

**Как обеспечить максимальный
CDN-offload эффект для
платформы P2P-стриминга
на базе WebRTC
Опыт Teleport Media**

Алексей Клименко, СТО

alex@teleport.media

Maximize CDN-offload with WebRTC-based P2P-streaming

Введение в WebRTC p2p-стриминг

CDN + P2P(WebRTC)

WebRTC

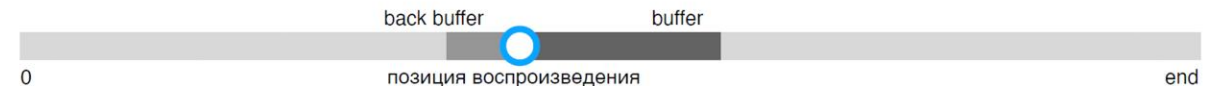
- передача видео, аудио и произвольных бинарных
- передача данных между клиентами по прямому со
- RTP и SCTP
- потребность в CDN снижается
- кросс-платформенный

Интеграция P2P модуля с WEB плеерами

1. Мониторинг позиции воспроизведения API плеера, либо напрямую
2. Некоторые плееры позволяют использовать протоколы транспорта(hls.js, shaka.js)
3. А некоторые нет. Но если рукава и написали "плагины"
4. Плееры с закрытым исходным кодом(Brightcove) как правило являются кросс-платформы и предлагают

Видеоплееры адаптивного стриминга

1. Адаптивный стриминг -- DASH, HLS, MSS, Adobe HDS(good bye, sweet prince)
2. Scheduler(Streaming engine) -- функция с аргументами "timestamp, quality", результат - URL сегмента из манифеста
3. Сегмент запрашивается с CDN и после скачивания попадает в буффер плеера
4. Размер буффера задается в конфиге плеера



Maximize CDN-offload with WebRTC-based P2P-streaming



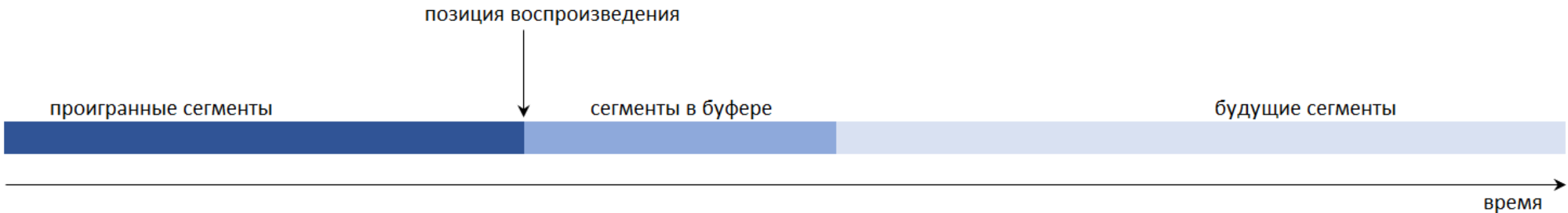
Maximize CDN-offload with WebRTC-based P2P-streaming



Maximize CDN-offload with WebRTC-based P2P-streaming

Teleport Media и Streamroot, архитектурные сходства:

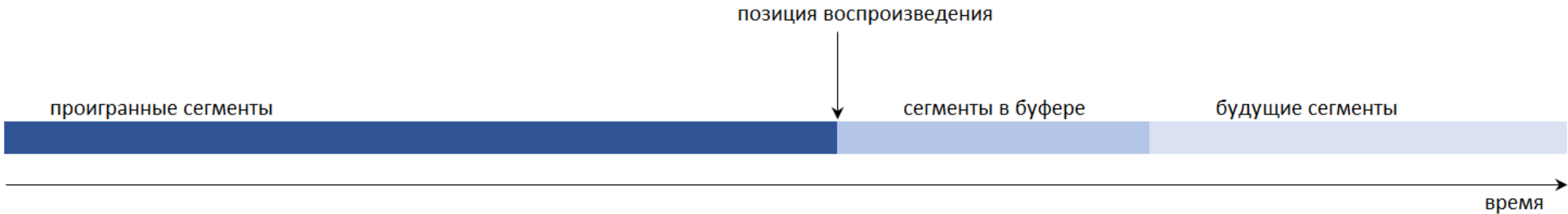
- контролируем buffer size (настройка `min_buffer_size`)
- ограничиваем время загрузки из пиринга (настройка `download_timeout`)



Maximize CDN-offload with WebRTC-based P2P-streaming

Teleport Media и Streamroot, архитектурные сходства:

- контролируем buffer size (настройка `min_buffer_size`)
- ограничиваем время загрузки из пиринга (настройка `download_timeout`)



Maximize CDN-offload with WebRTC-based P2P-streaming

Teleport Media и Streamroot, архитектурные сходства:

- контролируем buffer size (настройка `min_buffer_size`)
- ограничиваем время загрузки из пиринга (настройка `download_timeout`)

И различия:

- предсказание следующего сегмента / обработка запроса от плеера в момент его получения
- соединение пиров вокруг дорожки (LQ, SD, HD etc.) / соединение пиров вокруг url манифеста

Maximize CDN-offload with WebRTC-based P2P-streaming

Эксперименты Teleport Media начиная с 2016 года:

- Proof-of-concept на базе webtorrent (JavaScript) (октябрь 2016)

Maximize CDN-offload with WebRTC-based P2P-streaming

Эксперименты Teleport Media начиная с 2016 года:

- Proof-of-concept на базе webtorrent (JavaScript) (октябрь 2016)
- Альфа-версия на базе webtorrent (JavaScript) (март 2017)

Maximize CDN-offload with WebRTC-based P2P-streaming

Эксперименты Teleport Media начиная с 2016 года:

- Proof-of-concept на базе webtorrent (JavaScript) (октябрь 2016)
- Альфа-версия на базе webtorrent (JavaScript) (март 2017)
- Релиз на базе полностью «своего» кода (TypeScript) (март 2018)

Maximize CDN-offload with WebRTC-based P2P-streaming

Эксперименты Teleport Media начиная с 2016 года:

- Proof-of-concept на базе webtorrent (JavaScript) (октябрь 2016)
- Альфа-версия на базе webtorrent (JavaScript) (март 2017)
- Релиз на базе полностью «своего» кода (TypeScript) (март 2018)

Две версии кода back-end:

- Монолит на NodeJS (октябрь 2016)

Maximize CDN-offload with WebRTC-based P2P-streaming

Эксперименты Teleport Media начиная с 2016 года:

- Proof-of-concept на базе webtorrent (JavaScript) (октябрь 2016)
- Альфа-версия на базе webtorrent (JavaScript) (март 2017)
- Релиз на базе полностью «своего» кода (TypeScript) (март 2018)

Две версии кода back-end:

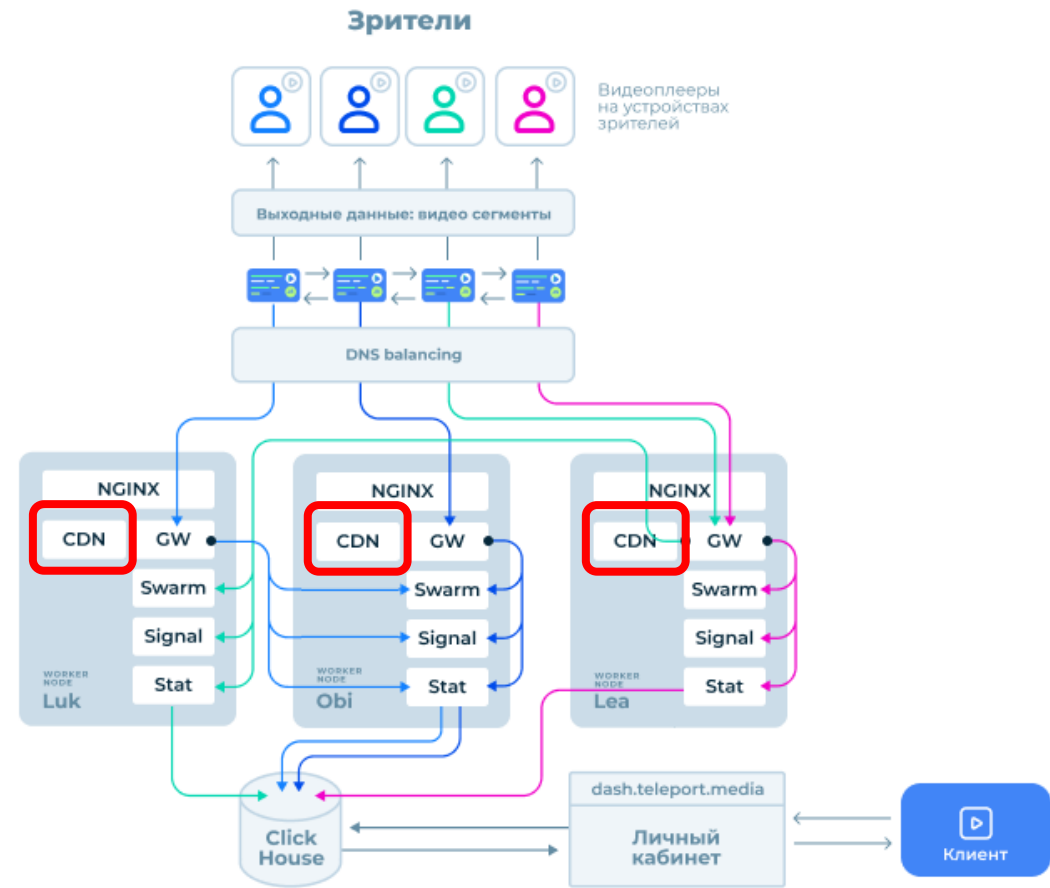
- Монолит на NodeJS (октябрь 2016)
- Микросервисы на Go (апрель 2018)

Maximize CDN-offload with WebRTC-based P2P-streaming

Текущая архитектура бэкенда

Teleport Media

cdn.teleport.media



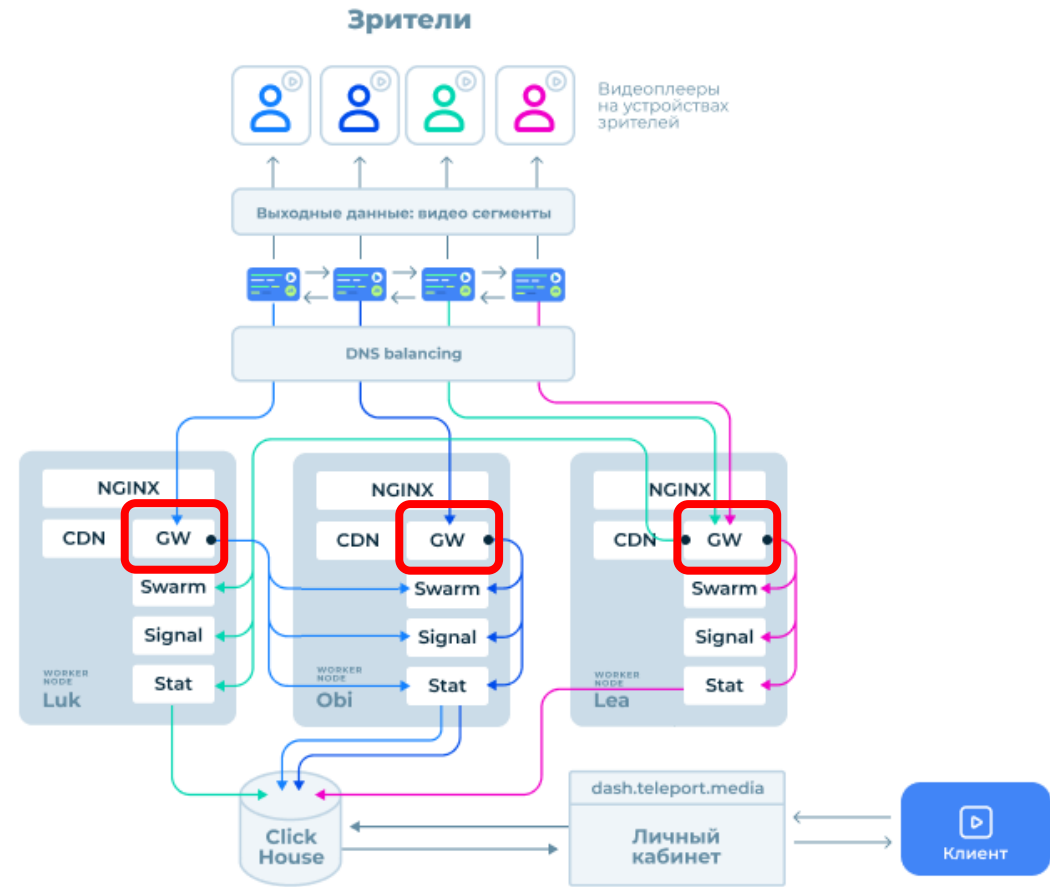
Maximize CDN-offload with WebRTC-based P2P-streaming

