

Wirtualna centralka telefoniczna P2P

opis protokołu

wersja druga, 2007-05-14

Spis treści

Spis treści.....	1
Streszczenie.....	2
Cel.....	2
Założenia.....	2
Definicje.....	2
Format komunikatów.....	3
Typy proste.....	3
Stałe używane w opisie komunikatów.....	3
Typy złożone i oznaczenia.....	4
Opis przesyłanych komunikatów.....	4
Dodatkowe założenia.....	4
Komunikaty.....	4
Bezpieczeństwo.....	4
Komunikaty ogólne.....	5
Komunikaty używane przy zestawianiu połączenia i sterowaniu nim.....	5
Komunikaty używane przy zajmowaniu WLT (w grupie wieloabonentowej).....	6
Założenia.....	7
Komunikaty.....	7
Synchronizacja.....	8
Inne komunikaty wymieniane przez abonentów z grupy.....	8
Poprawne sekwencje komunikatów, algorytmy działania.....	9
Zajmowanie WLT w grupie.....	9
Zestawianie połączenia.....	9
Odbieranie połączenia (reakcja na komunikat rozmowa).....	10
Zawieszanie połączenia.....	10
Kończenie połączenia.....	11
Włączanie się abonenta z grupy.....	11
Wyłączanie się abonenta z grupy.....	12
Odzyskiwanie WLT w grupie.....	12
Historia zmian.....	13

Streszczenie

Niniejszy dokument opisuje protokół sterujący połączeniami telefonicznymi w sieci IP. Protokół ma dostarczać funkcjonalność centralki biurowej – zestawianie, zawieszanie, rozłączanie połączeń, grupy odbiorców o wspólnych numerach, wielokrotne linie telefoniczne – w środowisku rozproszonym.

Cel

Protokół ma umożliwiać zestawianie, zawieszanie i rozłączanie połączeń, obsługiwać grupy użytkowników współdzielących wirtualne linie telefoniczne oraz umożliwiać posiadanie kilku linii telefonicznych związanych z jednym numerem.

Celem protokołu jest obsługa zestawiania rozmów i sterowania nimi. Sposób transmisji dźwięku po zestawionym połączeniu jest poza zakresem protokołu i tego dokumentu, protokół narzuca jedynie zastosowanie UDP do tego celu.

Założenia

Ponieważ zestawianie rozmów ma działać w sieci IP, numerem abonenta jest para (numer IP, numer portu), na którym można się z abonentem skomunikować. Ta wersja protokołu współpracuje z IPv4 i IPv6. Wsparcie dla IPv6 jest włączone w tej wersji protokołu, ponieważ nie utrudnia znacząco jego implementacji, IPv6 kiedyś wyprze IPv4, a wydania kolejnych wersji niniejszego protokołu są odłożone na czas nieokreślony.

Protokół ten działa w warstwie aplikacji. W założeniu model komunikacji jest „peer-2-peer”, bez centralnego serwera. Protokół używa protokołu transportowego UDP do transmisji dźwięku, ponieważ nie wiąże się to z narzutami na retransmisję utraconych komunikatów. Do przekazywania informacji kontrolnych i sterujących używany jest protokół zdalnego wywoływania procedur XML-RPC (<http://www.xmlrpc.com/spec>), dzięki czemu implementatorzy protokołu będą zwolnieni ze żmudnej pracy związanej z implementacją własnego schematu przesyłania komunikatów. Ponadto XML-RPC zapewnia przenośność protokołu – w chwili obecnej jest zaimplementowany dla wielu różnych języków programowania oraz na wielu różnych architekturach sprzętowych i środowiskowych.

Ta wersja protokołu nie obsługuje połączeń zestawianych z udziałem pośredników, obchodzeniem firewalli, itp.

Definicje

abonent – użytkownik instancji protokołu, identyfikowany przez parę (numer IP, numer portu)

grupa – zbiór użytkowników, znających wzajemnie swoje identyfikatory i wiedzących o fakcie przynależenia do grupy. Jeżeli abonent nie należy do żadnej grupy, przyjmuje się, że należy do grupy jednoabonentowej.

wirtualna linia telefoniczna (WLT) – logiczny kanał informacyjny umożliwiający prowadzenie rozmowy i sterowanie nią

Format komunikatów

Wszystkie przesyłane komunikaty są wywołaniami procedur XML-RPC lub odpowiedziami na nie, także zgodnymi ze standardem XML-RPC. Dane w kanale głosowym przesyłane są w postaci pakietów UDP, protokół nie przewiduje żadnej ingerencji w treść tych pakietów.

