

Igor Margasiński  
Instytut Telekomunikacji  
Politechnika Warszawska  
E-mail: I.Margasinski@tele.pw.edu.pl

# Prywatność w systemach P2P

W referacie podjęto dyskusję na temat powszechnych protokołów peer-to-peer w zakresie prywatności ich użytkowników. Przeanalizowano prywatność w popularnych aplikacjach: od implementacji prostych wariantów hybrydowych – do systemów P2P oferujących anonimowość. Zaprezentowano słabe punkty oraz nakreślono ewolucję zabezpieczeń systemów P2P i kierunki rozwoju protokołów w kontekście wzrastającego znaczenia tego typu aplikacji.

*Wśród wielkoludów staraj się być kartem,  
wśród kartów wielkoludem,  
ale wśród sobie równych staraj się być im równy.*

– Stanisław Jerzy Lec

## 1. Wstęp

Sieć Internet zawdzięcza swój unikalny sukces otwartości. Dzięki zdecentralizowanemu modelowi architektury sieci zyskujemy środek przekazu, który niwelując wpływ położenia geograficznego, finansowych czy formalnych ograniczeń, pozwala na swobodną wymianę informacji, stanowi katalizator dla rozwoju społeczeństwa informacyjnego. W wymiarze aplikacyjnym często jednak dystrybucja danych jest wciąż realizowana w architekturze scentralizowanej. Usługa WWW, stanowiąc trzon aplikacyjny Internetu, utrwaliła wśród użytkowników postrzeganie największej sieci rozległej przez pryzmat modelu klient-serwer, gdzie wyróżnione węzły sieci kontrolują dostarczane treści. W konsekwencji niespotykanej skali rozwoju i globalizacji, Internet jest utożsamiany dziś z wpływowym środkiem przekazu. Ceną jest ograniczanie jego niezależności. Na tym gruncie powstają nowe usługi niepoddające się kontroli czynników komercyjnych. Popularyzacja szerokopasmowego dostępu do Internetu ostatecznie przełamała praktyczne ograniczenia hamujące rozwój zdecentralizowanych aplikacji sieciowych. Sygnałem potwierdzającym ten kierunek jest migracja samej usługi WWW, która w obecnej formule wydaje się być wyczerpana. Obserwujemy decentralizację WWW i budowę serwisów w formie kompilacji wielu niezależnych źródeł – zjawisko Web 2.0. Jednak **prawdziwym motorem zmian są nowe aplikacje sieciowe peer-to-peer**. Obecnie wolumen ruchu P2P w sieci Internet stanowi ok. 80%. W opracowaniu będziemy zajmowali się właśnie tym popularnym nurtem – tzw. nakładkowymi sieciami zdecentralizowanymi (*Decentralized Overlay Networks*) [1], potocznie nazywanymi *sieciami peer-to-peer* (*peer* – dosłownie: *równy, rówieśnik, szlachcic*).

## 2. Słabe punkty

**Architektura P2P** – kusząc swoją niezawodnością, skalowalnością, niskimi kosztami, a szerzej wolnością wymiany informacji – nie pozostaje pod kontrolą jakichkolwiek pojedynczych ośrodków. Nasuwa się jednak obawa o bezpieczeństwo, związana z niejasnym umiejscowieniem funkcji administracyjnych. Groźne wydaje się rozproszenie – a często w praktyce eliminacja –

odpowiedzialności za bezpieczeństwo. Przyjmując czysty model P2P pozwalamy by funkcje administracyjne sprawowali poszczególni użytkownicy jej węzłów. Osoby takie nie posiadają zazwyczaj odpowiednich kwalifikacji, narażając siebie oraz pozostałych uczestników wymiany na niebezpieczeństwo. Rozważając zadanie dystrybucji danych, warto powrócić to do porównania z systemem WWW. Tu rola administratora jest traktowana niezwykle poważnie, a zarządzanie bezpieczeństwem informacji w architekturze klient-serwer jest znacznie prostsze. W architekturze zdecentralizowanej – zdecentralizowane jest również zaufanie. Należy więc ze szczególną uwagą projektować protokoły aplikacyjne P2P, dbając by kompromitacja poszczególnych węzłów nie wpływała znacząco na jakość i bezpieczeństwo usług.