Текущая архитектура бэкенда

Teleport Media

cdn.teleport.media

gateway.teleport.media



Maximize CDN-offload with WebRTC-based P2P-streaming

Текущая архитектура бэкенда

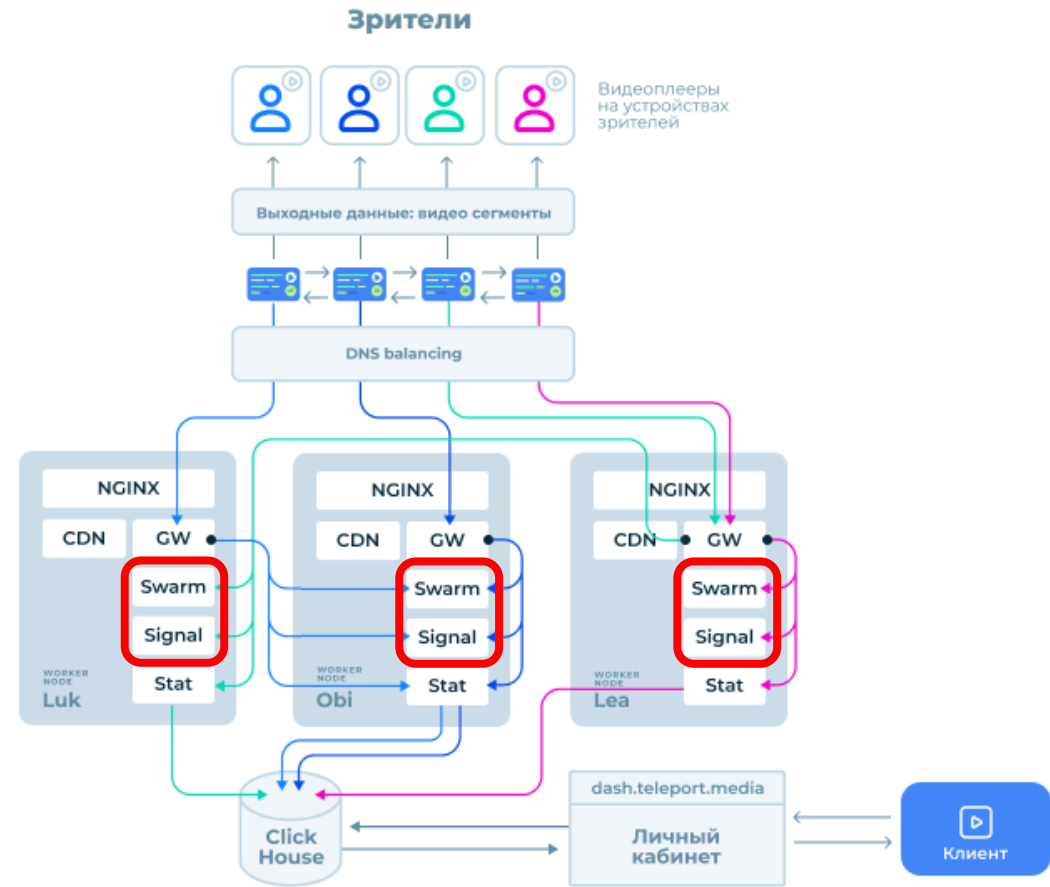
Teleport Media

cdn.teleport.media

gateway.teleport.media

swarm-xxx.teleport.media

signal-xxx.teleport.media



Maximize CDN-offload with WebRTC-based P2P-streaming

Текущая архитектура бэкенда

Teleport Media

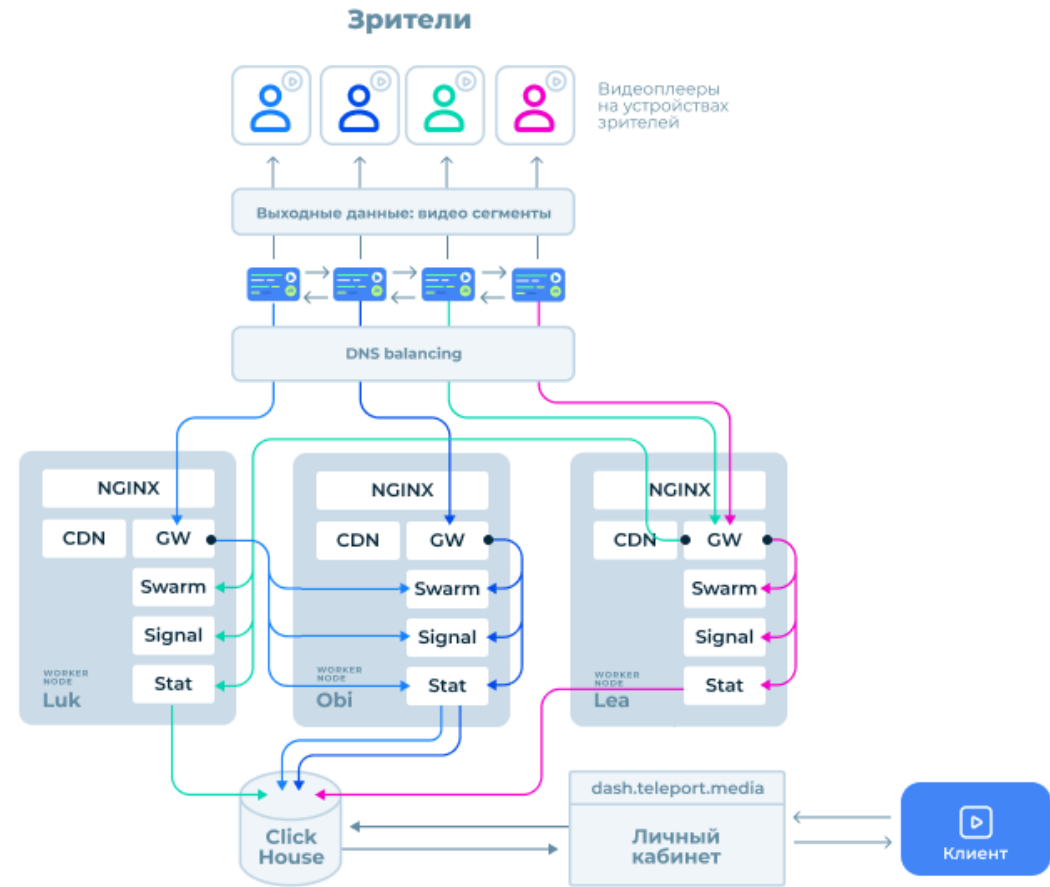
cdn.teleport.media

gateway.teleport.media

swarm-xxx.teleport.media

signal-xxx.teleport.media

stun.teleport.media



Maximize CDN-offload with WebRTC-based P2P-streaming

Текущая архитектура бэкенда

Teleport Media

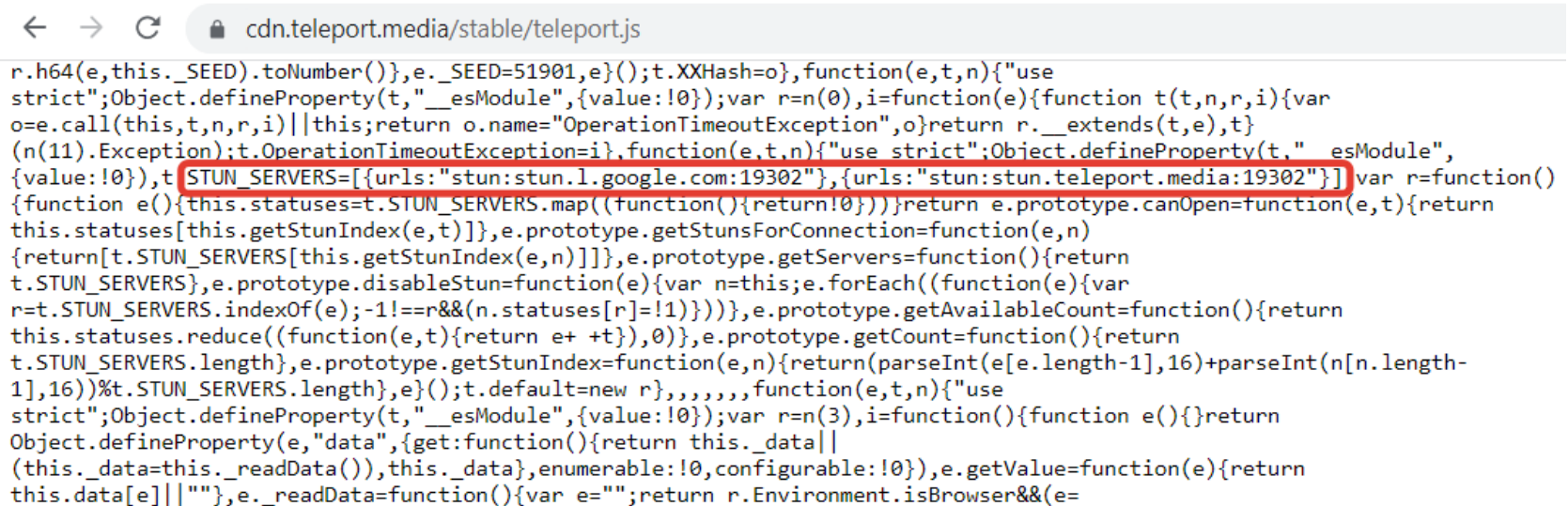
cdn.teleport.media

gateway.teleport.media

swarm-xxx.teleport.media

signal-xxx.teleport.media

stun.teleport.media



```
← → ↻ 🔒 cdn.teleport.media/stable/teleport.js
r.h64(e,this._SEED).toNumber()),e._SEED=51901,e}();t.XXHash=o},function(e,t,n){"use
strict";Object.defineProperty(t,"__esModule",{value:!0});var r=n(0),i=function(e){function t(t,n,r,i){var
o=e.call(this,t,n,r,i)||this;return o.name="OperationTimeoutException",o}return r.__extends(t,e),t}
(n(11).Exception);t.OperationTimeoutException=i},function(e,t,n){"use strict";Object.defineProperty(t,"
__esModule",{value:!0}),t.STUN_SERVERS=[{urls:"stun:stun.l.google.com:19302"},{urls:"stun:stun.teleport.media:19302"}]
var r=function()
{function e(){this.statuses=t.STUN_SERVERS.map((function(){return!0}))}return e.prototype.canOpen=function(e,t){return
this.statuses[this.getStunIndex(e,t)]},e.prototype.getStunsForConnection=function(e,n)
{return[t.STUN_SERVERS[this.getStunIndex(e,n)]],e.prototype.getServers=function(){return
t.STUN_SERVERS},e.prototype.disableStun=function(e){var n=this;e.forEach((function(e){var
r=t.STUN_SERVERS.indexOf(e);-1!==r&&(n.statuses[r]=!1))}),e.prototype.getAvailableCount=function(){return
this.statuses.reduce((function(e,t){return e+ +t}),0)},e.prototype.getCount=function(){return
t.STUN_SERVERS.length},e.prototype.getStunIndex=function(e,n){return(parseInt(e[e.length-1],16)+parseInt(n[n.length-
1],16))%t.STUN_SERVERS.length},e}();t.default=new r},,,,,,function(e,t,n){"use
strict";Object.defineProperty(t,"__esModule",{value:!0});var r=n(3),i=function(){function e(){return
Object.defineProperty(e,"data",{get:function(){return this._data||
(this._data=this._readData()),this._data},enumerable:!0,configurable:!0)},e.getValue=function(e){return
this.data[e]||""},e._readData=function(){var e="";return r.Environment.isBrowser&&(e=
```

