくケースでは単に動作が遅く見えるだけで、ユーザは気付かないこともある。

この事象がどのくらいの頻度で起きるかは、ISP ごとの LSN の収容設計により異なる。つまり 1 グローバル IPv4 アドレスを何人で共有するかにより、この事象が起きやすいかどうかが決まる。一般にこのような ISP 内部の設計情報は非公開であるため、アプリケーション側からは、この問題の頻度・重要度は今のところ全く予想がつかないばかりか、2011 年以降も起こっている障害の事象から状況を推測するしかないものと考えられる。

また、ブラウザによって動作が違うことも確認されている。今後調査も必要となる。

5. 各プログラミング言語における整理

本章では、各言語バージョンと IPv6 対応状況の整理および、その言語固有の IPv4 依存の関数やライブラリがある場合に、注意点を列挙する。

5.1. C/C++/C#

<IPv4 依存の API (型、関数、構造体、ソケット) >

inet_addr, inet_aton, in_anaof, in_makeaddr, inet_netof, inet_network, inet_ntoa, inet_ntop, inet_pton, addr, ntoa, network, getservbyport, gethostbyname, gethostbyname2, gethostbyaddr, getservbyname, sockaddr_in, struct sockaddr, struct in_addr, INADDR_LOOPBACK, INADDR_ANY, IP_TTL, rresvport, rcmd, AF_INET, PF_INET

<IPv4 依存のコードを発見する方法>

1) 以下のコマンドをたたいてみる

```
grep gethostby *.c *.h
grep inet_aton *.c *.h
grep sockaddr_in *.c *.h
grep in_addr *.c *.h
```

2) また、Sun 上には Socket Scrubber for C/C++というツールがあり、依存関数・依存構造体をチェックできる。

<使うべき構造体/API>

sockaddr_strage, getaddrinfo(3), getnameinfo(3)

アドレスを扱う際には sockaddr 構造体を用いる


sockaddr 構造体を利用する際の注意点は Linux/Solaris には addr_len がない事

5.2. Java

<対応状況>

Solaris と Linux では、J2SE 1.4(2002年2月リリース)から対応済み。Windows では、J2SE 5.0 で対応済み。

基本的に IPv6 対応のために、Java アプリケーションコードを移植する必要はない。その前提として、IPv4 のリテラルアドレス直接参照のではなく、ホスト名を指定してあげばよい。(そして、ホストが IPv6 での通信が可能になっている事)

	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	
--	----------------------------------	---

< IPv6 関連のシステムプロパティの活用 >

優先されるアドレスやソケットの型をシステムプロパティに設定することができる。

```
java.net.preferIPv4Stack=<true|false>
```

```
java.net.preferIPv6Addresses=<true|false>
```

< IPv6 固有の新しいクラスや API の活用 >

J2SE 1.4 での IPv6 サポートにあたって、InetAddress クラスのサブクラスとして、Inet4Address と Inet6Address が新設されている。アドレスファミリー固有の動作にアクセスする必要がある場合は、サブクラスを活用できる。基本的には、プロトコルバージョンを意識しない、InetAddress を使うことが推奨される。

5.3. Perl

バージョン 5.10.0 で対応済み。

CPAN モジュールを利用するには IPv6 対応しているか確認する必要がある。

5.4. Ruby

バージョン 1.9.1 では、デュアルスタック通信が不可能。

IPSocket.getaddress で IPv6 アドレスを指定すると IPv6 シングルスタックでしか動作しない。また、ソケットライブラリである、UDPSocket, TCPServer が、IPv4 プロトコル依存となっている。バージョン 1.9.2 で、これらの問題を改修予定である。

	INTEC Systems Institute, Inc.	23/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	

6. 関連システムの IPv6 サポート状況


表 1 製品の IPv6 対応バージョン

ベンダ	製品名	対応バージョン	備考
Microsoft	.NET Framework	1.1 以降	
Microsoft	SQL Server	2005 以降	
Microsoft	IIS	6.0 以降	
Oracle		11g R2	Oracle RAC, Oracle Clusterware, Oracle Fail Safe については未対応
—	Perl	5.10.0	CPAN などのライブラリは未対応
—	Ruby	1.9.1/1.9.2	1.9.1 では、最初に IPv6 で宣言すると、IPv6 Only となり、Dual Stack にならない。これは 1.9.2 で現在修正中
—	Java	J2SE 1.4	Solaris, Linux
		J2SE 5.0	Windows
—	PostgreSQL	8.2.3 以降	

