Beheben von QoS-Problemen mit abgehackten Sprachfunktionen

Inhalt

Einführung

Damit Packet Voice ein realistischer Ersatz für standardmäßige Telefondienste im öffentlichen Telefonnetz (PSTN) sein kann, muss die empfangene Qualität von Packet Voice mit der von grundlegenden Telefondiensten vergleichbar sein. Dies bedeutet konsistent hochwertige Sprachübertragungen. Wie andere Echtzeitanwendungen verfügt Packet Voice über eine große Bandbreite und ist verzögerungsempfindlich. Damit Sprachübertragungen für den Empfänger verständlich (nicht abgehackt) sind, können Sprachpakete nicht verworfen, übermäßig verzögert oder mit variierender Verzögerung (auch als Jitter bezeichnet) verarbeitet werden. In diesem Dokument werden verschiedene QoS-Aspekte (Quality of Service) beschrieben, die bei der Behebung von unübersichtlichen Sprachproblemen helfen. Die Hauptgründe für abgehackte Sprachprobleme sind Verlust und Verspätung von Sprachpaketen.

Voraussetzungen

Anforderungen

Die Leser dieses Dokuments sollten über folgende Punkte Bescheid wissen:

- Grundkonfiguration von Packet Voice (VoIP, Voice over Frame Relay (VoFR) oder Voice over ATM (VoATM) entsprechend der Anforderungen).
- Grundlegendes Verständnis von Priorisierung von Sprachfunktionen, Fragmentierung, verschiedenen Codecs und ihren Bandbreitenanforderungen.

Verwendete Komponenten

Die Informationen in diesem Dokument gelten für alle Software- und Hardwareversionen von Cisco Sprach-Gateways.

Die in diesem Dokument enthaltenen Informationen wurden aus Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Sie in einem Live-Netzwerk arbeiten, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen, bevor Sie es verwenden.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie in den Cisco Technical Tips Conventions.

Ursachen der abgehackten Stimme

Die choppe Sprachqualität wird dadurch verursacht, dass Sprachpakete im Netzwerk variabel verzögert oder verloren gehen. Wenn ein Sprachpaket sein Ziel verzögert, geht dem Ziel-Gateway Echtzeitinformationen verloren. In diesem Fall muss das Ziel-Gateway vorhersagen, wie der Inhalt des verpassten Pakets sein kann. Die Prognose führt dazu, dass die empfangene Stimme nicht die gleichen Eigenschaften wie die übertragene Stimme hat. Dies führt zu einer empfangenen Stimme, die robotisch klingt. Wenn sich ein Sprachpaket über die Vorhersagefunktion eines empfangenden Gateways hinaus verzögert, bleibt die Echtzeit-Lücke leer. Da diese Lücke am Empfängerende nicht geschlossen werden kann, geht ein Teil der übertragenen Sprache verloren. Dies führt zu einer abgehackten Stimme. Viele der abgehackten Sprachprobleme werden gelöst, indem sichergestellt wird, dass die Sprachpakete nicht sehr verzögert (und nicht variabel verzögert) werden. Manchmal führt die Sprachaktivitätserkennung (Voice Activity Detection, VAD) zu Front-End-Clipping bei einer Sprachkommunikation. Dies ist eine weitere Ursache für abgehackte (oder abgeschnittene) Stimme.

In den verschiedenen Abschnitten dieses Dokuments wird veranschaulicht, wie die Instanz der abgehackten Stimme auf ein Minimum reduziert werden kann. Für die meisten dieser Maßnahmen muss sichergestellt werden, dass in Ihrem Sprachnetzwerk mindestens Jitter installiert ist.

Bandbreitenanforderung

Bevor Sie Maßnahmen zur Minimierung von Jitter in Erwägung ziehen, sollten Sie eine ausreichende Netzwerkbandbreite zur Unterstützung von Echtzeit-Sprachdatenverkehr bereitstellen. So klingt beispielsweise ein G.711-VoIP-Anruf mit 80 Kbit/s (Payload mit 64 Kbit/s und Header mit 16 Kbit/s) über eine 64-Kbit/s-Verbindung schlecht, da mindestens 16 Kbit/s der

Pakete (das sind 20 Prozent) verworfen werden. Die Bandbreitenanforderungen variieren je nach dem für die Komprimierung verwendeten Codec. Verschiedene Codecs haben unterschiedliche Payloads und Header-Anforderungen. Die Verwendung von VAD wirkt sich auch auf die Bandbreitenanforderung aus. Wenn die Header Compression (cRTP) mit Real Time Protocol (RTP) verwendet wird, kann die Bandbreitenanforderung weiter verringert werden.

Die Bandbreite, die für einen Sprachanruf mit dem G.729-Codec (Standard-Payload von 20 Byte) mit cRTP erforderlich ist, ist beispielsweise wie folgt:

- Voice-Payload + komprimiert (RTP-Header + User Datagram Protocol (UDP)-Header + IP-Header) +Layer-2-Header

Dies entspricht:

- 20 Bytes + komprimiert (12 Byte + 8 Byte + 20 Byte) + 4 Byte

Dies entspricht:

- 28 Byte, da die Header-Komprimierung den IP-RTP-Header auf maximal 4 Byte reduziert. Dies ergibt 11,2 Kbit/s bei einer Codec-Geschwindigkeit von 8 Kbit/s (50 Pakete pro Sekunde).

Weitere Informationen finden Sie unter Bandbreitennutzung pro Anruf unter Voice-over-IP.

Priorität des Sprachdatenverkehrs

Bei der Priorisierung von Sprachfunktionen gibt es zwei wichtige Komponenten. Die erste besteht in der Klassifizierung und Kennzeichnung von interessantem Sprachdatenverkehr. Die zweite Möglichkeit besteht in der Priorisierung des markierten interessanten Sprachdatenverkehrs. In den beiden Unterabschnitten werden verschiedene Ansätze zur Klassifizierung, Kennzeichnung und Priorisierung von Sprachdaten erläutert.

Klassifizierung und Kennzeichnung

Um die Bandbreite für VoIP-Pakete zu garantieren, muss ein Netzwerkgerät in der Lage sein, die Pakete im gesamten IP-Datenverkehr zu identifizieren, der durch das Gerät fließt. Netzwerkgeräte verwenden die Quell- und Ziel-IP-Adresse im IP-Header oder die Quell- und Ziel-UDP-Portnummern im UDP-Header, um VoIP-Pakete zu identifizieren. Dieser Identifizierungs- und Gruppierungsprozess wird als Klassifizierung bezeichnet. Sie bildet die Grundlage für die Bereitstellung von QoS.

Die Paketklassifizierung kann prozessorintensiv sein. Daher muss die Klassifizierung so weit wie möglich am Netzwerk-Edge erfolgen. Da jeder Hop immer noch die Behandlung bestimmen muss, die ein Paket erhalten soll, muss eine einfachere, effizientere Klassifizierungsmethode im Netzwerkkern vorhanden sein. Diese einfachere Klassifizierung wird durch Markieren oder Festlegen des ToS-Bytes (Type of Service) im IP-Header erreicht. Die drei signifikantesten Bits des ToS-Bytes werden als IP Precedence-Bits bezeichnet. Die meisten Anwendungen und Anbieter unterstützen derzeit die Einstellung und Erkennung dieser drei Bits. Die Markierung entwickelt sich, sodass die sechs wichtigsten Bits des ToS-Bytes, der so genannte Differentiated Services Code Point (DSCP), verwendet werden können. Siehe Request for Comments (RFC).

Differentiated Services (DiffServ) ist ein neues Modell, bei dem Datenverkehr von zwischengeschalteten Systemen mit relativen Prioritäten behandelt wird, die auf dem ToS-Feld basieren. Der in RFC 2474 und RFC 2475 definierte DiffServ-Standard ersetzt die ursprüngliche

Spezifikation für die Definition der Paketpriorität, die in RFC 791 beschrieben wird. DiffServ erhöht die Anzahl der definierbaren Prioritätsstufen, indem Bits eines IP-Pakets für die Prioritätsmarkierung neu zugewiesen werden. Die DiffServ-Architektur definiert das DiffServ-Feld. Sie ersetzt das ToS-Byte in IP V4, um Entscheidungen über Paketklassifizierung und Datenverkehrsaufbereitung wie Messing, Marking, Shaping und Richtlinienvergabe im Per-Hop Behavior (PHB) zu treffen. Zusätzlich zu den zuvor genannten RFCs definiert der RFC 2597 die Assured Forwarding (AF)-Klassen. Dies ist eine Aufschlüsselung der DSCP-Felder. Weitere Informationen zu DSCP finden Sie unter Implementieren von Quality of Service-Richtlinien mit DSCP.

