講演資料

【実践】IPv6版ポート解放

～家庭内ネットワークにおけるIPv6通信パターンの変化の波を乗りこなそう～

株式会社コナミデジタルエンタテインメント

技術開発部　佐藤元彦

©Konami Digital Entertainment

「ポート解放」という言葉、

どれだけ説明できますか？

「ポート解放」なんて今の時代もう必要ない。
必要に応じてUDPHolePunchingを使用したベスト
エフォードな接続確立で、無理ならサーバ経由の
通信へフォールバックすればいいだけだ

こんなお気持ちの方、多いと思います。

（そして、これはある意味では正しい…）

しかし、ポート解放を活用した通信パターンならではの

メリットや、代替できない用途があります。

＋

IPv6でも、ポート解放が活用できる舞台が整ってきました。

今日は適切な場面でIPv6のポート解放を活用して、
より良い通信体験が可能な未来を設計できるように
なろう、というビジョンを議論します！

Agenda

- ★ IPv4/IPv6それぞれにおける、ポート解放の定義・意味
- ★ 家庭内ネットワークの通信パターンの変化
  - ★ ゲーム開発の視点より
  - ★ ルータベンダーからの視点より
- ★ 議論パート
- ★ まとめ

自己紹介

名前：佐藤 元彦

略歴：2008年 株式会社コナミデジタルエンタテインメント入社

仕事：オンラインゲームのネットワーク技術開発/サポート

> 研究：NAT越えアルゴリズム、IPv6、IPv4/IPv6共存技術、モバイルブロードバンド、クラウド技術

> 開発：NAT越え+IPv4/v6デュアルスタック P2P通信ライブラリ、WANエミュレータ

自己紹介

過去の講演資料

- [CEDEC 2025] IPv6向けNATの台頭と新たなNAT挙動を理解・対応実装を学び、変化するIPv6環境に備えよう！
  - https://cedec.cesa.or.jp/2024/session/detail/s65fbdad43ad54/
- [JANOG53] ついにIPv6向けUPnPが実運用フェーズに！〜 ゲームのP2Pオンライン対戦での活用フィードバックを添えて 〜
  - https://www.janog.gr.jp/meeting/janog53/upnp/
- [CEDEC 2022] ゲームにおけるIPv6向けUPnPの活用可能性と実装検証
  - https://cedec.cesa.or.jp/2022/session/detail/87.html
- [CEDEC 2021] ゲームトラフィックの動向と課題、それに対する5G関連技術の可能性
  - https://cedec.cesa.or.jp/2021/session/detail/s609c9048ac70c.html
- [CEDEC 2020] 次世代機開発におけるIPv6実用のために必要な環境構築・検証・調査手法
  - https://cedec.cesa.or.jp/2020/session/detail/s5e9be5c4270c9
- [CEDEC 2019] [JANOG×CEDECコラボセッション] ネットワーク事業者と語るインターネットのゲーム通信
  - https://cedec.cesa.or.jp/2019/session/detail/s5cd41730206fa.html
- [Internet Week 2019] ゲームにおけるIPv4の品質変化と対策事例
  - https://www.nic.ad.jp/iw2019/program/s02/
- [JANOG43] IPv4/IPv6デュアルスタックなリアルタイムP2P通信を行うオンラインゲームにおける現在の国内/海外ネットワーク環境とそれに対する検証環境の構築手法
  - https://www.janog.gr.jp/meeting/janog43/program/p2pv4v6
- [CEDEC 2018] コンシューマー・モバイルタイトルでIPv4/IPv6デュアルスタックなP2P通信をサポートしてきた中でやった事
  - https://cedil.cesa.or.jp/cedil_sessions/view/1820
- [CEDEC 2015] 多様なモバイルブロードバンド環境でリアルタイム通信を行なう上で考えるべき遅延特性
  - https://cedil.cesa.or.jp/cedil_sessions/view/1377
- [CEDEC 2014] モバイルブロードバンド時代におけるP2P通信の落とし穴
  - https://cedil.cesa.or.jp/cedil_sessions/view/1236
- [CEDEC 2013] Router & Network Report 2013 for P2P Online Game
  - https://cedil.cesa.or.jp/cedil_sessions/view/1041
- [CEDEC 2012] IPv4-IPv6 移行期のP2Pゲームクライアントに求められる技術
  - https://cedil.cesa.or.jp/cedil_sessions/view/903

オープンソース活動

- EM-uNetPi（WANエミュレーター）
  - https://github.com/KONAMI/EM-uNetPi

IPv4/IPv6それぞれにおける
ポート解放の定義・意味

まえおき：よく混同される話

ポート開放に限った話ではないですが、DS-LiteやMAP-E環境における ”IPv4の” 通信やルータの設定を “IPv6の” と表現されているケースを、多々見かけます。

正確には「IPv4共存技術下(IPv4 over IPv6環境下)におけるIPv4」であり、今日の話において、これらは「IPv4のポート解放」に該当する環境です。勘違いのなきよう！

IPv4共存技術をIPv6と混同するのをやめよう

IPv4/IPv6それぞれにおける
ポート解放の定義・意味

復習：

そもそも「ポート解放」とは何か？

関連資料：「IPv6ポート開放」という概念を整理・理解しよう
https://www.janog.gr.jp/meeting/janog51.5/doc/janog51.5-sato-lt.pdf

ポート解放とは何か？

標準化された用語ではないので、今日は家庭用ネットワークで用いられている背景を考慮しつつ、NAPTと接続性の観点から、以下の要素を実現する処理を指すものとします。

IPv4+IPv6	要素1	A. フィルタリング挙動 (Filtering Behavior => RFC4787) をEIFへ
IPv4	要素2	B. マッピング挙動 (Mapping Behavior => RFC4787) をEIMへ
IPv4	要素3	C. アドレスバインディング (NAT Binding => RFC2663) を生成する
IPv6	要素4	D. SPI (Stateful Packet Inspection) セッションを生成する

IPv6にNAPTがあるのは現状レアケースだが
SPIは存在するため、このような定義とする。
（一部地域でNAT66の増加は観測されているが、一旦忘れよう…）

IPv4+IPv6	要素1	A. フィルタリング挙動 (Filtering Behavior => RFC4787) をEIFへ
IPv4	要素2	B. マッピング挙動 (Mapping Behavior => RFC4787) をEIMへ
IPv4	要素3	C. アドレスバインディング (NAT Binding => RFC2663) を生成する
IPv6	要素4	D. SPI (Stateful Packet Inspection) セッションを生成する

ポ

標準化

いる背

る処理を指すものとします。

クライアントから見て、これらを実現する処理が
本来の通信に先んじて行われることで、その後の
接続処理・通信がより安定して行えるようになる

れて

現す

IPv4+IPv6 要素1 A. フィルタリング挙動 (Filtering Behavior => RFC4787) をEIFへ

IPv4 要素2 B. マッピング挙動 (Mapping Behavior => RFC4787) をEIMへ

IPv4 要素3 C. アドレスバインディング (NAT Binding => RFC2663) を生成する

IPv6 要素4 D. SPI (Stateful Packet Inspection) セッションを生成する

つまり、ポート解放の意味とは？

ポート解放との意味とは？

クライアントから見て、前述の要素を実現する処理を行なうことで、その後の接続処理・通信が安定して行えるようになり、アプリケーションの取れる **通信パターンの選択肢が増え** 、ユーザの **通信体験がより向上** する。

P1 クライアントから通信を開始する
クライアント<=>サーバ間の通信

P2 サーバから通信を開始する
サーバ<=>クライアント間の通信

P3 P2P通信

P1 接続確立にかかる時間の短縮

P2 Binding Table / Session Table のタイムアウト事故を防止

P3 対応機種においてはQoSが適用され、帯域・遅延品質が安定する

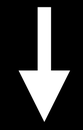
ポート解放との意味とは？

これらのポイントを、以下の3つの区分から比較します。

- A. 何もしない：特にNAT/Firewall越えの処理をしない場合
- B. UDP HolePunching：UDPHolePunchingのみを使って接続する場合
- C. ポート解放：ポート解放（+必要に応じてUDPHolePunchingを併用）を使って接続処理を行う場合

