

# Systemy konferencyjne.

Transmisja multimediiów w sieciach IP.

Obraz i dźwięk.

mgr inż. Łukasz Kosikowski

# Systemy konferencyjne

- Konferencje telefoniczne (*ang. Call conferences*)
- Konferencje internetowe (*ang. Web conferences*)
- Konferencje hybrydowe

# Wstęp

... do psychofizjologii percepcji 🤪

– Opóźnienie (akceptowalne)

- W aplikacjach wymagających interakcji człowieka <150ms (4,5 ramki dla 30fps)
  - wideokonferencje,
  - gry online,
  - nadzorowanie dóbr znacznej wartości.
- 30 ms w przypadku:
  - instalacji samochodowych,
  - zastosowań przemysłowych,
  - zastosowań medycznych.

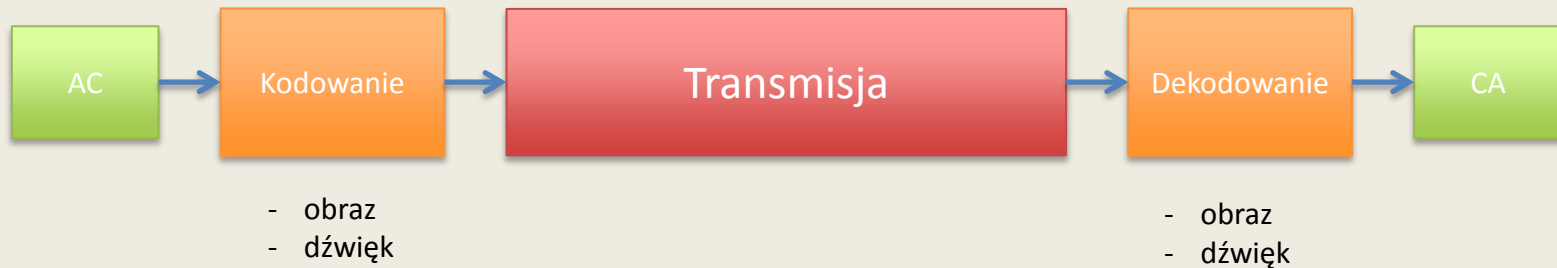
– Echo

- Opóźnienie między sygnałem oryginalnym a odbitym >50ms jest identyfikowane jako echo.

– Przerwy w transmisji.

# Tor komunikacyjny

wybrane elementy

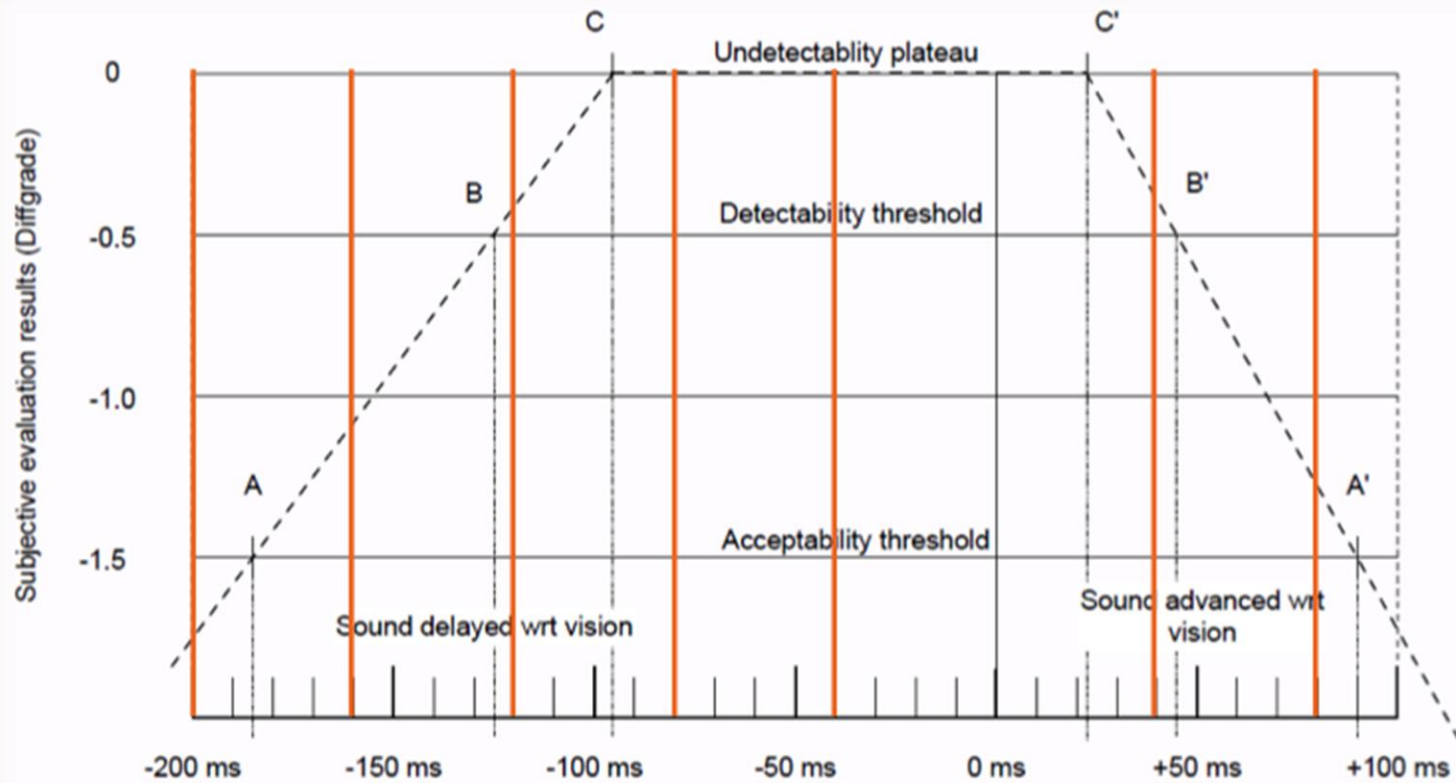


- Jakie opóźnienie wprowadza kodowanie i dekodowanie (coder/decoder = codec)?
- Jakie opóźnienie wprowadza transmisja?

# Opóźnienie dźwięku względem obrazu

- na skutek kompresji dźwięku i obrazu, a także dodatkowych etapów przetwarzania może pojawić się przesunięcie między dźwiękiem a obrazem
- wg normy ITU-R BT1359-1 (1998)
  - dopuszczalne wartości przesunięcia między dźwiękiem a obrazem: od +90ms do -185 ms
- wg zalecenia R37 EBU z 2006 roku:
  - -40ms/+60ms na wyjściu nadajnika

# Opóźnienie dźwięku względem obrazu



ITU-R BT.1359 Figure 2

# Dźwięk czy obraz...?

- Co jest ważniejsze, z punktu widzenia użytkowników systemu,
  - dźwięk,
  - czy obraz?