Dodatkowo **aplikacje P2P**, często utożsamiane z *podziemiem* Internetu, balansują na krawędzi legalności. Ta *niezależność* jest w określonych środowiskach (np. w sieciach korporacyjnych) jednoznacznie odbierana jako funkcja dywersyjna. W efekcie, z niebezpieczeństwem są utożsamiane same aplikacje P2P. Jest to często praktycznie uzasadnione. Rozwiązania budowane niejednokrotnie w duchu działań hobbystycznych, często posiadają liczne podatności oraz niosą zagrożenia związane z występowaniem ukrytej funkcjonalności. Należy zwrócić uwagę, że aplikacje P2P *nakładają (Overlaying)* na sieć TCP/IP odrębny, specyficzny model sieci (*Overlay Network*), dodając w warstwie aplikacji m.in. funkcjonalność routingu. *Wirtualna sieć*, nałożoną na sieć fizyczną, posiada bardzo szerokie możliwości.

\* \* \*

Wskazane słabe punkty mogą stanowić tylną furtkę do prywatności użytkowników systemów P2P. Czy ich wyeliminowanie wystarczy, by popularne schematy P2P mogły stanowić odpowiedni środek pozwalający na wolność wypowiedzi i wymianę informacji bez groźby śledzenia? Czy P2P mogą być odpowiednim środowiskiem zapewniającym prywatność?

### 3. Ewolucja zabezpieczeń prywatności

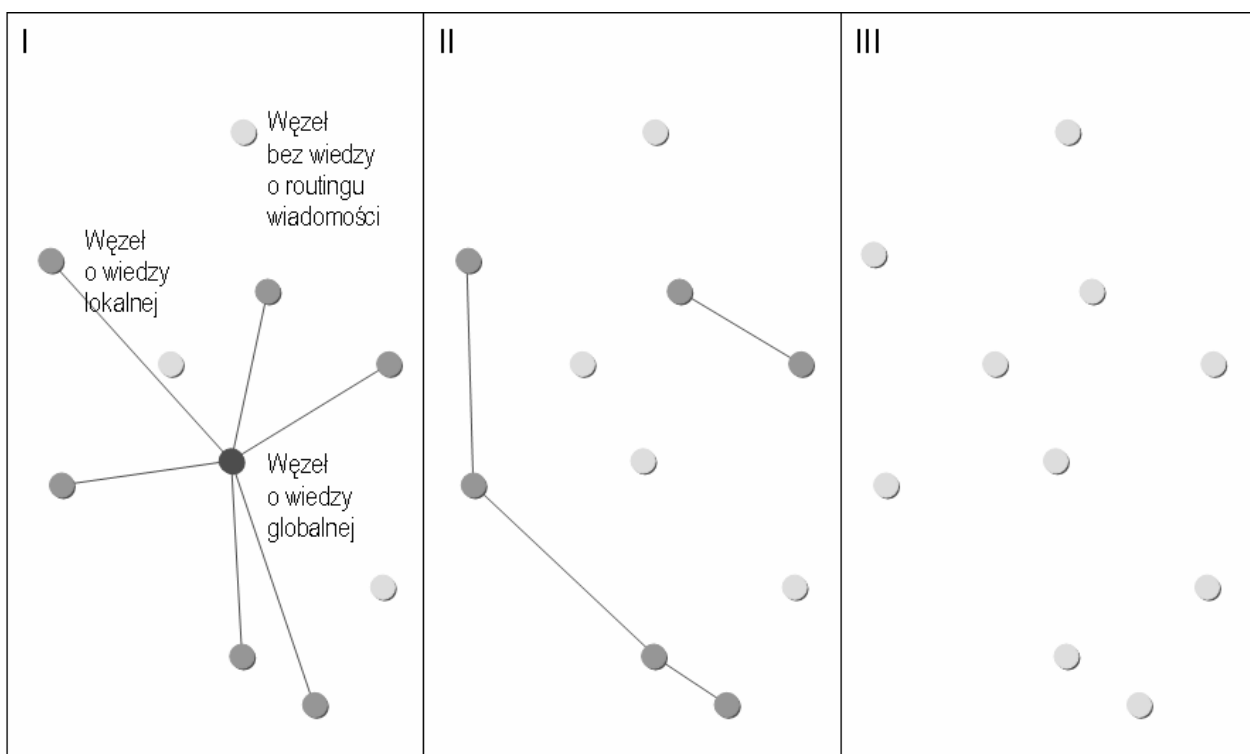
Rozwój popularnych systemów P2P wiąże się bezpośrednio z prywatnością węzłów – *peer-ów* i ich użytkowników. Obserwując rozwój popularnych systemów P2P pod kątem prywatności, można wyróżnić trzy podstawowe klasy (Rysunek 1):