ToS Byte - P2 P1 P0 T3 T1 T0 CU

IP-Rangfolge: drei Bit (P2-P0), ToS: vier Bit (T3-T0), CPU: ein Bit

DiffServ-Feld - DS5 DS4 DS3 DS2 DS1 DS0 ECN ECN

DSCP: 6 Bit (DS5-DS0), ECN: zwei Bit

XXX0000 Bit 0, 1, 2 (DS5, DS4, DS3) sind Prioritätsbits, wobei:

- 111 = Netzwerksteuerung = Rangfolge 7
- 110 = Internetwork Control = Rangfolge 6
- 101 = CRITIC/ECP = Rangfolge 5
- 100 = Flash Override = Rangfolge 4
- 011 = Flash = Rangfolge 3
- 010 = Sofort = Rangfolge 2
- 001 = Priorität = Rangfolge 1
- 000 = Routine = Rangfolge 0

000XXX00 Bit 3, 4, 5 (DS2, DS1, DS0) sind Bit für Verzögerung, Durchsatz und Zuverlässigkeit.

- Bit 3 = Verzögerung [D] (0 = Normal; 1 = Niedrig)
- Bit 4 = Durchsatz [T] (0 = Normal; 1 = Hoch)
- Bit 5 = Zuverlässigkeit [R] (0 = Normal; 1 = Hoch)

000000XX Bits 6, 7: ECN

In diesen beiden Abschnitten werden zwei Arten der Klassifizierung und Kennzeichnung erläutert.

Voice Dial Peers zur Klassifizierung und Markierung von Paketen

Bei Cisco VoIP-Gateways werden die VoIP-Pakete in der Regel mithilfe von Voice-Dial-Peers klassifiziert und die IP-Prioritätsbits markiert. Diese Konfiguration zeigt, wie die IP-Prioritätsbits markiert werden:

```
dial-peer voice 100 voip
destination-pattern 100
session target ipv4:10.10.10.2
ip precedence 5
```

Im obigen Beispiel sind für alle VoIP-Anrufe, die mit dem Befehl **dial-peer voice 100 voip** übereinstimmen, alle Sprach-Payload-Pakete mit IP Precedence 5 festgelegt. Das bedeutet, dass die drei signifikantesten Bits des IP ToS-Bytes auf 101 festgelegt sind.

```
dial-peer voice 100 voip
destination-pattern 100
session target ipv4:10.10.10.2
ip qos dscp ef media
ip qos dscp af31 signaling
```

Im obigen Beispiel sind bei jedem VoIP-Anruf, der dem Befehl **dial-peer voice 100 voip** entspricht, alle seine Media Payload Packets (Sprachpakete) mit dem Bitmuster Expedited Forwarding (EF) 101110 festgelegt. Alle Signalisierungspakete sind mit dem AF-Bitmuster 011010 eingestellt.

Hinweis: Der Befehl **ip qos dscp** wird seit Version 12.2(2)T der Cisco IOS®-Software unterstützt. Die IP Precedence ist in der Cisco IOS Software, Version 12.2T, nicht mehr verfügbar. Dasselbe wird jedoch durch den Befehl **ip qos dscp** erreicht. IP-Rangfolge 5 (101) entspricht IP DSCP 101000. Weitere Informationen finden Sie unter Klassifizieren von VoIP-Signalisierung und Medien mit DSCP für QoS.

Modulare QoS-CLI zur Klassifizierung und Markierung von Paketen

Die empfohlene Klassifizierungs- und Markierungsmethode ist die modulare QoS-CLI. Dies ist eine vorlagenbasierte Konfigurationsmethode, die die Klassifizierung von der Richtlinie trennt. Dadurch können mehrere QoS-Funktionen für mehrere Klassen gemeinsam konfiguriert werden. Verwenden Sie einen Befehl **class-map**, um Datenverkehr anhand verschiedener Anpassungskriterien zu klassifizieren, und einen Befehl **policy-map**, um zu bestimmen, was für die einzelnen Klassen geschehen muss. Wenden Sie die Richtlinie auf ein- oder ausgehenden Datenverkehr an einer Schnittstelle an, indem Sie den Befehl **service-policy** eingeben. In diesem Konfigurationsbeispiel wird die Verwendung der modularen QoS-CLI zum Klassifizieren und Markieren von Paketen veranschaulicht:

```
access-list 100 permit udp any any range 16384 32767
access-list 101 permit tcp any any eq 1720
!
class-map match-all voip
match access-group 100
class-map match-all control
match access-group 101
!
policy-map mqc
class voip
set ip precedence 5
class control
set ip precedence 5
class class-default
set ip precedence 0
!
interface Ethernet0/0
service-policy input mqc
```

In diesem Konfigurationsbeispiel wird jeder Datenverkehr, der mit der Zugriffssteuerungsliste (ACL) 100 übereinstimmt, als "class voip" klassifiziert und mit der IP-Rangfolge 5 festgelegt. Das bedeutet, dass die drei signifikantesten Bits des IP ToS-Bytes auf 101 festgelegt sind. ACL 100 entspricht den allgemeinen UDP-Ports, die von VoIP verwendet werden. Ebenso stimmt die ACL 101 mit dem H.323-Signalisierungsverkehr überein (Transmission Control Protocol (TCP)-Port 1720). Der gesamte andere Datenverkehr wird mit IP Precedence 0 (IP-Rangfolge 0) festgelegt. Die Richtlinie wird als "mqc" bezeichnet. Sie wird auf eingehenden Datenverkehr über Ethernet 0/0 angewendet.

Priorisierung

Nachdem jeder Hop im Netzwerk in der Lage ist, die VoIP-Pakete zu klassifizieren und zu identifizieren (entweder über Port-/Adressinformationen oder über das ToS-Byte), stellen diese Hops jedem VoIP-Paket die erforderliche QoS zur Verfügung. Konfigurieren Sie an diesem Punkt spezielle Techniken, um Prioritätswarteschlangen bereitzustellen, um sicherzustellen, dass große Datenpakete die Sprachdatenübertragung nicht stören. Dies ist in der Regel bei langsameren WAN-Verbindungen erforderlich, bei denen eine hohe Überlastungswahrscheinlichkeit besteht. Sobald der gesamte interessante Datenverkehr auf Basis der QoS-Anforderungen in QoS-Klassen eingeordnet ist, stellen Sie Bandbreitengarantien und priorisierte Services über einen intelligenten Ausgabewarteschlangenmechanismus bereit. Für VoIP ist eine Prioritätswarteschlange erforderlich.

Hinweis: Verwenden Sie jeden Warteschlangenmechanismus, der VoIP eine hohe Priorität einräumt. Low Latency Queuing (LLQ) wird jedoch empfohlen, da es flexibel und einfach zu konfigurieren ist.

LLQ verwendet die modulare QoS CLI-Konfigurationsmethode, um bestimmten Klassen Priorität einzuräumen und eine garantierte Mindestbandbreite für andere Klassen bereitzustellen. In Zeiten von Überlastungen wird die Prioritätswarteschlange mit der konfigurierten Geschwindigkeit geregelt, sodass der Prioritätsverkehr nicht die gesamte verfügbare Bandbreite beansprucht. (Wenn der Prioritätsverkehr die Bandbreite monopolisiert, verhindert er, dass Bandbreitengarantien für andere Klassen erfüllt werden.) Wenn Sie LLQ korrekt bereitstellen, sollte der Datenverkehr, der in die Prioritätswarteschlange geleitet wird, niemals die konfigurierte Rate überschreiten.