*To zależy czy sześć liczb, które padną w losowaniu Lotto, ktoś chce nam pokazać czy podyktować.*



# Rozmowa czy monolog...?

Wymagania zmieniają się w zależności od sposobu wykorzystania systemu...

- Rozmowa dwóch osób (Meeting)
  - Dopuszczalne, maksymalne opóźnienie 150 ms.
- Spotkanie wielu osób N:N (Meeting)
  - Dopuszczalne, maksymalne opóźnienie 150 ms – pomiędzy każdą parą uczestników.
  - Wymagane zaawansowane zarządzanie transmisją (opóźnieniem).
  - Czasami wymagana:
    - zmiana parametrów kodowania (online),
    - zmiana formatu,
    - zmiana protokołu transmisji (jeśli to możliwe),
    - zmiana konfiguracji (logicznej) sieci szkieletowej służącej do transmisji danych na rozległych obszarach (konfiguracja edge – origin).





# Rozmowa czy monolog...?

- Webinar (1:N)

- Dopuszczalne większe opóźnienie (większa przepływność bitowa).
- Można użyć bardziej wyrafinowanych formatów kompresji wymagających znacznie większych buforów (out/in) – a tym samym zwiększających opóźnienie.
- Zazwyczaj można transmitować obraz i dźwięk o lepszych parametrach.
- Co w sytuacji gdy równolegle wyświetlamy użytkownikom strumień wizyjno-foniczny lektora oraz sterowaną przez lektora prezentację PPT...? – wymagana synchronizacja.
- Co z użytkownikami, którzy mają niską przepustowość łącza?
  - przekodowanie strumienia do mniejszych przepływności,
  - równoległa transmisja strumieni o różnych przepływnościach.



# Jakie kodeki i formaty...

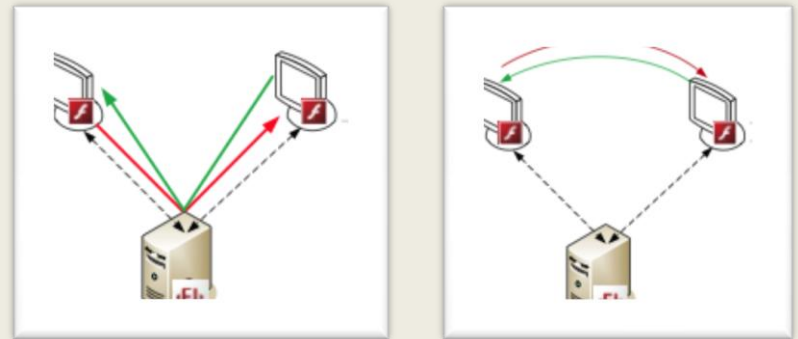
- Powinny być dopasowane do urządzenia/systemu operacyjnego.
- Szczególnie, jeżeli zależy nam na dostępności platformy na urządzeniach mobilnych.
- Np. iOS
  - Pełne wsparcie dla formatu H.264, AAC, Mp3
  - Natywnie wspierany protokół transmisji HLS (HTTP live streaming)
  - Wykorzystanie natywnych komponentów do odtwarzania strumienia foniczno-wizyjnego znacznie obniża zużycie baterii

# H.264/MPEG-4 Part 10 - AVC

- Czy i jaki poziom i profil...?
- Dekodowanie nie jest kosztowne obliczeniowo.
- MPEGLA licencja

# TCP vs UDP

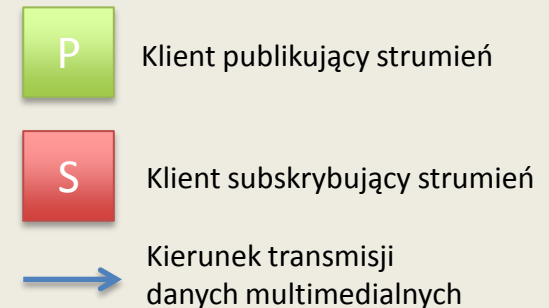
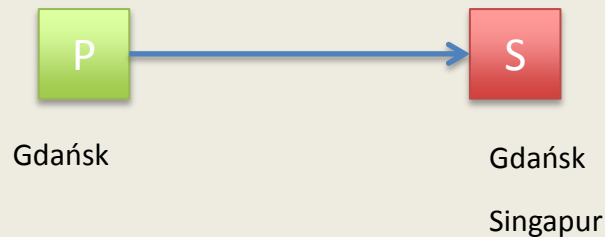
- Protokoły bazujące na UDP (*User Datagram Protocol*) mają znaczną przewagę nad protokołami bazującymi na TCP (*Transmission Control Protocol*):
  - Większa skuteczność dostarczania mediów strumieniowych na żywo,
  - Mniejsze opóźnienia,
  - Zwiększona jakość dźwięku i mowy,
  - Większa niezawodność połączenia.



O tym jeszcze będzie mowa na dalszych wykładach.

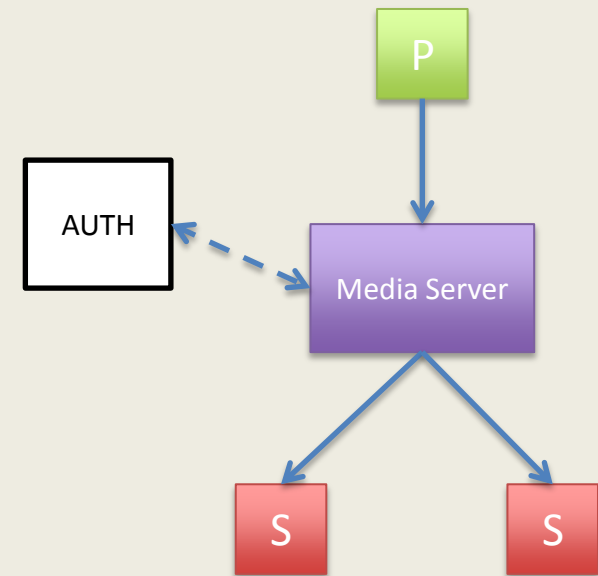
# Cel?

- Dostarczenie strumienia od osoby publikującej (*publisher*) do osoby oglądającej (*subscriber*).