Typy proste

Typy proste przesyłane w wywołaniach XML-RPC są dokładnie takie jak w XML-RPC: int, boolean, string, double, dateTime.iso8601, base64, struct, array. Dokładny opis jest na stronie <http://www.xmlrpc.com/spec>.

Stałe używane w opisie komunikatów

falsz – XML-RPC *boolean* 0

prawda – XML-RPC *boolean* 1

$T_{zyje} = 40$ sekund

Od tego zależy czas odzyskiwania zasobów w grupie. Zasoby będą odyskane po czasie

T_{nie_zyje} , zależącym od T_{zyje}

$T_{nie_zyje} = 100$ sekund

musi być większe niż $2 * T_{zyje}$, by nie rozpoczynać odzyskiwania zasobów wcale nie osieroconych poprzez awarię abonenta z grupy

$T_{nie_uzywam} = 65$ sekund

dotkliwe zabezpieczenie przed używaniem linii używanych przez innych abonentów w grupie; na wypadek sytuacji, w której tylko niektóre komunikaty nie dochodzą; powinno być większe niż T_{zyje} , by oczekiwanie miało sens

$T_{synchr} = \frac{1}{10}$ sekundy

czas, po którym możliwe jest odrzucenie komunikatu bez odpowiedzi; musi być nieduże

$T_{przedawnienie} = 5$ sekund

czas, po którym nadawca komunikatu uznaje odbiorcę za nieaktywnego; musi co najmniej kilka razy większe niż T_{synchr}

$T_{obsługa} = 4$ sekundy

czas, po którym odbiorca komunikatu definitywnie odmawia odpowiedzi; musi być wyraźnie mniejsze od $T_{przedawnienie}$

$P_{synchr} = \frac{\lceil \log(N) \rceil}{N}$, gdzie N jest liczbą abonentów w grupie

musi być prawdopodobieństwem; ma być łatwo obliczalne i ma mniej więcej minimalizować liczbę retransmitowanych komunikatów w algorytmie przyznawania linii w grupie

Typy złożone i oznaczenia

numer portu przekazywany jest jako *int*.

numer linii (WLT) przekazywany jest jako *int*.

lista przekazywana jest jako tablica z kolejnymi elementami listy.

numer IP przekazywany jest jako *string* postaci

„*nnn.nnn.nnn.nnn*” dla adresów IPv4, gdzie każde „*n*” to cyfra oraz całość tworzy poprawny adres,

„*[nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnnn]*” dla adresów IPv6, gdzie każde *n* to cyfra lub litera z przedziałów A-F i a-f; dopuszczalne są skróty zapisu jak opisano w [RFC 4291](#). (np. „*127.0.0.1*” i „*::ffff:7f00:1*” to w obu przypadkach *localhost*)

identyfikator abonenta przekazywany jest jako *struct* o polach „*ip*”, „*port*” i ewentualnie „*cookie*”. Numer portu i adres przekazywane są jak opisano wyżej. *Cookie* to dowolny napis (*string*) stały dla pary nadawca, odbiorca (powinien być różny dla jednego nadawcy i różnych odbiorców). *Cookie* przesyłane jest w identyfikatorze abonenta tylko wtedy, gdy identyfikator dotyczy nadawcy komunikatu, w pozostałych przypadkach pola tego nie ma.

stan linii (por. „Komunikaty używane przy zajmowaniu WLT w grupie”) przekazywany jest jako *struct* o polach „*stan*”, „*abonenci*”. Stan określony jest przez nazwę (*string* w kodowaniu UTF-8), *abonenci* to lista identyfikatorów abonentów będących parametrem stanu.

błąd(N, D) odpowiada odpowiedzi *fault* z XML-RPC, będącej rodzajem specjalnej wartości zwracanej, oznacza błąd wykonania o numerze N, opisie D.

błąd oznacza **błąd(0, „Error”)**

Opis przesyłanych komunikatów

Dodatkowe założenia

Poza parametrami komunikatów opisanymi przy poszczególnych komunikatach nadawca komunikatu zawsze podaje jako pierwszy parametr swój identyfikator abonenta (poniższej oznaczany jako parametr *ja*). Identyfikator abonenta związany z parametrem *ja* musi posiadać niepuste pole *cookie*.