- I. *Hybrydowe P2P* – (łącznie cechy architektury *peer-to-peer* i klient-serwer) – działające w oparciu o węzeł centralny.
- II. *Czyste P2P* – bez hierarchii węzłów.
- III. *Anonimowe P2P* – dedykowane do zapewniania anonimowości (a dokładniej pseudoanonimowość) nadawcy i odbiorcy.

W systemach P2P pierwszej generacji występował centralny węzeł, bezpośrednio narażony na śledzenie jego aktywności i tym samym aktywności wszystkich pozostałych węzłów. Po serii procesów sądowych w Stanach Zjednoczonych o naruszanie praw autorskich właścicieli dystrybuowanych treści, najbardziej rozpowszechniona aplikacja tej architektury (*Napster*) została wycofana z użytku publicznego.

Naturalnym kierunkiem rozwoju było wyrównywanie roli poszczególnych *peer-ów* (np. w systemie *Gnutella* [8] wszystkie *peer-y* posiadają równe prawa). W tym wariantcie nie istnieje już pojedynczy punkt mogący stanowić miejsce śledzenia globalnej aktywności w sieci. Możliwe jest jednak śledzenie poszczególnych użytkowników systemu.

Kolejna klasa systemów P2P ma eliminować również tę słabość. W praktyce zazwyczaj poprzez wprowadzenie koncepcji węzłów pośredniczących [4]. Poszczególne *peer-y*, stanowiąc proxy, budują anonimowość nadających i odbierających *peer-ów* (np. system *Freenet* [5]).



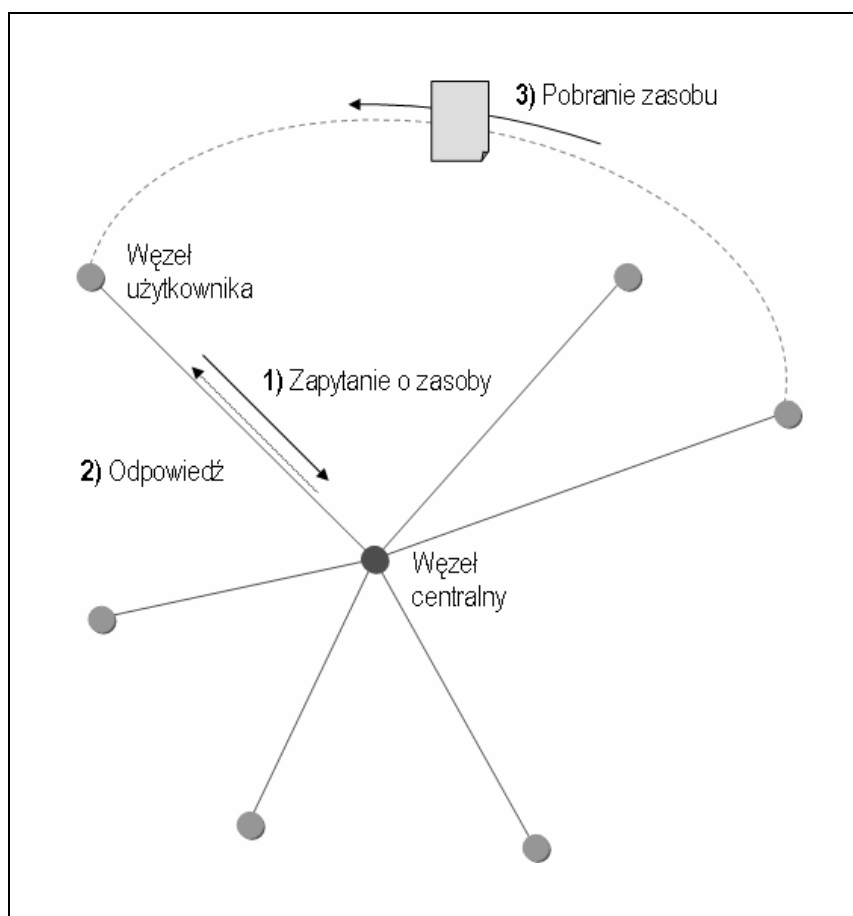
Rys. 1. Klasy systemów P2P z uwzględnieniem możliwości śledzenia aktywności użytkowników.