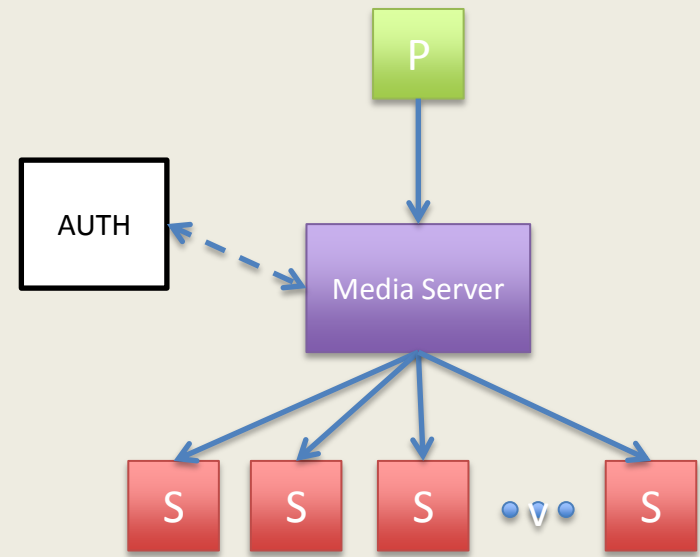
# Publikowanie poprzez serwer – do kilku odbiorców

- Publikujący
  - Wymagane odpowiednie łącze do serwera (up)
- Odbierający:
  - Wymagane odpowiednie łącze z serwera (down)
- Osoby siedzące przy jednym biurku komunikują się przez serwer.
- Dane multimedialne są transmitowane przez serwer.

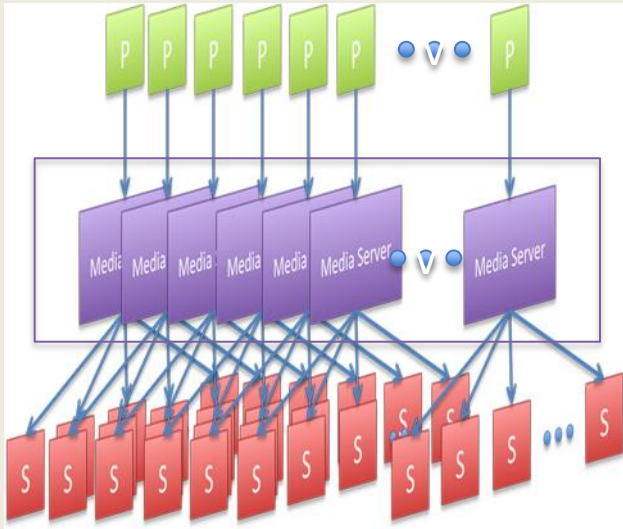


# Publikowanie poprzez serwer – do wielu odbiorców

- Publikujący
  - Wymagane odpowiednie łącze do serwera (up)
- Odbierający:
  - Wymagane odpowiednie łącze z serwera (down)
- Osoby siedzące przy jednym biurku komunikują się poprzez serwer.
- Serwer wymaga dostępu do sieci o znaczącej przepustowości (down).
- Wysyłanie 1 ramki danych od *publisher*a wymaga  $1 \times S_{\text{COUNT}}$  zapisów na socket.



# A co jeśli jest tak...?

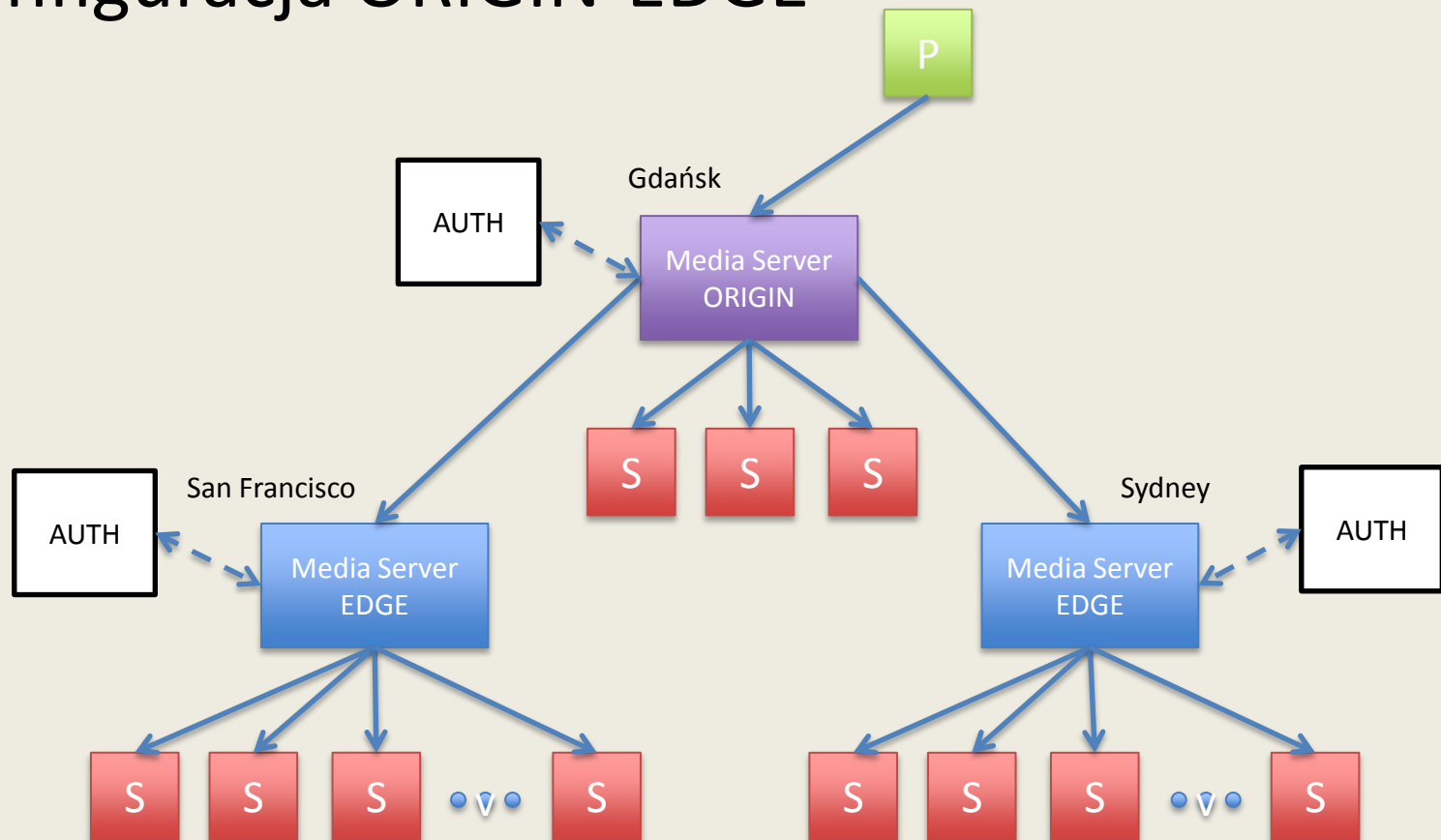


- Co z pasmem dostępnym na serwerze?
- Co ze złożonością obliczeniową?
- Dla protokołu bazującego na TCP, chwilowe zaburzenie w działaniu sieci w której znajduje się serwer (spadek wydajności, wysycenie łącza), może spowodować lawinowy wzrost opóźnień w transmitowanym sygnale ze względu na konieczność retransmisji ramek (wymaganie na przepustowość sieci zostaje dodatkowo, chwilowo zwiększone).



