

BRIC テクノロジー入門

BRIC テクノロジーについて分かりやすく説明します。

1) Circuit Switched Data Networks

ISDN、POTS コーデックでは図1のようなコンセプトに基づいており、これらのシステムは Circuit Switched Data Networks(CSD) がベースになっています。CSD ネットワークはベルトコンベアのように考えることができます。2地点間を連続して、正確に且つ同じスピードで運ばれていきます。

しかしながらベルトコンベアのスピードは非圧縮のデジタルオーディオを届けるのに十分ではありません。そこでエンコーダとデコーダが登場し、ベルトコンベアの両サイドに立ってベルトコンベアに合うスピードでデータを送り合うのです。

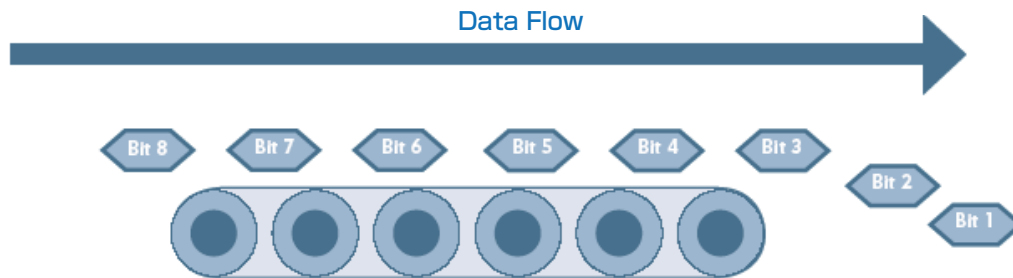


図1 CSD Network

図2の右例は片方向の一連の流れです。大抵のCSDネットワークは双方向ですので、このプロセスはそれぞれの方向で繰り返されます。エンコーダとデコーダは、デジタルオーディオを1ビットごと（言葉に例えると「一語一語」）に分けて処理しているわけではありません。エンコードする場合、エンコード処理前に入力オーディオの「窓」、つまりグループを作ります。この窓は「オーディオのスナップ写真」が入ったフレームに例えられます。

そしてアルゴリズムに依存して1/100秒から数百ミリ秒の時間をかけて処理をします。

エンコーダは、ネットワーク（ベルトコンベア）がデータの流を作っている間にフレームを貯めて一気に出力します。デコーダはフレームを貯めて一気に取り込みます。システムによって生み出されたオーディオコーディングの遅延は、上記のように説明できます。

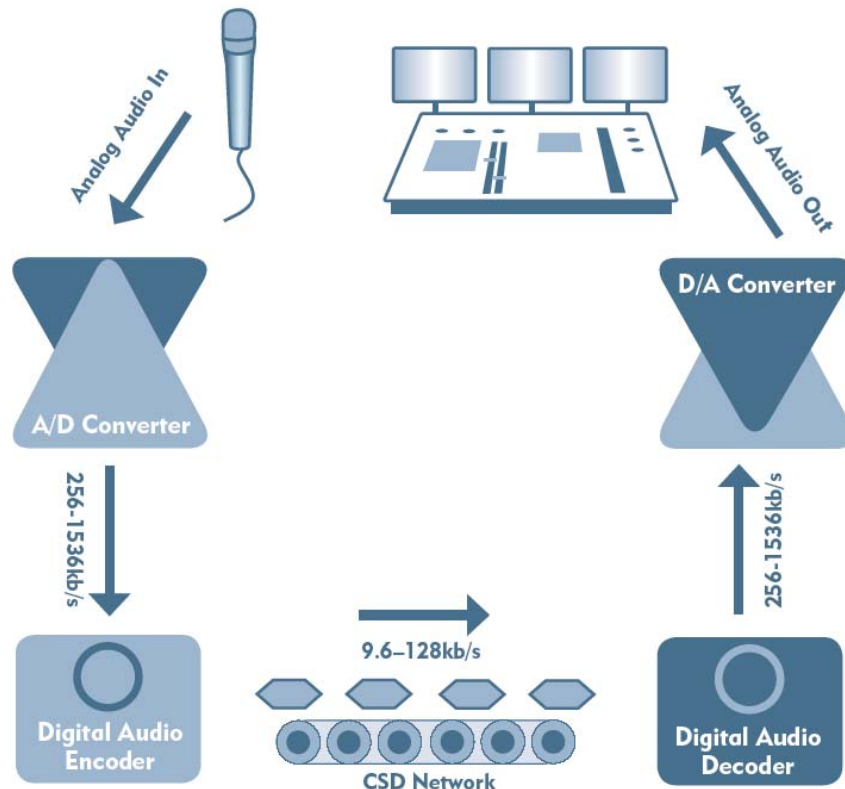


図2 Compressed Audio Over a CSD Network

2) IP Networks for Audio

インターネットのような IP ネットワークは、パケットによってデータのやり取りが行われています。図3に示されるように、2地点間を一転で接続するというよりは、IP ネットワークはルーターで知られる多数の「中継局」を経由して送られます。データは目的地を示すアドレス「IP ヘッダー」が付けられ、パケットにまとめられてネットワークに送られます。このヘッダーは、