ポート解放との意味とは？

クライアントから見て、前述の要素を実現する処理を行なうことで、その後の接続処理・通信が安定して行えるようになり、アプリケーションの取れる 通信パターンの選択肢が増え 、ユーザの 通信体験がより向上 する。



P1 クライアントから通信を開始する
クライアント<=>サーバ間の通信

P2 サーバから通信を開始する
サーバ<=>クライアント間の通信

P3 P2P通信

何もしない	UDPHolePunching	ポート解放
○	○	○
×	○ ※1	○ ※2
×	○ ※1	○ ※2

同じ「◯」でも、事前に必要な通信処理が異なるので、パターンとしては別物である。UDPHolePunchingはマッチングサーバ等でIP交換 + 同期処理をした上でUDPHolePunchingを実施する。一方でポート解放はクライアント単体でルータに対して処理を実施し、通信を開始する側に対してIPアドレスを通知する。

=> 処理の実装にかかるコスト・実現のためのインフラコストが異なる

P1 クライアントから通信を開始する
クライアント<=>サーバ間の通信

P2 サーバから通信を開始する
サーバ<=>クライアント間の通信

P3 P2P通信

何もしない	UDPHolePunching	ポート解放
◯	◯	◯
×	◯ ※1	◯ ※2
×	◯ ※1	◯ ※2

ポート解放との意味とは？

クライアントから見て、前述の要素を実現する処理を行なうことで、その後の接続処理・通信が安定して行えるようになり、アプリケーションの取れる 通信パターンの選択肢が増え 、ユーザの 通信体験がより向上 する。

何もしない	UDPHolePunching	ポート解放
×	×	○
×	×	○
×	×	○

P1 接続確立にかかる時間の短縮

P2 Binding Table / Session Table のタイムアウト事故を防止

P3 対応機種においてはQoSが適用され、帯域・遅延品質が安定する

用途によっては、UDPHolePunchingが都度行われる時間を許容できないこともある。また、いわゆるKeepAlive的な処理をクライアントを実装している場合には、環境依存であるルータのTimeoutの閾値を下回ると事故につながる。

=> 妥協できないシーンへの対応、細かいところでもQoE向上に貢献

何もしない	UDPHolePunching	ポート解放
×	×	○
×	×	○
×	×	○

P1 接続確立にかかる時間の短縮

P2 Binding Table / Session Table のタイムアウト事故を防止

P3 対応機種においてはQoSが適用され、帯域・遅延品質が安定する

ポート解放に問題がないわけでもない

ポート解放の問題点

使えるならば使った方がベターなケースが多いと感じられたかもしれませんが、ポート開放処理による恩恵をクライアントが受けるには、3つの課題があり、それらをクリアしなくてはなりません。

P1 クライアント・ルータの実装が大変！（とりわけ、IPv6向けのUPnPは標準仕様の策定が十分ではない上に、こなれた実装が普及していない）

P2 あくまで、クライアントの直上のルータに対して行う処理なので、さらに上流にNAPTが存在する場合（CGNATなど）、効果が出ない or 薄れる

P3 クライアント・ルータの実装の組み合わせによっては、不具合が出ることがあり、十分な検証と、適切なフォールバック処理が必要

ポート解放の問題点

国内初の実装・運用について、および、これから実装する人向けのメッセージはJANOG53のプログラム資料を参照
=> 「ついにIPv6向けUPnPが実運用フェーズに！～ ゲームのP2Pオンライン対戦での活用フィードバックを添えて ～」（https://www.janog.gr.jp/meeting/janog53/upnp/）

P1 クライアント・ルータの実装が大変！（とりわけ、IPv6向けのUPnPは標準仕様の策定が十分ではない上に、こなれた実装が普及していない）

P2 あくまで、クライアントの直上のルータに対して行う処理なので、さらに上流にNAPTが存在する場合（CGNATなど）、効果が出ない or 薄れる

P3 クライアント・ルータの実装の組み合わせによっては、不具合が出ることがあり、十分な検証と、適切なフォールバック処理が必要

ポート解放の問題点

クライアントに関しては、C#でのサンプル実装を「ゲーム・エンタメのネットワーク接続性に関する課題検討WG IPv6対応UPnP実装・検証SWG」より、Githubにて公開を開始
=> upnpv6-swg / upnpv6-router-list（https://github.com/upnpv6-swg/upnpv6-router-list/blob/main/TestClient/ImplementationGuide.md）

P1 クライアント・ルータの実装が大変！（とりわけ、IPv6向けのUPnPは標準仕様の策定が十分ではない上に、こなれた実装が普及していない）

P2 あくまで、クライアントの直上のルータに対して行う処理なので、さらに上流にNAPTが存在する場合（CGNATなど）、効果が出ない or 薄れる

P3 クライアント・ルータの実装の組み合わせによっては、不具合が出ることがあり、十分な検証と、適切なフォールバック処理が必要

ポート解放の問題点

ネットワークの経路上で、Firewallをかけるべきなのは誰なのか、どこの透過性は維持しなくてはならないのか？
=> IPv4のCGNATが流行り始めた時期に、EIM/EIFのサポートが重要視された過去もありました。議論パートで話題にあげたいネタです。

P1 クライアント・ルータの実装が大変！（とりわけ、IPv6向けのUPnPは標準仕様の策定が十分ではない上に、こなれた実装が普及していない）

P2 あくまで、クライアントの直上のルータに対して行う処理なので、さらに上流にNAPTが存在する場合（CGNATなど）、効果が出ない or 薄れる

P3 クライアント・ルータの実装の組み合わせによっては、不具合が出ることがあり、十分な検証と、適切なフォールバック処理が必要

ポート解放の問題点

（議論パートの話を先取りして、軽く触れると…）
「ポート解放なんて不要だ！」とISPやCGNベンダーの方が思ってしまうと、上流でフィルタリング等をかけることが常態化してしまい「ポート解放」というパターン自体がネットワーク設計から消滅し滅びます。今日の議論はそうならないためのものでもあります。

P1 クライアント・ルータの実装が大変！（とりわけ、IPv6向けのUPnPは標準仕様の策定が十分ではない上に、こなれた実装が普及していない）

P2 あくまで、クライアントの直上のルータに対して行う処理なので、さらに上流にNAPTが存在する場合（CGNATなど）、効果が出ない or 薄れる

P3 クライアント・ルータの実装の組み合わせによっては、不具合が出ることがあり、十分な検証と、適切なフォールバック処理が必要

ポート解放の問題点

先ほどのリポジトリでは、検証用のクライアントも配布中。ルータベンダーの方は、ぜひこちらを使って検証いただければ！
=> upnpv6-swg / upnpv6-router-list（https://github.com/upnpv6-swg/upnpv6-router-list/blob/main/TestClient/ImplementationGuide.md）

P1 クライアント・ルータの実装が大変！（とりわけ、IPv6向けのUPnPは標準仕様の策定が十分ではない上に、こなれた実装が普及していない）

P2 あくまで、クライアントの直上のルータに対して行う処理なので、さらに上流にNAPTが存在する場合（CGNATなど）、効果が出ない or 薄れる

P3 クライアント・ルータの実装の組み合わせによっては、不具合が出ることがあり、十分な検証と、適切なフォールバック処理が必要

ポート解放の問題点

UDPHolePunchingに一本化した方がよい、とよく言われるのも、こういった課題のクリアが大変＆リスクになってしまうからですね…

補足：リポジトリ紹介

upnpv6-swg / upnpv6-router-list
https://github.com/upnpv6-swg/upnpv6-router-list/