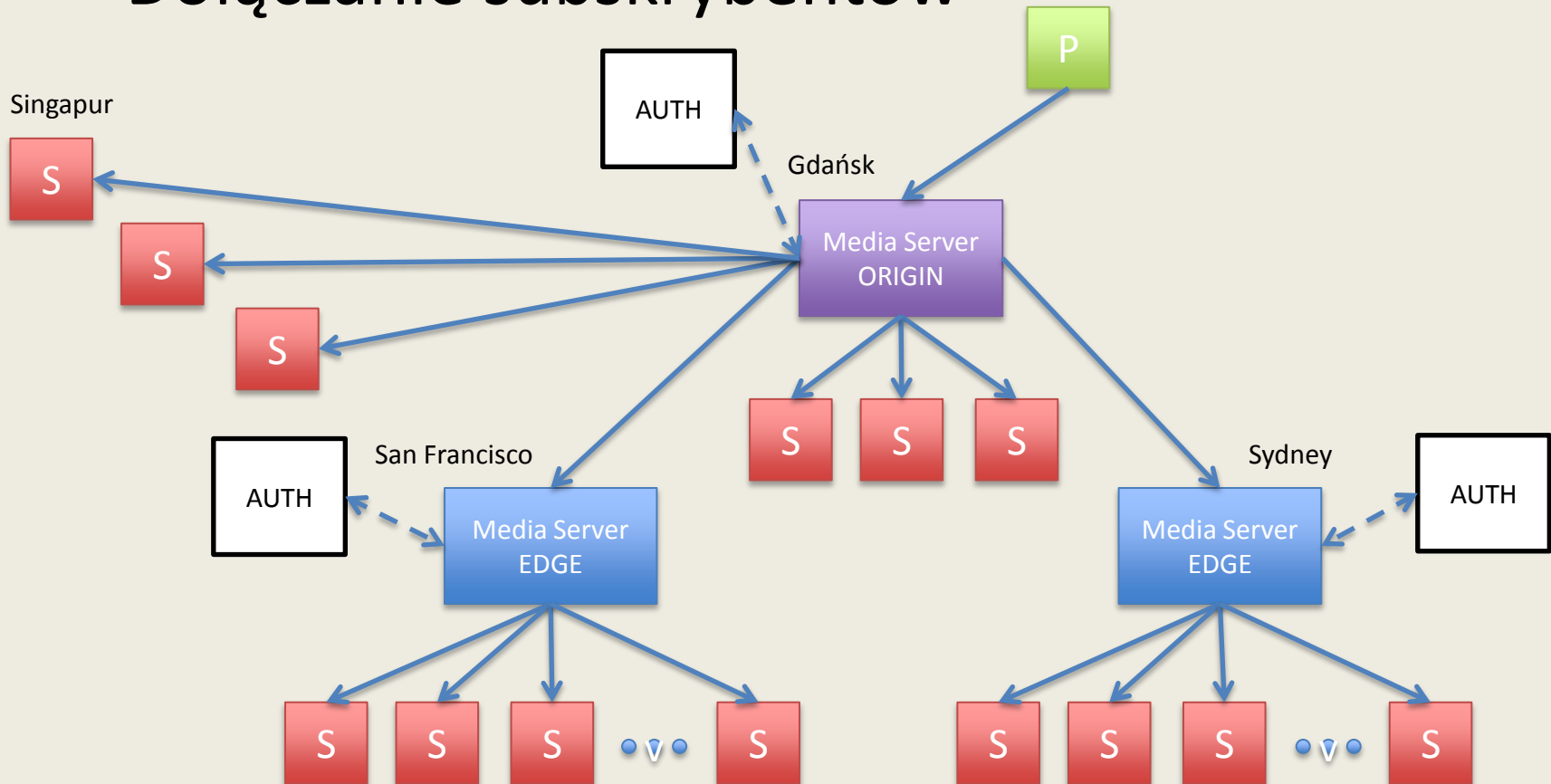
# Transmisja poprzez serwer – duża skala

- Konfiguracja ORIGIN-EDGE



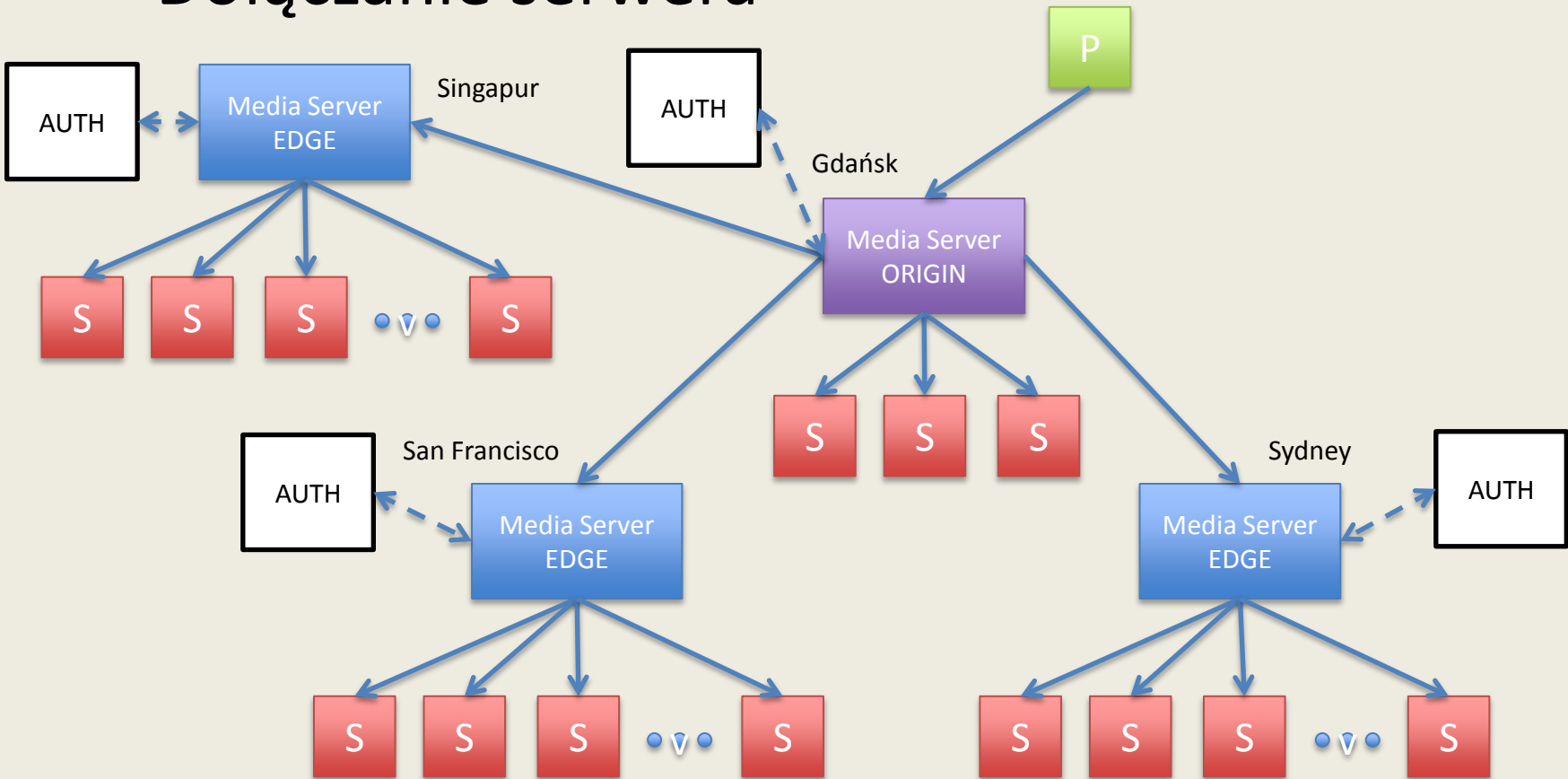
# Transmisja poprzez serwer – duża skala