### 1.1. Hybrydowe P2P

Boom dystrybucji danych (głównie plików multimedialnych) za pomocą systemów P2P zapoczątkowały rozwiązania hybrydowe. Zakładają one wyjątkowo prostą architekturę. Tam gdzie to wygodne, opierają się na komunikacji klient-serwer (najczęściej w procesie wyszukiwania węzłów i zasobów). Tam gdzie obecność serwera nie jest już potrzebna, następuje komunikacja *peer-to-peer*, pomiędzy równouprawnionymi węzłami (zazwyczaj przy wymianie danych).

Prześledźmy sposób działania tego wariantu na przykładzie do niedawna wpływowego systemu P2P *współdzielenia plików – Napster*. W systemie *Napster* wszyscy użytkownicy łączą się z węzłem centralnym. Zadaniem węzła centralnego jest prowadzenie i dystrybucja aktualnej listy węzłów sieci. Dodatkowo wprowadzono funkcjonalność komunikatora – centralny węzeł stanowi tzw. *chat room*. Podstawą systemu jest protokół działający w architekturze klient-serwer z wykorzystaniem transportu TCP. Użytkownicy za pośrednictwem odpowiedniego oprogramowania klienckiego łączą się z serwerem, gdzie mogą dokonać rejestracji (podając dane osobowe). Administrator węzła centralnego może dokonać odrzucenia (*ban*) poszczególnych użytkowników. Klienci informują serwer o posiadanych zasobach i wystosowują o nie zapytania. Serwer przekazuje klientowi adres węzła z dostępnymi zasobami. Następnie użytkownik łączy się bezpośrednio z tym węzłem, który posiada żądany plik (w oryginalnej implementacji tylko typu MP3). Jednocześnie *peer-y* przesyłające plik powiadamiają węzeł centralny o swoim aktualnym stanie, by zapewnić podstawowe sterowanie przepływem danych w sieci. Węzły deklarują ograniczenia w szerokości pasma dedykowanego do wymiany P2P. Na tej podstawie węzeł centralny podejmuje decyzję o wskazaniu najbardziej odpowiedniego węzła posiadającego żądane zasoby. Oprogramowanie klienckie może wykorzystywać dowolny port TCP. Dodatkowo

wprowadzono tzw. tryb pasywny (działający podobnie jak analogiczny wariant protokołu FTP) umożliwiający obejście starszych systemów ścian przeciwogniowych (*firewall bypass*).



Rys. 2. Przykład działania hybrydowego systemu P2P.

\* \* \*

Zaprezentowany schemat jest bardzo prosty i jednocześnie skuteczny w działaniu. Skuteczność, oparta na połączeniu architektury klient-serwer i *peer-to-peer*, okazała się kluczowa w popularyzacji aplikacji sieciowych P2P. Jednak schemat szybko okazał się niepraktyczny ze względu na brak prywatności. Węzeł centralny stanowi punkt gromadzący informacje o wszystkich uczestnikach wymiany danych. Skupia dane o adresach poszczególnych węzłów, wyszukiwanych hasłach oraz o pobieranych i udostępnianych zasobach. Wariant pomimo powszechnego zakwalifikowania do grupy aplikacji *peer-to-peer*, zależy od usług serwera (lub serwerów) i działa pod jego pełną kontrolą. Ze wszystkimi tego konsekwencjami – włączając niemożność przetrwania.