Mit LLQ können auch Warteschlangentiefen festgelegt werden, um zu bestimmen, wann der Router Pakete verwerfen muss, wenn in einer bestimmten Klassenwarteschlange zu viele Pakete warten. Es gibt auch einen **Klassenstandardbefehl**, mit dem die Behandlung des gesamten Datenverkehrs bestimmt wird, der nicht durch eine konfigurierte Klasse klassifiziert wurde. Der Klassenstandard wird mit einem **fair-queue-**Befehl konfiguriert. Das bedeutet, dass jeder nicht klassifizierte Datenfluss einen ungefähr gleich großen Anteil an der verbleibenden Bandbreite erhält.

In diesem Beispiel wird die Konfiguration von LLQ veranschaulicht. Weitere Informationen finden Sie unter Low Latency Queuing:

```
access-list 100 permit udp any any range 16384 32000
access-list 101 permit tcp any any eq 1720
access-list 102 permit tcp any any eq 80
access-list 103 permit tcp any any eq 23
!
class-map match-all voip
match access-group 100
class-map match-all voip-control
natch access-group 101
class-map match-all data1
match access-group 102
class-map match-all data2
match access-group 103
!
policy-map llq
class voip
priority 32
class voip-control
```

```
bandwidth 8
class data1
bandwidth 64
class data2
bandwidth 32
class class-default
fair-queue
!
interface Serial1/0
bandwidth 256
service-policy output llq
```

In diesem Beispiel wird jeder Datenverkehr, der mit ACL 100 übereinstimmt, als "Class voip" klassifiziert (d. h. Sprachdatenverkehr). Er erhält eine hohe Priorität von bis zu 32 Kbit/s. ACL 100 entspricht den allgemeinen UDP-Ports, die von VoIP verwendet werden. Die Zugriffsliste 101 entspricht dem H.323-Signalisierungsverkehr (TCP-Port 1720). Class data1 vergleicht den Web-Datenverkehr (TCP-Port 80, siehe Zugriffsliste 102) und garantiert 64 Kbit/s. Die Klasse data2 vergleicht den Telnet-Datenverkehr (TCP-Port 23, wie in ACL 103 gezeigt) und garantiert 32 Kbit/s. Die Standardklasse wird so konfiguriert, dass ein gleich hoher Anteil der verbleibenden Bandbreite für nicht klassifizierte Datenflüsse bereitgestellt wird. Die Richtlinie wird als "llq" bezeichnet. Sie wird auf ausgehenden Datenverkehr auf Serial1/0 angewendet, der eine Gesamtbandbreite von 256 Kbit/s aufweist.

Hinweis: Standardmäßig muss die garantierte Gesamtbandbreite und Prioritätsbandbreite für alle Klassen weniger als 75 Prozent der Schnittstellenbandbreite betragen. Ändern Sie diesen Prozentsatz, indem Sie den Befehl **für die maximale reservierte Bandbreite** eingeben.

In dieser Tabelle werden verschiedene Software-Warteschlangenmechanismen mit ihren jeweiligen Vorteilen und Einschränkungen verglichen.

Software Queuing-Mechanismus	Beschreibung	Vorteile	Einschränkungen
First-In-First-Out (FIFO)	Pakete kommen an und verlassen die Warteschlange in der gleichen Reihenfolge.	Einfache Konfiguration und schneller Betrieb.	Keine Services mit Priorität oder Bandbreitengarantien möglich.[1]
Weighted Fair Queuing (WFQ)	Ein Hash-Algorithmus, der in separate Warteschlangen fließt, in denen Gewichte verwendet werden, um zu bestimmen, wie viele Pakete gleichzeitig verarbeitet werden. Sie definieren	Einfache Konfiguration Standardeinstellung für Verbindungen mit weniger als 2 Mbit/s.	Keine Services mit Priorität oder Bandbreitengarantien möglich.[1]

	Gewichtungen, indem Sie IP Precedence- und DSCP-Werte festlegen.		
Custom Queuing (CQ)	Der Datenverkehr wird in mehrere Warteschlangen mit konfigurierbaren Warteschlangen beschränkungen klassifiziert. Die Warteschlangen beschränkungen werden basierend auf der durchschnittlichen Paketgröße, der maximalen Übertragungseinheit (Maximum Transmission Unit, MTU) und dem Prozentsatz der zuzuweisenden Bandbreite berechnet. Warteschlangen beschränkungen (in Byteanzahl) werden für jede Warteschlange aus der Warteschlange entfernt. Daher wird die zugewiesene Bandbreite statistisch bereitgestellt.	Bereits seit einigen Jahren verfügbar. Sie ermöglicht eine ungefähre Bandbreite nzuweisun g für verschiede ne Warteschla ngen.	Es ist keine Prioritätswa rtung möglich. Bandbreiten garantien sind ungefähre Werte. Es gibt eine begrenzte Anzahl von Warteschla ngen. Konfiguratio n ist relativ schwierig.[1]
Priority Queuing (PQ)	Der Datenverkehr wird in Warteschlangen mit hoher, mittlerer, normaler und niedriger	Bereits seit einigen Jahren verfügbar. Es bietet Services mit Priorität.	Datenverke hr mit höherer Priorität beanspruch t Warteschla ngen mit

	Priorität klassifiziert. Der Datenverkehr mit hoher Priorität wird zuerst verarbeitet, gefolgt von Datenverkehr mit mittlerer, normaler und niedriger Priorität.		geringerer Priorität an Bandbreite. Bandbreitengarantien sind nicht möglich.[2]
Class-Based Weighted Fair Queuing (CBWFQ)	Zur Klassifizierung des Datenverkehrs wird eine modulare QoS-CLI verwendet. Klassifizierter Datenverkehr wird in reservierte Bandbreitenwarteschlangen oder eine nicht reservierte Standardwarteschlange gestellt. Ein Scheduler steuert die Warteschlangen auf Basis von Gewichtungen, sodass die Bandbreitengarantien eingehalten werden.	Ähnlich wie bei LLQ, mit der Ausnahme, dass keine Prioritätswarteschlange vorhanden ist. Einfache Konfiguration und Möglichkeit zur Bereitstellung von Bandbreitengarantien	Keine Prioritätswartung möglich.[3]
Weighted Fair Queuing (PQ-WFQ) für Prioritätswarteschlangen, auch als IP RTP Priority bezeichnet	Mit einem Befehl für eine einzelne Schnittstelle wird eine Prioritätswartung für alle UDP-Pakete bereitgestellt, die für selbst Port-Nummern innerhalb eines bestimmten	Einfache Konfiguration mit nur einem Befehl. Bietet eine Prioritätswartung für RTP-Pakete.	Der gesamte andere Datenverkehr wird mit WFQ behandelt. Der Datenverkehr im Real-Time Conferencin

	Bereichs bestimmt sind.		g Protocol (RTCP) wird nicht priorisiert. Keine garantierte Bandbreite.[4]
LLQ, früher PQCBWFQ (Priority Queue Class-Based Weighted Fair Queuing) genannt	Zur Klassifizierung des Datenverkehrs wird eine modulare QoS-CLI mit Prioritätswarteschlange verwendet. Der klassifizierte Datenverkehr wird in eine Prioritätswarteschlange, reservierte Bandbreitenwarteschlangen oder eine nicht reservierte Standardwarteschlange gestellt. Ein Scheduler steuert die Warteschlangen auf Basis von Gewichtungen, sodass der Prioritätsverkehr zuerst gesendet wird (bis zu einem bestimmten Grenzwert bei Überlastung) und die Bandbreitengarantien eingehalten werden.	Einfache Konfiguration Möglichkeit zur Prioritätensetzung bei mehreren Datenverkehrsklassen und zur Festlegung von Obergrenzen bei der Bandbreitennutzung. Sie können auch garantierte Bandbreitenklassen und eine Standardklasse konfigurieren.	Es gibt noch keinen Mechanismus zur Bereitstellung mehrerer Prioritätsstufen. Der gesamte Prioritätsverkehr wird über dieselbe Prioritätswarteschlange gesendet. Bei Überlastungen können separate Prioritätsklassen separate Bandbreitengrenzen für oberste Priorität aufweisen. Die gemeinsame Nutzung von Prioritätswarteschlangen zwischen Anwendungen kann jedoch möglicherweise Jitter hervorrufen.[4]

1. Nicht für Sprache geeignet.
2. Benötigt garantierte Bandbreite für Sprache.

3. Latenz muss gewährleistet werden.
4. Ausreichend für Sprache.

Verzögerung der Serialisierung

Selbst wenn Warteschlangen optimal funktionieren und Sprachverkehr priorisieren, kann es vorkommen, dass die Prioritätswarteschlange leer ist und ein Paket aus einer anderen Klasse gewartet wird. Pakete aus garantierten Bandbreitenklassen müssen basierend auf ihrem konfigurierten Gewicht gewartet werden. Wenn ein Sprachpaket mit Priorität während der Verarbeitung dieser Pakete in der Ausgabewarteschlange eingeht, kann das Sprachpaket eine beträchtliche Zeit warten, bevor es gesendet wird. Bei Sprachpaketen kommt es zu Serialisierungsverzögerungen, wenn sie auf größere Datenpakete warten müssen.