封筒に書かれたアドレスとして役目を果たし、数々のルーターによって読み込まれて、多くのルートの中から一つのルートに向けて中継されていきます。それぞれのパケットは異なるルートで運ばれますが実際の接続は成立しません。決まった2地点間でのみ接続が成り立ちます。

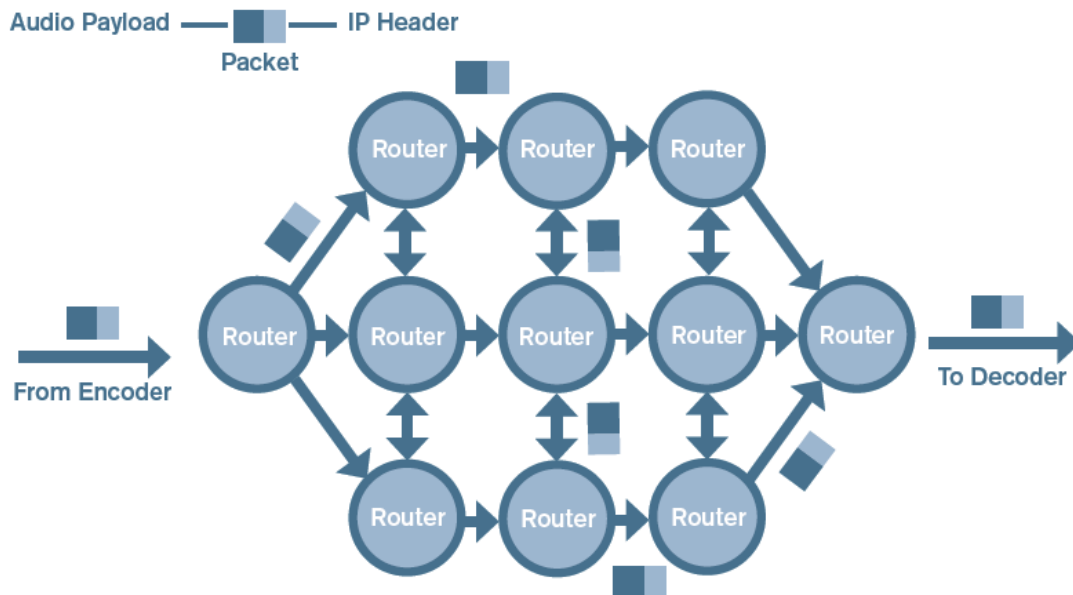


図3 IP Network

実際、パケットには3つのヘッダー「IP」「UDP」「RTP」が必要で、役に立つ情報から不必要なものまで様々な情報が含まれています。しかし厄介なインターネットは重要な情報が詰まったパケットを、不快な「パケット」の中に落としてしまいます。

IP ヘッダーの中のデータとアドレスは長さ 32 ビットの 2 進法で構成されています。そして直接インターネットに接続されると独自の数値を持つようになります。IP アドレスを表現するために、我々は通常これら 32 ビットを「8 ビットが 4 つ」というように分割して、それらの間にドットを打ち、0 から 255 の 10 進法で表現します。例えば、COMREX ACCESS のテスト用 IP アドレスは 70.22.155.133 です。

IP パケットには様々な長さがありますが、我々のエンコーダはオーディオデータのフレーム長さを固定して送ります。遅延を最小限にするために 1 パケットに対して 1 つのフレームに 1 つの音声で構成するように工夫しています。そしてエンコーダはオーディオフレームが出来上がったらすぐにネットワークに送り出せるようにしています。

3) Bandwidth in IP

CSD ネットワークではベルトコンベアのスピードは 2 地点間で正確なスピードで動いています。モデムを使った POTS コーデックであれば通常 24 ~ 56kb/s ですし、ISDN では 128kb/s となります。

IP ネットワークでは一定のスピードというものがなく、「最大値」と言う数値で表します。これは CSD ネットワークよりはるかに高く、通常 Mb/s の数値で表すことが出来ます。なぜオーディオデータを圧縮する必要があるのでしょうか。ネットワークを