- Dołączanie subskrybentów



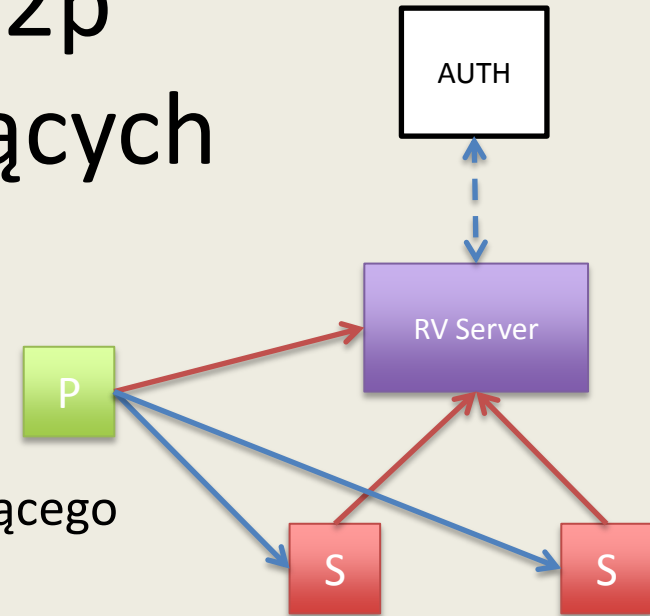
# Transmisja poprzez serwer – duża skala

- Dołączanie serwera



# Publikowanie p2p - do kilku oglądających

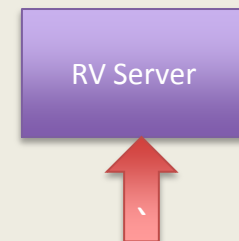
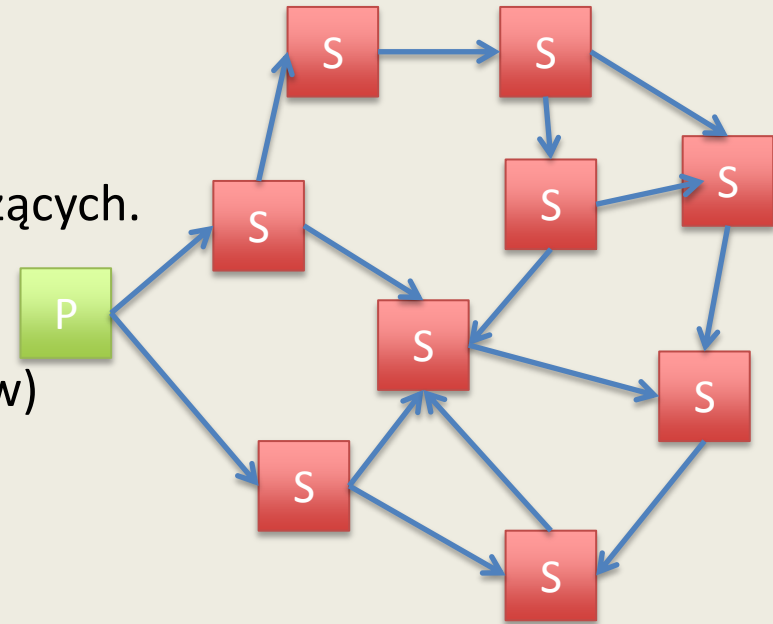
- Publikujący
  - Wymagane odpowiednie łącze:  
 $P_{REQ\_BW} = P_{STREAM\_RATE} \times S_{COUNT}$
- Odbierający:
  - Wymagane odpowiednie pasmo do publikującego
- Potrzebny serwer rendezvous (RVS)
  - Dostarcza stabilnych adresów IP wymaganych do transmisji danych multimedialnych od P do S (pulling from P).
- Minimalizacja opóźnień
- Większe wymagania na łącze publikującego (*up*)
  - Z tego względu nie jest to rozwiązanie dowolnie skalowalne
- Większe wymagania na widoczność w sieci