本表はあくまで目安として記述しているものです。正確・詳細な情報については各製品のホームページなどを直接お調べください。

その他、関連情報として以下の情報が参考になります。

- 1) Basic Socket Interface Extensions for IPv6, RFC 3493
<http://tools.ietf.org/html/rfc3493> - <http://tools.ietf.org/html/rfc3493>
- 2) Advanced Sockets Application Program Interface (API) for IPv6, RFC3542
<http://tools.ietf.org/html/rfc3542> - <http://tools.ietf.org/html/rfc3542>
- 3) Sun “IPv6 とインターネットの将来”
<http://jp.sun.com/practice/software/solaris/jp/wp/wp-ipv6/>

	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	
--	----------------------------------	---

7. 本ドキュメントに関して

7.1. 非営利目的での利用について

本ガイドラインは、「(2) 営利目的等での利用について」に記載するケースに該当する場合を除き、クリエイティブコモンズライセンス（表示-非営利-改変禁止 2.1 日本）の下、無償で利用することができます。（営利目的でソフトウェアを開発する場合のガイドラインとして、利用することができます。）

(1) 【クリエイティブコモンズライセンス】

あなたは以下の条件に従う場合に限り、自由に本作品を複製、頒布、展示、実演することができます。あなたの従うべき条件は以下の通りです。

- ・表示. あなたは原著作者のクレジットを表示しなければなりません。
- ・非営利. あなたはこの作品を営利目的で利用してはなりません。
- ・改変禁止. あなたはこの作品を改変、変形または加工してはなりません。
- ・再利用や頒布にあたっては、この作品の使用許諾条件を他の人々に明らかにしなければなりません。
- ・著作[権]者から許可を得ると、これらの条件は適用されません。

正式なクリエイティブコモンズライセンスについては、下記の URL をご覧ください。

<http://creativecommons.org/licenses/by-nc-nd/2.1/>

(2) 営利目的等での利用について

以下に該当する場合は、弊社窓口までお問い合わせください。

- ・営利を目的とした企業または団体において、顧客等に対し複製を用いた研修、講義、出版、頒布、若しくは公衆送信等を行う場合
- ・営利を目的としない団体（特定非営利活動法人等）が営利事業として研修、講義、出版、頒布、若しくは公衆送信等を行う場合


営利利用をご希望される場合は、弊社窓口までお問い合わせください。また本ライセンスに記載されていない利用形態（本ガイドラインの全部または一部を改変、変形または加工して利用しようとする場合等を含みます）をご希望の場合もご遠慮なくご相談下さい。

弊社窓口

株式会社インテックシステム研究所

ガイドライン関連窓口（電子メールのみ）: hiromi_ruri@intec-si.co.jp

	INTEC Systems Institute, Inc.	25/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	

	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	
--	----------------------------------	---


7.2. 内容の充実に関する貢献についてのお願い

本ガイドラインの内容の充実に関してご協力いただける場合は、弊社窓口までご連絡ください。弊社編集規定により内容を精査させていただき、改版時に反映させていただくことがございます。その際、原則として対価をお支払いすることはございませんが、ご希望によりご貢献いただいた旨とともに企業名または団体名若しくは個人のお名前を本ガイドライン中に記載させていただきます。

	INTEC Systems Institute, Inc.	26/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	

8. チェックリスト一覧表

(I) 動作環境のチェック	
I-1	アプリケーション自体が IP ネットワークを利用しているものかどうか？
I-2	ネットワークを利用するために、なんらかのパッケージやフレームワークを使っているか？
I-3	ミドルウェア、データベースなどの関連システムが IPv6 対応をしておき、かつ動作可能な状態になっているか？
I-4	コールする他のプログラムは IPv6 に対応しているか？
(II) アドレス変換による影響チェック	
II-1	UPnP を利用した接続手法を利用していないか？
II-2	NAT トラバーサル仕組みを利用した接続手法を利用していないか？
II-3	大量の同時セッションによる IPv4 通信を行っていないか？
II-4	サーバ側から利用ユーザを IP アドレス (IPv4) で特定する仕組みを有していないか？
(III) プロトコル変換による影響チェック	
III-1	通信データ中に IP アドレスを用いていないか？
III-2	IPv6 アプリケーションとして IPv6 特定の機能を利用していないか？
(IV) IPv4 依存部分のチェック	
IV-1	GUI における入力項目として IPv4 だけを考慮していないか？
IV-2	IP アドレスをデータとして扱う際に、32 ビット (4 バイト) もしくは文字列 15 文字で扱っていないか？
IV-3	プログラムコード内に IPv4 アドレスが埋め込まれていないか？
IV-4	アプリケーションの設定ファイルで IPv4 アドレスを指定する項目がないか？
IV-5	URL 内に IPv4 アドレスを指定されていないか？
IV-6	アドレスの範囲で動作を変える仕様を含んでいないか？
(V) プログラム言語におけるチェック	
V-1	IPv4 しか扱えない API を利用していないか？
V-2	IPv6 対応した古い実装の API を使っていないか？
V-3	IPv6 のソケットオープン時に IPv6_IPV6ONLY を指定しているか？
(VI) その他のチェック	
VI-1	IPv4 アプリケーションで DF ビットを指定していないか？