Serialisierungsverzögerungen können die schlimmste Form von Jitter für Sprachpakete hervorrufen. Wenn die Sprachpakete hinter einem Datenpaket warten müssen, das 1500 Byte groß ist, auf einer langsameren Verbindung, bedeutet dies eine enorme Verzögerung. Die Serialisierungsverzögerung unterscheidet sich erheblich, wenn das Datenpaket 80 Byte umfasst, wie im folgenden Beispiel gezeigt:

- Serialisierungsverzögerung auf einer 64-Kbit/s-Verbindung aufgrund eines 1500-Byte-Pakets = 1500*8/64000 = 187,5 ms.
- Serialisierungsverzögerung bei einer 64-Kbit/s-Verbindung aufgrund eines 80-Byte-Pakets = 80*8/64000 = 10 ms.

Daher muss ein Sprachpaket möglicherweise bis zu 187,5 ms warten, bevor es gesendet wird, wenn es hinter einem einzelnen 1500-Byte-Paket auf einer 64-Kbit/s-Verbindung steckt. Andererseits muss ein anderes Sprachpaket nur 10 ms auf das Ziel-Gateway warten. Dies führt zu einem enormen Jitter, der aufgrund der Varianz der Verzögerung zwischen den Paketen auftritt. Auf dem ursprünglichen Gateway werden Sprachpakete in der Regel alle 20 ms gesendet. Mit einem End-to-End-Verzugsbudget von 150 ms und strengen Jitter-Anforderungen ist eine Lücke von mehr als 180 ms inakzeptabel.

Stellen Sie einen Fragmentierungsmechanismus vor, der sicherstellt, dass die Größe einer Übertragungseinheit unter 10 ms liegt. Alle Pakete mit einer Serialisierungsverzögerung von mehr als 10 ms müssen in 10-ms-Chunks fragmentiert werden. Ein 10-ms-Chunk oder -Fragment ist die Anzahl der Bytes, die über den Link in 10 ms gesendet werden. Berechnen Sie die Größe mithilfe der Verbindungsgeschwindigkeit, wie in diesem Beispiel gezeigt:

- Fragmentierungsgröße = (0,01 Sekunden * 64.000 Bit/s) / (8 Bit/Byte) = 80 Byte

Das Senden eines 80-Byte-Pakets oder die Fragmentierung über eine 64-Kbit/s-Verbindung dauert 10 ms.

Bei mehreren ATMs oder Frame Relay Permanent Virtual Circuits (PVCs) auf einer einzigen physischen Schnittstelle müssen die Fragmentierungswerte (auf allen PVCs) auf der Basis der PVC konfiguriert werden, die über die niedrigste verfügbare Bandbreite verfügt. Wenn es beispielsweise drei PVCs mit einer garantierten Bandbreite von 512 Kbit/s, 128 Kbit/s und 256 Kbit/s gibt, konfigurieren Sie alle drei PVCs mit einer Fragmentgröße von 160 Byte (die niedrigste Geschwindigkeit beträgt 128 Kbit/s, die eine Fragmentgröße von 160 Byte erfordert). Diese Werte werden für verschiedene Verbindungsgeschwindigkeiten empfohlen:

Link Speed (kbps)	Fragmentation Size (bytes)
56	70

64	80
128	160
256	320
512	640
768	960
1024	1280
1536	1920

Hinweis: Wenn die Fragmentgröße größer als die MTU-Größe der Verbindung ist, ist keine Fragmentierung erforderlich. Beispielsweise beträgt die Fragmentgröße für eine T1-Verbindung mit einer MTU von 1500 Byte 1920 Byte. Daher ist keine Fragmentierung erforderlich. Die Paketfragmentierungsgröße sollte nie niedriger sein als die VoIP-Paketgröße. VoIP-Pakete nicht fragmentieren. Die Fragmentierung dieser Pakete führt zu zahlreichen Problemen bei der Anrufeinrichtung und -qualität.

Derzeit sind drei Mechanismen zur Fragmentierung und Verschachtelung von Verbindungen verfügbar. Weitere Erläuterungen zu verschiedenen Verzögerungen in einem Paketnetzwerk finden Sie unter Verzögern von Paketen-Sprachnetzwerken. In dieser Tabelle sind die Vorteile und Einschränkungen aufgeführt:

LFI-Mechanismus (Link Fragmentation and Interleaving)	Beschreibung	Vorteile	Einschränkungen
MTU-Fragmentierung mit WFQ	Befehl auf Schnittstellenebene zum Ändern der MTU-Größe oder der IP-MTU-Größe. Werden zur Fragmentierung großer IP-Pakete auf eine bestimmte MTU-Größe verwendet. LFI verwendet WFQ, um Echtzeit-Pakete zwischen den Fragmenten zu übertragen.	Einfache Konfiguration	Fragmente werden nur durch die empfangende Anwendung wieder zusammengebaut. Daher ist eine ineffiziente Nutzung des Netzwerks erforderlich. Nur IP-Pakete, deren DF-Bit (Do Not Fragment) nicht festgelegt ist, können

			die Fragmentierung gut handhaben. Hoch prozessorintensiv. Nicht empfohlen.
Multilink Point-to-Point Protocol (MLPPP)-LFI	Bei seriellen Point-to-Point-Verbindungen muss zunächst MLPPP konfiguriert werden. Anschließend muss eine Fragmentierungsgröße in ms festgelegt werden. Interleaving muss auch auf der Multilink-Schnittstelle aktiviert sein.	Pakete werden auf einem Ende der Verbindung fragmentiert und am anderen Ende wieder zusammengesetzt. Mehrere Verbindungen können kombiniert werden, um als große virtuelle Leitung zu fungieren.	Nur für Verbindungen verfügbar, die für PPP konfiguriert sind. Lösungen für PPP over Frame Relay oder PPP over ATM werden auch in Version 12.1(5)T oder höher der Cisco IOS-Software unterstützt.
Frame Relay Fragmentation (FRF.12)	Auf Frame Relay-PVCs muss der Befehl **Frame-Relay Traffic Shaping** aktiviert und unter der Map-Class eine Fragmentierungsgröße festgelegt werden.	Die Pakete sind auf einem Ende der PVC fragmentiert und am anderen Ende wieder zusammengesetzt.	Nur bei Frame Relay-PVCs verfügbar, bei denen der **Frame-Relay Traffic-Shaping-**Befehl aktiviert ist.

VAD

Eine regelmäßige Sprachkommunikation besteht aus mehreren Momenten der Stille. Eine typische Sprachkommunikation besteht zu 40 bis 50 Prozent aus Schweigen. Da 40 % eines Sprachanrufs nicht über das Netzwerk übertragen werden, kann durch die Bereitstellung von VAD ein Teil der Bandbreite eingespart werden. Mit VAD sucht das Gateway nach Sprachlücken. Diese Lücken werden durch Komfort-Geräusch (Hintergrundgeräusche) ersetzt. So wird eine Menge Bandbreite eingespart. Es gibt jedoch einen Kompromiss. Es gibt eine kleine Zeit (in der Größenordnung von Millisekunden), bevor die Codecs Sprachaktivität erkennen, gefolgt von einer Pause. Diese kleine Zeit führt zum Front-End-Clipping empfangener Stimme. Um die Aktivierung während sehr kurzer Pausen zu vermeiden und das Clipping auszugleichen, wartet VAD ca. 200 ms nach dem Stoppen der Sprache, bevor die Übertragung beendet wird. Nach dem Neustart der Übertragung werden die vorherigen 5 ms Sprache zusammen mit der aktuellen Sprache angezeigt. VAD deaktiviert sich automatisch bei einem Anruf, wenn das Umgebungsgeräusch die Unterscheidung zwischen Sprache und Hintergrundgeräusch verhindert. Wenn die Bandbreite jedoch kein Problem darstellt, schalten Sie das VAD aus.