Maximize CDN-offload with WebRTC-based P2P-streaming

Текущая архитектура бэкенда

Teleport Media

cdn.teleport.media

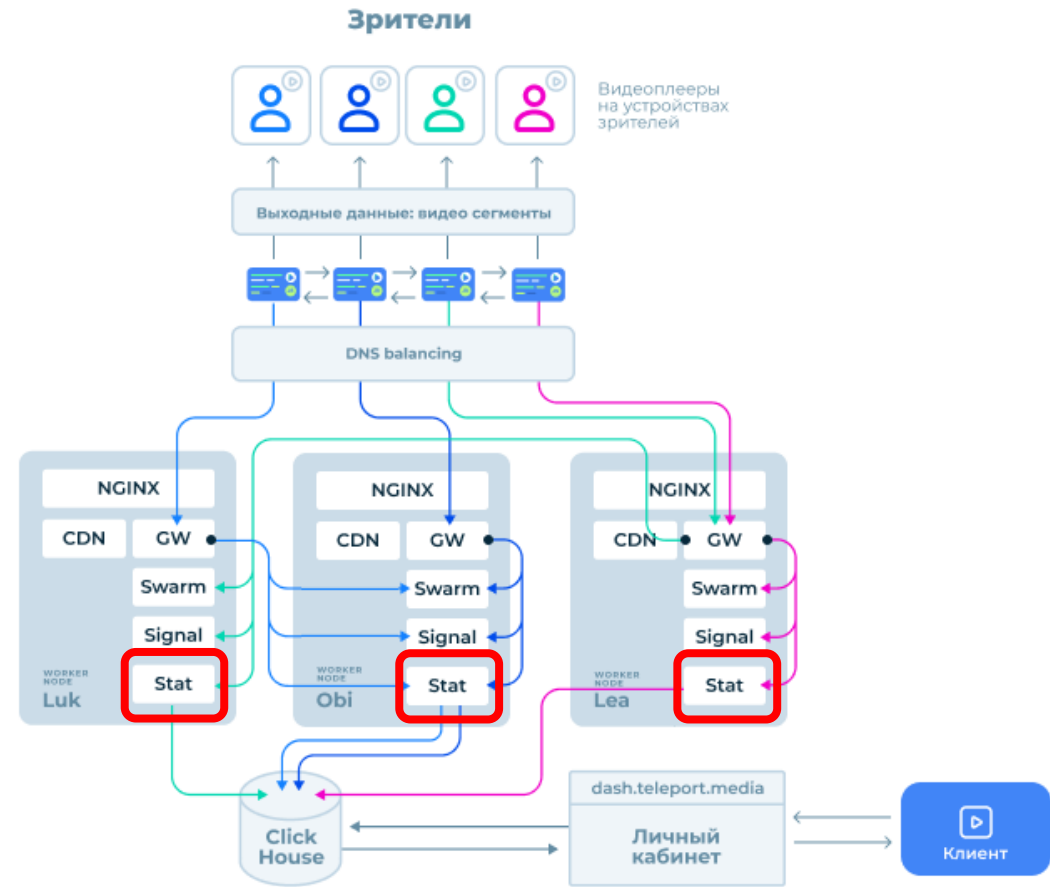
gateway.teleport.media

swarm-xxx.teleport.media

signal-xxx.teleport.media

stun.teleport.media

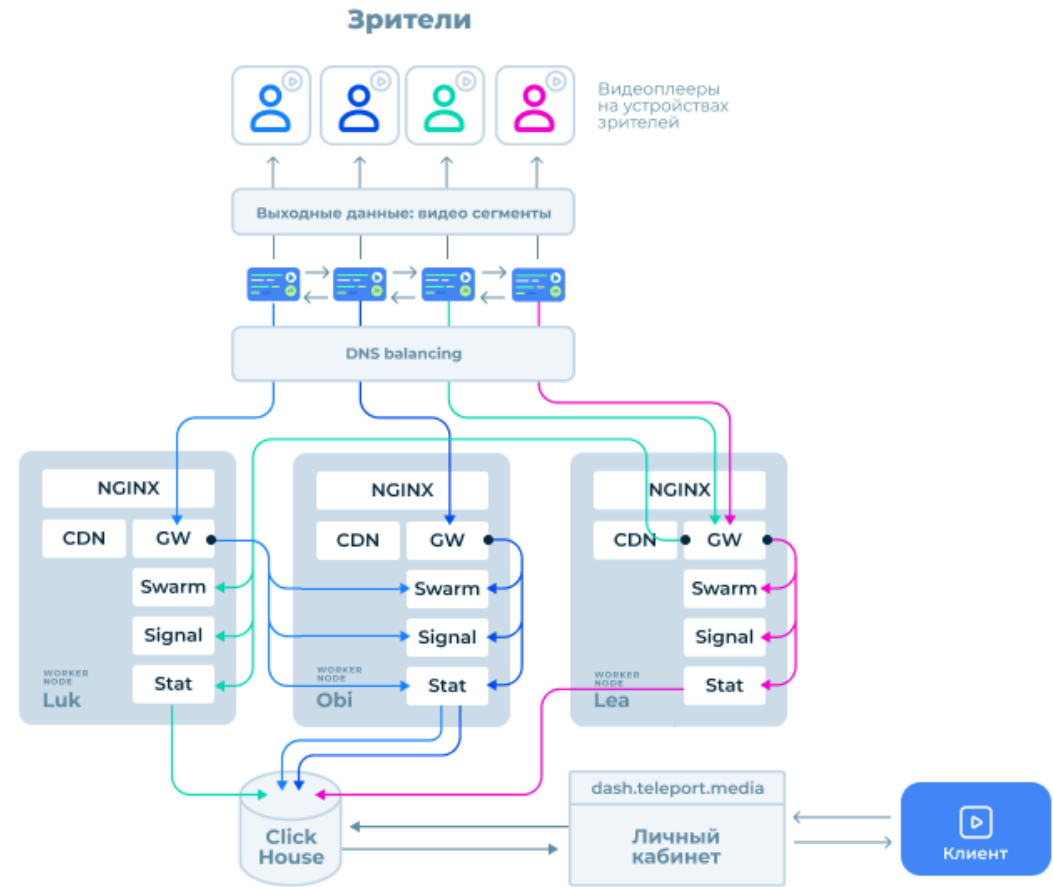
stat.teleport.media



Maximize CDN-offload with WebRTC-based P2P-streaming

Текущая архитектура бэкенда

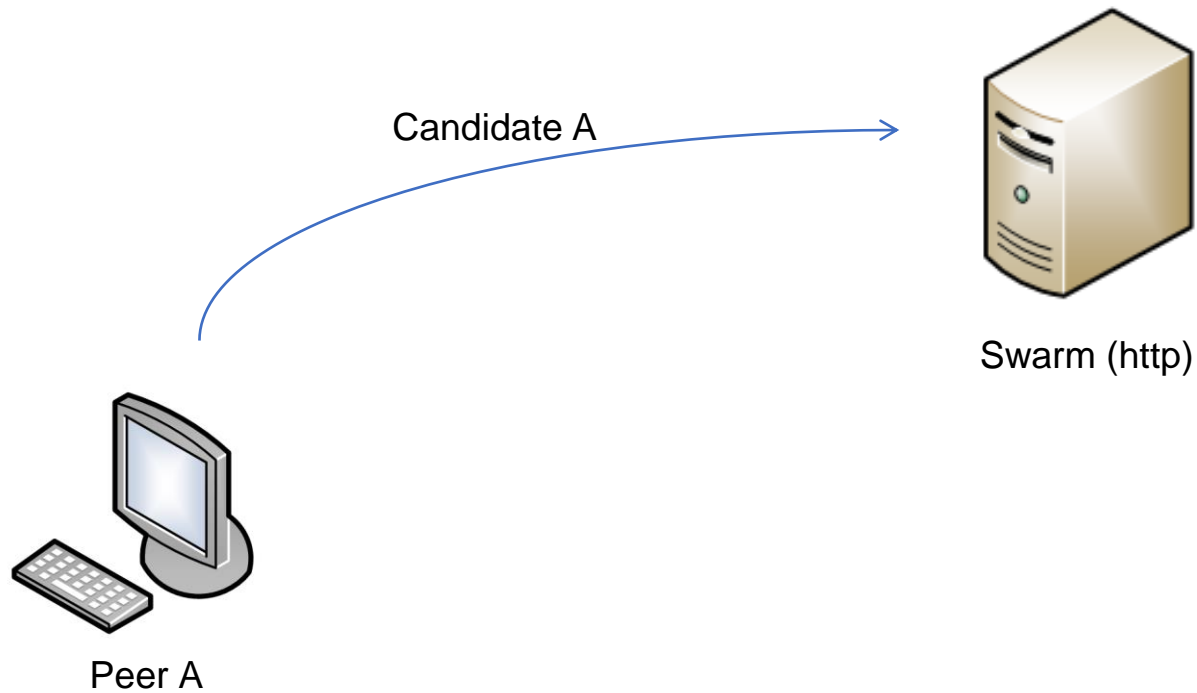
Teleport Media



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

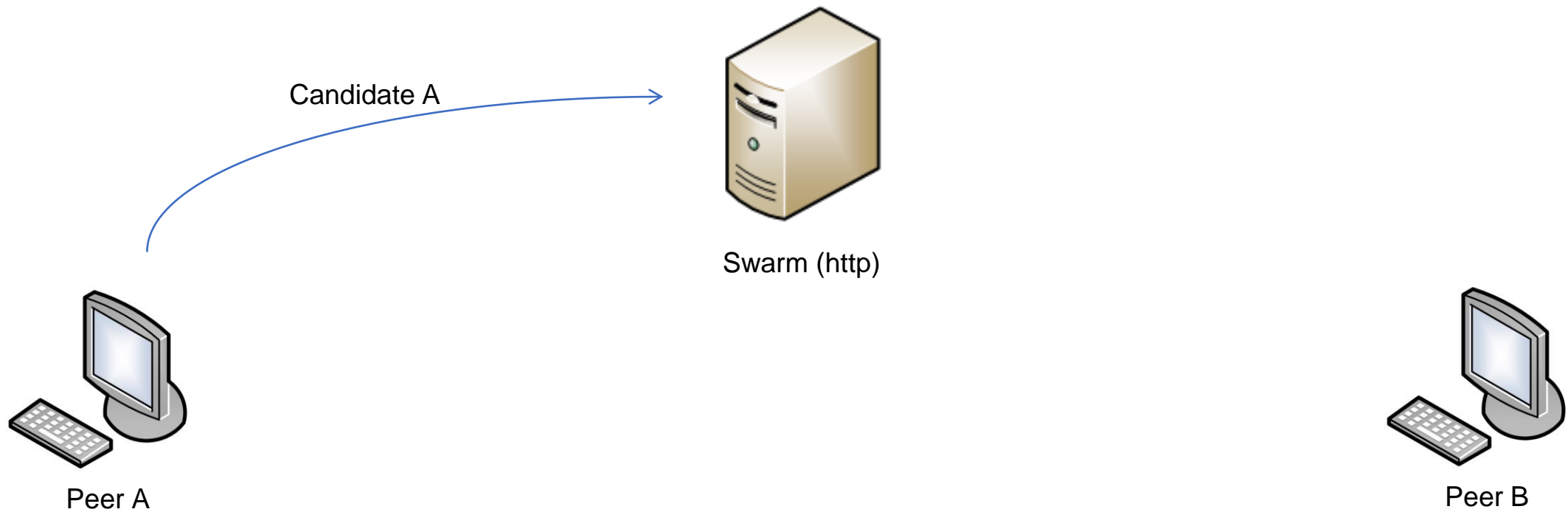
- Поиск и соединение пиров



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

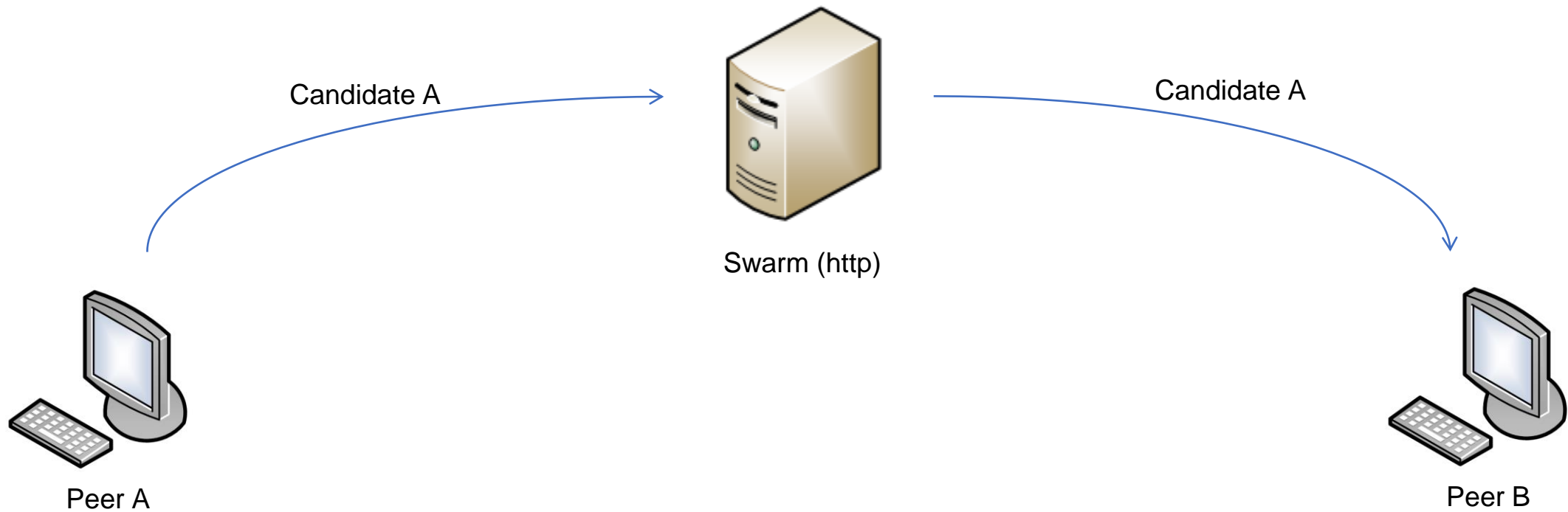
- Поиск и соединение пиров



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

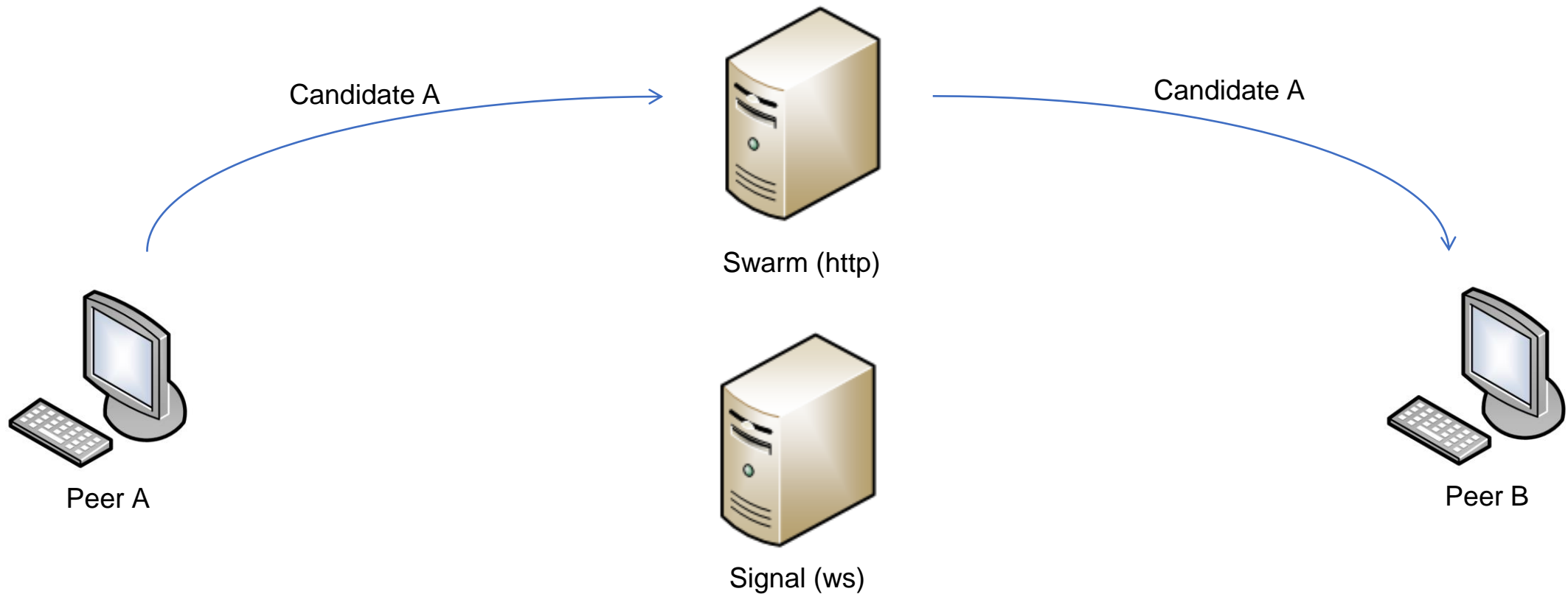
- Поиск и соединение пиров



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

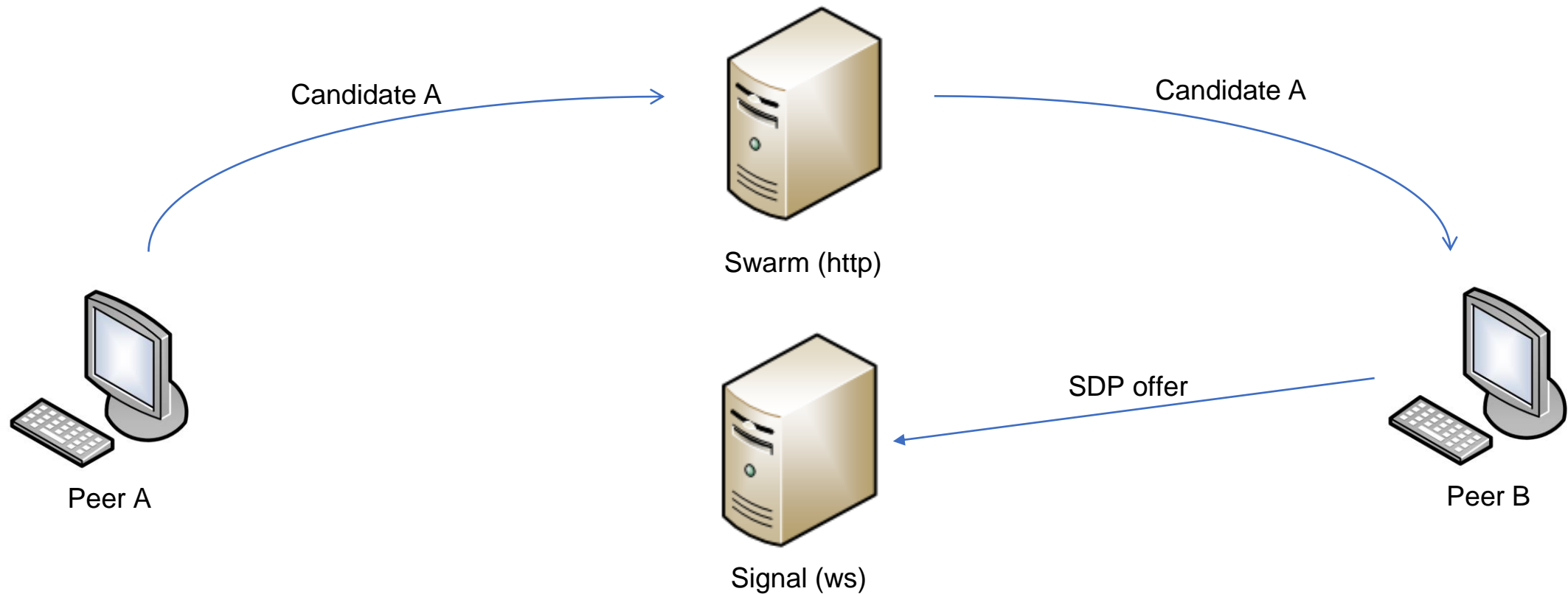
- Поиск и соединение пиров



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

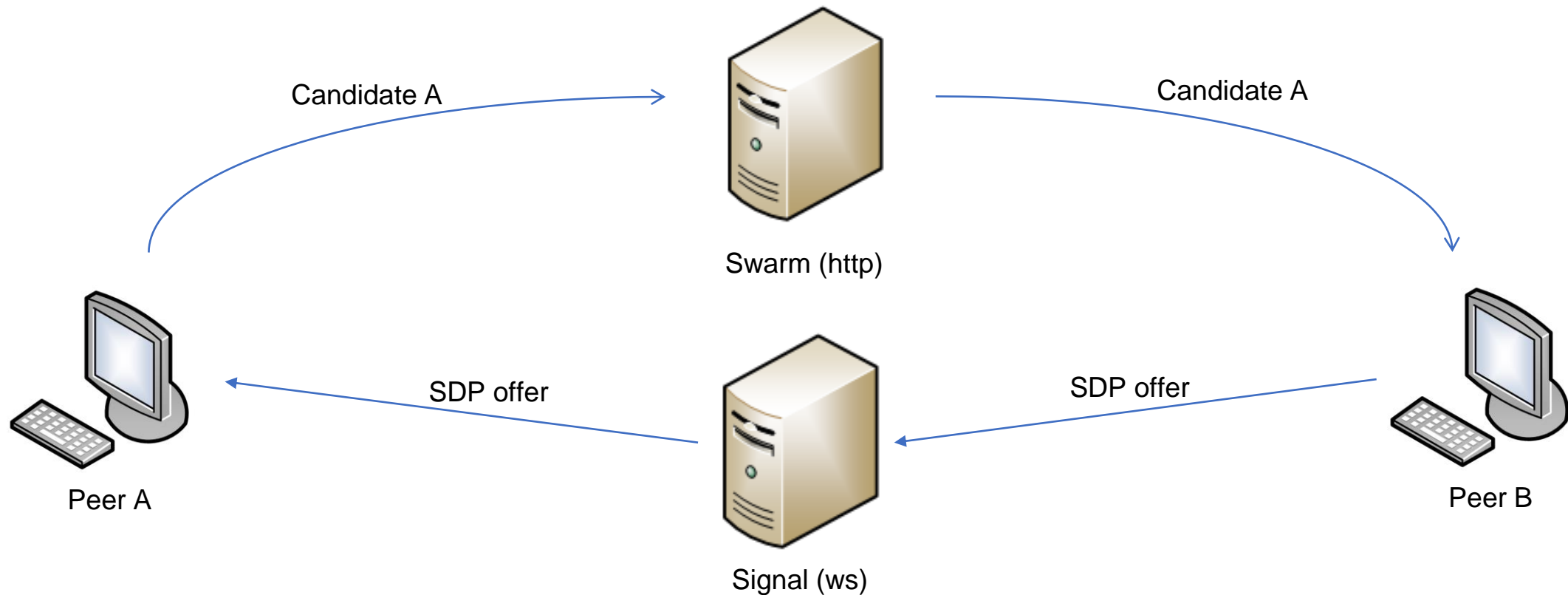