Jeśli nie określono inaczej, wartością zwracaną jest *prawda*. Jeżeli komunikat nie jest obsłużony (brak poprawnej odpowiedzi XML-RPC) przez czas $T_{\text{przedawnienie}}$, nadawca powinien uznać odbiorcę za wyłączzonego.

Komunikaty

Bezpieczeństwo

Każdy przychodzący poprawny komunikat wyposażony jest w identyfikator nadawcy z polem *cookie*. *Cookie* jest kluczem autoryzującym nadawcę, który można zweryfikować komunikatem **autoryzuj**. Ponieważ *cookie* jest stałe dla pary (nadawca, odbiorca) wystarczy zweryfikować jego pierwsze

wystąpienie. Ze względu na możliwość awarii nadawcy lub zmianę *cookie* z dowolnej innej przyczyny, pierwszą reakcją na nadejście komunikatu z nieprawidłowym *cookie* powinna być próba jego autoryzowania. Wyjątek stanowią oczywiście sytuacje ataków typu DoS itp., lecz poprawna reakcja na nie leży poza obszarem objętym tym protokołem.

Komunikaty ogólne

autoryzuj(ja, cookie)

Przesyłane na główny port odbiorcy.

Wartość zwracana jest typu *boolean* i informuje, czy odbiorca w komunikacji z nadawcą używa w swoim identyfikatorze *cookie*.

Odpowiedź powinna być jak najszybsza, nie może to być komunikat blokujący, ponieważ w zależności od implementacji może być on wysyłany bardzo często.

Komunikaty używane przy zestawianiu połączenia i sterowaniu nim

rozmowa(ja, port)

Przesyłane na główny port odbiorcy.

port jest numerem portu kanału sterowania WLT wykorzystywanej przez nadawcę do sterowania nowym połączeniem.

Wartość zwracana: *falsz* lub lista identyfikatorów wszystkich członków grupy odbiorcy.

rozmowa() oznacza chęć rozpoczęcia rozmowy z odbiorcą (bądź dowolnym członkiem grupy odbiorcy).

Nadawca po otrzymaniu *falsz* powinien uznać połączenie za odrzucone, po otrzymaniu listy identyfikatorów powinien czekać na porcie *port* na komunikat **odrzucenie** lub

rozmawiamy.

odrzucenie(ja)

Przesyłane na port kanału sterowania WLT.

Wartość zwracana: *prawda*.

Oznacza odrzucenie połączenia.

Odbiorca powinien sprawdzić, czy nadawca jest na liście zwróconej w odpowiedzi na komunikat **rozmowa**. Powinien zwrócić **błąd**, jeżeli otrzymał wcześniej jakiś komunikat

odrzucenie lub **rozmawiamy** w tym kanale sterowania od ostatniego swojego komunikatu

rozmowa(*, port'), gdzie *port'* jest numerem portu kanału sterowania, na którym odbierany jest komunikat.

rozmawiamy(ja, port)

Przesyłane na port kanału sterowania WLT.

port jest numerem portu UDP kanału głosowego WLT nadawcy wykorzystywanej w danej rozmowie.

Wartość zwracana: numer portu UDP, który będzie używany przez odbiorcę do komunikacji w kanale głosowym.

Oznacza pozytywną odpowiedź na żądanie rozmowy. Od momentu wysłania odpowiedzi

zwrotnej nadawca i odbiorca mogą się ze sobą komunikować na odpowiednich portach UDP

– połączenie głosowe jest zestawione.

Odbiorca powinien zwrócić błąd, jeżeli otrzymał wcześniej jakiś komunikat **odrzućcie** lub **rozmawiamy** w tym kanale sterowania od ostatniego swojego komunikatu **rozmowa(*, port')**, gdzie *port'* jest numerem portu kanału sterowania, na którym odbierany jest komunikat.

zawieszenie(ja, grupa)

Przesyłane na port kanału sterowania WLT.

grupa jest listą identyfikatorów członków grupy nadawcy.