VAD-Parameter einstellen

Die Funktionsweise von VAD wird durch zwei Parameter bestimmt. Dies sind die Befehle **für Musik** und **Vad-Time für Sprache**.

Musikschwelle

Es wird zunächst festgelegt, welcher Schwellenwert für die Aktivierung von VAD festgelegt wird. Dies wird gesteuert, indem der **Befehl music-threshold *threshold_value* auf einem Sprach-Port** definiert wird, wie in diesem Beispiel gezeigt. Der Bereich hierfür liegt zwischen -70 Dezimalstellen pro Milliwatt (dBm) und -30 dBm. Der Standardwert hierfür ist -38 dBm. Wenn ein niedrigerer Wert (in Richtung -70 dBm) konfiguriert wird, wird VAD bei einer wesentlich geringeren Signalstärke aktiv (die Lautstärke muss wirklich niedrig fallen, bevor sie als Stille angesehen wird). Wenn ein höherer Wert (näher an -30 dBm) konfiguriert wird, wird VAD selbst bei einem kleinen Rückgang der Sprachsignalstärke aktiv. Es treibt den Playout an, um Komfort-Rauschpakete öfter abzuspielen. Dies führt jedoch manchmal zu einem geringfügigen Abschneiden der Audiowiedergabe.

```
3640-6#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
3640-6(config)#voice-port 3/0/0
3640-6(config-voiceport)#music-threshold ?
WORD Enter a number b/w (-70 to -30)
3640-6(config-voiceport)#music-threshold -50
3640-6(config-voiceport)#end
3640-6#
3640-6#show run | be voice-portvoice-port 3/0/0 music-threshold -50
```

Voice Vad-Time

Sobald die VAD aktiviert ist, wird die Komponente für Hintergrundgeräusche und Komfortgeräusche durch Konfigurieren des **Befehls voice vad-time *timer_value* unter der globalen Konfiguration** gesteuert, wie in diesem Beispiel gezeigt. Dies ist die Verzögerungszeit in Millisekunden für die Pausenerkennung und die Unterdrückung der Sprachpaketübertragung. Der Standardwert für die Haltezeit ist 250 ms. Das bedeutet, dass innerhalb von 250 ms Komfortgeräusch beginnt. Der Bereich für diesen Timer beträgt 250 ms bis 65.536 ms. Wenn dafür ein hoher Wert konfiguriert ist, kommt viel später Komfortgeräusch ins Spiel (Hintergrundgeräusche werden weiterhin gespielt). Wenn dies für 65536 ms konfiguriert ist, wird

das Komfortgeräusch deaktiviert. Ein höherer Wert für diesen Timer ist wünschenswert, um den Übergang zwischen Hintergrundgeräuschen und Komfortgeräusch zu erleichtern. Der Nachteil bei der Konfiguration des Befehls **Voice Vad-Time** besteht darin, dass er die angestrebte Bandbreiteneinsparung von 30 bis 35 Prozent nicht erreicht.

```
3640-6#
3640-6#
3640-6#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
3640-6(config)#voice vad-time ?
<250-65536> milliseconds
3640-6(config)#voice vad-time 750
3640-6(config)#end
3640-6#
3640-6#
3640-6#
3640-6#show run | be vad-time voice vad-time 750
```

Typische Konfigurationsbeispiele für QoS

Ein typisches Szenario für die Einrichtung von VoIP-Anrufen ist entweder eine Frame-Relay-Verbindung oder eine PPP-Verbindung. Dies sind Konfigurationsbeispiele für diese Szenarien.

VoIPoFR - QoS-Konfigurationsbeispiel

In diesem Beispiel (das nur relevante Abschnitte der Konfiguration enthält) wird davon ausgegangen, dass die Frame-Relay-Schaltgeschwindigkeit 256 Kbit/s beträgt. Die garantierte Committed Information Rate (CIR) für PVC 100 beträgt 64 Kbit/s und für PVC 200 192 Kbit/s. PVC 100 wird für die Übertragung von Daten und Sprache verwendet. PVC 200 wird nur zur Übertragung von Daten verwendet. Es stehen zu jedem Zeitpunkt maximal vier gleichzeitige Sprachanrufe zur Verfügung. Konfigurieren Sie die Fragmentierung auf beiden PVCs auf Basis der CIR des PVC mit der niedrigsten Bandbreite (PVC mit Sprachübertragung). Basierend auf den Beispielen in diesem Dokument bedeutet dies, dass die Fragmentierungsgröße auf der Grundlage der CIR von PVC 100 (64 Kbit/s) festgelegt wird. Wie in der Tabelle im Abschnitt "Serialization Delay" (Serialisierungsverzögerung) gezeigt, ist für eine 64-Kbit/s-Verbindung eine Fragmentierungsgröße von 80 Byte erforderlich. Dieselbe Fragmentierungsgröße muss für PVC 200 konfiguriert werden.

Weitere Informationen zur Konfiguration von VoIP über Frame Relay finden Sie unter VoIP over Frame Relay mit Quality of Service (Fragmentierung, Traffic Shaping, LLQ/IP RTP Priority).

```
3660-1#show run
Building configuration...
!
class-map match-any voip
match ip rtp 16384 16383
match ip dscp 26 46
class-map match-all voip-control
match access-group 101
!
!
policy-map VoIPoFR
class voip
priority 48
```

```
class voip-control
bandwidth 8
class class-default
fair-queue
!
voice call send-alert
voice rtp send-recv
!
!
interface Serial4/0:0
bandwidth 256
no ip address
encapsulation frame-relay
frame-relay traffic-shaping
!
interface Serial4/0:0.1 point-to-point
bandwidth 64
ip address 10.10.10.10 255.255.255.0
frame-relay ip rtp header-compression
frame-relay interface-dlci 100
 class voice
!
interface Serial4/0:0.2 point-to-point
bandwidth 192
ip address 20.20.20.20 255.255.255.0
frame-relay interface-dlci 200
class data
!
map-class frame-relay data
frame-relay fragment 80
frame-relay adaptive-shaping becn
frame-relay cir 256000
frame-relay bc 32000
frame-relay be 0
frame-relay mincir 192000
frame-relay fair-queue
!
map-class frame-relay voice
frame-relay fragment 80
no frame-relay adaptive-shaping
frame-relay cir 64000
frame-relay bc 640
frame-relay be 0
frame-relay mincir 64000
service-policy output VoIPoFR
!
!
access-list 101 permit tcp any any eq 1720
!
!
voice-port 3/1/0
!
voice-port 3/1/1
!
!
dial-peer voice 10 voip
incoming called-number .
destination-pattern 1408.......
session target ipv4:10.10.10.11
dtmf-relay h245-signal h245-alphanumeric
no vad
!
dial-peer voice 20 pots
destination-pattern 1234
```

```
port 3/1/0
!
dial-peer voice 21 pots
destination-pattern 5678
port 3/1/1
```

