

# Create groups without establishing direct connections

## Problems

Current groups design involves creating two connections per member - a group and a direct connection (for member's contact), even though current setting for allowing direct messages between members is off by default. This leads to double the traffic on subscriptions and especially inefficient in large groups.

Another problem is that after deleting member's contact there is no way to re-establish direct connection with member, except out-of-band.

See also [Group contacts management](#).

## Solution

Alternatives:

- [Use group connections for sending direct messages](#).
- [Don't create direct connections during group handshake, allow to create on demand](#).
- [Only create direct connections during group handshake if group allows direct messages between members](#).

### Solution 1. Use group connections for direct messages

Pros:

- Doesn't require additional negotiation.
- No additional traffic on subscriptions.

Cons:

- Breaks chat model - currently group connections and direct connections are treated as different entities on chat business logic level, this would require reworking group connections to allow to function as direct contact connections:
  - Sending messages, files, saving chat items.
  - Assigning connection to both contact and group member in connections table. Without complex schema migration, deletion of either contact or group member triggers cascade deletion of the connection. Probably no downgrade path.
  - Additional field in chat protocol message envelopes to separate group and direct messages.
  - Leaving group (and deleting group members) would require keeping connection if contact was used. We do the same thing for

- profiles currently, but profiles are local only and knowledge about connections would remain on servers; etc.
- Agent abstractions will also leak:
    - Integrity errors, skipped errors - can be received "to contact" even if bad/skipped previous messages were sent "to group member" and vice versa.
    - On decryption errors it's impossible to know intended entity - contact or group member.
    - No connection level isolation between activity as contact and activity as member.
  - Unclear contact merging. Connecting with the same person in different groups would merge their contacts, which would lead to using the same connection for different groups. Connection negotiated via host in one group would be used for another group, likely with a different threat model. Even further breaks model and connection level isolation.
  - In new planned group protocol (sparsely connected group) hosting members would be relaying messages between unconnected group members, which means not only direct connection negotiation will be carried out via host, but messages intended for contact would be passed via host in plaintext - this would be an unacceptable tradeoff.

## **Solution 2. Only create direct connections on demand**

Pros:

- Additional traffic on subscriptions only for created contacts.
- Less required changes to chat protocol and model. Connection isolation between member and contact.

Cons:

- Negotiation of a new direct connection via group connection involves several steps. In general case it will be delayed by clients availability. During negotiation sending messages should either be prohibited or messages should be created as pending.
- More traffic on subscription than with solution 1. Still probably orders less traffic for contacts than currently in large groups.

See Group contacts management rfc (link above) for design thoughts in first approximation. Though it's slightly outdated some changes to protocol messages will likely apply currently. Additional protocol messages would be required for sending invitation to a direct connection via group connection, e.g.:

```
haskell XGrpDirectInv :: ConnReqInvitation -> ChatMsgEvent 'Json
```

Both clients should take into account whether they're incognito in group - if yes, share the same profile.

### **Solution 3. Create direct connections only if group setting allows**

Same pros and cons as for solution 2.

Additional pros:

- In groups that allow direct messages between members no additional negotiation is required. This can be convenient for small groups of people who know each other, or for connecting groups of devices. On the other hand in such groups it can be easy to establish them on demand without much delay anyway.

Additional cons:

- More traffic on subscriptions than with solution 2. Since the biggest decrease in traffic is still expected to come from large groups, this may be a good trade-off. Additionally direct connection creation can be disabled not only based on group setting, but also based on current member count (same as for group receipts). The host would include it in invitation and introduction messages and member clients would know not to create additional connections.

Group state may be de-synchronized and clients can have different value for "direct messages allowed" setting - there's two options how to process such case:

- Either don't establish connection if any of group members has it as off;
- Or establish or not based on group member introduction - establish if it has direct connection even if the setting is off, since it means that's what was communicated by host to the introduced member. Optionally host can remove direct connection invitation from introduction if the rule doesn't allow it but invited member still offered it - in case of race with preferences update or incorrect behavior of invited member client.