- Поиск и соединение пиров



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

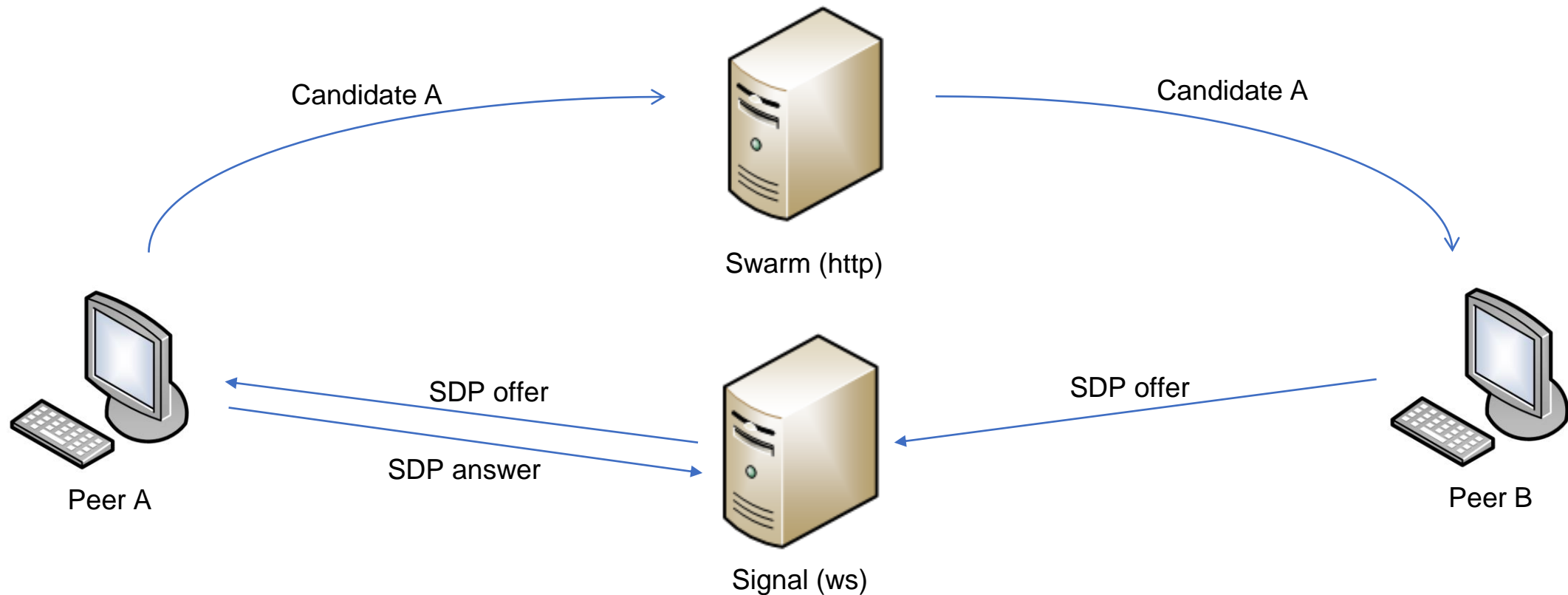
- Поиск и соединение пиров



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

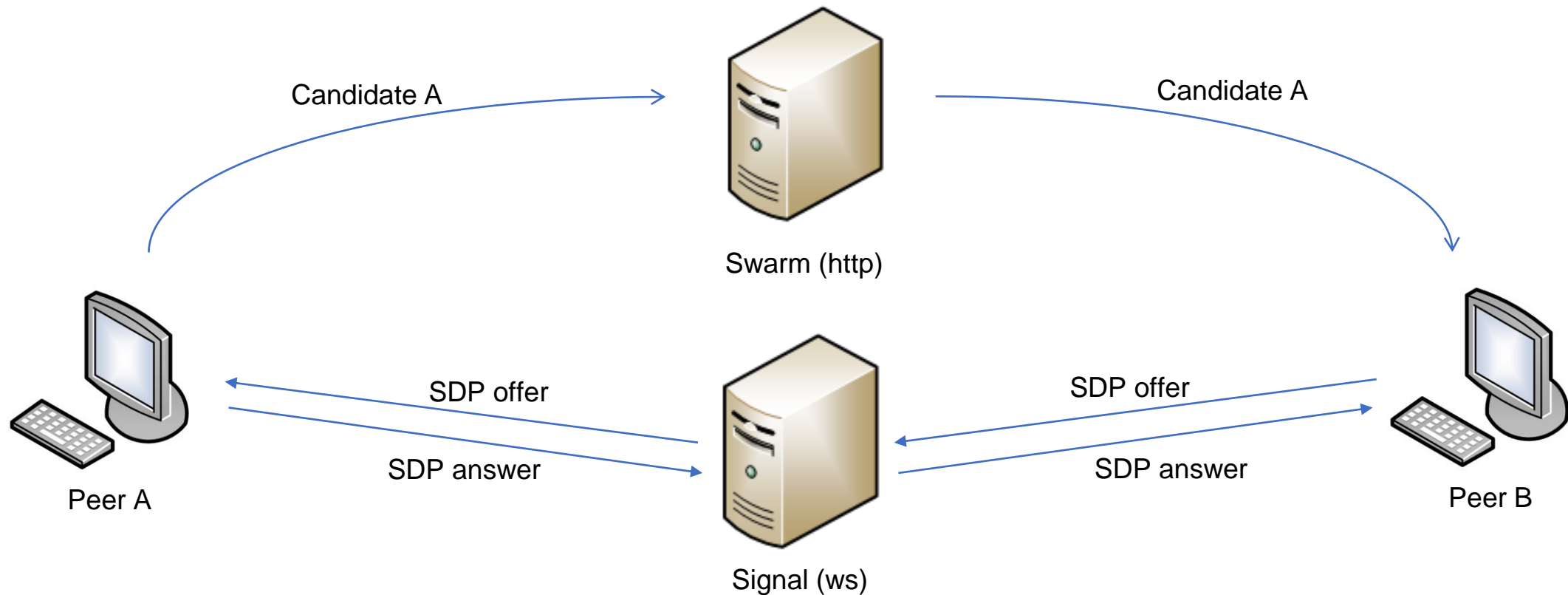
- Поиск и соединение пиров



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

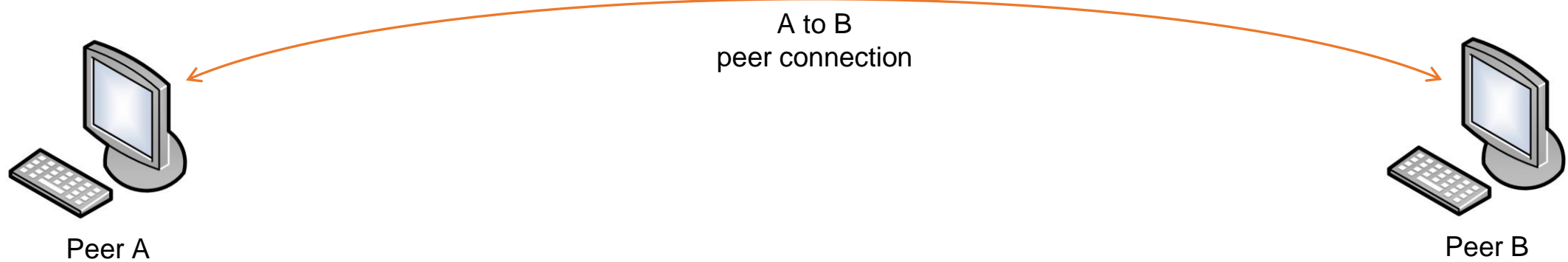
- Поиск и соединение пиров



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

- Поиск и соединение пиров



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

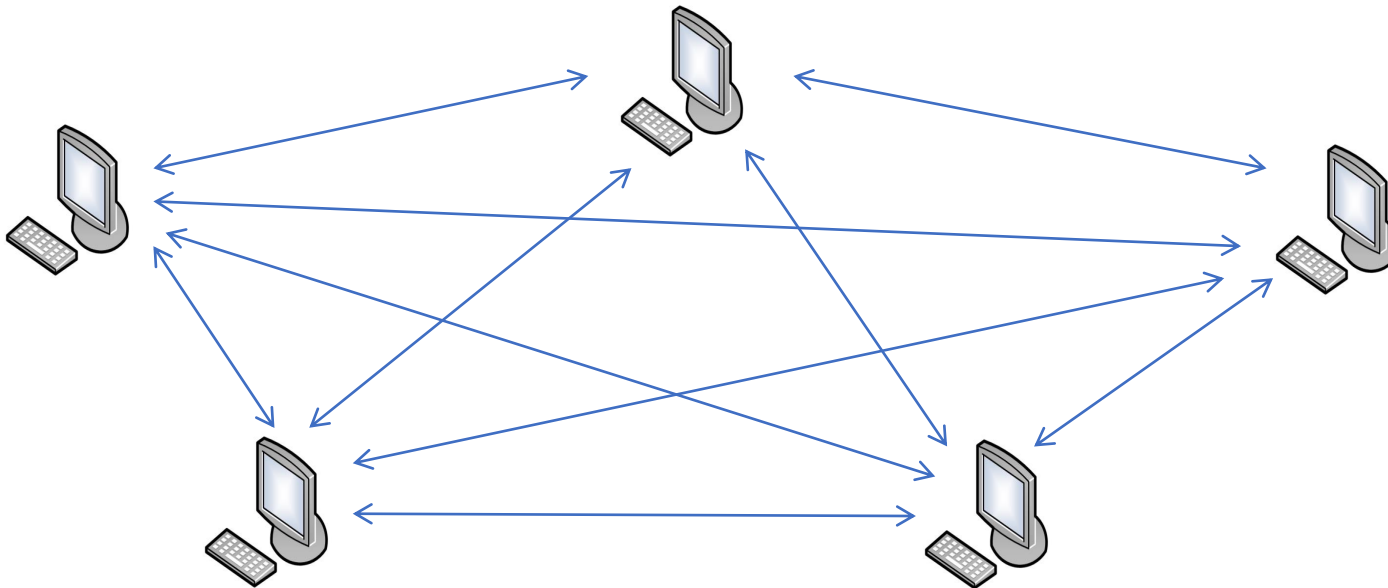
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

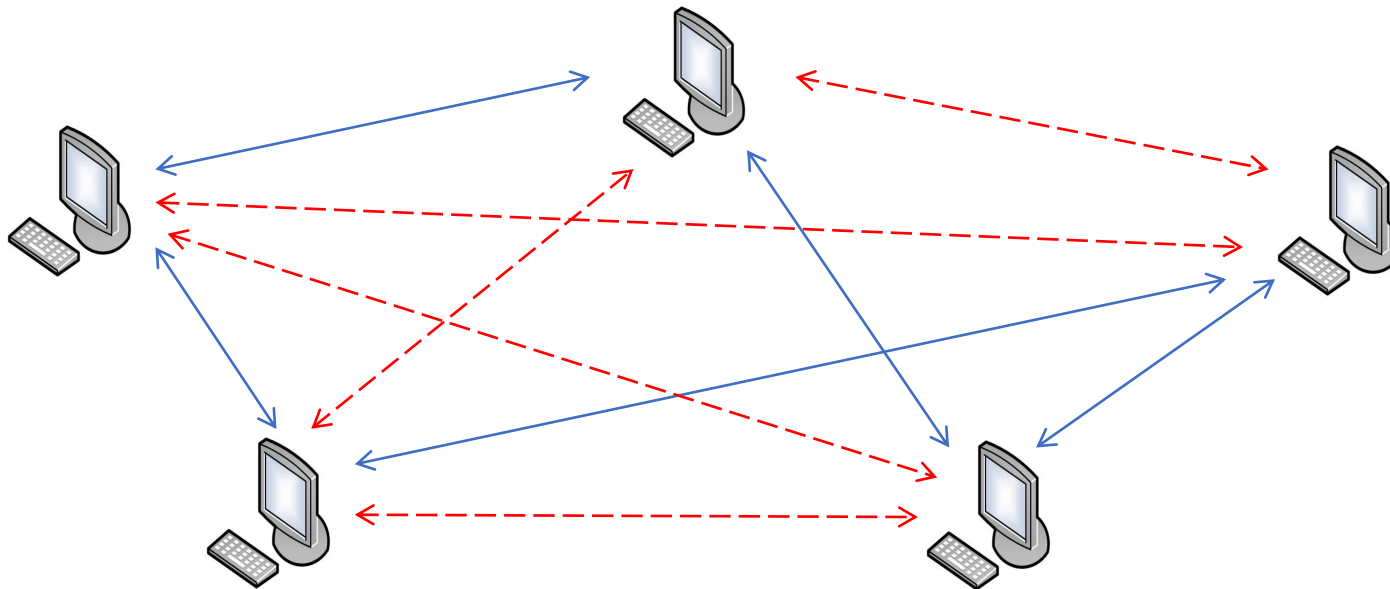
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

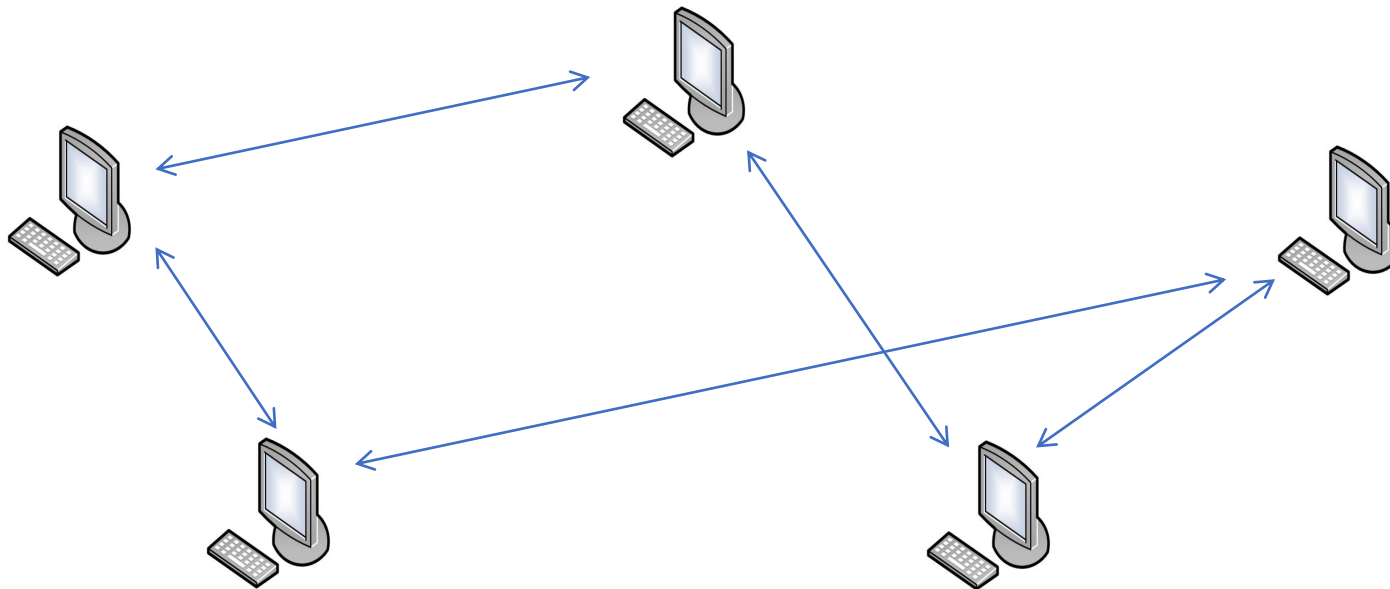
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

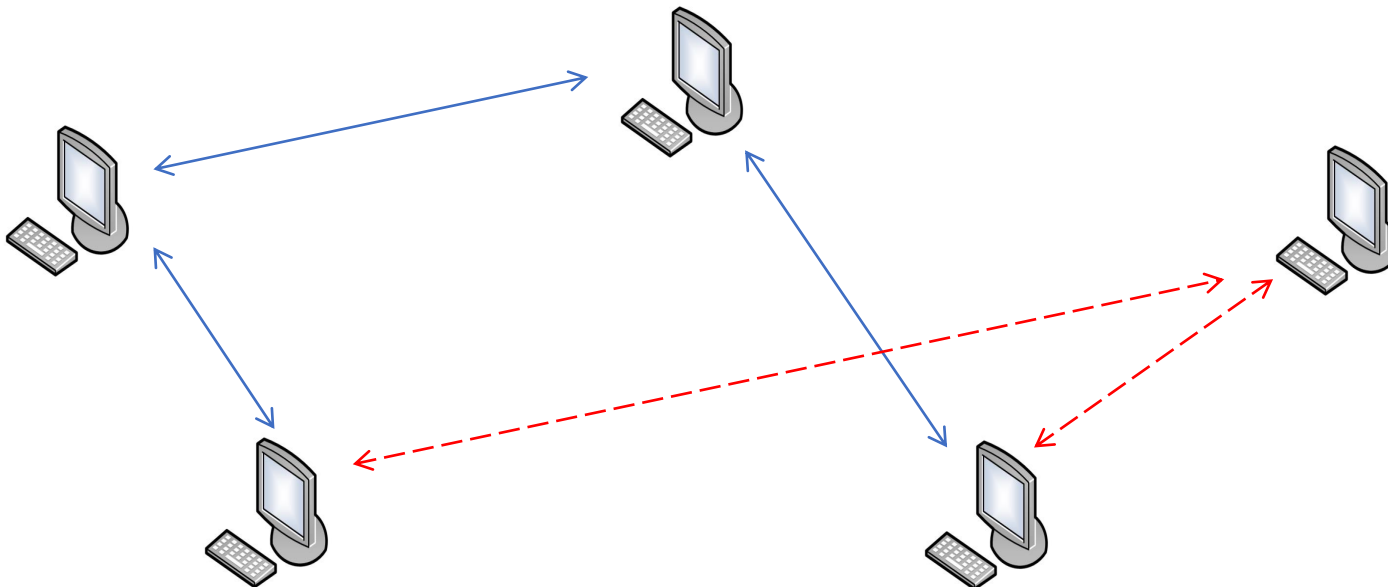
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

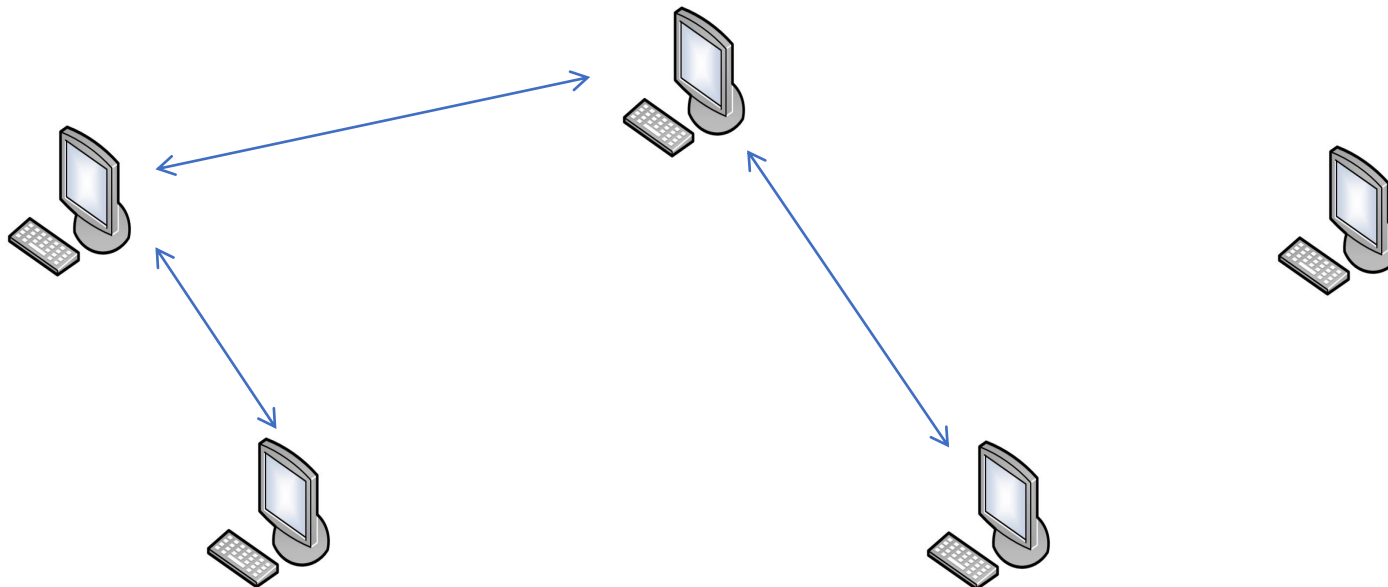
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