自分でコントロールできるかどうか焦点を当ててみましょう。管理されたネットワーク (LAN や WAN 等) では、どんなトラフィック状況下でも Quality of Service(QoS) に優先順位を付けたりと、自由にコントロールできます。しかし公共のインターネットでは QoS は信用できません。すべてのトラフィックは平等に扱われるため、パケットを送ったら無事届いてくれることを願うしかないので。

図4は典型的なインターネット接続の様子を表しています。ユーザーAとユーザーBが互いに接続するところです。各々が太いインターネット回線を持っています。しかしこの回線は屋内にてLANで繋がっており、複数のユーザーと共有しています。プロバイダー(ISP)まで繋がると、集まったパイプを一つにまとめます。

インターネットに送られると膨大な数のユーザーとルーター、またルーター間のリンクを共有しなくてはならないのです。太い回線も当てにすることは難しいと言えます。

結果としてAB間の接続は不確実で、様々なパターンの中でも変化している状況下に置かれるのです。

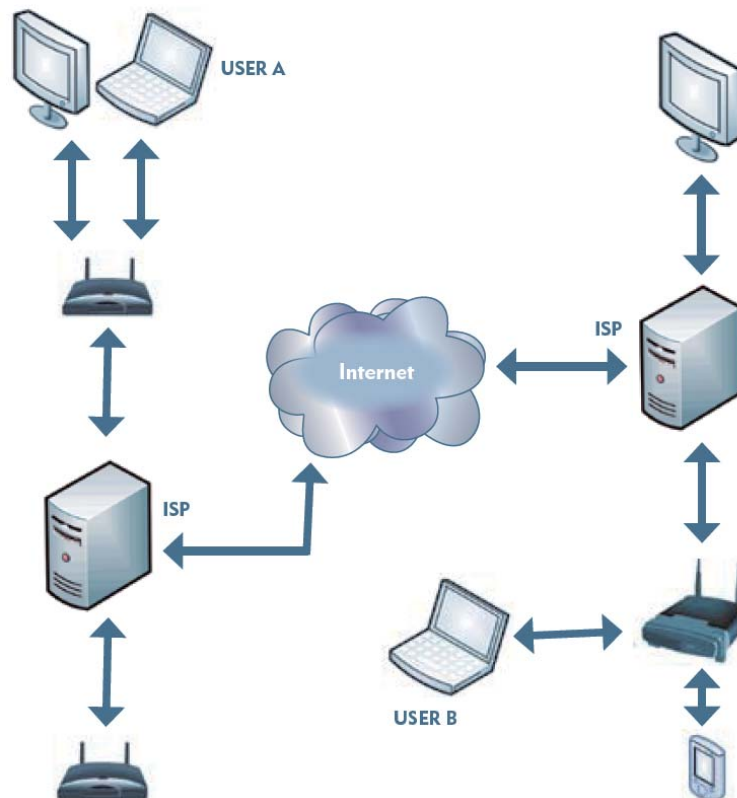


図4 Typical Internet Sharing

混雑したネットワークにIPオーディオを確実に通す、単純で有効な方法は多くを期待しないことです。圧縮を必要とする理由はここにあります。短期間メガビットの接続ができたとしても意味がありません。BRICによる圧縮がいかに有効で安定した通信を可能にしているかは、読み進めていただくとお分かりになるかと思います。

他にもIPネットワークに速く確実にパケットを送る方法がありますが、一つは何度もパケットを送ることです。太い回線を使っているならばForward Error Correction(FEC)、つまりエラー訂正という技術が安定した通信をもたらしてくれます。このFEC

は送るパケットにコピーのパケットを1つとそれ以前に送られたパケットのコピーも付けて送ることができます。これにより紛失してしまったパケットや遅れて届かないパケットが復元できるのです。

図5ではFECについて説明していますが、失われた情報が特別な解読アルゴリズムを通して復元できるように同じパケットが付けられているのが分かります。しかし注意したいのは、FECはその分データ量が多くなるわけですからより太い回線が必要になってきます。これは最良の方法とは言えません。