	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	
--	----------------------------------	---

付録

チェックリスト項目の事例集


	INTEC Systems Institute, Inc.	28/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	

通番	1	確認日付	2009. 11
チェックリスト項目	II-3 大量のセッションを張るアプリケーション		
事例内容	<p>Google Map アプリケーションを、セッション数を限定した LSN 上で動作させると、画面が欠けたり、動作が安定しなくなった。</p> <p>一方、別の大量のセッションを張るアプリケーションにおいても同様の実験をしたが、動作はかなり異なるものとなった。アプリ側で生成するセッション数より少ないセッション数の設定の LSN で動作させたにもかかわらず、</p> <ul style="list-style-type: none"> ・動作が若干遅くなる ・たまに表示が崩れるがコンテンツをリロードすると OK <p>という動作になり、あまり大きな問題は起きなかった。</p> <p>動作解析の結果、後者のアプリケーションにおいては、コンテンツを短時間で取得後、セッションを解放しており、他のセッションはタイムアウト前にリソースを掴めるといったような動きとなっていた。一方で、Google Map はセッションを張りっぱなしになっているため、他のセッションがタイムアウトになってしまっていた。</p>		
対処法	<p>不必要なセッションは張らない、不要にセッションを張りっぱなしにしない、などの配慮がされているほうが枯渇に強い。</p>		
備考			

	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	
--	----------------------------------	---


通番	2	確認日付	2009.11
チェックリスト項目	II-3 大量のセッションを張るアプリケーション		
事例内容	<p>項目1の「別の大量のセッションを張るアプリケーション」において、たまに出た不具合の事象として、表示の一部が欠けるのではなく、全体が崩れるような表示になることがあった。これは、表示の枠組み・フォーマットを取得するセッションがたまたまタイムアウトしたからであると推測された。</p>		
対処法	<p>重要なセッションは先に張る、張りっぱなしにするなどの対応が有効な場合もある</p>		
備考			

	INTEC Systems Institute, Inc.	30/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	

	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	
--	----------------------------------	---

通番	3	確認日付	2009.12
チェックリスト項目	その他		
事例内容	<p>とある業務イントラアプリのサブシステムにおいて、メールシステム（通常の SMTP/POP によるメールシステム）を装備していた。このメールシステムの運用の仕方によっては、問題が生じる可能性がある。</p> <p>世の中のメールシステムの運用では、SMTP の認証として直前の POP の認証を持って代替するやり方 (POP Before SMTP) が一部でなされているが、LSN 配下では誤動作する可能性がある。</p>		
対処法			
備考	<p>基本的にはアプリケーションの話ではないが、アプリケーション運用上で関連するので記しておく。</p>		


	INTEC Systems Institute, Inc.	31/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	

		
	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	

通番	4	確認日付	2010.1
チェックリスト項目	II-1 UPnP II-2 NAT トラバーサル		
事例内容			
<p>今まで動作していた一重 NAT の配下のウェブカメラが二重 NAT 配下になったら動作しなくなった。</p>			
対処法			
備考			
<p>試験環境で確認。</p> <p>基本的にはアプリケーションの話ではないが、関連するので記しておく。</p>			

	INTEC Systems Institute, Inc.	32/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	

通番	5	確認日付	2009.10-2010.1
チェックリスト項目	NA		
事例内容	<p>いくつかの典型的かつそれなりに複雑なウェブアプリ (http ベース)いくつかを、</p> <ul style="list-style-type: none"> • IPv4-NAT-IPv4 (NAT ポート制限なし) • IPv4-NAT-NAT-IPv6 (NAT ポート制限なし) • IPv6-IPv6 • IPv4-Translator-IPv6 • IPv6-Translator-IPv4 <p style="text-align: center;">(Client 環境—middle boxes—Server 環境)</p> <p>の環境で動作させたが、動作上問題となるところはほとんどなかった</p>		
対処法			
備考	<p>厳密には、詳細な実験条件を示さないと情報として意味がないが、詳細については2010.3月以降に某所より報告書を発表予定。</p> <p>ほぼ問題なく動くこと自身も情報となるので、あえて挙げておく</p>		

	IPv4 アドレス枯渇対応 アプリケーションチェックリスト	
--	----------------------------------	---

通番		確認日付	
チェックリスト項目			
事例内容			
<p>****事例募集中****</p> <p>実ネットで判明したこと、 実証実験してわかったこと、 机上でわかったこと、 なんでも関連事例を募集中</p>			
対処法			
備考			

	INTEC Systems Institute, Inc.	34/34
ファイル名	IP アドレス枯渇対応アプリケーションチェックリストα3	
最終更新日時	2010-07-20	