# Publikowanie do wielu oglądających

## - z wykorzystaniem grup

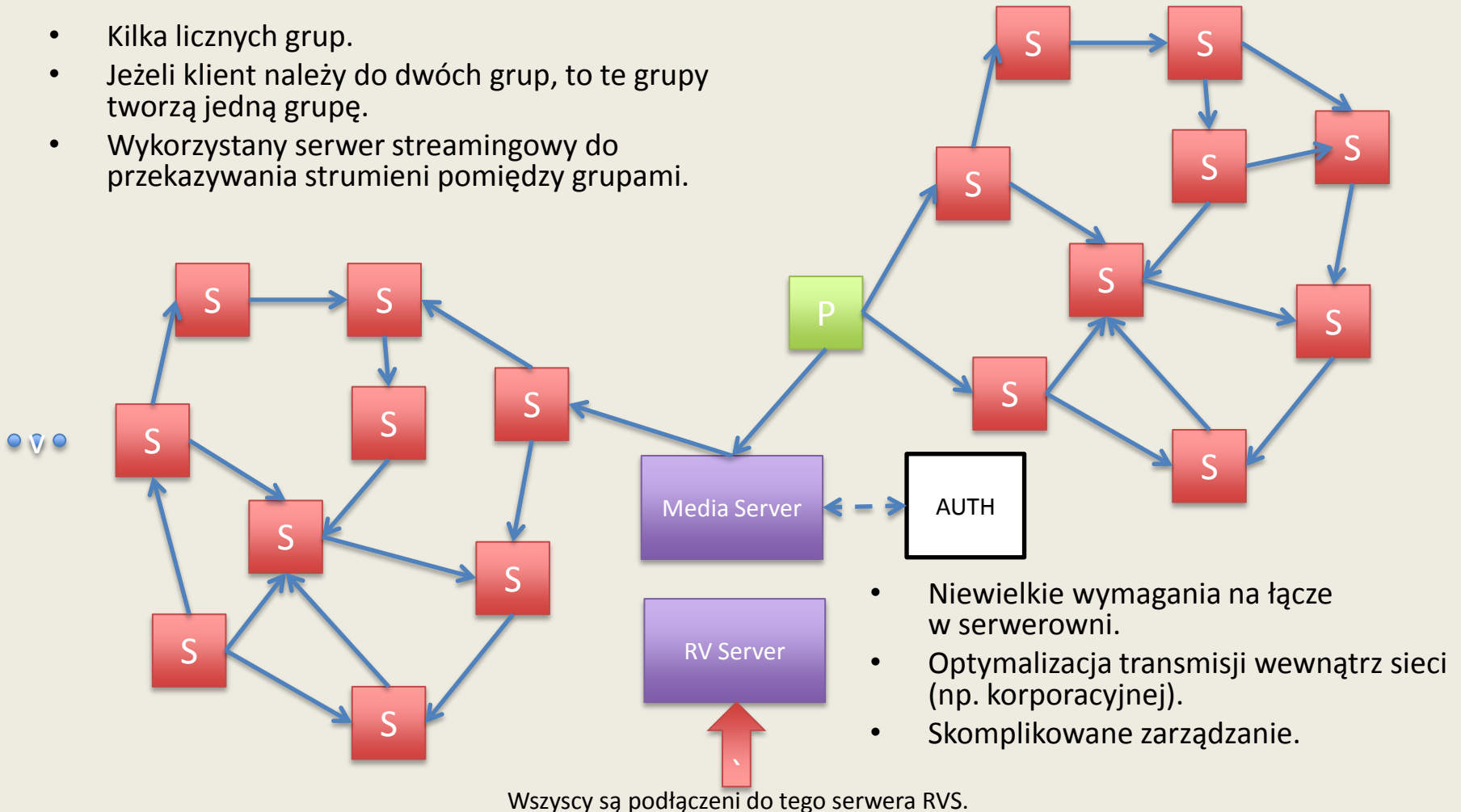
- Publikujący
  - Wymagane odpowiednie łącze.
  - Kontrolowana liczba połączeń wychodzących.
- Odbierający:
  - Wymagane odpowiednie pasmo do publikującego, lub do sąsiada (sąsiadów) w zależności od przyjętego kryterium optymalizacji ruchu wewnątrz grupy.
- Minimalizacja opóźnień.
- Zaawansowane algorytmy optymalizacji połączeń.
- Sterowanie czasem dostępności poszczególnych pakietów wewnątrz sieci (grupy).



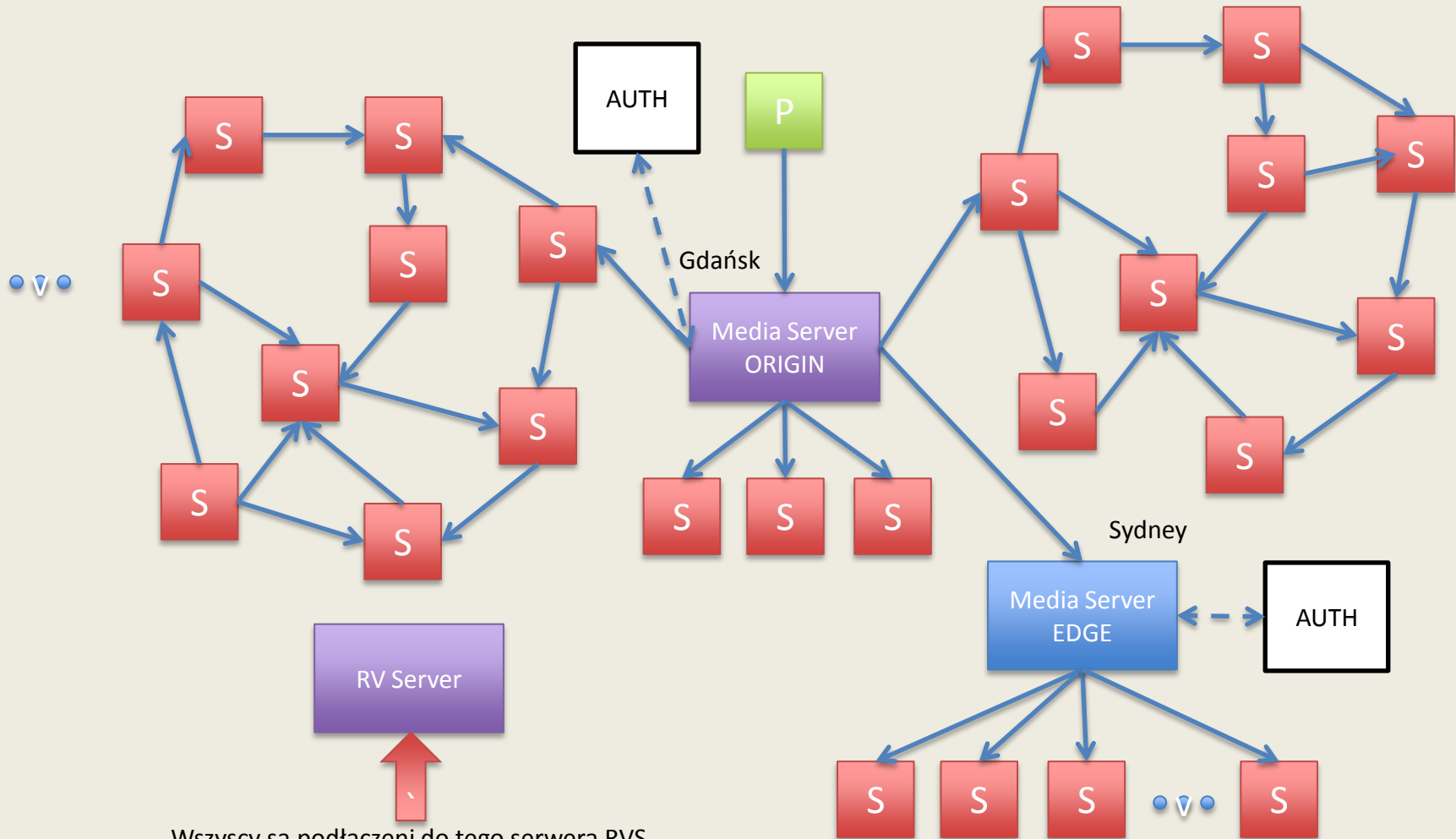
Wszyscy są podłączeni do tego serwera RVS.