このような場合に用いられるのは、Pinhole機能（IPv6対応UPnP）ではなく、従来のIPv4向けのUPnPであるため、注意すること

なお、IPv4 over IPv6環境下において、従来のIPv4対応UPnP機能が機能するか否かについても、実際のネットワーク環境と

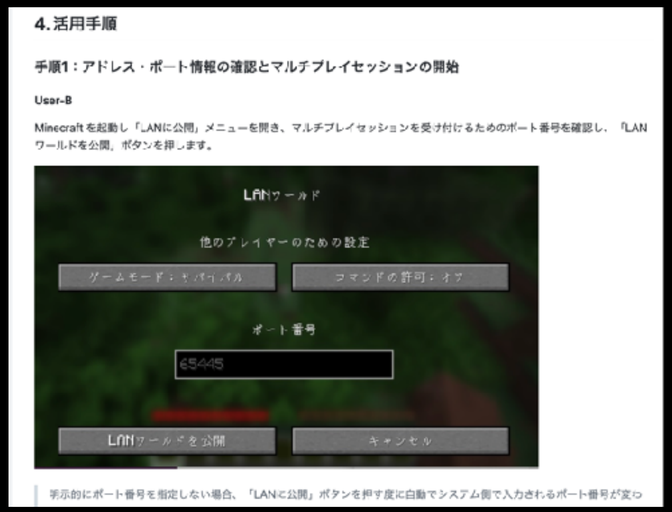
3. IPv6対応UPnP 検証済ルータリスト

掲載順はローマ字表記のA-Za-zとします。

3.1 NECプラットフォームズ株式会社

機種	ファームウェア	検証日	検証結果	製品情報	備考
Aterm WX5400HP	Ver2.0.1	2025.03.28	A	Webサイト	
Aterm WX5400T6	Ver1.2.4	2025.03.28	A	Webサイト	
Aterm 7200D3BE	Ver1.1.1	2025.04.09	A	Webサイト	

```
42  namespace jpn {
46  public class UPnPv6Client : MonoBehaviour
118     public async Task AddPinhole(){
139         logMsgTf.text += "No IPv6 UPnP service found.\n";
140         goto wayout;
141             }
142
143         logMsgTf.text += " >> OK\n";
144             Debug.Log("Fetching control URL...");
145
146             controlUrlCache = await GetControlUrlAsync(location);
147             if (string.IsNullOrEmpty(controlUrlCache))
148             {
149         Debug.Log("Control URL not found.");
150         logMsgTf.text += "Control URL not found.\n";
151         goto wayout;
152             }
153     }
154     else {
155         logMsgTf.text += "Cache Exist. Skip Discovery Phase.\n";
156     }
157
158     logMsgTf.text += $"Control URL: {controlUrlCache}\n";
159
160         // HttpClientが使用する送信元IPv6アドレスを取得
161     // 機種依存処理になるため、プロダクト実装時には要検証
162         internalAddress = GetLocalIPv6Address(controlUrlCache);
163         Debug.Log($"Http Client Using IPv6 address: {internalAddress}");
164     if(internalAddress == ""){
165         logMsgTf.text += "Host Address not found.\n";
166         goto wayout;
167     }
168     logMsgTf.text += $"Host Address: {internalAddress}\n";
169
```



4. 活用手順

手順1：アドレス・ポート情報の確認とマルチプレイセッションの開始

User-B

Minecraftを起動し「LANに公開」メニューを開き、マルチプレイセッションを受け付けるためのポート番号を確認し、「LANワールドを公開」ボタンを押します。

- WGで検証を行ったルータのリスト
- IPv6対応UPnPのサンプル実装（クライアント）／クライアント
- ゲームでの活用事例集

ゲームから見た
ポート解放のメリット

ポート解放のメリット

前述のように、ポート解放を活用した通信パターンは、UDP HolePunchingだけを使ったものに比べて優位性がある。有効な場面ではコストと天秤にかけて、採用する通信パターンを決めていきたいですね。

P1	インフラをより削減しやすい構成、サーバ経由のフォールバック削減が可能
P2	通信確立までの時間をより抑えることができる
P3	タイムアウトがシビアに設定された機器でのトラブル削減
P4	暗号化されて、QoS制御が適切に機能しずらいプロトコルでも、適切なQoS処理が可能になる（対応機器の場合）

ゲームのプレイ環境において
ポート解放を行う手段

ポート解放を行う手段

現在主流となっているポート解放の方法は

A. 「ルータの管理画面からPortForward・Firewallの設定を操作する」
B. 「UPnPを使用して制御する」

IPv4ではA/Bいずれもよく知られた手法だが、IPv6では、まだまだ普及段階。

- A：IPv4に比べてアドレス体系・記法が複雑であり、手動での入力が難しい
- B：対応ルータ・クライアントがまた普及過程

※ ちなみに、UPnPによる制御に関しては「ゲームプラットフォームが制御する」「ゲームソフトウェアが制御する」「ユーザが任意のソフトウェアを使用する」という3つのパターンにさらに分類できる。

このパートのまとめ

ぼんやりとした「ポート解放」という言葉について
しっかりと整理と理解が完了！

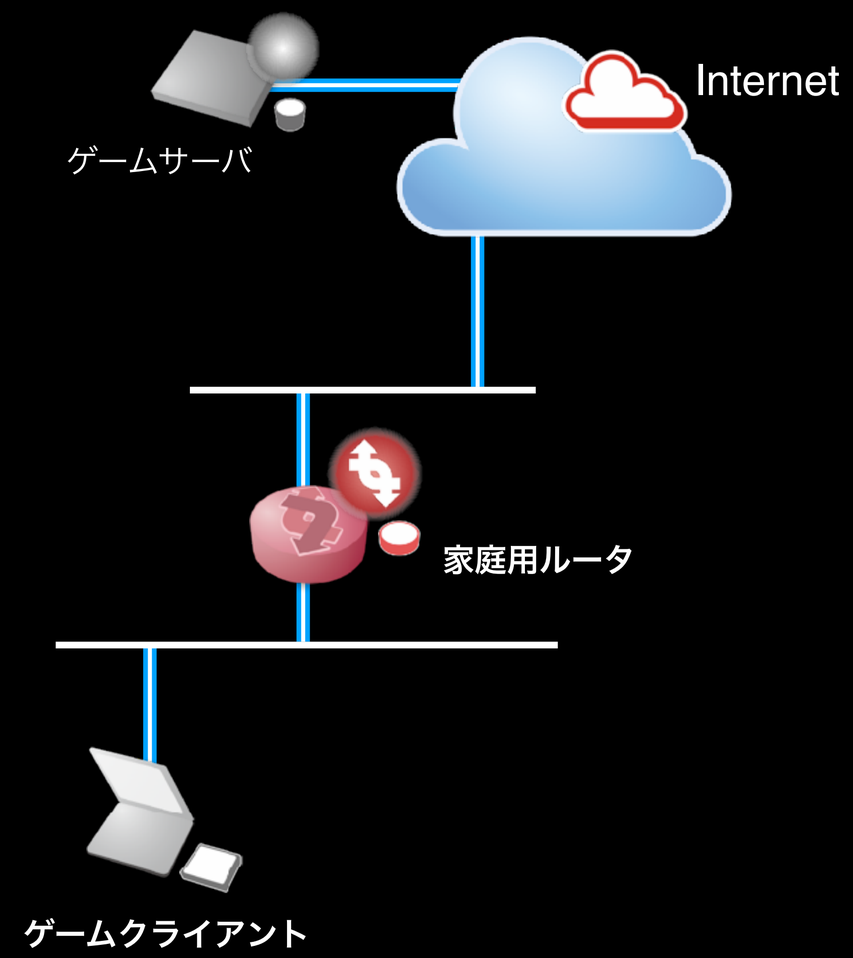
ポート解放によって実現可能な通信パターンの理解、
使わない場合との比較を考えられるようになり、通信
設計のための知識が1段階あがったことでしょう。