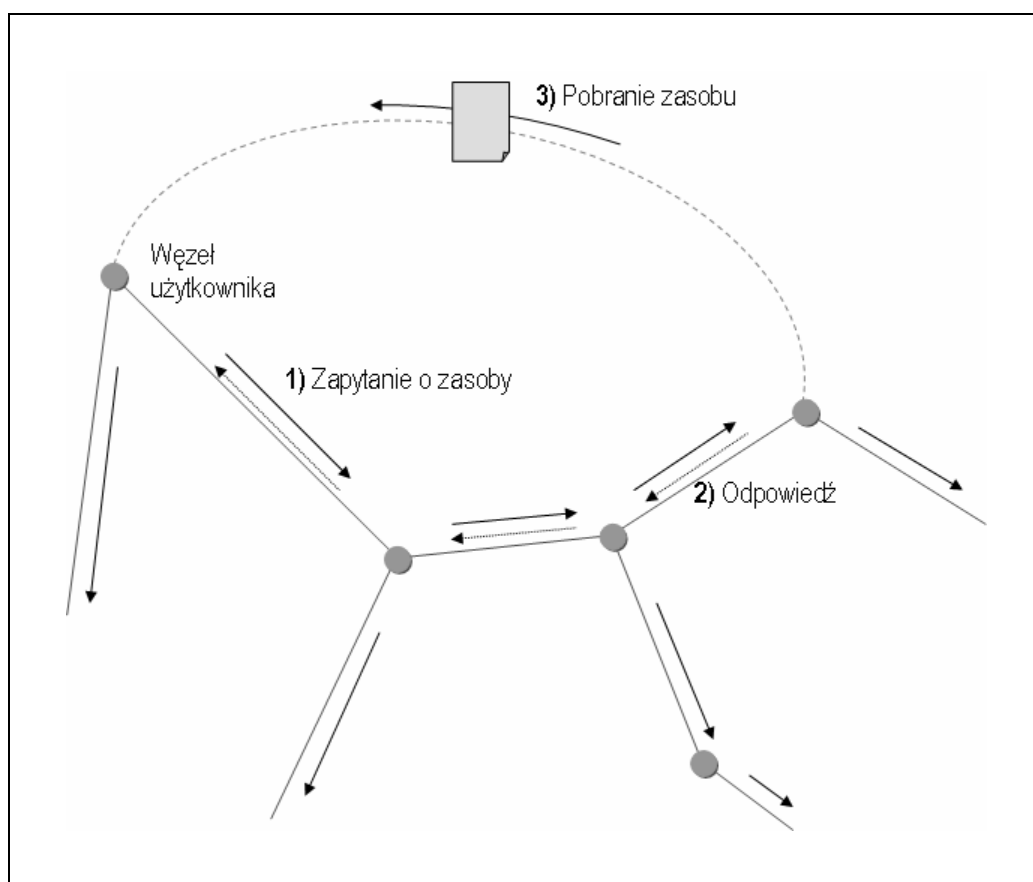
## 1.2. Czyste P2P

Sposób działania tej klasy aplikacji P2P można prześledzić na przykładzie otwartego (ogólnie dostępnego) protokołu *Gnutella*. W systemie *Gnutella* nie ma wyróżnionego węzła centralnego, wszystkie węzły posiadają taką samą funkcjonalność. Sieć nie zawiera również repozytorium z listą węzłów, zasobów, itp. Nowy uczestnik, aby dołączyć do sieci, musi znać przynajmniej jeden adres IP lub przynajmniej jedną nazwę hosta węzła sieci *Gnutella*. Dopiero wtedy będzie mógł wystosować zapytanie, które zostanie sukcesywnie przekazane do kolejnych węzłów. Węzły, które posiadają żądane zasoby prześlą odpowiedzi zwrotne. Ostatecznie plik zostanie pobrany od jednego z *peer-ów*.