Wartość zwracana: *prawda*, jeżeli rozmowę można zawiesić (w szczególności nie jest zawieszona), wpp. *fałsz*.

Komunikat ten nie może być obsługiwany równoległe z wysyłaniem lub obsługiwaniem innych komunikatów na danej WLT.

Odbiorca powinien sprawdzić, czy komunikat pochodzi od abonenta, z którym rozmawia. Oznacza chęć tymczasowego zawieszenia rozmowy, tj. zerwania kanału głosowego. Po wymianie tego komunikatu odbiorca i nadawca mogą zamknąć porty UDP używane w kanale głosowym aktywnej rozmowy. Odbiorca rozpoczyna periodyczne wysyłanie komunikatu **zyje** do nadawcy co czas T_{zyje} .

odwieszenie(ja, port)

Przesyłane na port kanału sterowania WLT.

port jest numerem portu UDP kanału głosowego WLT nadawcy wykorzystywanej w danej rozmowie.

Wartość zwracana: numer portu UDP, który będzie używany przez odbiorcę do komunikacji w kanale głosowym.

Odbiorca powinien sprawdzić, czy komunikat pochodzi od abonenta z listy *grupa* przesłanej z komunikatem **zawieszenie**. Od momentu wysłania odpowiedzi zwrotnej nadawca i odbiorca mogą się ze sobą komunikować na odpowiednich portach UDP – połączenie głosowe jest zestawione.

Jeżeli rozmowa nie jest zawieszona, odbiorca zwraca **błąd**.

zakonczenie(ja)

Przesyłane na port kanału sterowania WLT.

Wartość zwracana: *prawda*.

Oznacza zakończenie rozmowy. Po wymianie tego komunikatu odbiorca i nadawca mogą zamknąć porty UDP używane w tej rozmowie i zwalniają WLT używane w rozmowie.

Nadawca od momentu wysłania komunikatu uznaje rozmowę za zakończoną i odrzuca z błędem wszystkie komunikaty **zakonczenie**, **zawieszenie**, **odwieszenie**.

Komunikaty używane przy zajmowaniu WLT (w grupie wieloabonentowej)

Abonent tworzący grupę jednoabonentową sam zarządza swoimi WLT i nie wymaga to wymiany dodatkowych komunikatów. Dwaj lub więcej abonentów tworzących grupę abonentów współdzielili WLT przy odbieraniu i dzwonieniu, więc istnieje potrzeba wzajemnego wykluczania przy korzystaniu z WLT.

Założenia

Abonent należący do grupy przechowuje informację o wszystkich WLT posiadanych przez grupę. Każda WLT niezależnie może być w stanie:

wolna

zajmowana przez określony zbiór abonentów ubiegających się o dostęp do danej WLT

zajęta przez określonego abonenta, wykorzystywana w rozmowie lub w zestawianiu rozmowy

Komunikaty

Poniższe komunikaty muszą być przesyłane na główny port odbiorcy i obsługiwane synchronicznie: W czasie od wysłania komunikatu do odebrania odpowiedzi (wartości zwracanej) nadawca nie może obsługiwać innych komunikatów z poniższej listy. Opis metody unikania zakleszczeń znajduje się niżej. Wymiana poniższych komunikatów służy zapewnieniu wzajemnego wykluczania przy dostępie do WLT wg algorytmu „Zajmowanie WLT w grupie” opisanego niżej.

W poniższych opisach „linia” oznacza WLT.

pozwol(ja, linia)

Nadawca prosi odbiorcę o przydzielenie linii *linia*. Jeżeli odbiorca także rywalizuje o dostęp do WLT o numerze *linia*, to rezygnuje z tej rywalizacji powiadamiając o tym odpowiednich abonentów. Odbiorca zaznacza WLT jako **zajmowana** i dodaje nadawcę do zbioru abonentów ubiegających się o dostęp do danej WLT. Od tego momentu nadawca jest zobowiązany do wysłania komunikatu **rezygnacja** do odbiorcy jeśli tylko zrezygnuje z ubiegania się o daną WLT.

Nadawca może wysłać komunikat **pozwol** tylko wtedy, gdy nie wie nic o tym, by ktokolwiek inny ubiegał się o linię *linia*.