家庭内ネットワークの通信パターンの変化

- ゲーム開発者からみた場合 -

家庭内ネットワークの通信パターン

※ あくまで一例です



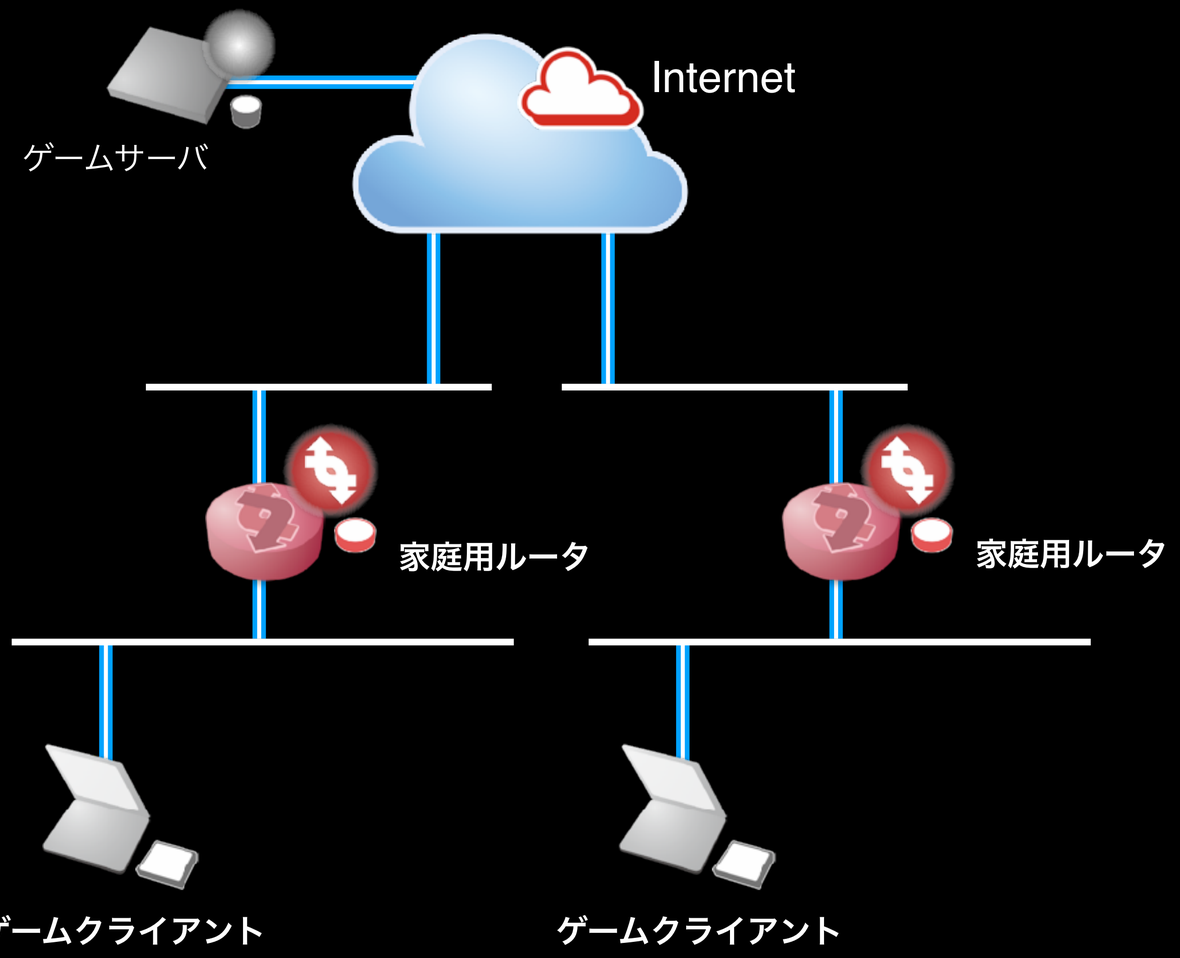
IPv4

IPv6

黎明期

家庭内ネットワークの通信パターン

※ あくまで一例です



IPv4

IPv6

P2P通信の普及

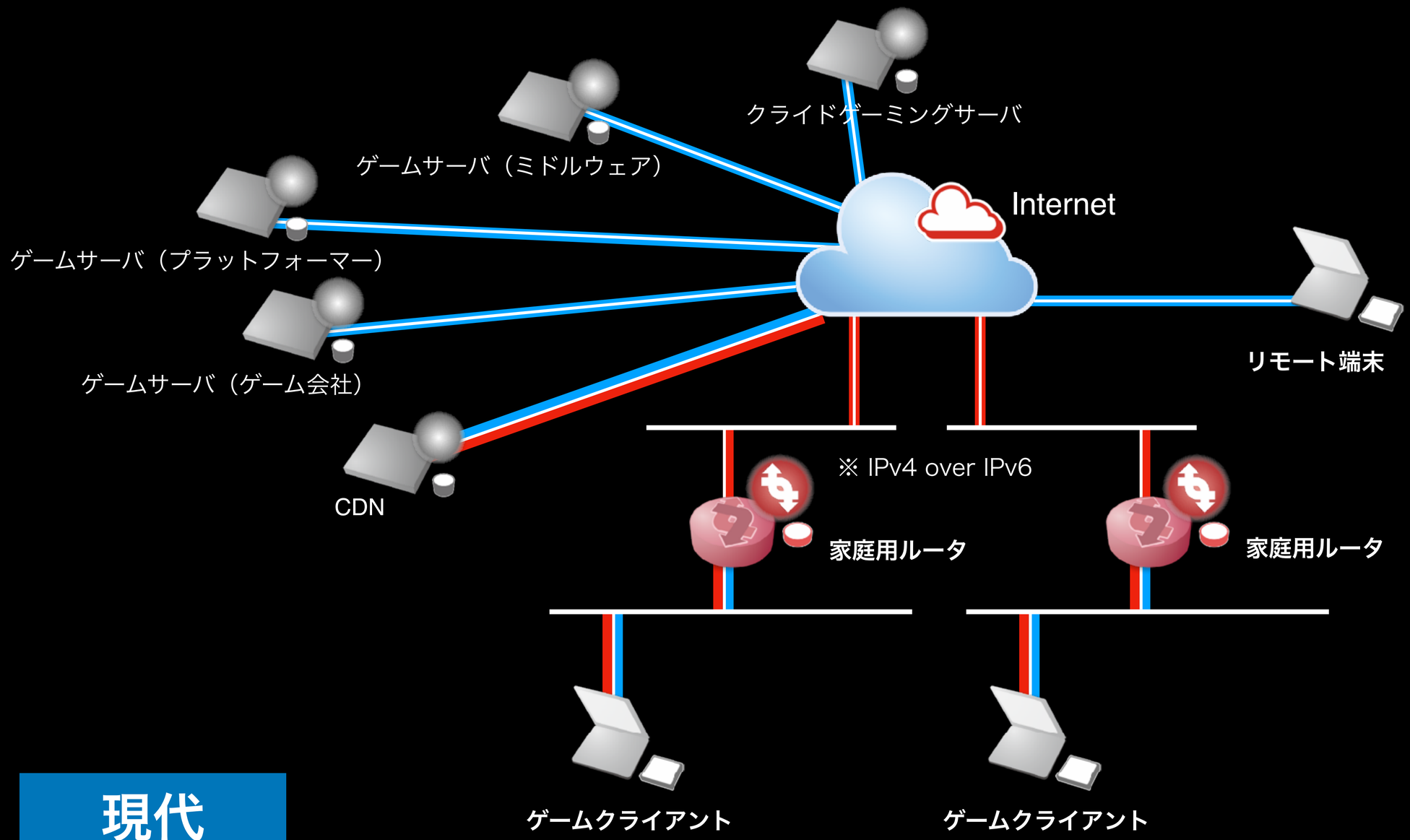
日が暮れるので

20年が経過した現代…

家庭内ネットワークの通信パターン

※ あくまで一例です

！？



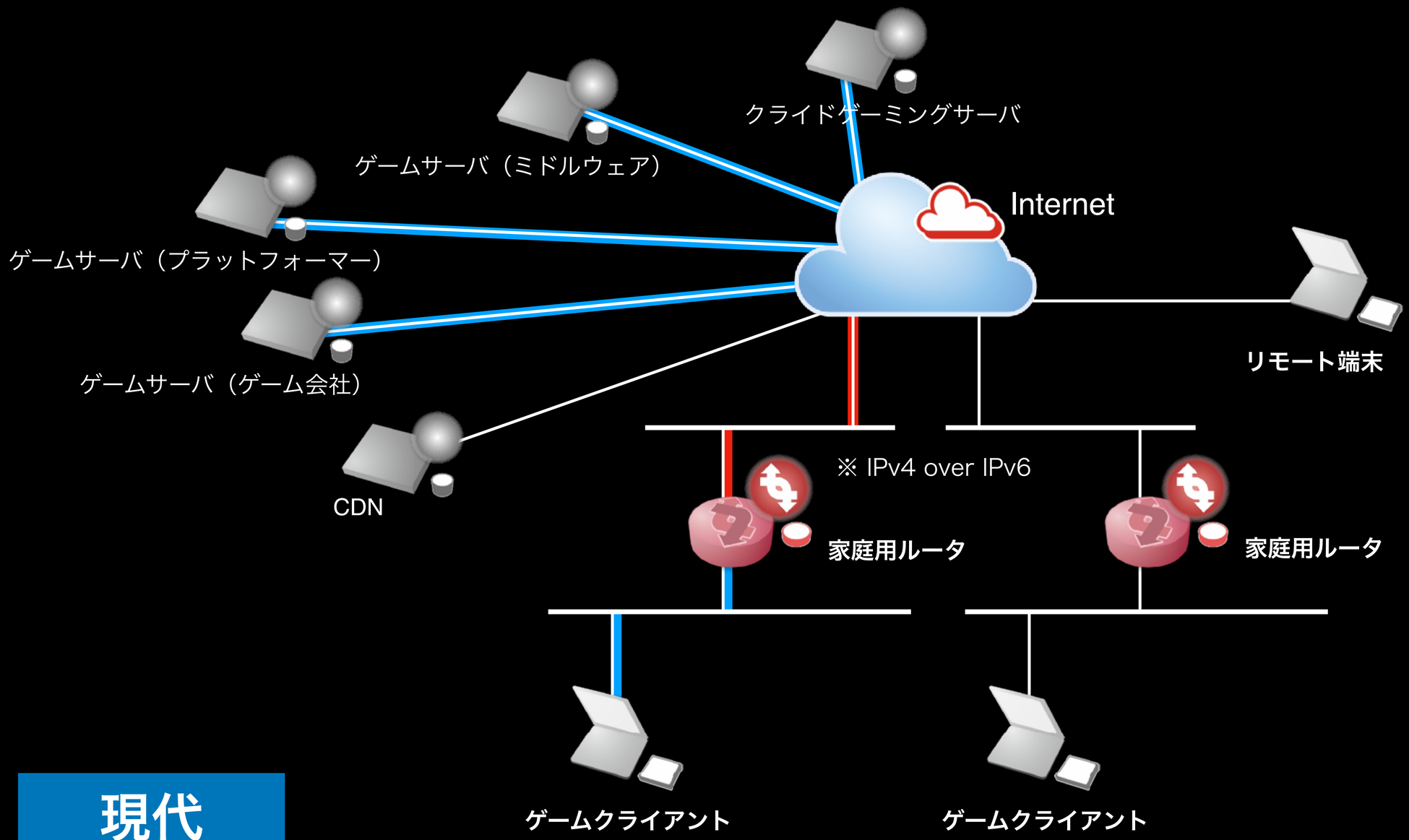
IPv4

IPv6

現代

家庭内ネットワークの通信パターン

※ あくまで一例です