Protokół *Gnutella* oparto na pięciu wiadomościach:

- Ping i Pong – wyszukiwanie węzłów w sieci,
- Query i Query hits – wyszukiwanie zasobów,
- Push – pobranie zasobu (w przypadku komunikacji z węzłami za firewall-em).

Wszystkie wiadomości zawierają pole TTL (*Time to Live*), które wraz z identyfikatorem wiadomości pozwala na eliminację pętli w sieci.



Rys. 3. Przykład działania czystego systemu P2P.

Rysunek 3 obrazuje proces wyszukiwania węzłów posiadających interesujące zasoby (wiadomości Query i Query hit). Wyszukiwanie węzłów (wiadomości Ping i Pong) zachodzi analogicznie (oczywiście z pominięciem kroku 3). Przesyłanie zasobów (krok 3) realizowane jest w oparciu o protokołu HTTP (*HyperText Transfer Protocol*).

\* \* \*

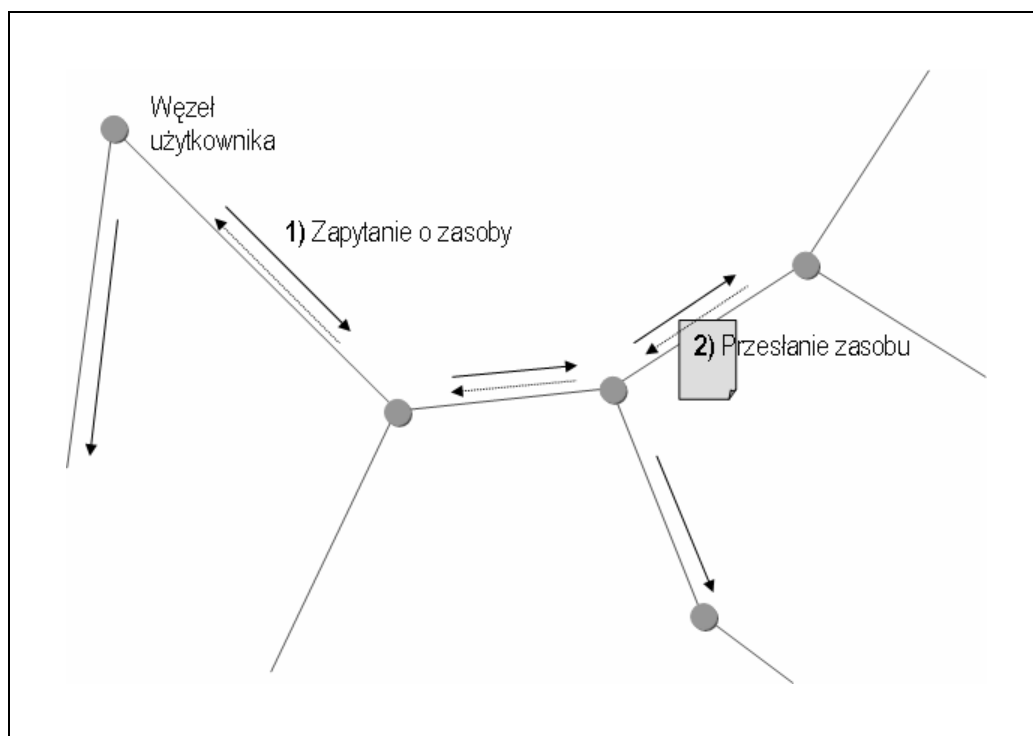
Schematy działające w czystej architekturze *peer-to-peer* znacznie utrudniają globalne śledzenie użytkowników sieci. Dodatkowo uzyskujemy sieć bardziej niezawodną, bo odporną na awarie poszczególnych węzłów (*Single Point of Failure Resistance*).