Odbiorca odpowiada **błąd**, jeśli wg niego linia jest **zajęta**.

rezygnacja(ja, linia)

Powiadamia odbiorcę o tym, że nadawca zrezygnował z ubiegania się o linię *linia*. Odbiorca powinien usunąć *ja* ze zbioru abonentów ubiegających się o dostęp do niej. Jeżeli zbiór ten stał się pusty, odbiorca oznacza linię jako *wolna*.

Jeżeli wg odbiorcy linia jest już **zajęta** przez *ja*, oznacza linię jako **wolna**.

zajalem(ja, linia)

Powiadamia odbiorcę o tym, że *ja* korzysta z linii *linia*. Linia *linia* powinna być u odbiorcy zaznaczona jako **zajmowana**, a *ja* powinien należeć do zbioru abonentów ubiegających się o dostęp do niej – w przeciwnym przypadku zgłaszany jest **błąd**.

Odbiorca zaznacza linię *linia* jako **zajęta** przez *ja*.

zwalniam(ja, linia)

Powiadamia odbiorcę o tym, że linia *linia* jest teraz **wolna**. Jeżeli wg odbiorcy linia nie jest **zajęta** lub jest **zajęta** przez kogoś innego, zgłaszany jest **błąd**.

stan_linii(ja)

Odbiorca zwraca nadawcy listę stanów wszystkich linii.
Jeżeli nadawca nie jest w grupie odbiorcy, zgłaszany jest **błąd**.

Synchronizacja

Ponieważ komunikaty z powyższej grupy są obsługiwane synchronicznie i po jednym na raz na każdego abonenta, możliwa jest sytuacja zakleszczenia – cyklicznego oczekiwania na obsłużenie komunikatu. Konieczny jest więc mechanizm wykrywania zakleszczeń, który swoją prostotą odpowiadałby prostocie całego protokołu, a zarazem pozwalał na efektywne obsługiwane grup abonentów o liczebności nie przekraczającej 100.

Jeżeli odbiorca komunikatu wymienianego synchronicznie (**pozwol**, **rezygnacja**, **zajalem**, **zwalniam**, **stan_linii**) przez czas T_{synchron} nie może go obsłużyć, to z prawdopodobieństwem P_{synchron} zwraca **błąd(1, „EAGAIN”)**, w przeciwnym przypadku zeruje licznik czasu (tj. jeśli znowu upłynie czas T_{synchron} , ponownie podejmuje losową decyzję). Jeżeli odbiorca nie może obsłużyć komunikatu przez czas $T_{\text{obsługa}}$ (licząc od faktycznego początku obsługiwanego), bezwarunkowo zwraca **błąd(1, „EAGAIN”)**.

Nadawca po otrzymaniu odpowiedzi **błąd(1, „EAGAIN”)** obsługuje wszystkie zaległe komunikaty przychodzące i – o ile w trakcie obsługi komunikatów przychodzących nie zmieni zdania – ponownie wysyła swój komunikat.

Inne komunikaty wymieniane przez abonentów z grupy

Poniższe komunikaty są przesyłane na główny port odbiorcy.

dzwoni(ja, linia, on, port)

Informuje odbiorcę, że grupa na linii *linia* otrzymała żądanie połączenia od abonenta *on*. *port* jest numerem portu kanału sterowania WLT użytej przez *on* do nawiązania rozmowy. W reakcji na ten komunikat odbiorca może wysłać do *on* na port *port* komunikat **odrzucenie** lub **rozmawiamy**. Jeżeli komunikat zostanie zaakceptowany (odpowiedź będzie *prawda*), to powiadamia grupę o tym, że zajął linię (komunikat **zajalem**), a następnie jeżeli wysłał do *on* **odrzucenie** powiadamia grupę o tym, że linia jest wolna (komunikat **zwalniam**).

rozmawiam(ja, linia, on)

Informuje odbiorcę, że *ja*, członek jego grupy, na linii *linia* rozmawia z abonentem o identyfikatorze *on*. Odbiorca sprawdza, czy linia *linia* jest wg niego **zajęta** i ewentualnie sygnalizuje **błąd**. Oznacza linię jako **zajęta** przez *ja*.

zawiesilem(ja, linia, on, port)

Informuje odbiorcę, że nadawca zawiesił kanał głosowy rozmowy prowadzonej na linii *linia*. Odbiorca może od teraz do otrzymania **rozmawiam** lub **zakonczyłem** wysłać do abonenta *on* na port *port* komunikat **odwieszenie**, a w przypadku sukcesu wysyła do całej grupy komunikat **rozmawiam**.

zyje(ja)

Nadawca informuje odbiorcę o tym, że działa.