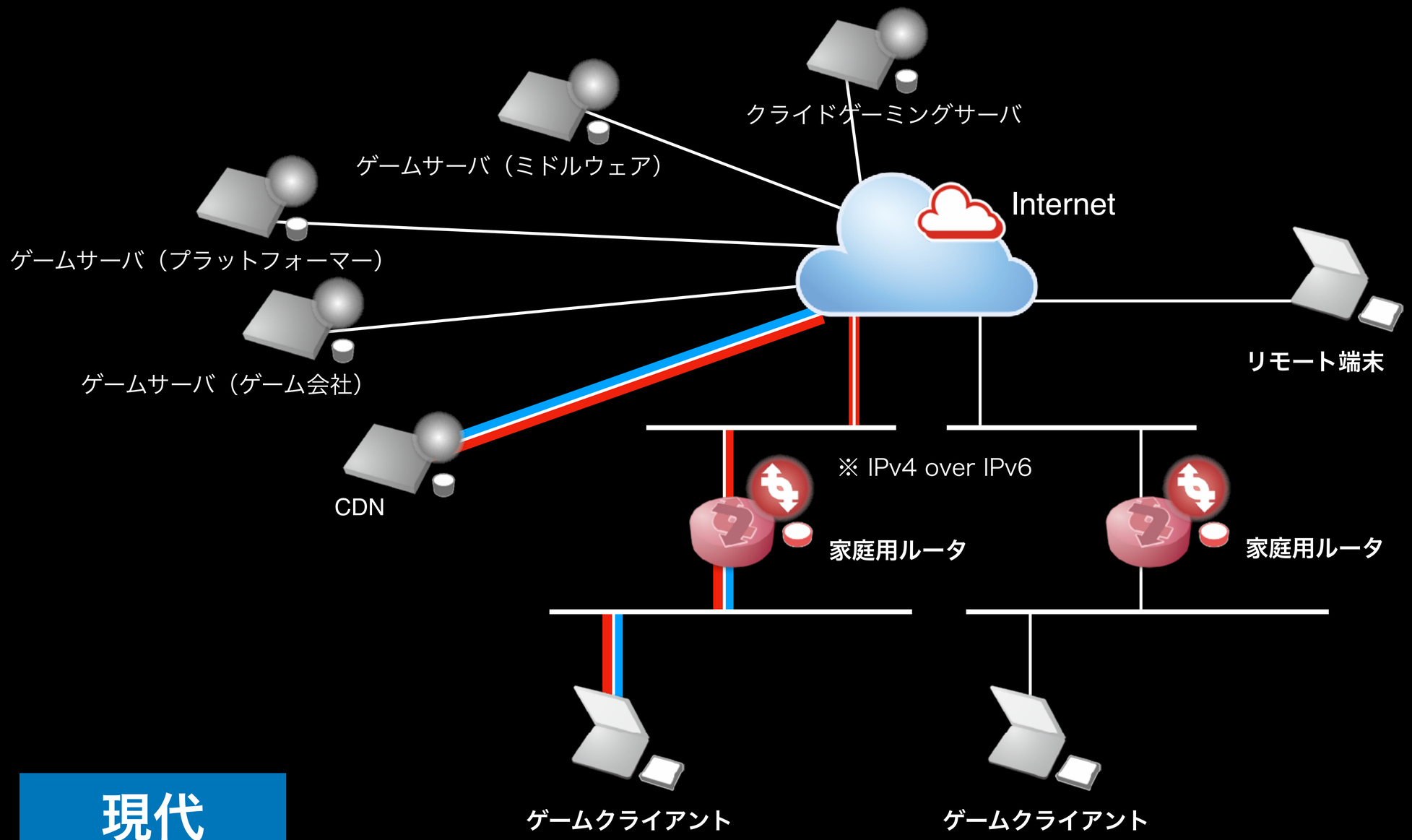
IPv4

IPv6

現代

家庭内ネットワークの通信パターン

※ あくまで一例です



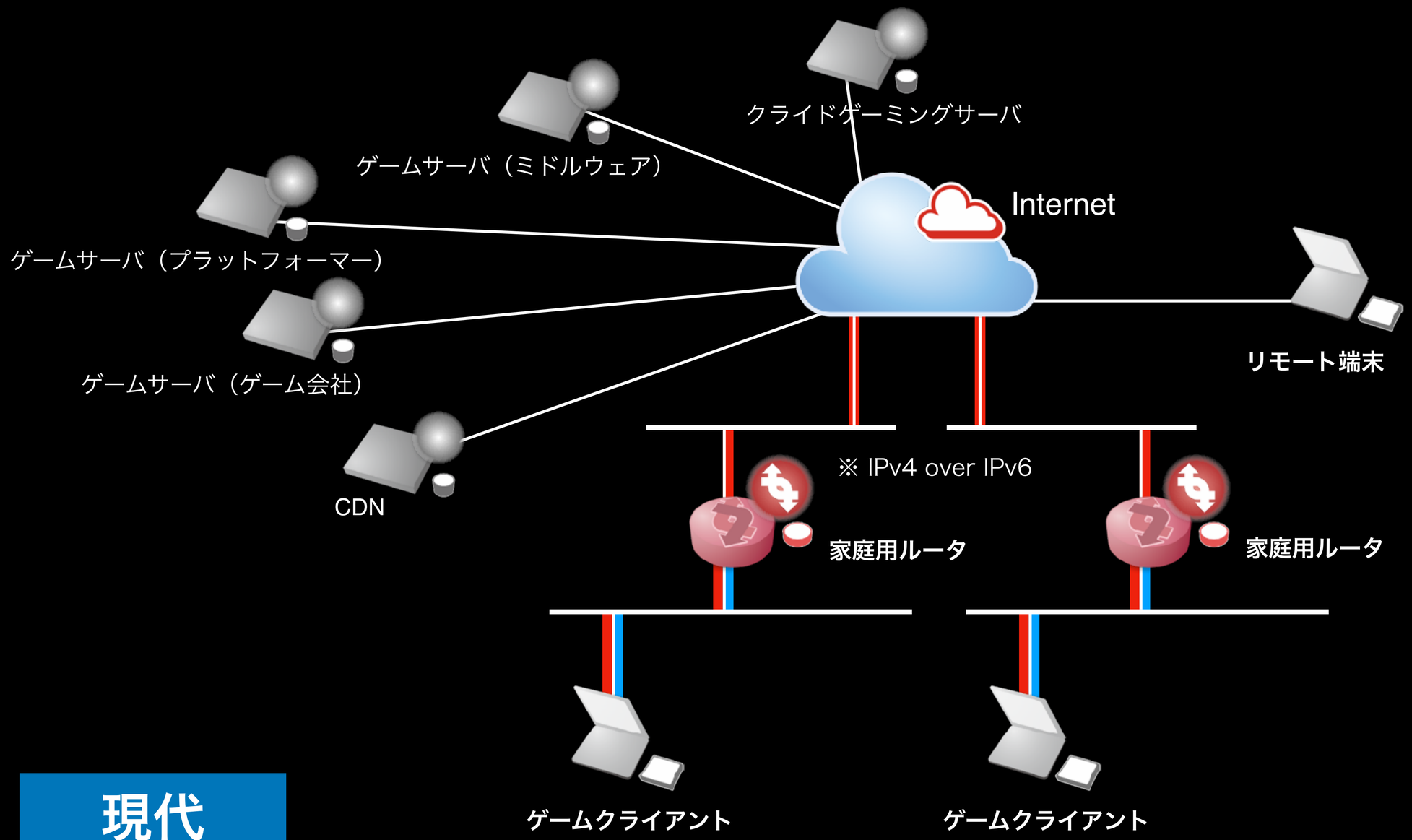
IPv4

IPv6

現代

家庭内ネットワークの通信パターン

※ あくまで一例です



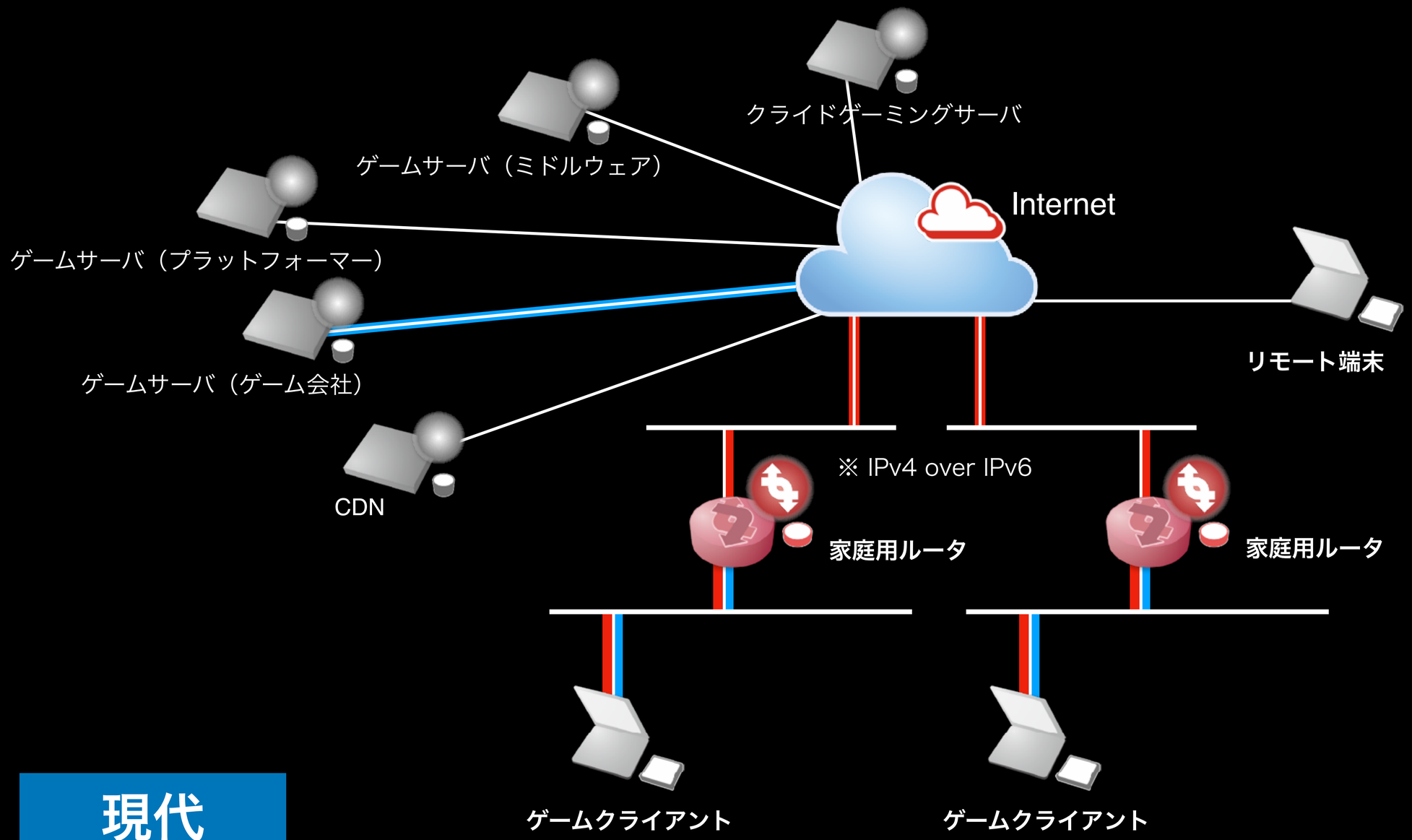
IPv4

IPv6

現代

家庭内ネットワークの通信パターン

※ あくまで一例です



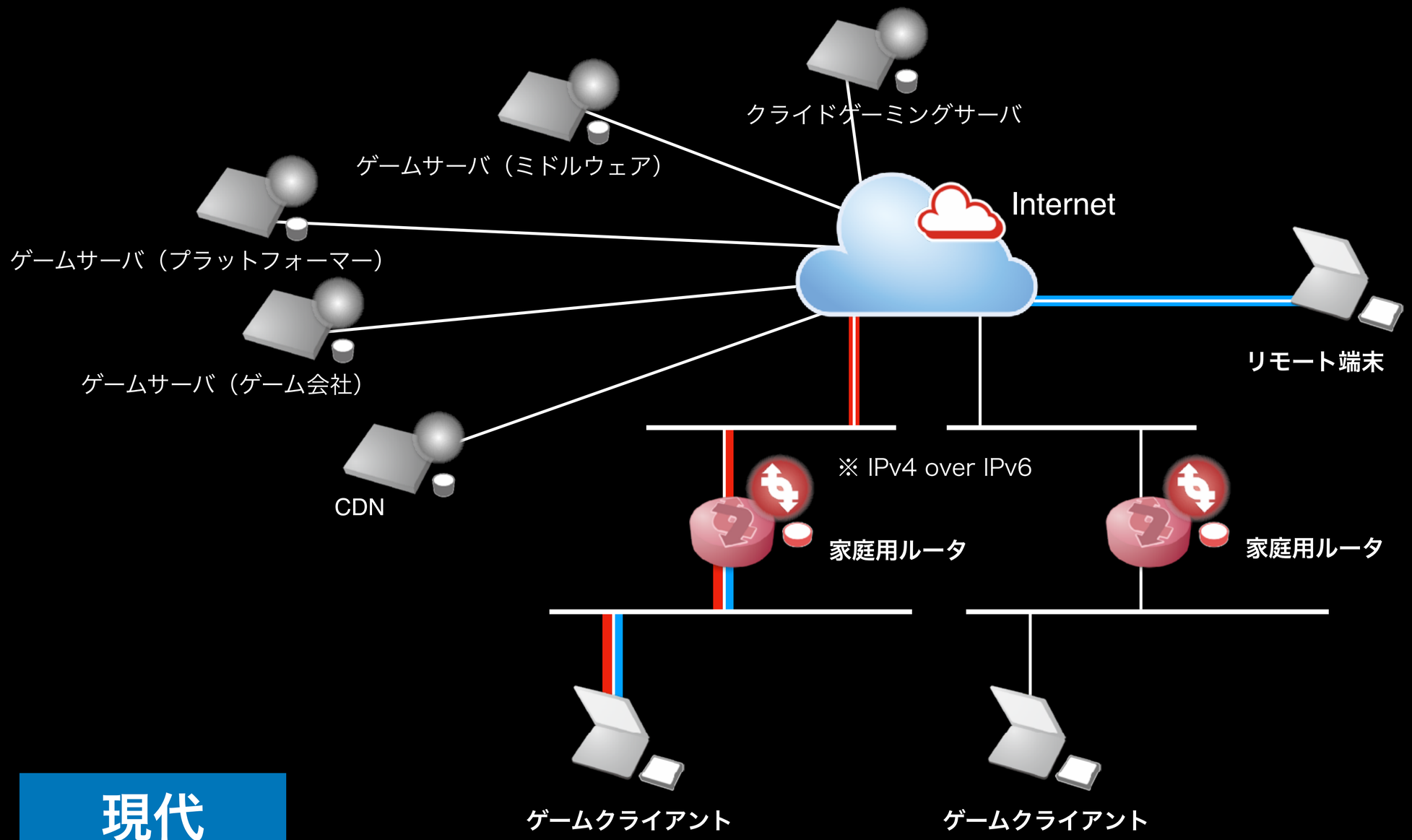
IPv4

IPv6

現代

家庭内ネットワークの通信パターン

※ あくまで一例です