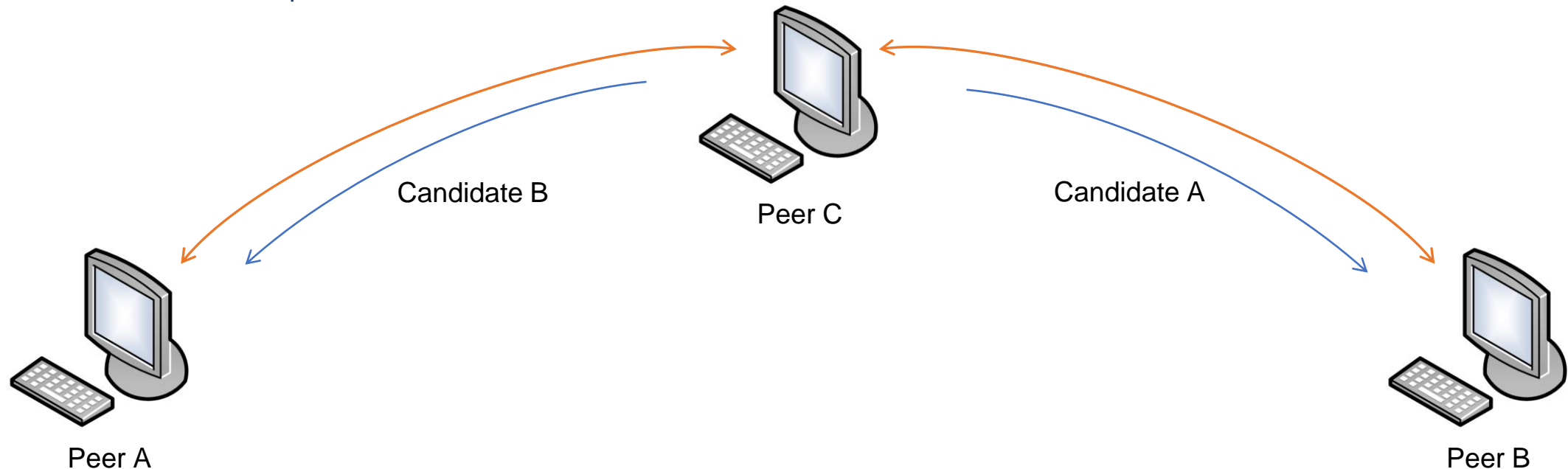
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

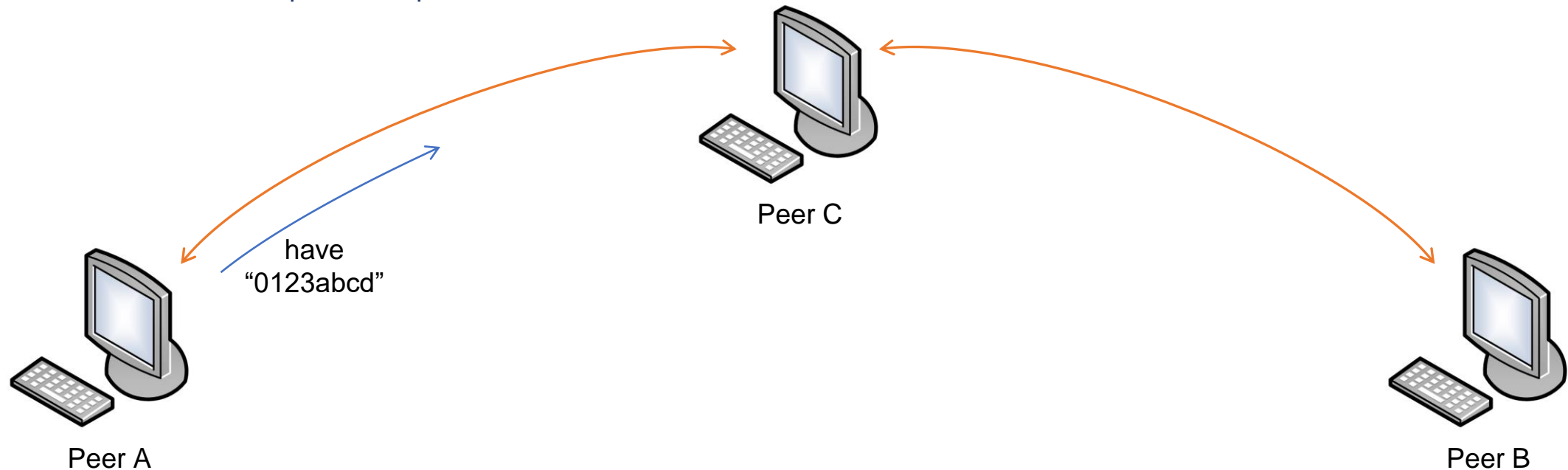
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме (score, timeslot)



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

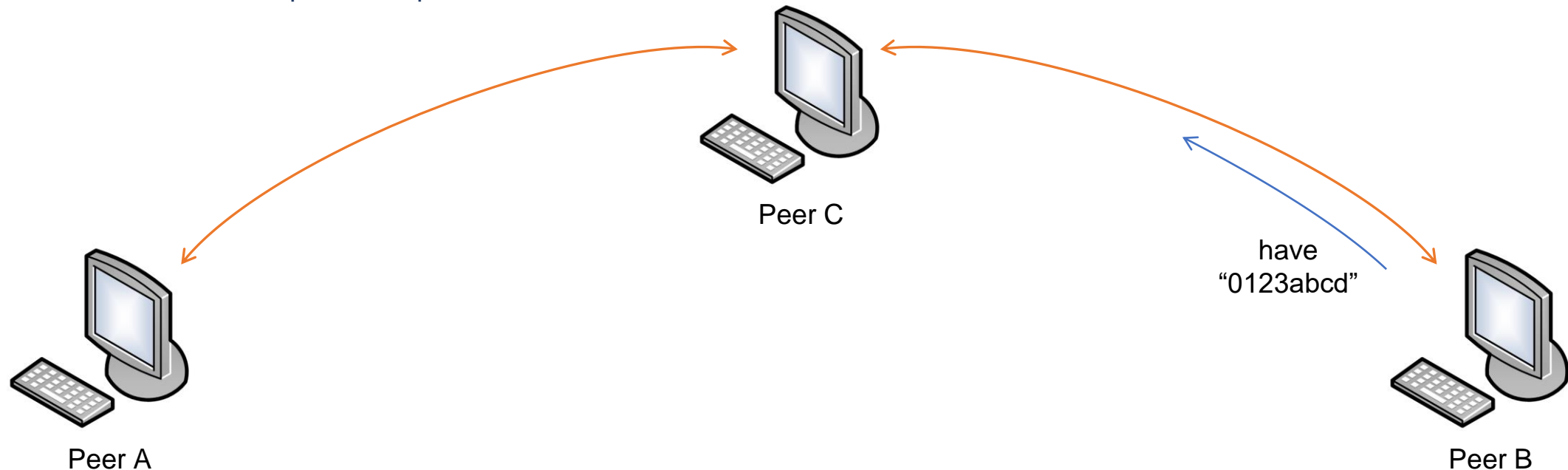
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме (impact match)



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

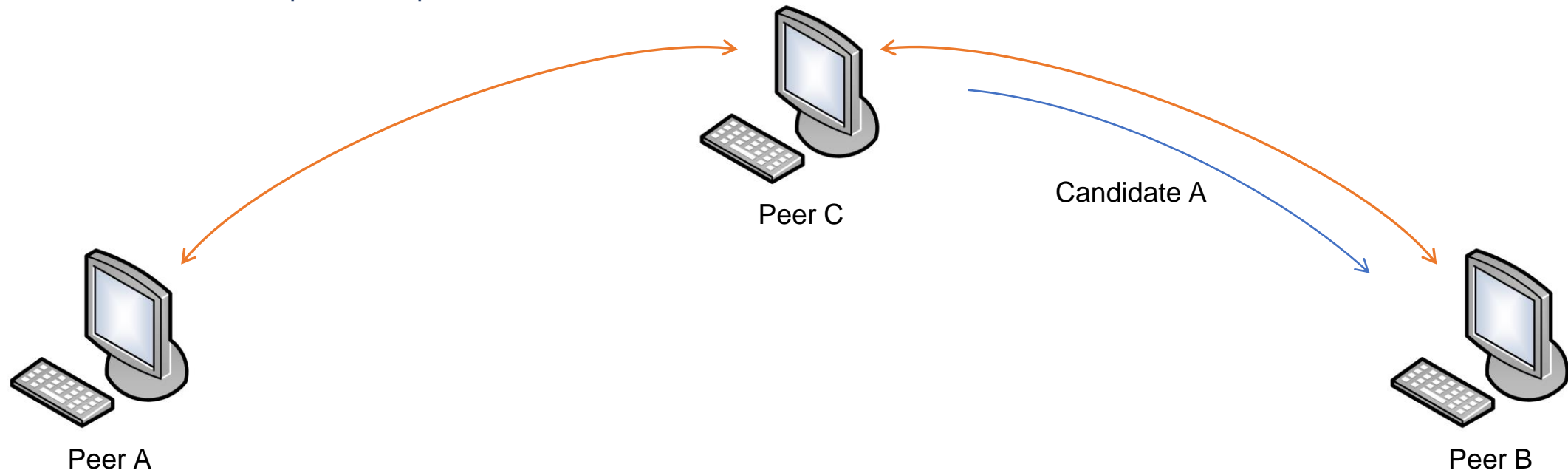
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме (impact match)



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

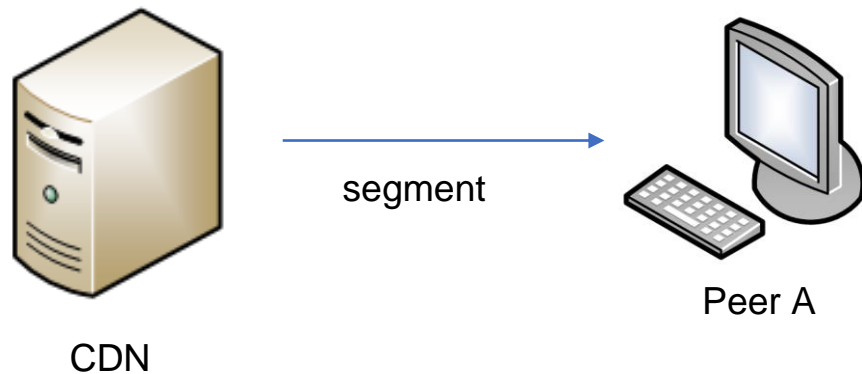
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме (impact match)



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

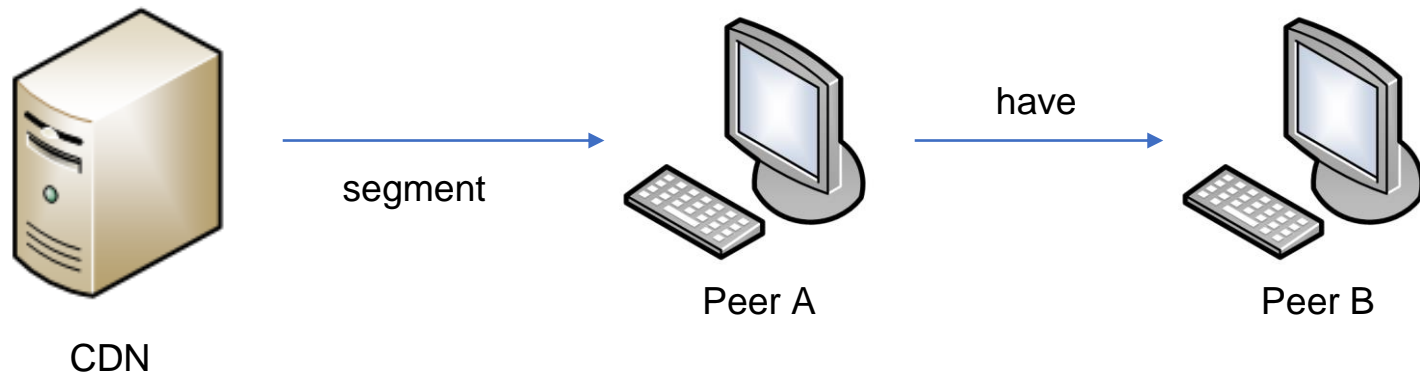
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме
- Стриминг и ре-стриминг (ускоряет проталкивание сегмента в пиринге)



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

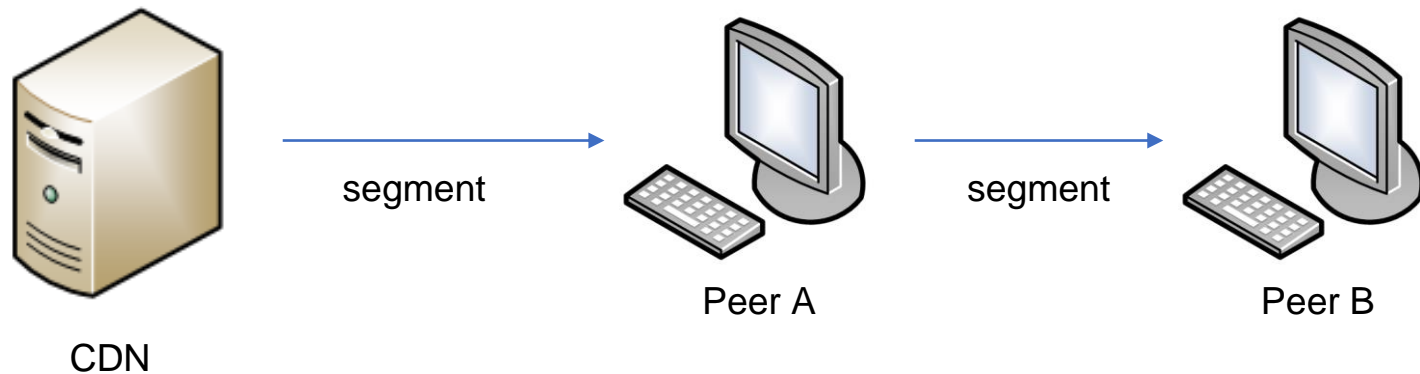
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме
- Стриминг и ре-стриминг (ускоряет проталкивание сегмента в пиринге)



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

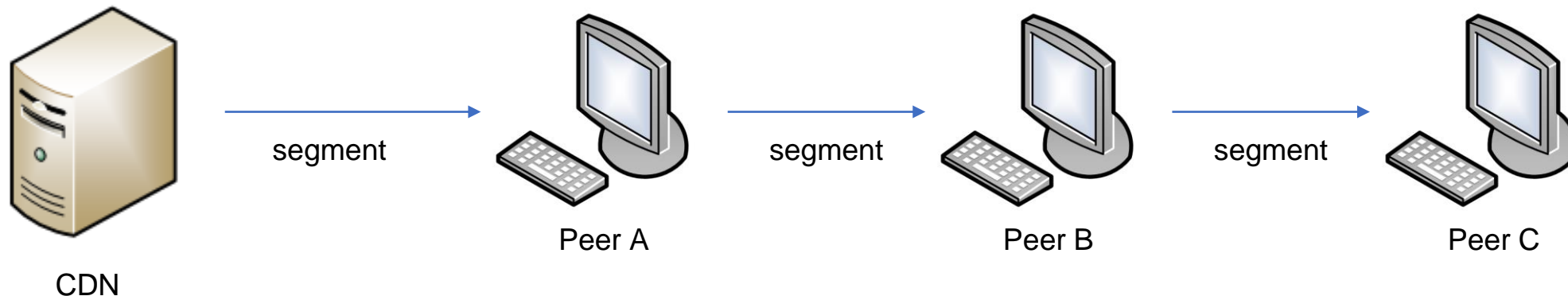
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме
- Стриминг и ре-стриминг (ускоряет проталкивание сегмента в пиринге)