Każdy członek grupy powinien wysyłać do pozostałych członków grupy takie komunikaty nie rzadziej niż co czas T_{zyje} .

Poprawne sekwencje komunikatów, algorytmy działania

Zajmowanie WLT w grupie

Jeżeli abonent będący członkiem grupy wieloabonentowej chce zająć WLT powinien wykonać następujące kroki:

- 1 Wybrać jedną z WLT, o której wie, że jest wolna (tj. nie wie nic o tym, by była zajęta). Jeżeli takiej linii nie ma, rezygnuje z zajmowania linii i kończy wykonywanie algorytmu.
- 2 Niech n będzie numerem linii. Do wszystkich członków grupy w losowej kolejności wysyła komunikat **pozwol(ja, n)** w międzyczasie obsługując inne przychodzące komunikaty.
 - 2.1 Jeżeli w międzyczasie otrzyma od kogoś komunikat **pozwol(*, n)**, to rezygnuje z zajmowania linii, odpowiednio obsługuje komunikat.
 - 2.2 Jeżeli w międzyczasie otrzyma jeden z komunikatów **rezygnacja(*, n)**, **zajalem(*, n)**, **zwalam(*, n)**, **dzwoni(*, n, *, *)**, **rozmawiam(*, n, *, *)**, **zawiesilem(*, n, *)**, to rezygnuje z zajmowania linii, odpowiednio obsługuje komunikat, a fakt jego nadejścia zapisuje w logu błędów, jeśli taki istnieje.
- 3 Jeżeli abonent rezygnuje z zajmowania linii n , wysyła do wszystkich tych, do których wcześniej wysłał **pozwol**, komunikat **rezygnacja(ja, n)** i kończy wykonywanie algorytmu.
- 4 Jeżeli abonent wysłał już do wszystkich członków grupy **pozwol**, zajmuje linię i wysyła do wszystkich komunikat **zajalem(ja, n)**, a przypadku otrzymania odpowiedzi **bład**, rezygnuje z korzystania z linii wysyłając do wszystkich **rezygnacja**. Jeżeli wszystkie **zajalem** zakończyły się sukcesem (brak połączenia lub odpowiedź prawda), od teraz abonent ma wyłączność w dysponowaniu linią n aż do zrzeczenia się jej (poprzez komunikat **zawiesilem**, **zwalam**).

Zestawianie połączenia

- 1 Abonent wybiera i zajmuje wolną WLT.
 - 1.1 Jeżeli abonent należy do grupy wieloabonentowej, ten punkt oznacza wykonanie kroków algorytmu „Zajmowanie WLT w grupie”, a w przypadku niepowodzenia przerwanie wykonywania kroków niniejszego algorytmu.
- 2 Wybranie abonenta docelowego. Abonent wysyła do niego komunikat **rozmowa(ja, port)**, gdzie *port* jest numerem portu kanału sterowania wybranej linii WLT. W przypadku sukcesu, abonent zapamiętuje listę abonentów z grupy abonenta docelowego.
 - 2.1 W przypadku porażki (odpowiedź negatywna lub brak poprawnej odpowiedzi XML-RPC przez czas $T_{przedawnienie}$), abonent rezygnuje z dzwonięcia i zwalnia linię.
- 3 Oczekiwanie na porcie *port* na komunikat **odrzućcie** lub **rozmawiamy** pochodzący od członka grupy abonenta docelowego.