# Transmisja z wykorzystaniem struktury połączonych grup – duża skala

- Kilka liczych grup.
- Jeżeli klient należy do dwóch grup, to te grupy tworzą jedną grupę.
- Wykorzystany serwer streamingowy do przekazywania strumieni pomiędzy grupami.



# Wiele konfiguracji łącznie – realna sytuacja



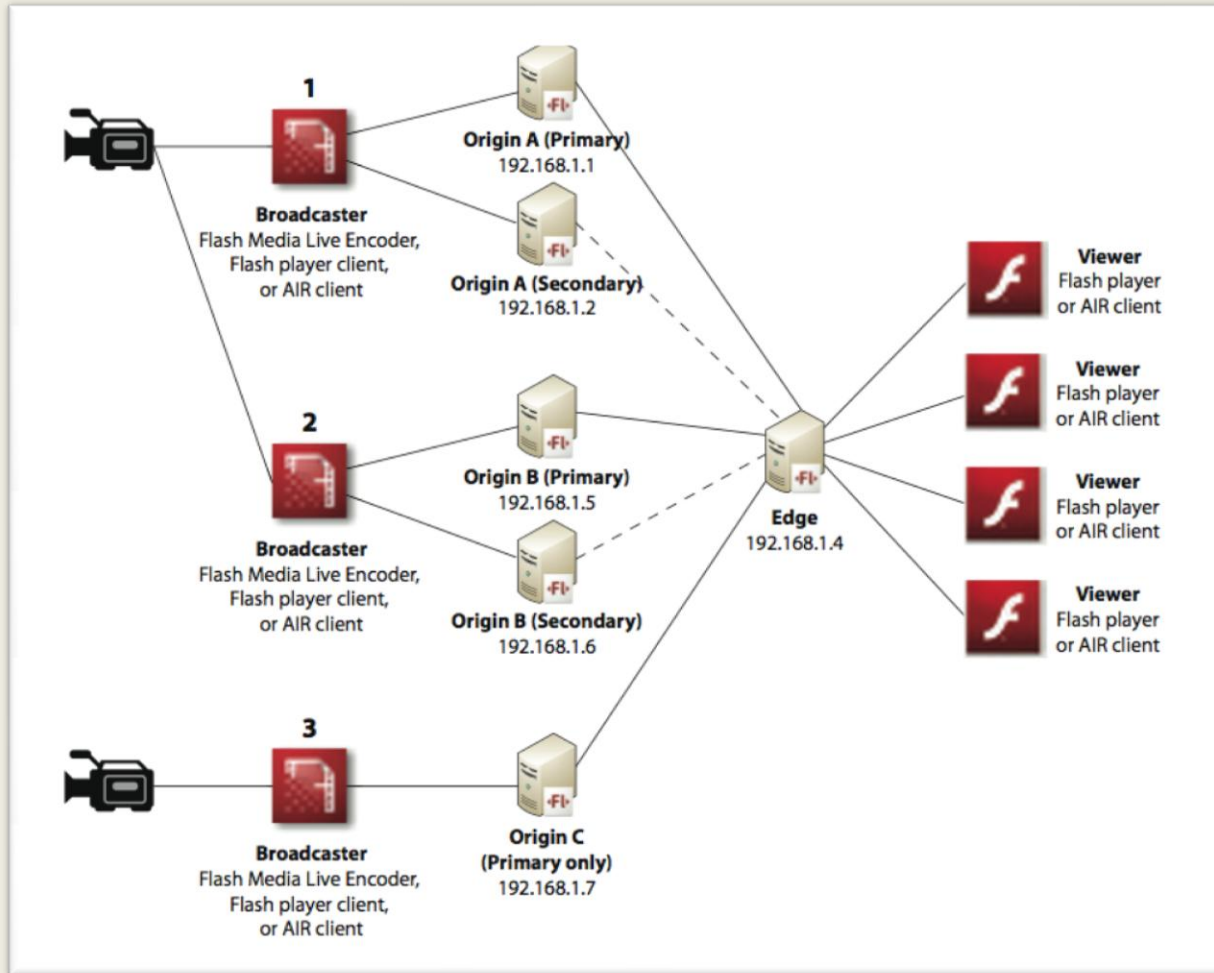
Wszyscy są podłączeni do tego serwera RVS.

# Oprogramowanie...

- Adobe Media Server
- Nginx rtmp-module
- Red5
- Wowza Media Server
- I wiele innych *open source* i komercyjnych...



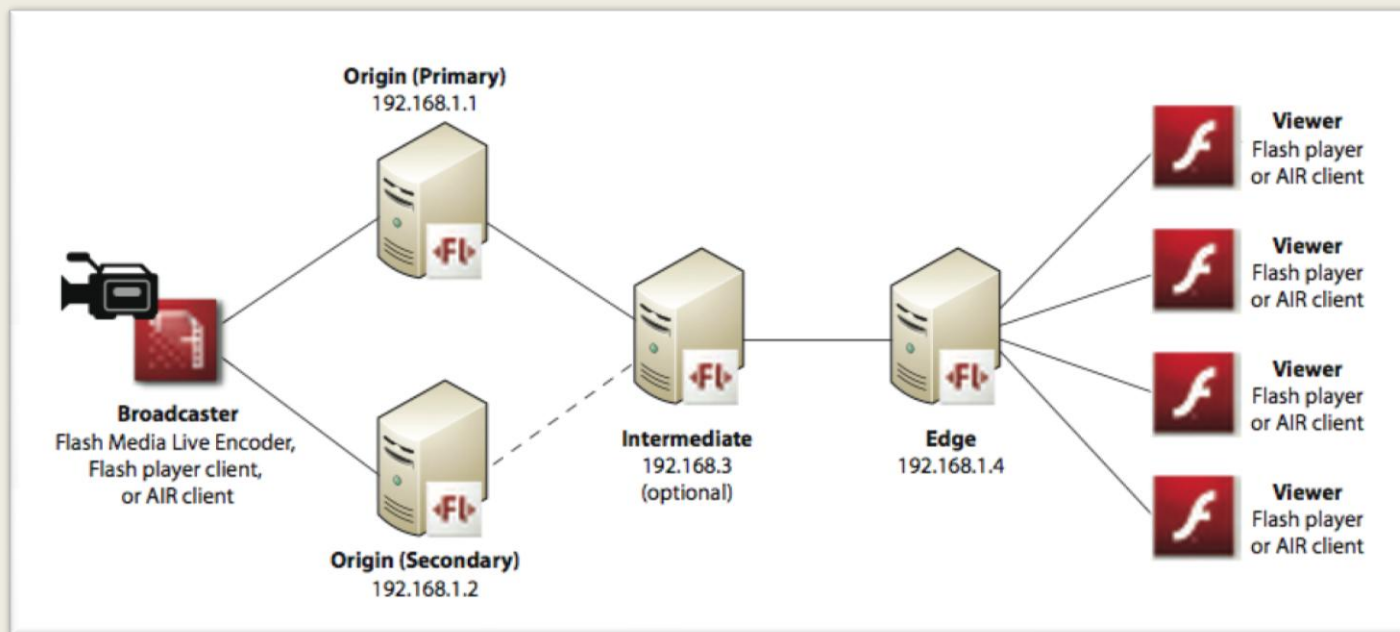
# Przykład konfiguracji do transmisji strumieniowej (1)



Konfiguracja z kilkoma grupami serwerów źródłowych i kilkoma koderami.