VoIP over PPP - QoS-Konfigurationsbeispiel

In diesem Beispiel (das nur relevante Abschnitte der Konfiguration enthält) wird davon ausgegangen, dass die QoS für einen Punkt-zu-Punkt-T1-Controller (der zwölf Kanäle umfasst) konfiguriert werden muss. Es stehen zu jedem Zeitpunkt maximal vier gleichzeitige Sprachanrufe zur Verfügung. Die Konfigurationsaufgabe besteht darin, diese serielle Schnittstelle mit PPP-Kapselung zu konfigurieren, sie zu einer Multilink-Gruppe zu machen, eine Multilink-Schnittstelle (die zur gleichen Multilink-Gruppe gehört) zu erstellen und die gesamte QoS auf der Multilink-Schnittstelle zu konfigurieren. Weitere Informationen zur Konfiguration von VoIP über PPP finden Sie unter VoIP over PPP Links with Quality of Service (LLQ/IP RTP Priority, LFI, cRTP).

```
3660-1#show run
Building configuration...
!
class-map match-any voip
match ip rtp 16384 16383
match ip dscp 26 46
class-map match-all voip-control
match access-group 101
!
!
policy-map VoIPoPPP
class voip
priority 48
class voip-control
bandwidth 8
class class-default
fair-queue
!
voice call send-alert
voice rtp send-recv
!
!
interface Multilink7
bandwidth 768
ip address 10.10.10.10 255.255.255.0
ip tcp header-compression iphc-format
service-policy output VoIPoPPP
no cdp enable
ppp multilink
ppp multilink fragment-delay 10
ppp multilink interleave
multilink-group 7
ip rtp header-compression iphc-format
!
!
interface Serial4/0:0
bandwidth 768
no ip address
encapsulation ppp
no fair-queue
ppp multilink
multilink-group 7
!
```

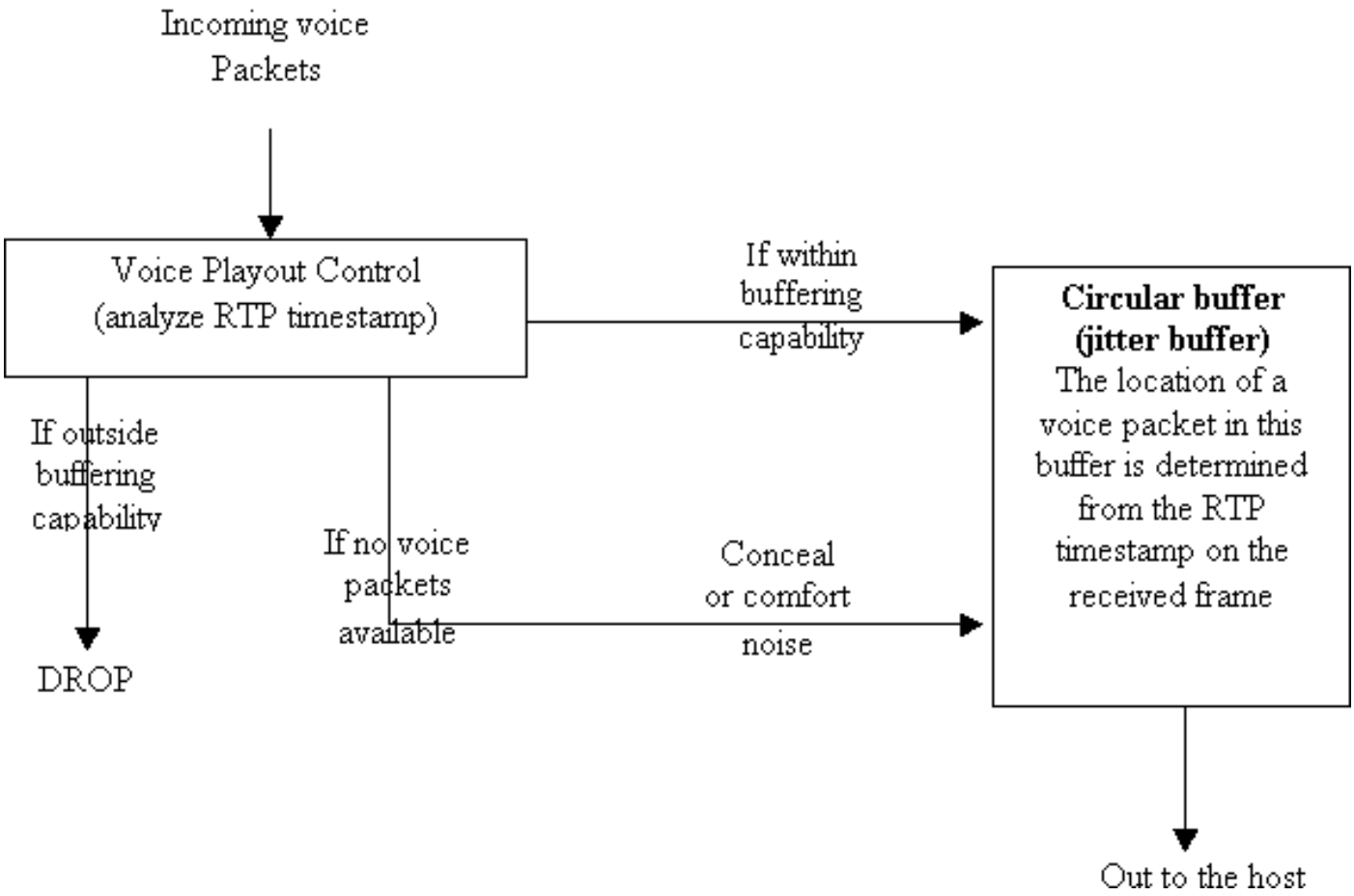
```
!
access-list 101 permit tcp any any eq 1720
!
voice-port 3/1/0
!
voice-port 3/1/1
!
!
dial-peer voice 10 voip
incoming called-number .
destination-pattern 1408.......
session target ipv4:10.10.10.11
dtmf-relay h245-signal h245-alphanumeric
no vad
!
dial-peer voice 20 pots
destination-pattern 1234
port 3/1/0
!
dial-peer voice 21 pots
destination-pattern 5678
port 3/1/1
!
```

Jitter- und Layoutmechanismus

Es gibt immer einige unkontrollierte Einheiten in einem Netzwerk, die zu weiteren Verzögerungen und Jitter in den empfangenen Sprachpaketen beitragen. Durch Ändern des Jitter-Puffers am Terminierungs-Gateway wird der unkontrollierte Jitter im Sprachnetzwerk aufgelöst.

Playout-Mechanismus

Der Jitter-Puffer ist ein Zeitpuffer. Das Terminierungs-Gateway sorgt für eine effektivere Wiedergabe. Dies ist ein Funktionsdiagramm des Wiedergabemechanismus:



Wenn das Play-Out-Steuerelement ein Sprachpaket empfängt, analysiert es den RTP-Zeitstempel. Wird das Sprachpaket über die Haltekapazität des Jitter-Puffers hinaus verzögert, wird das Paket sofort verworfen. Wenn das Paket in der Pufferung enthalten ist, wird es im Jitter-Puffer abgelegt. Der Speicherort dieses Pakets im Jitter-Puffer hängt vom für dieses Paket berechneten RTP-Zeitstempel ab. Falls kein Sprachpaket verfügbar ist, versucht das Play-Out-Steuerelement, es zu verbergen (prognostiziert das verpasste Paket). Wenn VAD aktiviert ist, wird Komfortgeräusch abgespielt.

Die Verantwortung des Play-Out-Steuerelements besteht darin, die Ereignisse von verlorenen Paketen, duplizierten Paketen, beschädigten Paketen und Paketen zu verarbeiten, die nicht in der angegebenen Reihenfolge sind. Diese Ereignisse werden durch die zeitliche Ausrichtung der getarnten Sprachpakete, die Wiedergabe von Komfortgeräuschen (sofern VAD konfiguriert ist) oder sogar die Wiederherstellung von Mehrfrequenzwahltönen (DTMF) für die Wiedergabe mit dem Host bewältigt.