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

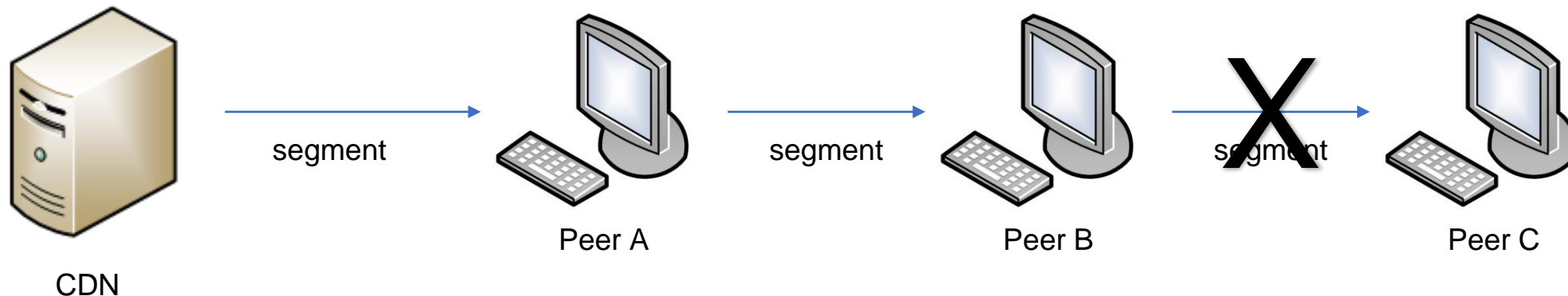
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме
- Стриминг и ре-стриминг (ускоряет проталкивание сегмента в пиринге)



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

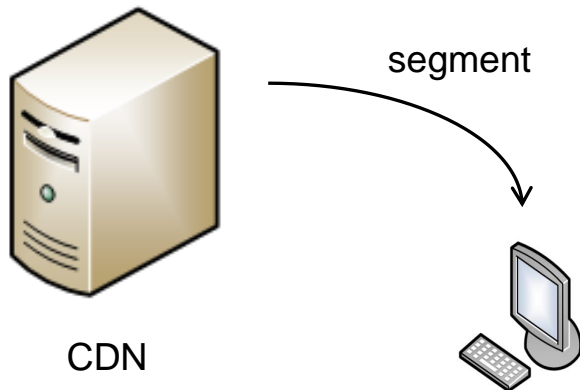
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме
- Стриминг и ре-стриминг (ускоряет проталкивание сегмента в пиринге)



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

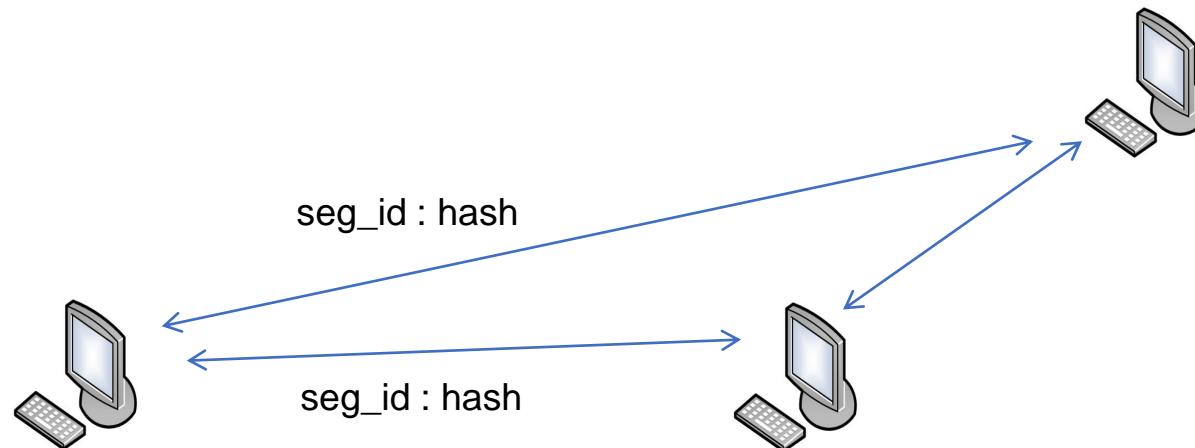
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме
- Стриминг и ре-стриминг
- Проверка целостности сегмента (hash broadcast)



Maximize CDN-offload with WebRTC-based P2P-streaming

Ключевые аспекты, на которых необходимо фокусироваться:

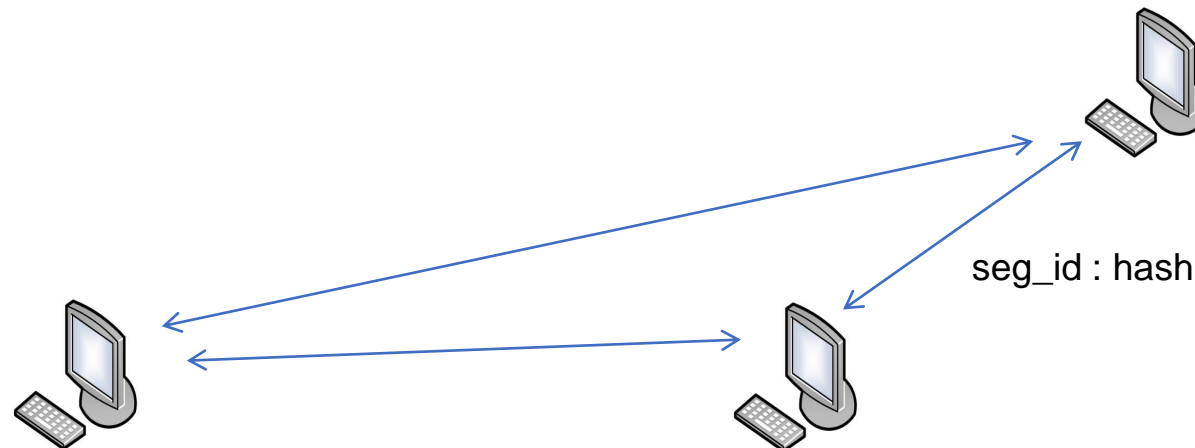
- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме
- Стриминг и ре-стриминг
- Проверка целостности сегмента (hash broadcast)



Maximize CDN-offload with WebRTC-based P2P-streaming

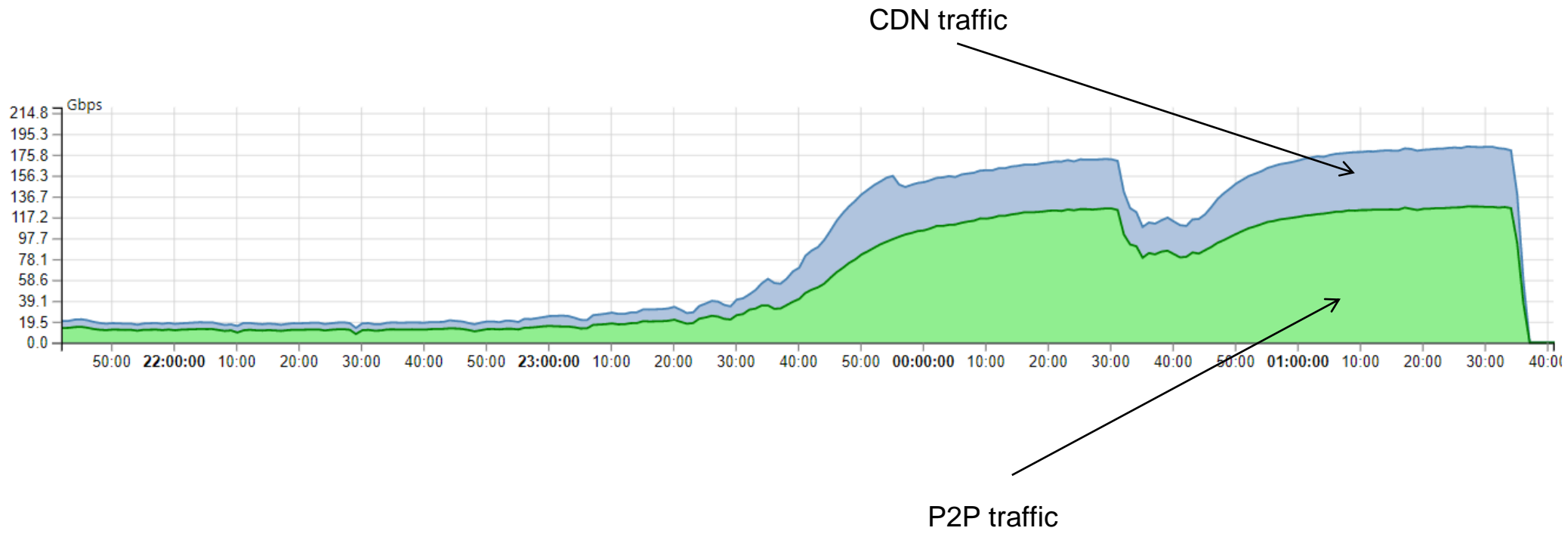
Ключевые аспекты, на которых необходимо фокусироваться:

- Поиск и соединение пиров
- Стратегия рассылки «have» сообщений
- Блэклист
- Поиск в сворме
- Стриминг и ре-стриминг
- Проверка целостности сегмента (hash broadcast)



Maximize CDN-offload with WebRTC-based P2P-streaming

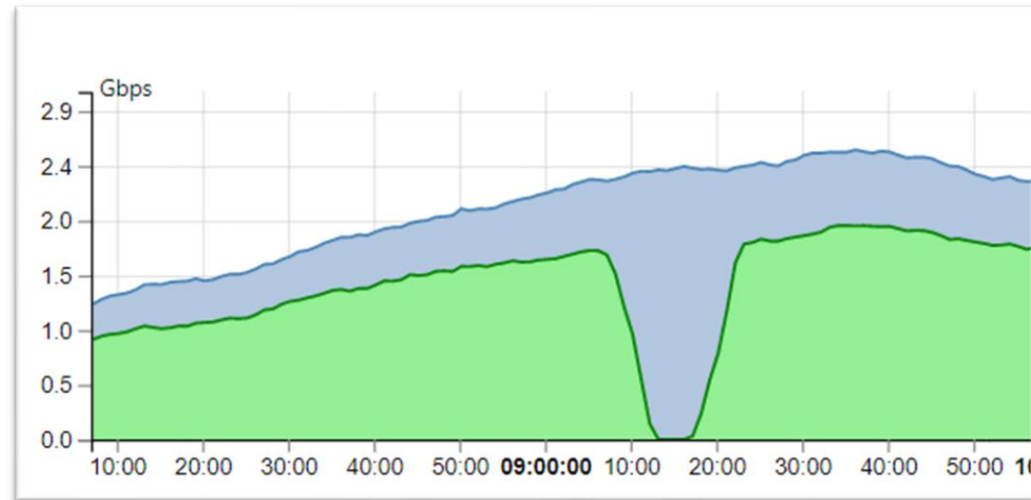
Что делать с большими нагрузками?



Maximize CDN-offload with WebRTC-based P2P-streaming

Что делать с большими нагрузками?

- Резервный поиск



Maximize CDN-offload with WebRTC-based P2P-streaming

Что делать с большими нагрузками?

- Резервный поиск



Swarm (http)

index	Bucket
000	02f9282511f98c03
001	247458e5485a6159
010	51e08a77d0263857
011	7e12d606c7ee53d0
100	801af0c490d8455c
101	abcbf9030d2ac35b
110	c9b4704b024e9e08
111	e362e43af665b89a

Maximize CDN-offload with WebRTC-based P2P-streaming

Что делать с большими нагрузками?

- Резервный поиск



Swarm (http)

index	Bucket	Prefix	
		from	to
000	02f9282511f98c03	00	1F
001	247458e5485a6159	20	3F
010	51e08a77d0263857	40	5F
011	7e12d606c7ee53d0	60	7F
100	801af0c490d8455c	80	9F
101	abcbf9030d2ac35b	A0	BF
110	c9b4704b024e9e08	C0	DF
111	e362e43af665b89a	E0	FF

Maximize CDN-offload with WebRTC-based P2P-streaming

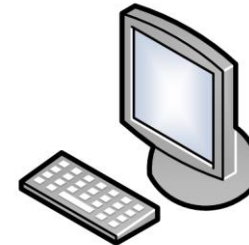
Что делать с большими нагрузками?

- Резервный поиск

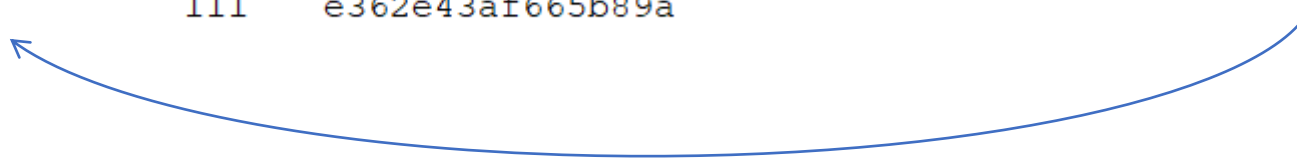


Swarm (http)

index	Bucket
000	02f9282511f98c03
001	247458e5485a6159
010	51e08a77d0263857
011	7e12d606c7ee53d0
100	801af0c490d8455c
101	abcbf9030d2ac35b
110	c9b4704b024e9e08
111	e362e43af665b89a



Peer A



73989005b0b7afe3

Maximize CDN-offload with WebRTC-based P2P-streaming

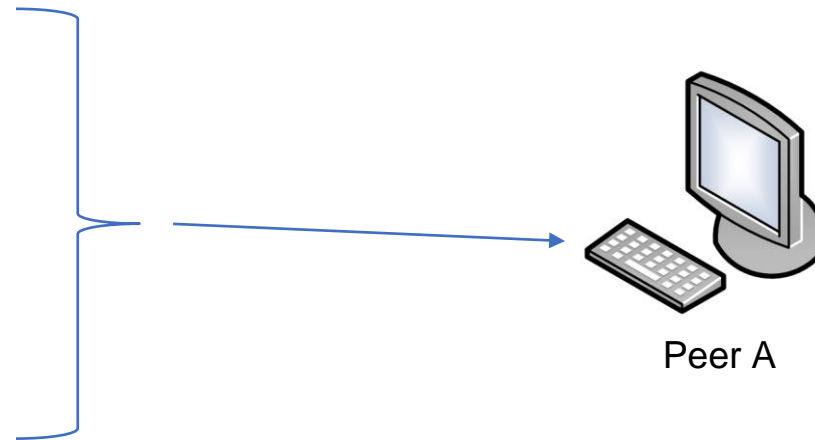
Что делать с большими нагрузками?

- Резервный поиск



Swarm (http)

index	Bucket
000	02f9282511f98c03
001	247458e5485a6159
010	51e08a77d0263857
011	7e12d606c7ee53d0
100	801af0c490d8455c
101	abcbf9030d2ac35b
110	c9b4704b024e9e08
111	e362e43af665b89a



Maximize CDN-offload with WebRTC-based P2P-streaming

Что делать с большими нагрузками?

- Резервный поиск



Swarm (http)

index	Bucket
000	02f9282511f98c03
001	247458e5485a6159
010	51e08a77d0263857
011	73989005b0b7afe3
100	801af0c490d8455c
101	abcbf9030d2ac35b
110	c9b4704b024e9e08
111	e362e43af665b89a

73989005b0b7afe3

Maximize CDN-offload with WebRTC-based P2P-streaming

Что делать с большими нагрузками?

- Резервный поиск
- Автоматический скейлинг бэкенда

Maximize CDN-offload with WebRTC-based P2P-streaming

Что делать с большими нагрузками?

- Резервный поиск
- Автоматический скейлинг бэкенда

 **Aggregation manager 2.0** APP 8:39 AM

Changing parameters of the cluster. MIN_NODE: 1; MAX_NODE: 1

Avg_cpu_idle is 78.3885%

 **Autoscaler** APP 8:39 AM

The minimum size of cluster is 1 nodes. There is/are only 0 node(s) in the cluster right now. Starting new nodes.

The new node stat-node-93 has been added to the cluster

Maximize CDN-offload with WebRTC-based P2P-streaming

Что делать с большими нагрузками?

- Резервный поиск
- Автоматический скейлинг бэкенда
- Агрегация статистики
для уменьшения нагрузки на stat.teleport.media

Maximize CDN-offload with WebRTC-based P2P-streaming

Что делать с большими нагрузками?

- Резервный поиск
- Автоматический скейлинг бэкенда
- Агрегация статистики
для уменьшения нагрузки на stat.teleport.media

```
Headers Payload Preview Response Initiator Timing
▼ Query String Parameters view source view URL-encoded
ts: 1663758129933
qq: 17
qt: 1
qd: 5
qbf: 29.629
rseg: 48353ea5
rsrc: PDN
rtrg: cf97f0f0da4fd69f
rr: 0-
rf: 16786437
rh: 101
rwt: -1
rdt: 4800
rs: 1054505
rsps: 2
rss: 19
rsst: 22
rc: true
rm: 3
ra: true
ttrs: 1663758128252
tthv: 0
ttsn: 217
ttfb: 277
ttlb: 1665
ttfn: 1679
```

Maximize CDN-offload with WebRTC-based P2P-streaming

А что на счёт VOD стриминга?