Przy zastosowaniu tak prostych protokołów jak *Gnutella* pojawiają się jednak problemy związane ze skalowalnością, odpornością na ataki typu odmowa usługi (*DOS – Denial of Service*) oraz wydajnością dodatkowo ograniczoną przez dużą nadmiarowość ruchu (wiadomości ping i query). Opracowano oczywiście już szereg bardziej praktycznych schematów, które w większym lub mniejszym stopniu stanowią powrót do wariantu hybrydowego, np. *eDonkey*, *FastTrack*

(Kazaa). Sieci takie nie posiadają czystej struktury P2P. Opierają się zazwyczaj na strukturze hierarchicznej, gdzie wyróżnia się uprzywilejowane węzły o dodatkowej funkcjonalności (zwane np. *super węzłami* – *super nodes*). Jednak sieci te również nie mogą zapewniać pełnej swobody wypowiedzi, ponieważ nie chronią przed śledzeniem aktywności poszczególnych węzłów.

### 1.3. Anonimowe P2P

Prześledźmy sposób działania tej klasy systemów P2P na przykładzie jednego z najpopularniejszych (rozpowszechnionego w ponad 2 mln. kopii) działających publicznie systemów – *Freenet*. System *Freenet* jest przeznaczony do zapewniania anonimowości stronie publikującej i odbiorczej. Anonimowość zapewniana jest poprzez wzajemne przekazywanie zapytań pomiędzy poszczególnymi węzłami sieci by ostatecznie, na bazie metod heurystycznych, dostarczyć zapytanie do węzła docelowego. Transmisja pomiędzy węzłami jest szyfrowana. Podobnie jak w systemie *Gnutella* zastosowano pole TTL do ograniczenia zakresu propagacji zapytań w sieci.



Rys. 4. Przykład działania anonimowego systemu P2P.

Każdy dokument jest przechowywany w kilku węzłach sieci. Użytkownicy nie mogą decydować o zawartości przechowywanych przez siebie dokumentów. Pliki są indeksowane za pomocą funkcji skrótu, tzw. kluczy *content-hash keys*. W systemie *Freenet* wyróżniono specjalną usługę odpowiadającą za korelację funkcji skrótu z nazwami dokumentów. Każdy węzeł sieci utrzymuje tablicę routingu zawierającą informacje o drogach do kilku innych węzłów. Tablica routingu zawiera również listę kluczy zasobów, które prawdopodobnie mogą być dostarczone przez węzły sąsiadujące. Aby odnaleźć dokument, węzeł wysyła zapytanie do sąsiada z najbardziej podobnym kluczem. Zapytanie przekazywane jest dalej, do momentu odnalezienia zasobu (Rys. 4). W przypadku wykrycia pętli, zapytanie jest przekazywane do kolejnego węzła o najbardziej zbliżonym kluczu. Po pomyślnie przekazaniu zapytaniu tablice routingu są uaktualniane. Gdy wielkość tablicy routingu przekroczy dopuszczalną pojemność danego węzła, usuwane są najmniej popularne wpisy. Dokument jest przesyłany do zainteresowanej strony ścieżką, którą przekazywane było zapytanie. Węzły pośredniczące kopiują lokalnie dokument zapewniając funkcję *cache-u*.

P2P anonimowe mają stanowić odpowiedź na obecne oczekiwania z zakresu prywatności dystrybucji danych. Proponowane rozwiązania jednak wciąż borykają się z problemem wydajności. Zastosowanie *peer-ów* (zazwyczaj domowych komputerów osobistych) jako węzłów pośredniczących w warstwie aplikacji przy przesyłaniu dokumentów, stanowi poważną barierę prędkości transmisji. Wciąż opracowywane są rozwiązania mające poprawić efektywność dystrybucji danych w anonimowych P2P. Dla systemu *Freenet* jest to nowy protokół routingu aplikacyjnego, nazwany algorytmem nowej generacji – *next generation routing algorithm*. Istotna poprawa wydajności dokonywana jest tam jednak kosztem samej anonimowości... Co więcej, anonimowość w systemie *Freenet* nawet bez modyfikacji routingu jest poważnie kwestionowana. W systemie *Freenet* brak ścisłych ram, w jakich może formułować się struktura sieci oraz niedeterministyczny przebieg nawiązywania połączeń wiąże się z brakiem gwarancji, co do poziomu anonimowości. Opierając się na wynikach symulacji z [3] schemat zapewnia usługę anonimowości jedynie w pewnej liczbie przypadków. Nie należy, zatem traktować proponowanej w systemie usługi anonimowości za niezawodną.