Das Verbergen eines Sprachpakets erfolgt entweder durch Verbergen von Vorhersagen oder durch Verbergen von Pausen. Die Prognose-Verschleierung basiert auf dem vorherigen und dem nächsten Paket (sofern verfügbar). Er eignet sich am besten für Codecs mit niedriger Bitrate (5 Kbit/s bis 16 Kbit/s). Der Verlust von Sprachpaketen für einen Codec mit hoher Bitrate (32 Kbit/s bis 64 Kbit/s) kann möglicherweise zu unzureichenden Prognosen für Verdeckungen führen. Die Prognose beginnt bei geringen und seltenen Verzögerungen oder bei einer geringeren Anzahl von Paketverlusten. Zu viel Verheimlichung von Vorhersagen kann zu roboter Sprachqualität führen. Das Schweigen ist die schlimmste Form von Vorhersagen, die verbergen. Sie kommt zum Tragen, wenn keine Informationen vorliegen, die vorhergesagt werden können. Es ist einfach ein Hintergründen. Es beginnt bei hohen Verzögerungen und einer größeren Anzahl von Paketverlusten. Zu viel Verbergen von Schweigen führt zu abgehackter Sprachqualität. Das Verbergen der Vorhersage ist gut für 30 msekten, nach denen die Schweigedecke ins Spiel kommt.

Jitter-Puffer

Der Jitter-Puffer wird durch eine hohe Wassermarke und eine niedrige Wassermarke begrenzt. Die Wassermarke mit hohem Wasserstand ist die obere Zeitgrenze, innerhalb deren ein Paket für eine pünktliche Wiedergabe erwartet wird. Pakete, die nach der Markierung für hohe Wasserwerte eintreffen, werden als verspätete Pakete oder verlorene Pakete gekennzeichnet. Die Mindestdauer, innerhalb deren ein Paket für eine pünktliche Wiedergabe eintreffen soll, ist die Niedrigwassermarke. Pakete, die vor der Niedrigwasserkennzeichnung eintreffen, werden als frühe Pakete angesehen (sie können immer noch pünktlich ausgespielt werden).

Wenn das terminierende Gateway weiterhin eine Erhöhung der Anzahl verspäteter Pakete feststellt, erhöht sich die Wassermarke. Dieser Wert für die Markierung mit hohem Wasser bleibt während der gesamten Dauer des Anrufs gleich. Diese wird auf einen in der Konfiguration festgelegten Höchstwert erhöht. Auf ähnliche Weise erkennt das terminierende Gateway die Anzahl der frühzeitigen Pakete, die empfangen wurden. Wenn diese Pakete beginnen, das Gateway zu häufen, verringert dies die Wasserkennzeichnung. Dieser Wert bleibt während des Anrufs gleich. Dieser Jitter-Puffermodus wird als "Adaptive Mode" bezeichnet, bei dem das terminierende Gateway seinen Jitter-Puffer auf Basis des Datenverkehrsmusters anpasst. Der andere Modus ist "fester Modus". Im festen Modus gibt es einen Anfangswert für das Niedrigwasserzeichen und das Hochwasserzeichen. Dieser Wert basiert auf der geschätzten empfangenen Verzögerung (siehe Abschnitt show voice call <port-number> dieses Dokuments).

Weitere Informationen zum Jitter-Puffer finden Sie unter Understanding Jitter in Packet Voice Networks (Cisco IOS-Plattformen).

Verzögerung und Jitter identifizieren

In diesem Abschnitt wird beschrieben, wie Sie Jitter in Ihrem Netzwerk identifizieren.

aktive Anrufe anzeigen

Der Befehl **show call active voice brief** enthält viele Informationen über ein laufendes Gespräch. In dieser Ausgabe werden einige wichtige Punkte angezeigt, die mit diesem Befehl gelernt wurden:

```
11E4 : 2170927hs.1 +600 pid:10 Answer 1000 active
dur 00:08:43 tx:26157/522967 rx:7044/139565
Tele 3/0/0:9: tx:151310/755/0ms g729r8 noise:-62 acom:0 i/0:-56/-48 dBm
11E4 : 2171198hs.1 +329 pid:20 Originate 2000 active
dur 00:08:43 tx:7044/139565 rx:26165/523127
IP 30.30.30.29:18682 rtt:51ms pl:148590/290ms lost:0/0/15 delay:65/60/132ms g729r8
```

In der Befehlsausgabe für den Befehl **show call active voice brief** sehen Sie, dass alle auf dem Telefonieabschnitt empfangenen Nachrichten (rx:7044) an den IP-Abschnitt (tx:7044) übertragen werden. Dasselbe gilt für Pakete, die auf den IP-Beinen (26165) eingegangen sind und an den Telefonieabschnitt (26157) weitergeleitet werden. Die Differenz zwischen der Anzahl der auf der IP-Schicht empfangenen Pakete und der Anzahl der auf der Telefoniestufe übertragenen Pakete wird zu verspäteten Paketen beigetragen, die nicht rechtzeitig gesendet werden.

Diese Ausgabe des Befehls **show call active voice** (ohne das Schlüsselwort "brief") zeigt weitere Details zu Parametern an, die Jitter direkt identifizieren.

```
GapFillWithSilence=850 ms
GapFillWithPrediction=9230 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
```

show voice call <Port-Nummer>

Der **Befehl show voice call** *port-number* **(Portnummer für Sprachanrufe anzeigen)** liefert nützliche Informationen. Stellen Sie sicher, dass Sie entweder im Gateway konsolidiert sind, oder wenn Sie über Telnet mit einem Gateway verbunden sind, dass Sie den Befehl **terminal monitor** von der exec-Ebene aus ausgegeben haben.

Hinweis: Dieser Befehl ist auf den AS5x00-/AS5x50-Plattformen nicht verfügbar.

In dieser Ausgabe ist der Wert für Rx Delay Est (ms) 71. Dies ist der aktuelle Puffer-Wert für Jitter. Darauf wird ein Wert für die Wassermarke mit hohem Wasserstand und die Wassermarke mit niedrigem Wasserstand abgezogen. Der durchschnittliche Anfangswert für die Hochwassermarke beträgt 70 ms, während der für die Niedrigwassermarke 60 ms beträgt. Sobald ein Anfangswert festgelegt ist, verfolgt das Gateway alle frühzeitigen oder verspäteten Pakete, die eingegangen sind. Wie in der Ausgabe hier zu sehen ist, liegen die Vorhersagen bei etwa 250 ms, während die Stille Verbergen 30 ms beträgt. Es gibt immer einen höheren Wert für das Verbergen von Prognosen, da das Verbergen von Schweigen nur ein schlimmeres Szenario von Vorhersagen ist. Bei jedem Verbergen der Prognose ist ein Anstieg des Rückwurfs des Pufferüberlaufs zu verzeichnen.

Wenn man Puffer-Entsorgung sieht, bedeutet dies nicht unbedingt, dass man eine Erhöhung der Wassermarke sieht. Die obere Grenze des Jitter-Puffers ist die Wassermarke. Sie ändert sich nur, wenn ein Trend beobachtet wird. Anders ausgedrückt: Es sollte einen kontinuierlichen Fluss von verspäteten Paketen geben. Dies führt zu einer Erhöhung des Jitter-Puffers. In der Produktion ist ein solcher Trend vorhanden. Daher wird das hohe Wasserzeichen von 70 ms auf 161 ms erhöht. Wenn dieser Wert nicht geändert wird (und Sie immer noch 14 verspätete Pakete sehen), impliziert dies, dass es sich um sporadische verspätete Pakete handelt, die keinen Trend darstellen.