Maximize CDN-offload with WebRTC-based P2P-streaming

А что на счёт VOD стриминга?

- Эффективность для LIVE до 75%

Maximize CDN-offload with WebRTC-based P2P-streaming

А что на счёт VOD стриминга?

- Эффективность для LIVE до 75%
- Эффективность VOD Catch up до 60-70%
- Эффективность VOD Long tail до 0-10%

Maximize CDN-offload with WebRTC-based P2P-streaming

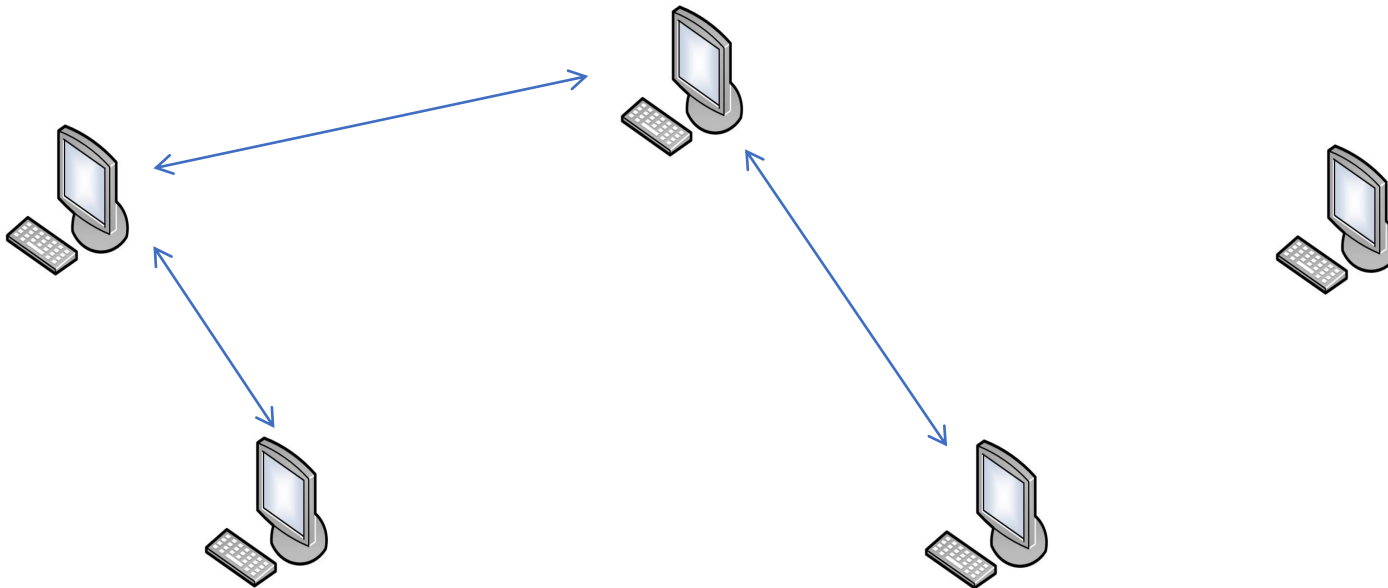
Так возник проект NODR с его позиционированием для:

- Пиров, попавших в блеклист (или вообще не имеющих webrtc)
- VOD стриминга

Maximize CDN-offload with WebRTC-based P2P-streaming

Так возник проект NODR с его позиционированием для:

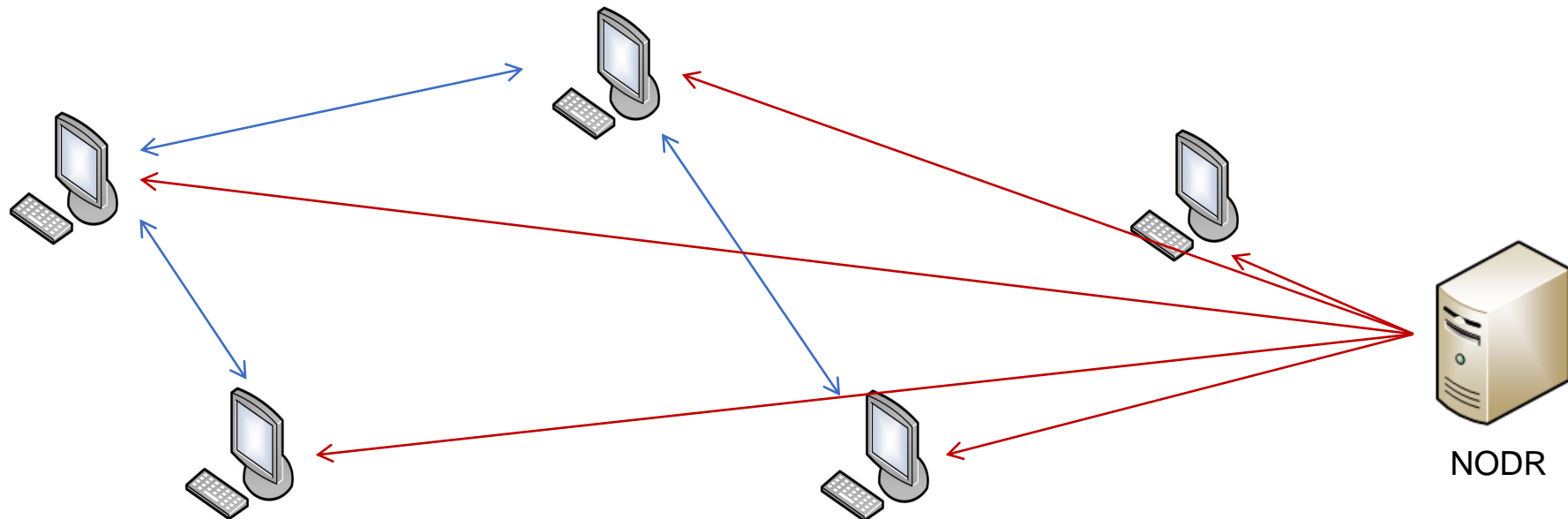
- Пиров, попавших в блеклист (или вообще не имеющих webrtc)
- VOD стриминга



Maximize CDN-offload with WebRTC-based P2P-streaming

Так возник проект NODR с его позиционированием для:

- Пиров, попавших в блеклист (или вообще не имеющих webrtc)
- VOD стриминга



Maximize CDN-offload with WebRTC-based P2P-streaming

В процессе R&D проекта NODR реализовали передачу сегментов через:

- WebRTC



Peer A



NODR

Maximize CDN-offload with WebRTC-based P2P-streaming

В процессе R&D проекта NODR реализовали передачу сегментов через:

- WebRTC

```
github.com/sctplab/usrctp/blob/7c31bd35c79ba67084ce029511193a19ceb97447/usrctplib/netinet/sctp_cc_functions.c

1145         net->ssthresh = (uint32_t)((uint64_t)t_cwnd - cc_delta);
1146     } else {
1147         net->ssthresh = net->mtu;
1148     }
1149 }
1150 if ((net->cwnd > t_cwnd) &&
1151     (net->ssthresh < net->mtu))
1152     net->ssthresh = net->mtu;
1153 }
1154 if (net->ssthresh < net->mtu)
1155     net->ssthresh = net->mtu;
1156 }
1157 } else {
1158     net->ssthresh = max(t_cwnd, net->mtu);
1159 }
1160 net->cwnd = net->mtu;
1161 net->partial_bytes_acked = 0;
```

```
github.com/sctplab/usrctp/blob/7c31bd35c79ba67084ce029511193a19ceb97447/usrctplib/netinet/sctp_cc_functions.c

2355 static void
2356 sctp_htcp_cwnd_update_after_timeout(struct sctp_tcb *stcb,
2357     struct sctp_nets *net)
2358 {
2359     int old_cwnd = net->cwnd;
2360
2361     /* JRS - reset as if the state were being changed to timeout */
2362     htcp_reset(&net->cc_mod.htcp_ca);
2363     net->ssthresh = htcp_recalc_ssthresh(net);
2364     net->cwnd = net->mtu;
2365     net->partial_bytes_acked = 0;
2366     if (SCTP_BASE_SYSCTL(sctp_logging_level) & SCTP_CWND_MONITOR_ENABLE) {
2367         sctp_log_cwnd(stcb, net, net->cwnd - old_cwnd, SCTP_CWND_LOG_FROM_RTX);
2368     }
2369 }
2370
```

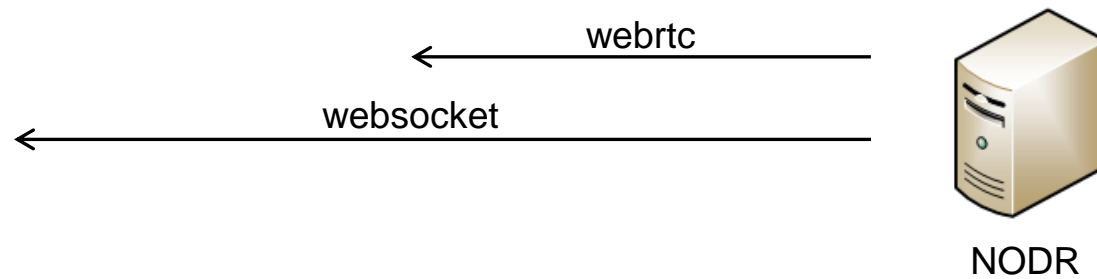
Maximize CDN-offload with WebRTC-based P2P-streaming

В процессе R&D проекта NODR реализовали передачу сегментов через:

- WebRTC
- WebSocket



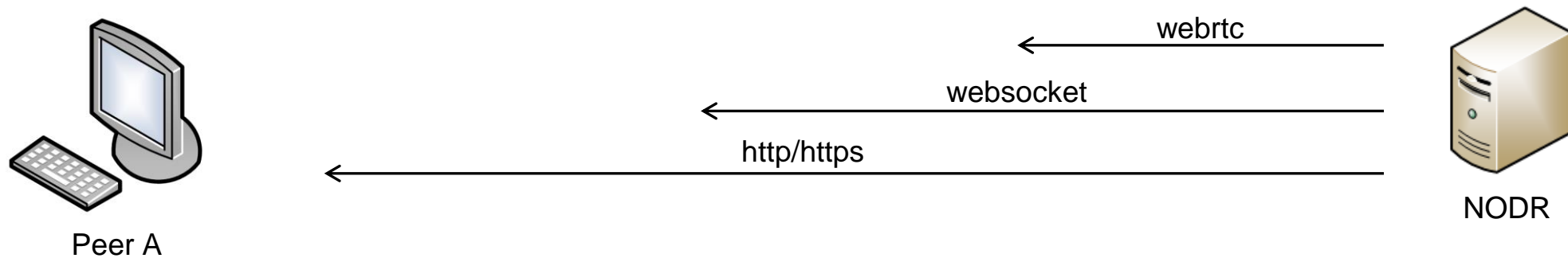
Peer A



Maximize CDN-offload with WebRTC-based P2P-streaming

В процессе R&D проекта NODR реализовали передачу сегментов через:

- WebRTC
- WebSocket
- HTTP/HTTPS



Maximize CDN-offload with WebRTC-based P2P-streaming

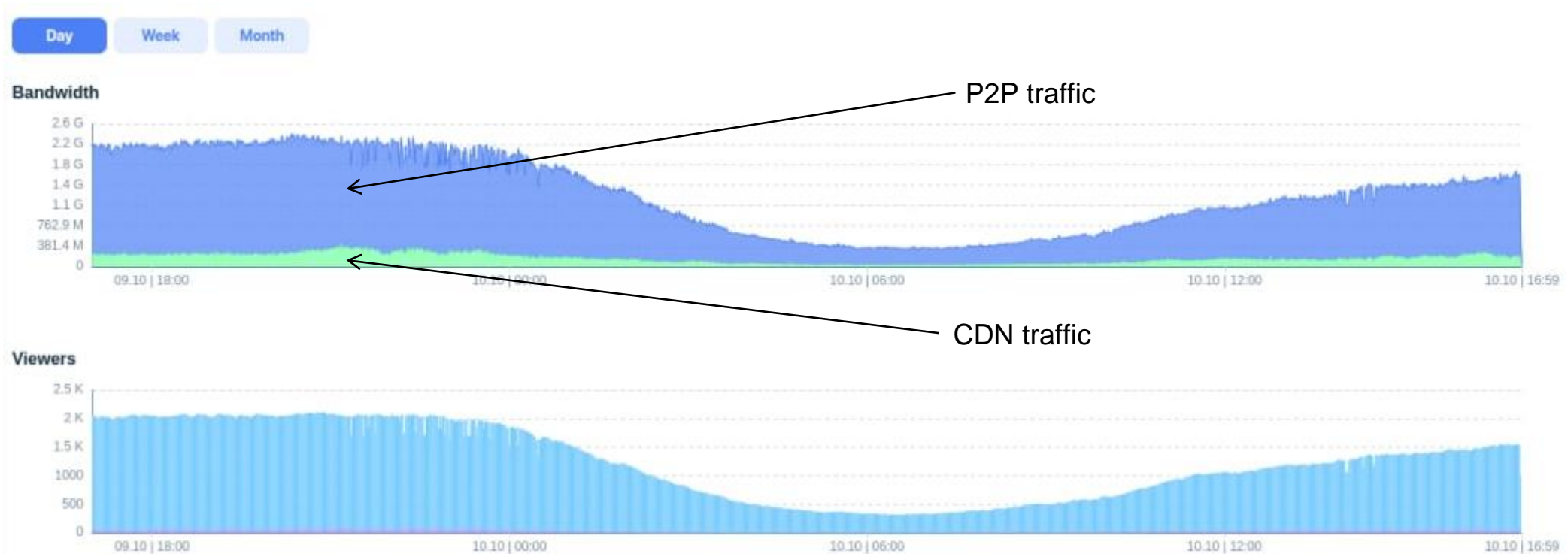
В настоящий момент NODR (C++):

- Использует только HTTP/HTTPS протокол
- Работает как проху или edge CDN сервер

Maximize CDN-offload with WebRTC-based P2P-streaming

В настоящий момент NODR (C++):

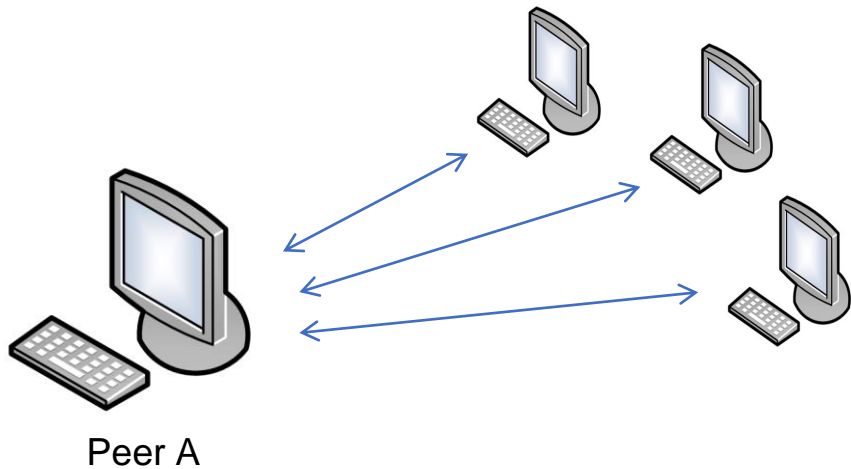
- Использует только HTTP/HTTPS протокол
- Работает как проху или edge CDN сервер
- Позволяет достичь до 95% эффективности CDN оффлоада



Maximize CDN-offload with WebRTC-based P2P-streaming

А скрипт teleport имеет иерархию из трёх уровней источников сегмента:

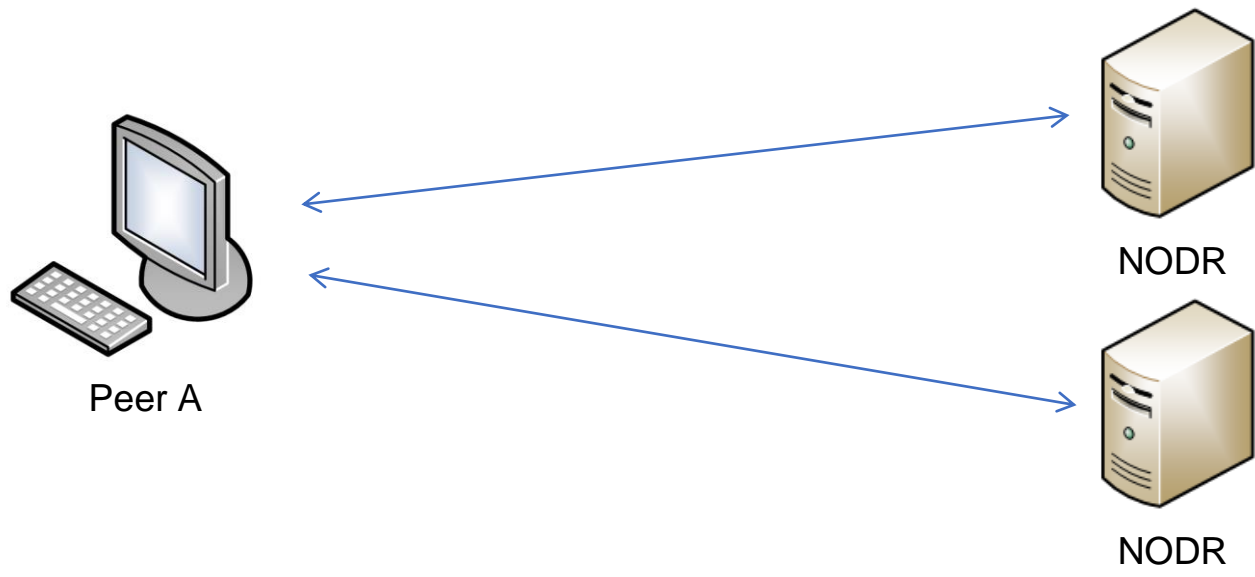
- Пиринг



Maximize CDN-offload with WebRTC-based P2P-streaming

А скрипт teleport имеет иерархию из трёх уровней источников сегмента:

- Пиринг
- Узлы NODR



Maximize CDN-offload with WebRTC-based P2P-streaming

А скрипт teleport имеет иерархию из трёх уровней источников сегмента:

- Пиринг
- Узлы NODR
- Оригинальный CDN (адреса, по которым плеер запрашивает сегменты)



Maximize CDN-offload with WebRTC-based P2P-streaming

Возвращаемся в Live стримингу

- Можно ли и как преодолеть слабые места webrtc?
- Пробуем концепцию race download

Maximize CDN-offload with WebRTC-based P2P-streaming

Разделим работу по загрузке сегмента на несколько пиоров
(тех, которые уже прислали have для нужного сегмента)



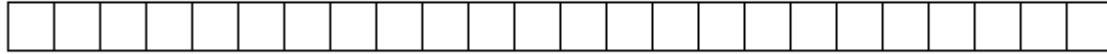
Maximize CDN-offload with WebRTC-based P2P-streaming

Разделим работу по загрузке сегмента на несколько пиров
(тех, которые уже прислали have для нужного сегмента)

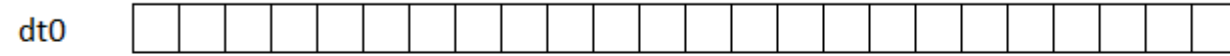


Maximize CDN-offload with WebRTC-based P2P-streaming

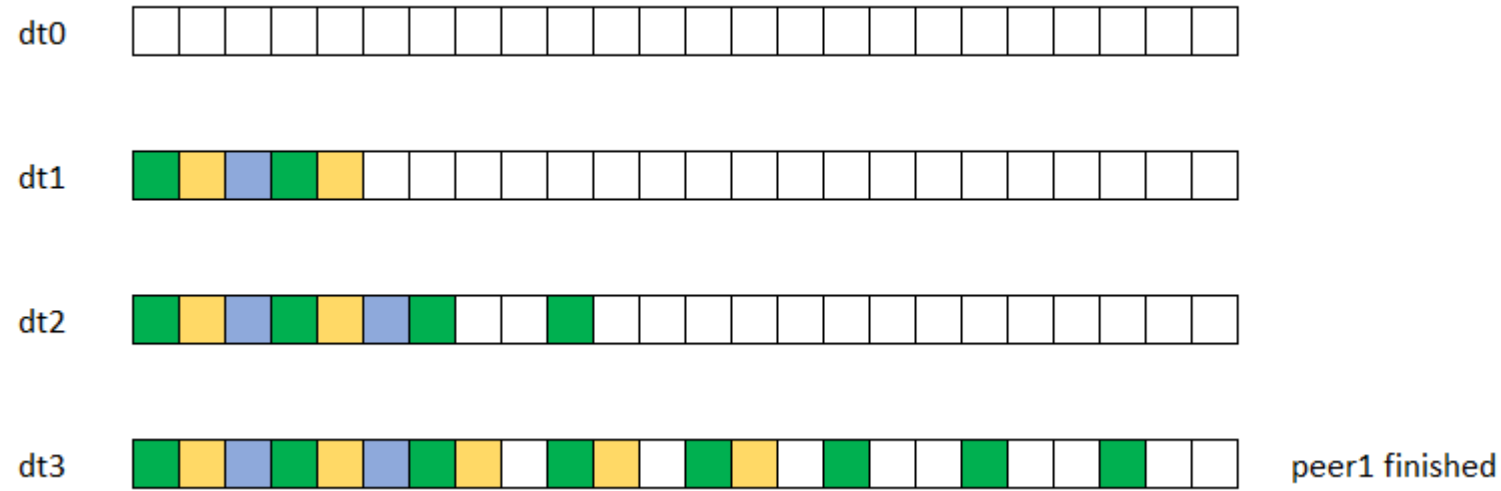
dt0



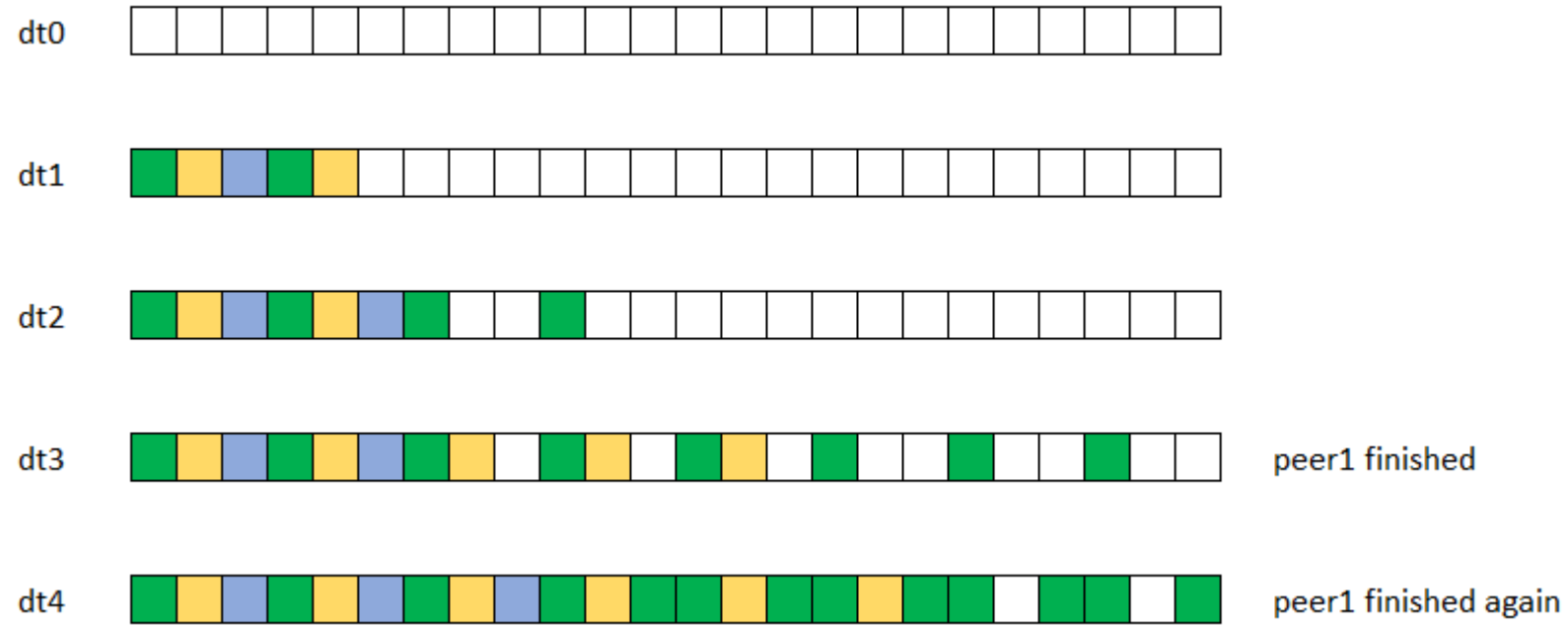
Maximize CDN-offload with WebRTC-based P2P-streaming



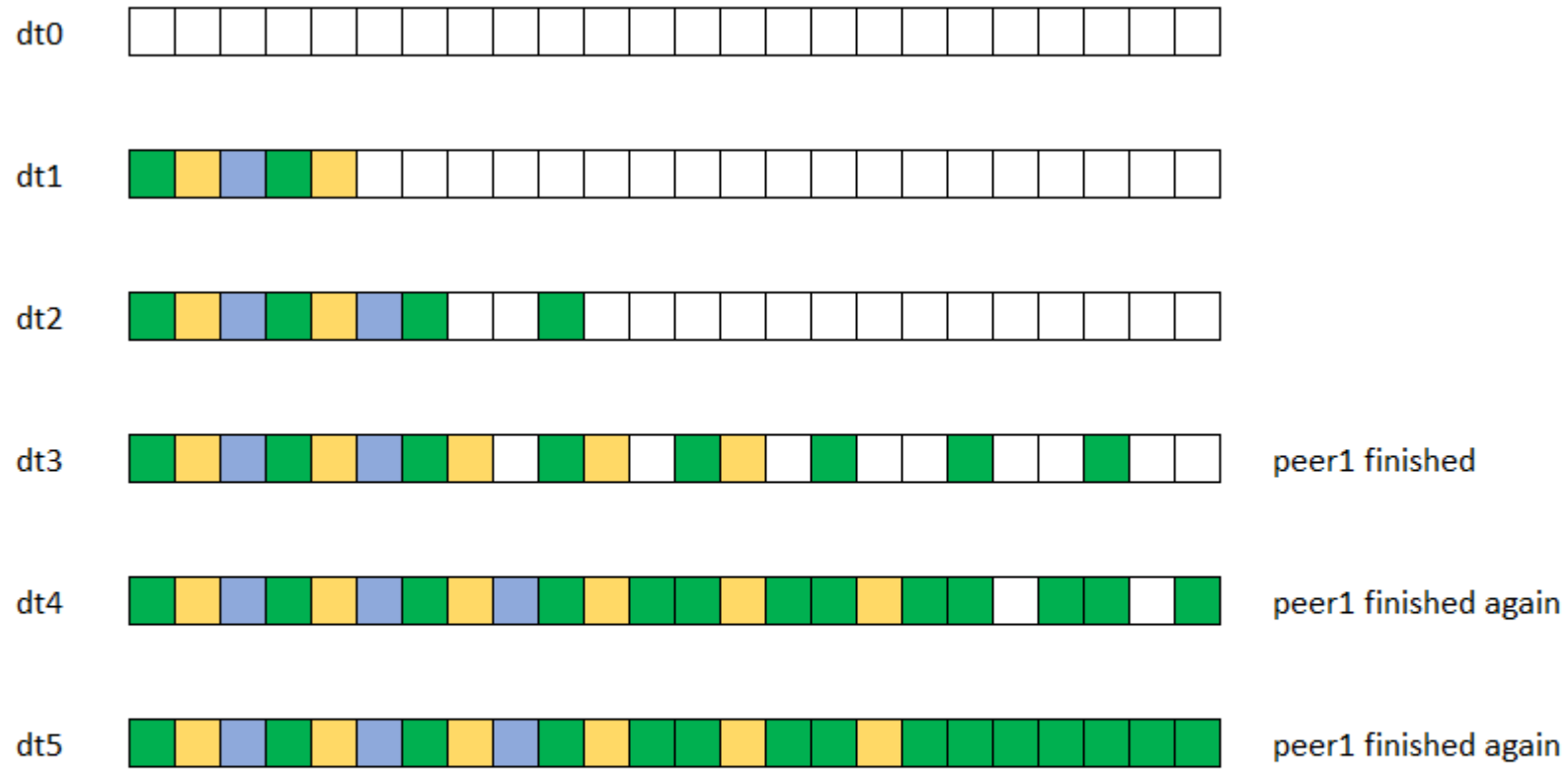
Maximize CDN-offload with WebRTC-based P2P-streaming



Maximize CDN-offload with WebRTC-based P2P-streaming



Maximize CDN-offload with WebRTC-based P2P-streaming



peer1	15	chunks
peer2	6	chunks
peer3	3	chunks

Maximize CDN-offload with WebRTC-based P2P-streaming

Как контролировать целостность собираемого из нескольких источников сегмента?

Maximize CDN-offload with WebRTC-based P2P-streaming

Как контролировать целостность собираемого из нескольких источников сегмента?

И как делать это «прямо в процессе» скачивания, так чтобы минимизировать время, которое будет потрачено впустую в случае ошибки проверки целостности?

Maximize CDN-offload with WebRTC-based P2P-streaming



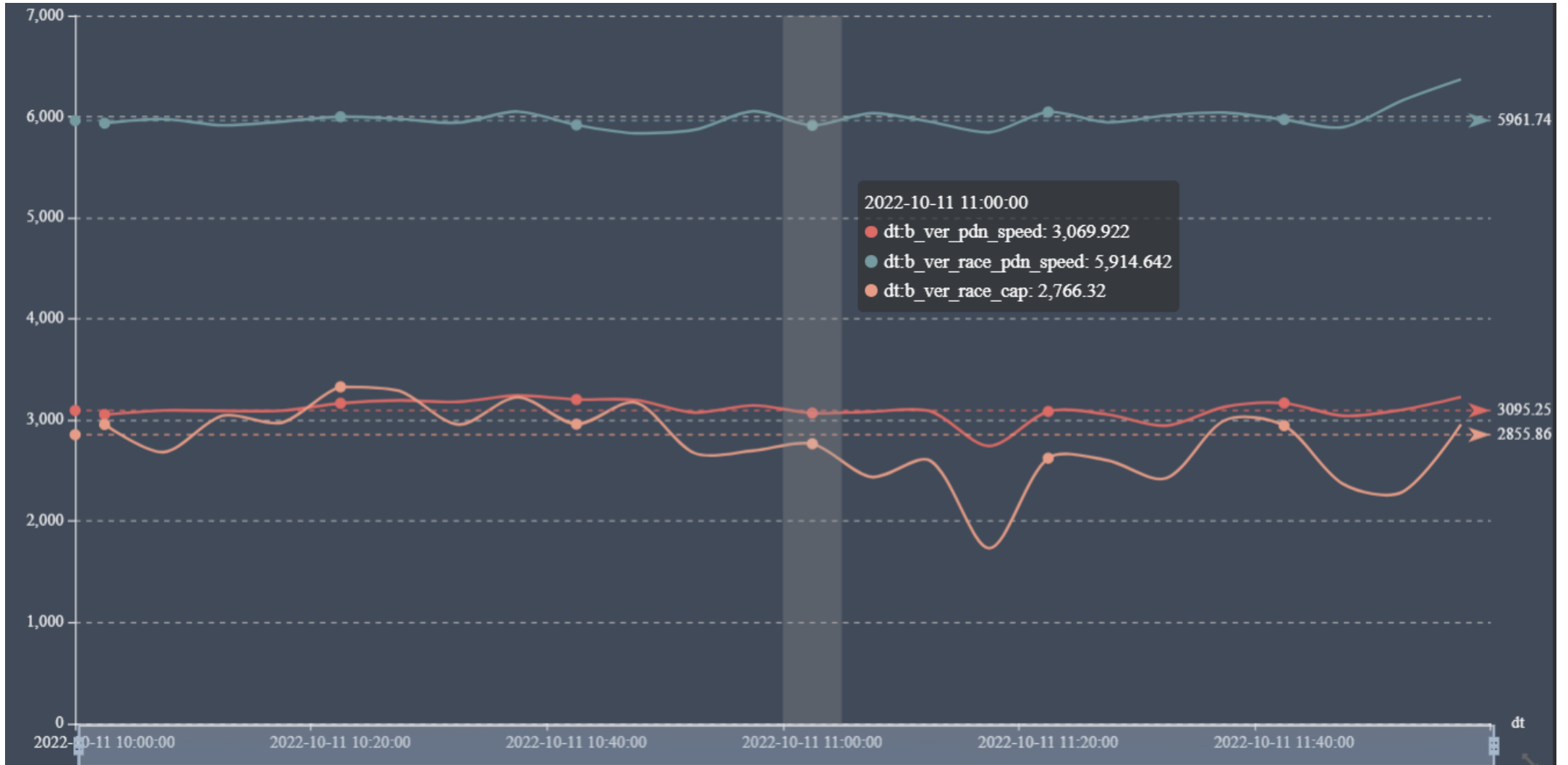
Maximize CDN-offload with WebRTC-based P2P-streaming

Таким образом Race Download:

- Помогает ускорить загрузку из пиринга за счёт нескольких источников
- В процессе перераспределяет работу на самые быстрые источники
- Обеспечивает целостность получаемых данных
- Минимизирует время, теряемое в случае ошибок проверки целостности

Race Download – ключ к Low Latency и High Definition видео стримингу

Maximize CDN-offload with WebRTC-based P2P-streaming



Maximize CDN-offload with WebRTC-based P2P-streaming

Спасибо за внимание!

Вопросы