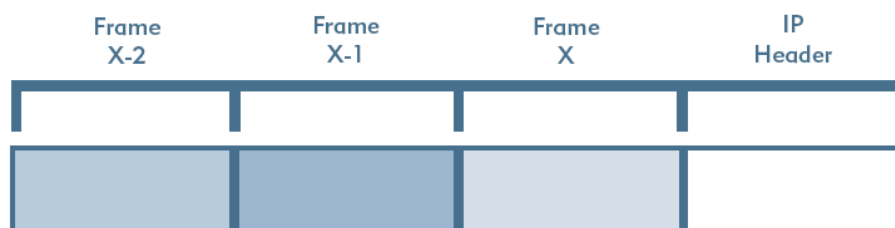


図5 FEC by Resending Packets

実際のところパケットが失われることはまれで、遅れて到着することがほとんどです。静的なネットワークであればパケットの遅れも少なく扱いても楽ですが、インターネット上では難しい話です。トラフィックやルートの違いでパケットが遅れたり早く到着したりします。忘れた頃に着いた、ということもあるかもしれません。インターネットでオーディオを送る時、一番早く到着したパケットと一番遅くに到着したパケット間の差をジッター (Jitter) と呼びます。このジッターを一定にする唯一の方法があります。

それは、デコーダに入ってくるパケットの貯めを作り、最初と最後のパケット間の差を極力なくすることです。これはもちろん流れ全体に遅れを加えることを意味しています。望んでいたことと異なってきてしまいます。

図6がジッターバッファを表しています。不定期にパケットが左から入って右に順調に流れていきます。バッファが長ければ長いほどデコーダはより連続的にデータを受け取れるのです。

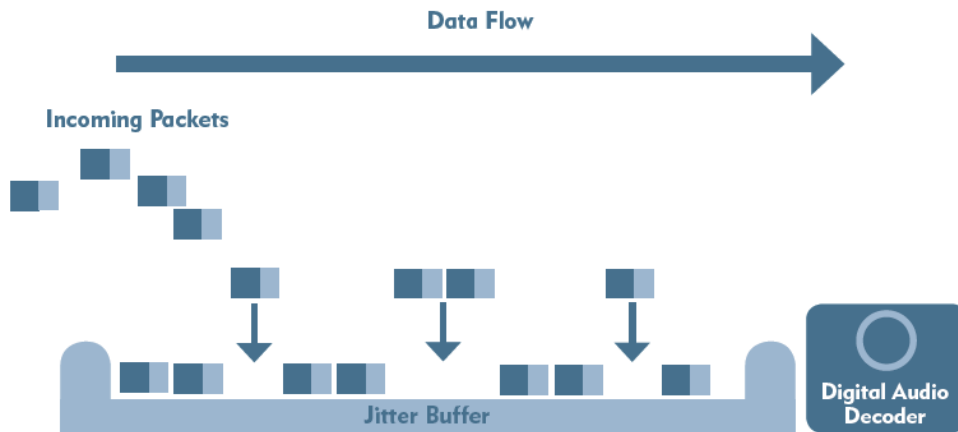


図6 Decoder Jitter Buffer



実は BRIC に隠されたマジックのひとつがこのバッファ処理なのです。バッファを小さくし遅延を最小限に抑えつつも、ジッターの変化に左右されない程度に最大限バッファをとるのが、この BRIC に隠されたテクニックです。BRIC コーディングアルゴリズムは圧縮に対して非常に復元力があるため、ジッターバッファを増やす前にかなりのパケットロスにも対応できるのです。

IP オーディオの基本をまとめたものが図7です。

- 1) アナログ / デジタル交換に続いてエンコーダに入ります。
- 2) パケットサイズが決められインターネット網に送られます。
- 3) パケットを受け取り、蓄積してデコードします。
- 4) デコードされたものをデジタル / アナログ変換します。

これらの流れはしっかり管理された LAN あるいは WAN のようなネットワークの場合です。実際の公共のインターネットはそんなりとはいきません。そこで IP 伝送のノウハウを凝縮した我々の製品が役に立つのです。BIRC のコンセプトを覗いてみてください。何が特別であるかを理解できます。