Achten Sie bei der Ausgabe des Befehls **show call active voice** auf verlorene Pakete. Für jedes verlorene Paket sehen Sie zwei Pakete, die nicht sequenziert sind. Dies wird auf der Ausgabe der Rx Non-Seq Pkts angezeigt. Da es sich nicht um einen positiven Wert handelt, wird der Schluss gezogen, dass auch keine Paketverluste aufgetreten sind.

```
3640-6# ***DSP VOICE TX STATISTICS***
Tx Vox/Fax Pkts: 195, Tx Sig Pkts: 0, Tx Comfort Pkts: 10
Tx Dur(ms): 192070, Tx Vox Dur(ms): 388, Tx Fax Dur(ms): 0
***DSP VOICE RX STATISTICS***
Rx Vox/Fax Pkts: 9604, Rx Signal Pkts: 0, Rx Comfort Pkts: 0
Rx Dur(ms): 192070, Rx Vox Dur(ms): 191560, Rx Fax Dur(ms): 0
Rx Non-seq Pkts: 0, Rx Bad Hdr Pkts: 0
Rx Early Pkts: 0, Rx Late Pkts: 14
***DSP VOICE VP_DELAY STATISTICS***
Clk Offset(ms): 0, Rx Delay Est(ms): 71
Rx Delay Lo Water Mark(ms): 60, Rx Delay Hi Water Mark(ms): 161
***DSP VOICE VP_ERROR STATISTICS***
Predict Conceal(ms): 250, Interpolate Conceal(ms): 0
Silence Conceal(ms): 30, Retroact Mem Update(ms): 0
Buf Overflow Discard(ms): 500, Talkspurt Endpoint Detect Err: 0
***DSP LEVELS***
TDM Bus Levels(dBm0): Rx -49.9 from PBX/Phone, Tx -41.7 to PBX/Phone
```

```
TDM ACOM Levels(dBm0): +2.0, TDM ERL Level(dBm0): +11.1
TDM Bgd Levels(dBm0): -58.9, with activity being voice
***DSP VOICE ERROR STATISTICS***
Rx Pkt Drops(Invalid Header): 0, Tx Pkt Drops(HPI SAM Overflow): 0
```

Beachten Sie die Tx Comfort Pkts und Rx Comfort Pkts. Wie bei den Beispielausgängen wird der Schluss gezogen, dass das mit diesem Router verbundene Telefon in der Regel leise bleibt, da es viele Tx Comfort Pkts gibt. Gleichzeitig gibt es keine Rx Comfort Pkts, d. h. das andere Ende spricht durchgehend.

Vergleichen Sie die Ausgabe hier mit der vorherigen Befehlsausgabe. Die Anzahl der verspäteten Rx-Pkte (von 14 auf 26) ist gestiegen. Der Wassergrenzwert erhöht sich jedoch nicht. Dies weist darauf hin, dass die 12 Pakete sporadisch verzögert werden. Der Pufferüberlaufabwurf wird auf 910 ms erhöht. Da jedoch kein Trend beobachtet wird, wird die Wassermarke nicht erhöht.

In der Ausgabe hier sind Rx Early Pkts: 3. Das bedeutet, dass ein Paket viel vor dem erwarteten Zeitpunkt eintrifft. Wie die Ausgabe hier zeigt, hat sich der Jitter-Puffer gedehnt, um noch frühere Pakete aufzunehmen, indem er die Niedrigwassermarke von 60 auf 51 reduziert.

```
3640-6# ***DSP VOICE TX STATISTICS***
Tx Vox/Fax Pkts: 209, Tx Sig Pkts: 0, Tx Comfort Pkts: 11
Tx Dur(ms): 337420, Tx Vox Dur(ms): 416, Tx Fax Dur(ms): 0
***DSP VOICE RX STATISTICS***
Rx Vox/Fax Pkts: 16843, Rx Signal Pkts: 0, Rx Comfort Pkts: 1
Rx Dur(ms): 337420, Rx Vox Dur(ms): 335920, Rx Fax Dur(ms): 0
Rx Non-seq Pkts: 0, Rx Bad Hdr Pkts: 0
Rx Early Pkts: 3, Rx Late Pkts: 26
***DSP VOICE VP_DELAY STATISTICS***
Clk Offset(ms): 0, Rx Delay Est(ms): 72
Rx Delay Lo Water Mark(ms): 51, Rx Delay Hi Water Mark(ms): 161
***DSP VOICE VP_ERROR STATISTICS***
Predict Conceal(ms): 510, Interpolate Conceal(ms): 0
Silence Conceal(ms): 70, Retroact Mem Update(ms): 0
Buf Overflow Discard(ms): 910, Talkspurt Endpoint Detect Err: 0
***DSP LEVELS***
TDM Bus Levels(dBm0): Rx -51.5 from PBX/Phone, Tx -44.1 to PBX/Phone
TDM ACOM Levels(dBm0): +2.0, TDM ERL Level(dBm0): +11.9
TDM Bgd Levels(dBm0): -61.3, with activity being voice
***DSP VOICE ERROR STATISTICS***
Rx Pkt Drops(Invalid Header): 0, Tx Pkt Drops(HPI SAM Overflow): 0
```

Konfigurieren des Jitter-Puffers auf einem Gateway

Die in diesem Dokument behandelten QoS-Richtlinien berücksichtigen das abgehackte oder verschlechterte Problem bei der Sprachqualität. Die Konfiguration eines Puffers mit Verzögerung bei der Wiedergabe ist eine Lösung für eine unsachgemäße QoS-Implementierung im Netzwerk. Verwenden Sie diese Methode nur als Stopp-Lücke-Fix oder als Tool zur Fehlerbehebung und Eingrenzung von Jitter-Problemen im Netzwerk.

Playout-Verzögerungsmodus

Der Jitter-Puffer ist entweder für den festen oder den adaptiven Modus konfiguriert. Im adaptiven Modus können Sie mit dem Gateway einen Mindestwert für den Jitter-Puffer, einen Maximalwert und einen Nominalwert konfigurieren. Der Jitter-Puffer erwartet, dass die Pakete innerhalb des Nominalwerts-Bereichs eintreffen. Der Nennwert muss entweder gleich oder größer als der Mindestwert und kleiner/gleich dem Höchstwert sein. Der Puffer wird bis zum konfigurierten

Maximalwert erweitert. Dies kann bis zu 1.700 ms betragen. Ein Problem bei der Konfiguration eines hohen maximalen Werts besteht in der Einführung einer End-to-End-Verzögerung. Wählen Sie den Wert der maximalen Wiedergabepause aus, sodass keine unerwünschten Verzögerungen im Netzwerk entstehen. Diese Ausgabe ist ein Beispiel für den Jitter-Puffer, der für den adaptiven Modus konfiguriert wurde:

```
3640-6#
3640-6#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
3640-6(config)#voice-port 3/0/0
3640-6(config-voiceport)#playout-delay mode adaptive
3640-6(config-voiceport)#playout-delay maximum 400
3640-6(config-voiceport)#playout-delay nominal 70
3640-6(config-voiceport)#playout-delay minimum low
3640-6(config-voiceport)#^Z
3640-6#
3640-6#
3640-6#show run | begin 3/0/0
voice-port 3/0/0
playout-delay maximum 400
playout-delay nominal 70
playout-delay minimum low
playout-delay mode adaptive
!
```

Im Festkonfigurationsmodus überprüft das Gateway den konfigurierten Wert auf "nominal". Obwohl es Ihnen ermöglicht, den Mindest- und Höchstwert für die Wiedergabepause zu konfigurieren, wird er bei der Konfiguration für den festen Modus ignoriert. Im festen Modus bleibt der hohe Wasserkennwert oder der niedrige Wasserkennwert immer konstant. Sie basiert auf dem Nominalwert und dem Rx Delay Est (ms)-Wert. Es ist also möglich, dass Sie im festen Modus den Wert als 200 ms konfigurieren. Liegt die geschätzte Empfangsverzögerung jedoch bei fast 100 ms, so werden die Wassermarke für die gesamte Dauer des Anrufs und die Wassermarke für die niedrigste Wassermarke festgelegt.

```
3640-6#
3640-6#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
3640-6(config)#voice-port 3/0/0
3640-6(config-voiceport)#playout-delay mode fixed
3640-6(config-voiceport)#playout-delay nominal 70
3640-6(config-voiceport)#^Z
3640-6#
3640-6#
3640-6#show run | begin 3/0/0
voice-port 3/0/0
playout-delay mode fixed
playout-delay nominal 70
!
```

Weitere Informationen zur Konfiguration der Wiedergabepause finden Sie unter Verbesserungen der Playout-Verzögerung für Voice over IP.

Zugehörige Informationen

- Implementierung von Quality of Service-Richtlinien mit DSCP
- Low Latency Queuing
- Vergleichen der Bandbreite und der Prioritätenbefehle einer QoS-Service-Richtlinie

- Konfigurieren von Link Fragmentation and Interleaving für Frame Relay- und ATM Virtual Circuits
- Konfigurieren von Link Fragmentation and Interleaving für Multilink PPP
- Unterstützung von Sprachtechnologie
- Produkt-Support für Sprach- und Unified Communications
- Fehlerbehebung bei Cisco IP-Telefonie
- Technischer Support - Cisco Systems