- 3.1 Komunikat **odrzucenie** oznacza odrzucenie rozmowy i zakończenie algorytmu.
- 3.2 Komunikat **rozmawiamy** musi być natychmiast obsłużony poprzez nawiązanie połączenia na kanale głosowym (por. opis komunikatu **rozmawiamy**). W przypadku, gdy nawiązanie tego połączenia jest niemożliwe, abonent rezygnuje z dalszego zestawiania połączenia i wysyła stronie rozmowy **zakonczenie**.
- 3.3 Brak komunikatu oznacza oczekiwanie na reakcję odbiorcy i nie wymaga działania.
- 4 Jeżeli poprzednie kroki się udały, abonent i abonent docelowy (lub inny członek grupy abonenta docelowego) mają nawiązane połączenie na kanale głosowym.

Odbieranie połączenia (reakcja na komunikat rozmowa)

- 1 Abonent podczas obsługiwanego komunikatu **rozmowa** wybiera i zajmuje wolną WLT. Jeśli jest to niemożliwe w czasie $T_{obsługa}$, zwraca *falsz*.
- 2 Jeżeli jest członkiem grupy wieloabonentowej, wysyła do pozostałych członków swojej grupy komunikat **dzwoni(ja, n, on, port)**, gdzie n jest numerem linii zajętej w punkcie 1, on jest identyfikatorem abonenta dzwoniącego, a $port$ jest numerem portu jego kanału sterowania WLT, z której dzwoni (informacja przekazana wraz z komunikatem **rozmowa**).
- 3 Abonent oraz każdy członek jego grupy po otrzymaniu **dzwoni(*, n, on, port)** może wysłać do on na port $port$ komunikat **odrzucenie** lub **rozmawiamy**.
 - 3.1 Wysłanie komunikatu **rozmawiamy** zakończone sukcesem (wartość zwracana równa *prawda*) oznacza przejęcie kontroli nad linią n i zobowiązuje do wysłania do całej grupy komunikatu **rozmawiam**.
 - 3.2 Wysłanie komunikatu **odrzucenie** zakończone sukcesem (wartość zwracana równa *prawda*) zobowiązuje do wysłania do całej grupy komunikatów **rozmawiam** i **zwalniam(*, n)**.
 - 3.3 Otrzymanie od członka grupy komunikatu **rozmawiam(*, n, *)** lub **zwalniam(*, n)** wywołuje rezygnację z rozmawiania lub odrzucania rozmowy.

Zawieszanie połączenia

Podczas rozmowy każda ze stron może zawiesić połączenie. W tym celu:

- 1 Wysyła do drugiej strony rozmowy komunikat **zawieszenie**. W przypadku odpowiedzi różnej niż *prawda* zawieszenie uznaje się za nieudane.
 - 1.1 Druga strona rozmowy rozpoczyna periodyczne co czas T_{zyje} wysyłanie komunikatu **zyje** do pierwszej strony. W przypadku braku poprawnej odpowiedzi XML-RPC wysyła do pierwszej strony komunikat **zakonczenie**, uznaje połączenie za zakończone. Sprawdzenie obecności pierwszej strony druga strona kończy także w przypadku odebrania komunikatu **rozmawiamy**.
- 2 Jeżeli jest członkiem grupy wieloabonentowej, wysyła do całej grupy komunikat **zawiesilem**.

3 Abonent zawieszający lub każdy członek grupy po otrzymaniu **zawiesilem(*, n, on, port)** może wysłać do abonenta *on* na port *port* komunikat **rozmawiamy**.

3.1 Wysłanie komunikatu **rozmawiamy** zakończone sukcesem (wartość zwracana równa *prawda*) oznacza przejście kontroli nad linią *n* i zobowiązuje do wysłania do całej grupy komunikatu **rozmawiam**.

3.2 Otrzymanie od członka grupy komunikatu **rozmawiam(*, n, *)** lub **zwalniam(*, n)** wywołuje rezygnację z odwieszenia rozmowy (i jest odpowiednio obsługiwane).

Kończenie połączenia

Zakończyć połączenie nie zawieszzone może każda strona rozmawiająca, a strona, która nie zawieszała, może zakończyć połączenie zawieszzone. Aby zakończyć połączenie:

1 Abonent wysyła komunikat **zakonczenie** na port *port* wskazany w ostatnio otrzymanym komunikacie **rozmawiamy(*, port)**.

1.1 Po otrzymaniu komunikatu **zakonczenie** abonent zwalnia linię WLT, na której otrzymał ten komunikat. Jeżeli należy do grupy, wysyła do wszystkich komunikat **zwalniam**.