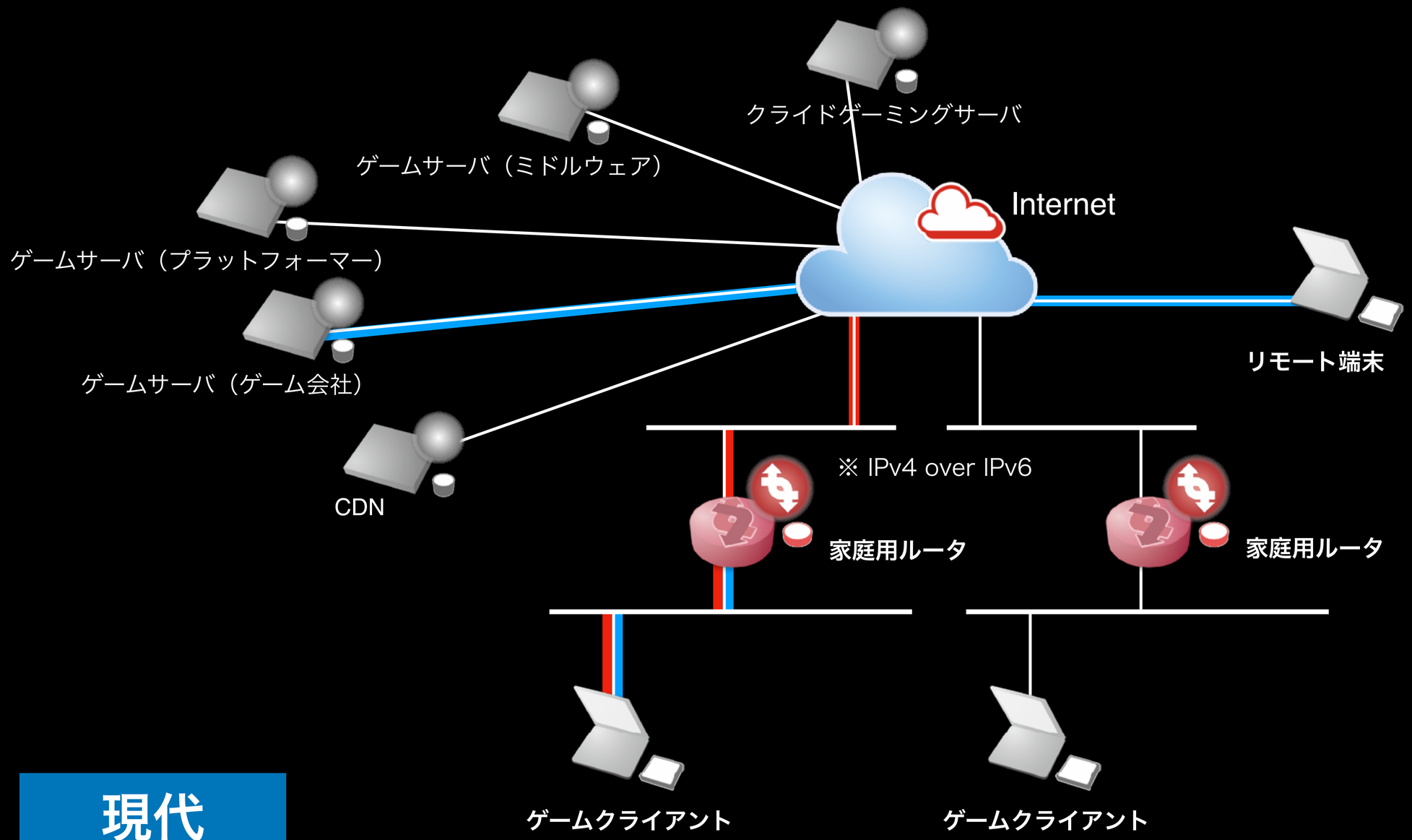
IPv4

IPv6

現代

家庭内ネットワークの通信パターン

※ あくまで一例です

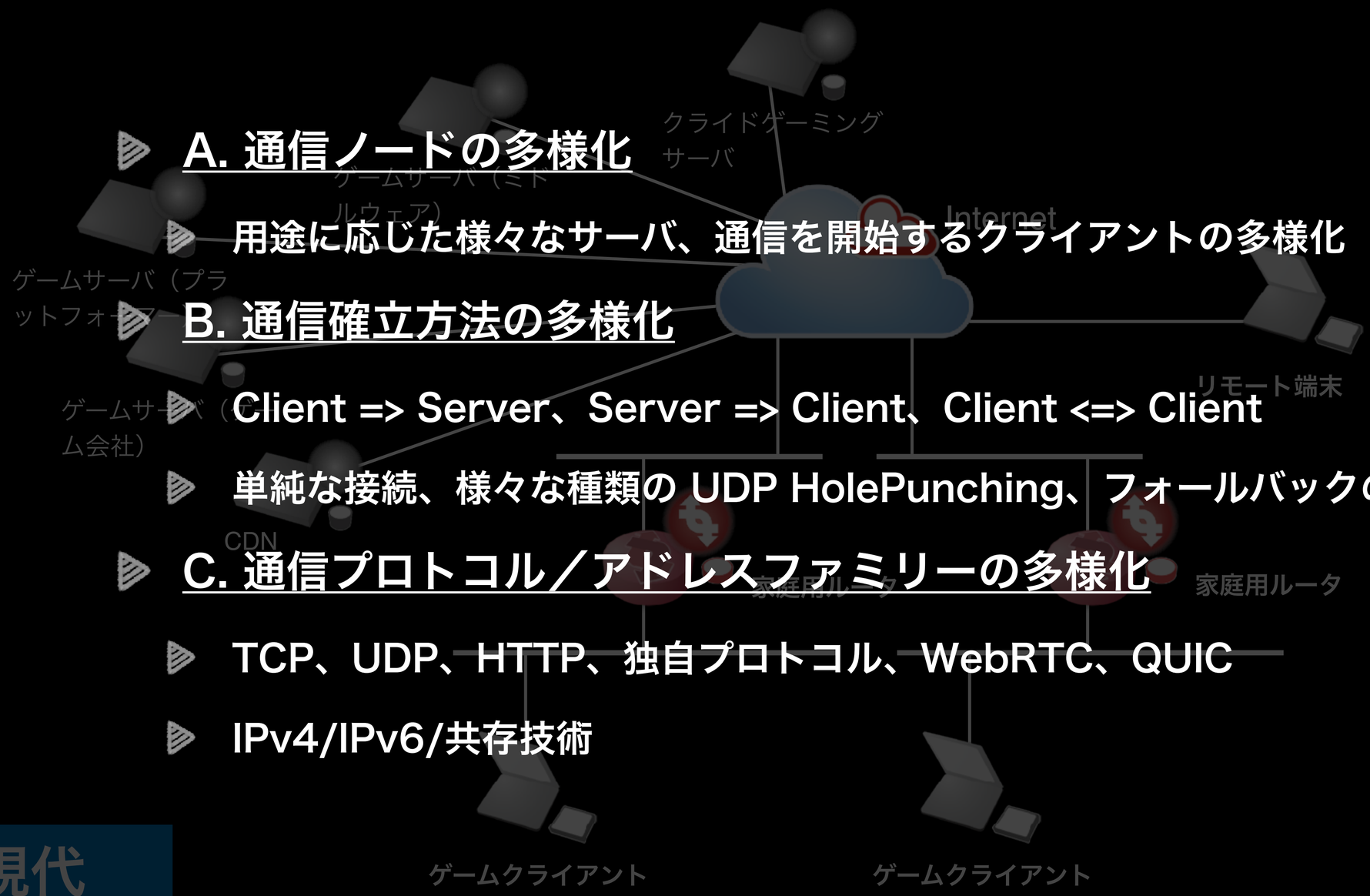


IPv4

IPv6

現代

家庭内ネットワークの通信パターン

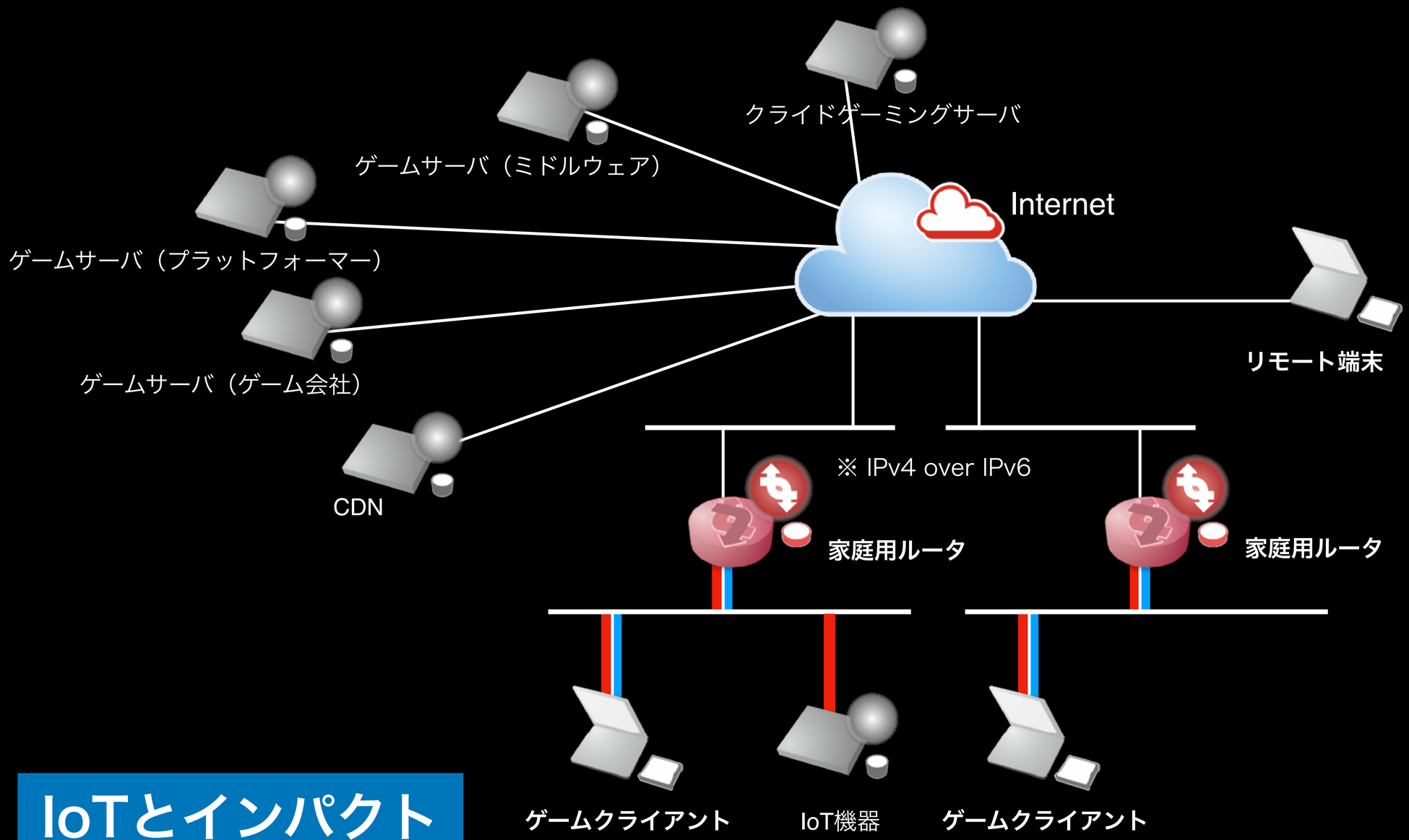


- **A. 通信ノードの多様化**
  - 用途に応じた様々なサーバ、通信を開始するクライアントの多様化
- **B. 通信確立方法の多様化**
  - Client => Server、Server => Client、Client <=> Client
  - 単純な接続、様々な種類の UDP HolePunching、フォールバックの多様化