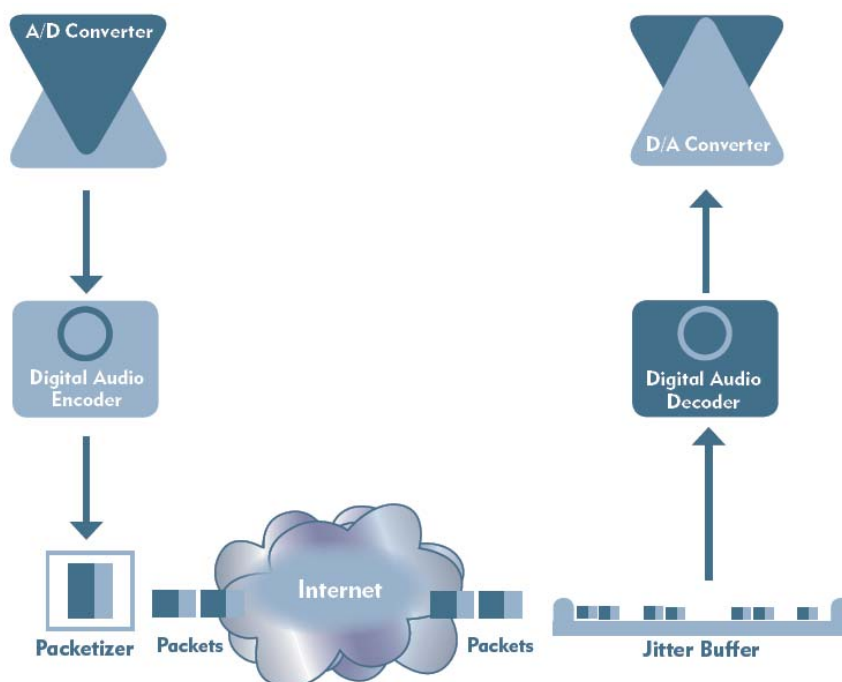


図7 Codec Building Blocks

BRIC (Broadcast Reliable Internet Codec)

1) Codec Algorithms

BRIC_ULB(Ultra Low Bitrate) Voice quality of BRIC_ULB BRIC_ULB の特長図 8 に示すように、BRIC_ULB のエンコーダがインターネットに送るパケットサイズを臨機応変に変えている点です。初めに入力オーディオの量によって全体的なアウト

プットフレームサイズを決めます。少ない場合には小さくし、多い場合には大きくします。次にエンコーダはデコーダからネットワークの混雑情報を受け取り、これをもとにパケットサイズを決定します。

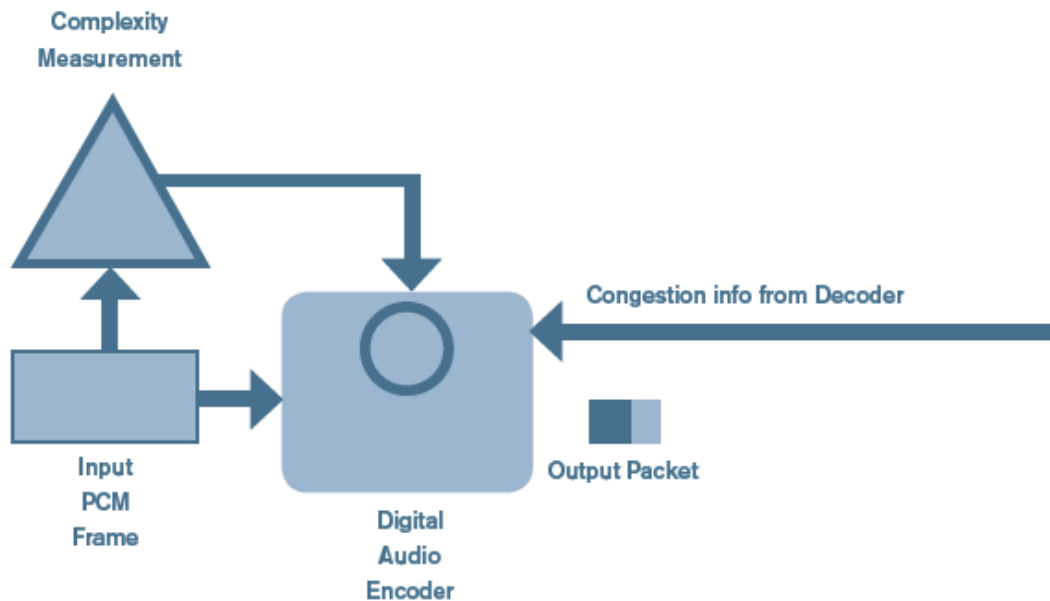


図 8 BRIC-ULB Variable Bit-Rate Decisions

2) Error Concealment/ Jitter management (Concealment : 隠すこと)

IP コーデックの技術の中で最も重要な決定事項の 1 つにジッターバッファをどれだけ確保するかがあります。この数値をどれほどにするかによってシステム全体の遅延と安定性に違いが出てきます。パケットロスと遅延の絶妙なバランス、いわゆるスイートスポットを探ることが非常に重要になってきます。BRIC アルゴリズムは多くのアルゴリズムと比べてもパケット損失に対して復元力があります。このパケットロスをうまく調整できる機能によっ

て、どんなにパケットロスが起きてもジッターバッファを調整して遅延を最小限に抑えることができます。たとえジッターバッファ訂正が必要なときでも、出力された音にはほとんど分からず調整することができます。仮に非常に大きな訂正が必要になったとしてもジッターバッファがすばやくそれに対応して拡張され、徐々に元の非常に低い値にコントロールされていきます。