#### 4. Przyszłość

Omówione systemy należą do tzw. niestrukturalnych sieci nakładkowych (*Unstructured Overlay Networks*). Obecnie, stanowią trzon powszechnie użytkowanych implementacji. Niestrukturalne rozwiązania *ad-hoc* cechują się brakiem matematycznych podstaw w zakresie relacji połączeń dokonywanych w sieci. Relacje, w jakich następuje połączenie, są bezpośrednio podporządkowane żądaniom użytkowników. Jednak główny nurt prowadzonych obecnie badań jest skierowany w innym kierunku – dotyczy projektowania rozwiązań strukturalnych (*Structured Networks*). Szereg propozycji takich jak *Chord* [7], *Pastry* [6], *Tapestry* [9] bazuje na matematycznych podstawach formułowania struktury sieci. Wykorzystuje się między innymi takie struktury jak drzewa PRR, grafy de Bruijna, Butterfly, Hypercube. Przypuszcza się, że ten kierunek pozwoli nie tylko na stworzenie wydajnej i niezawodnej sieci, ale jest również rozpatrywany w świetle szerokiej możliwości zapewniania anonimowości. Uważa się, że jasno sformułowane zasady przebiegu komunikacji w sieci zapewnią niezawodność w dostarczaniu usługi anonimowości, będącej podstawowym środkiem do elektronicznej prywatności.

#### 5. Podsumowanie

P2P jest dziś zjawiskiem o skali, której nie można ignorować. Możliwość prywatnej i pozbawionej cenzury komunikacji jest obecnie szczególnie istotna. Poszukiwane są rozwiązania, pozwalające na prywatne funkcjonowanie w otaczającym nas elektronicznym świecie. Sieci telekomunikacyjne, jak nigdy dotąd, pozwalają na szybką i wszechobecną wymianę informacji. Jednak również, jak nigdy dotąd, ogromny zakres danych, *logów*, ścieżek aktywności, jest automatycznie gromadzony i analizowany. Należy wierzyć, że możliwość wolnego dzielenia się informacją, może mieć miejsce we współczesnym, wspartym na telekomunikacji, świecie. Głównym obszarem otwierającym się na możliwości swobodnej wypowiedzi są nakładkowe sieci zdecentralizowane – *peer-to-peer*. Początkowo spontaniczność i zakulisowość tworzenia P2P potęgowały zagrożenie występowania luk w protokołach i aplikacjach. Po kilkuletniej ewolucji, ich architektura coraz silniej przypomina sieć niezależnych, ale dobrze organizujących się węzłów symetrycznych. Szeroko stosowane, praktyczne schematy niestrukturalne wciąż nie chronią w pełni prywatności ich użytkowników. Rozwiązania teoretycznie bezpieczne, opierające się na popartych aparatem matematycznym sieciach strukturalnych P2P, wciąż nie mogą przejść sprawdzianu praktycznego. Należy mieć jednak nadzieję, że wkrótce ten nurt wydobędzie się z zacisza laboratoriów, wkraczając do życia codziennego.

## Literatura

1. Aberer, K., Alima, L. O., Ghodsi, A., Girdzijauskas, S., Hauswirth, M., Haridi, S. The essence of P2P: A reference architecture for overlay networks. In: 5th IEEE International Conference on Peer-to-Peer Computing, 2005.
2. Adar, E., Huberman, B.A. Free riding on Gnutella. *First Monday*, 5(10), 2000.
3. Borisov, N. Anonymous Routing in Structured Peer-to-Peer Overlays. PhD Thesis, UC Berkeley, 2005.
4. Chaum, D. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 1981.
5. Clarke, I., Sandberg, O., Wiley, B., Hong, T. W. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, number 2009 in LNCS, 2001.
6. Rowstron, A., Druschel, P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, 2001.
7. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In *ACM SIGCOMM*, 2001.
8. The Gnutella Protocol Specification. 2001.
9. Zhao, B. Y., Kubiatowicz, J., Joseph, A. D. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley, 2001.