Włączanie się abonenta z grupy

Nowo włączający się abonent nie będący członkiem grupy wieloabonentowej bez wymiany komunikatów wie, że wszystkie jego WLT są wolne. Abonent z grupy wieloabonentowej włączając się:

1 Jeżeli był włączony w ciągu czasu T_{nie_zyje} , używał jakichś WLT i nie jest pewien, że zwolnił je przed wyłączeniem się, przed podłączeniem do sieci IP odczeka czas T_{nie_zyje} .

2 Rozpoczyna nasłuchiwanie na swoim porcie głównym, jednak na wszystkie komunikaty wysyła odpowiedzi puste (niezgodne z protokołem XML-RPC). Wysyłanie tych odpowiedzi także podlega synchronizacji jak opisano w „Komunikaty używane przy zajmowaniu WLT”.

3 Wysyła do kolejnych członków grupy w kolejności alfabetycznej rosnących identyfikatorów abonenta komunikat **stan_linii** aż do otrzymania odpowiedzi pozytywnej.

3.1 Jeżeli nie otrzymał żadnej odpowiedzi, uznaje, że wszystkie linie są wolne a inni członkowie grupy nie są podłączeni do sieci. Jeżeli to założenie jest w sprzeczności z otrzymywanymi komunikatami **zyje**, abonent rozłącza się i zapisuje odpowiednią informację do logu błędów, jeżeli ten istnieje.

3.2 Jeżeli na komunikat **stan_linii** otrzyma w odpowiedzi **błąd** lub cokolwiek innego od poprawnej odpowiedzi, rozpoczyna pkt. 2. od początku.

3.3 Jeżeli w międzyczasie otrzyma dowolny komunikat od abonenta, do którego wcześniej wysłał komunikat **stan_linii** nie otrzymując żadnej odpowiedzi, rozpoczyna pkt. 2. od początku.

3.4 Jeżeli w odpowiedzi na **stan_linii** otrzymał informacje, że zajmuje jakąś linię, rozłącza się i zapisuje odpowiednią informację do logu błędów, jeżeli ten istnieje. Odczeka czas

T_{nie_zyje} , rozpoczyna pkt. 2. od początku.

4 Atomowo: stan linii zwrócony przez **stan_linii** kopiuje do swojej tablicy stanu linii, rozpoczyna poprawne odpowiadanie na komunikaty nadchodzące na jego porcie głównym (atomowo z punktu widzenia synchronizacji opisanej w „Komunikaty używane przy zajmowaniu WLT”).

4.1 Jeżeli wszystkie komunikaty **stan_linii** nie doczekały się odpowiedzi i abonent w międzyczasie nie otrzymał żadnych innych komunikatów od swojej grupy, uznaje wszystkie linie za wolne i rozpoczyna poprawne odpowiadanie na komunikaty przychodzące na jego porcie głównym.

5 Od tego momentu abonent normalnie obsługuje komunikaty przychodzące, wysyła komunikaty **zyje**, może ubiegać się o dostęp do linii.

Wyłączanie się abonenta z grupy

Abonent odłączający się (kończący działanie) powinien przed zakończeniem działania zwolnić wszystkie zajmowane przez siebie WLT. Następnie przestaje odpowiadać na wszelkie komunikaty, zamyka port główny i porty kanałów sterowania.

Odzyskiwanie WLT w grupie

Każdy członek grupy przechowuje informację o stanie wszystkich WLT grupy, w szczególności wie, który członek grupy których linii używa. Jeżeli przez czas T_{nie_zyje} abonent *A* nie otrzyma od abonenta *B* żadnego komunikatu **zyje**, oznacza wszystkie linie używane przez *B* jako wolne, ale jeszcze przez czas T_{nie_uzywam} nie używa tych linii w kroku 1. algorytmu „Zajmowanie WLT w grupie”, tj. nie próbuje z tych linii korzystać.

Historia zmian

1. między wersją pierwszą a drugą
 - opis stałych przyjętych w protokole wraz z uzasadnieniem
 - argumentacja dla bazowania protokołu centralki na protokołach UDP i XML-RPC
 - uzupełnienie protokołu o mechanizm autoryzacji
 - zmiana formatu adresów IPv6
 - rozwiązanie problemu obustronnego zawieszania połączenia