- **C. 通信プロトコル／アドレスファミリーの多様化**
  - TCP、UDP、HTTP、独自プロトコル、WebRTC、QUIC
  - IPv4/IPv6/共存技術

現代

家庭内ネットワークの通信パターン（予想）

※ あくまで一例です



IPv4

IPv6

IoTとインパクト

このパートのまとめ

ゲームの通信の複雑さ、多様化の片鱗がなんとなくわかったと思います。

そして、IPv4が未だに根強く使われている、ゲームの動作する家庭内環境、ここに大きな変化が訪れようとしています。この変化とぶつかったとき、どうなってゆくのでしょうか？

ルータベンダーの視点から

- IoT Matter のインパクト -

議論パート

議論トピック

- A. 「IPv6版ポート解放」という言葉の意味・定義・活用方法・実装
  - 疑問点や別の視点からの提案
    - 活用パターン、実装方法、ユーザからの利用法の提案
  - UPnP以外でポート解放を実現するための技術／手法 => AI制御／MatterController連携
- B. 今後の家庭用ネットワークで行われる通信パターンの変化について
  - MatterやIPv6-Mostlyによる宅内のIPv6化とゲームプレイ環境の衝突によって何が起きてゆくのか => 共存できるようにお互い変わるのか、喧嘩し合うのか…
- C. ポート解放文化が廃れた場合に、ゲーム・IoT・ネットワーク業界が被る損害
  - サーバ経由／集約型のパターンしか考えられない環境になるとどうなるか => インフラコスト増、長期サービスの困難化、トラフィックの集中
- D. IoT Matterの仕様や活用法、家庭内の変化についてもっと！