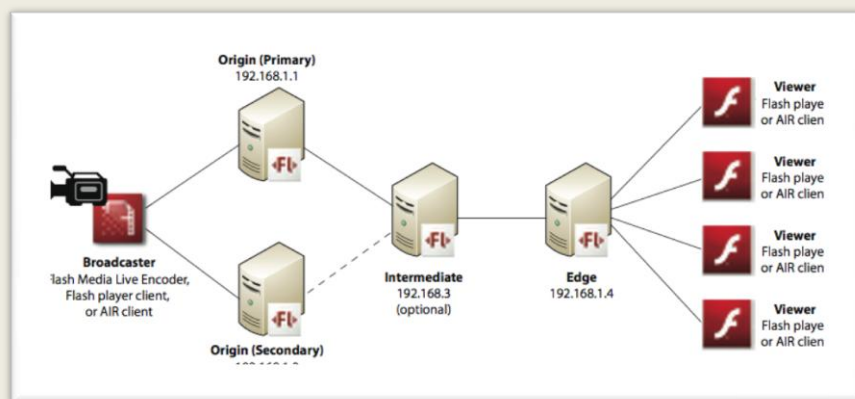
- Zapasowy serwer źródłowy jest dostępny tylko dla koderów 1 i 2.
- Serwer brzegowy zna lokalizację wszystkich serwerów głównych.
- Użytkownicy końcowi łączą się do serwera brzegowego.

# Przykład konfiguracji do transmisji strumieniowej (2)



- Konfiguracja z opcjonalnym węzłem pośredniczącym.
- Serwer brzegowy interpretuje główny serwer źródłowy i zapasowy serwer źródłowy jako ten sam serwer. (W konfiguracji ma zapisane tylko lokalizację serwera pośredniczącego).

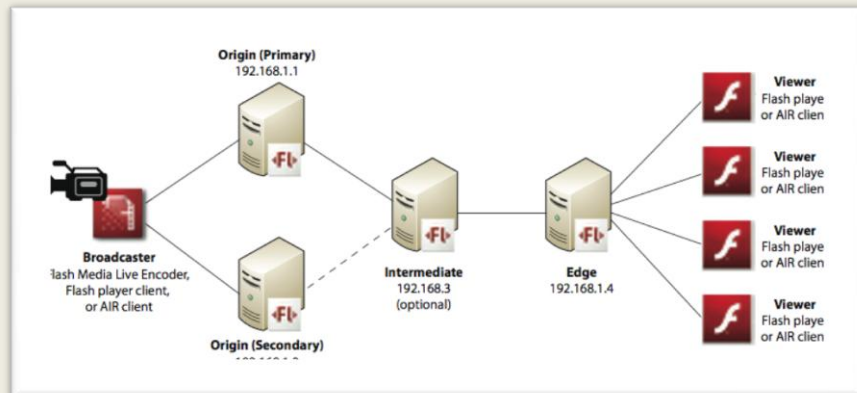
# Typy węzłów (1)



- **Serwer źródłowy**

- Aplikacja odbiera strumień wideo od nadawcy i dystrybuuje do wielu serwerów pośredniczących
- Ma możliwość akceptowania połączeń od nadawców. Przechowuje listę wszystkich nadawców.
- Akceptuje połączenie od serwerów pośredniczących i powiadamia te serwery o podłączeniu/rozłączeniu nadawcy.
- Akceptuje komendę *play* od serwera pośredniczącego i dystrybuuje dane do węzłów pośredniczących.

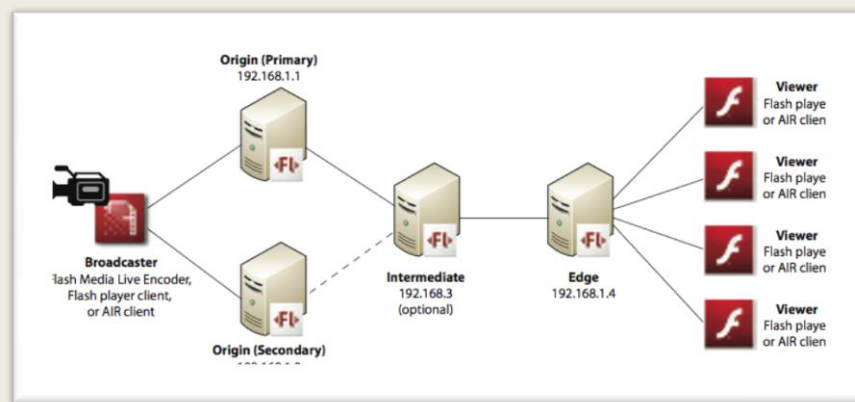
# Typy węzłów (2)



- **Serwer pośredniczący**

- Aplikacja serwerowa odbiera strumień wideo z serwera źródłowego i przekazuje dane do wielu serwerów brzegowych.
- Czyta adresy IP serwerów źródłowych z pliku konfiguracyjnego
- Łączy się do wszystkich serwerów źródłowych, których adresy są zawarte w pliku konfiguracyjnym i dba o ponowne połączenie w przypadku rozłączenia.
- Odbiera notyfikacje z serwerów źródłowych i przetrzymuje listą nadawców dla każdego serwera źródłowego

# Typy węzłów (2)



- **Serwer brzegowy**

- Aplikacja serwerowa odbiera żądania połączenia od klientów i przekazuje dane z serwerów pośredniczących do klientów.
- Czyta adresy IP serwerów źródłowych z pliku konfiguracyjnego
- Łączy się do wszystkich serwerów pośredniczących, których adresy są zawarte w pliku konfiguracyjnym i dba o ponowne połączenie w przypadku rozłączenia.
- Odbiera notyfikacje z serwerów pośredniczących i przetrzymuje listę nadawców dla każdego serwera pośredniczącego
- Wysyła komendę *play* i odbiera dane od serwerów pośredniczących.

Dziękuję za uwagę.