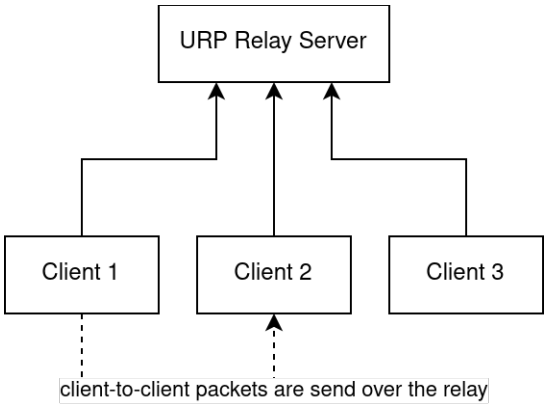
User Relay Protocol

Version 1.0

January 17, 2024

Preface

User Relay Protocol was developed to address the challenges faced by application developers, especially in the realm of multiplayer games and real time collaboration, URP introduces a novel approach to seamless and dynamic communication. It is intended to facilitate a creation of simple and interchangeable peer-to-peer like networks over TCP/IP based on a universal relay servers that have no application specific logic and can be easily implemented and deployed providing a robust solution for for decentralized communication.



The Concept

At its core, URP envisions a network architecture where application developers are liberated from the intricacies of managing server logic. Universal "Relay" servers serve as the backbone, providing a point for clients to connect. Once connected, clients have the ability to form "groups" - subsets of users within the larger network that can communicate between each other in a peer-to-peer like fashion, while the relay servers function just as facilitators.

The main benefits of this style of a network are:

- Sustainability - the client application does not rely on a centralized server but instead can facilitate any specification compliant URP relay, that can be started by any willing user.
- Decentralization - the nature of the URP network makes it so that even if a large shared relay goes down clients can still reestablish connection using any other available relay.
- Ease of use – the simplicity of the protocol and network design makes it so URP can be easily implemented and deployed.

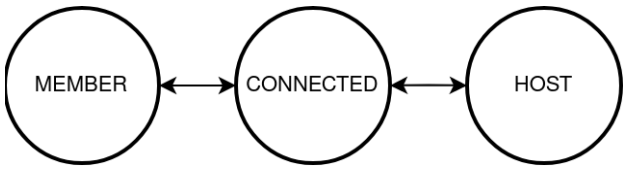
Users

Users on a URP relay are are a representation of each of the connected clients. Each user has 3 attributes:

- `UID` (User Identifier), a 32-bit, non-zero, unsigned integer unique among all the users being connected to the server at the same time, there is no requirement on the way those numbers should be chosen – the ordering is implementation defined.
- `GID` (Group Identifier) of the group this user belong to (or 0 if unused), learn more under “Groups”. Each user can be a host of just one group.
- Role, a 3 state enum describing the state the user is in, this role controls the packets the user can send to the server

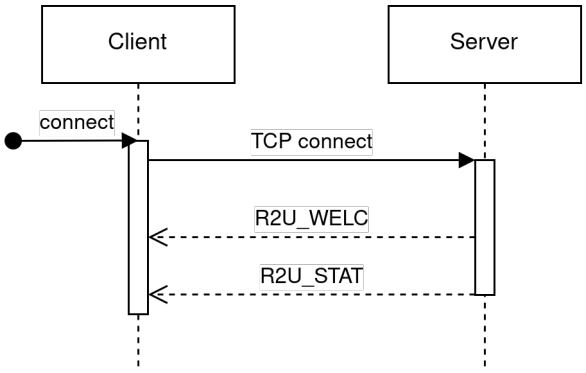
Role Name	Role Identifier
`CONNECTED`	0x01
`MEMBER`	0x02
`HOST`	0x03

The state of the role for a user can only ever change as shown on the graph below, from `CONNECTED` to `MEMBER` by sending the `U2R_JOIN` packet. And from `CONNECTED` to `HOST` by sending the `U2R_MAKE`. A user can reset it’s state back to `CONNECTED` by sending the `U2R_QUIT` packet (or by being kicked by the host using `U2R_KICK` packet). Each time the role of a user changes a `R2U_STAT` packet is send to the user to notify it of the new role. The maximum number of concurrently connected users is implementation defined.



When a new client connects with a relay the following actions are taken by the relay:

1. A new `UID` number is assigned.
2. The `CONNECTED` role is assigned.
3. A `R2U_WELC` packet is sent.
4. A `R2U_STAT` packet is sent.
5. The server starts listening for incoming packets from the new client.



Groups

Groups on a URP relay are collections of users that allow their members to communicate with each others. Each group has 7 attributes:

- `GID` (Group Identifier), a 32-bit, non-zero, unsigned integer unique among all the groups on the server, there is no requirement on the way those numbers should be chosen – the ordering is implementation defined.
- The list of `UID`s of all the members in the group, note that this list should also include the host.
- The `UID` of the host. Each group can have just one host.
- 32-bit group password.
- 32-bit group flags.
- 32-bit maximum number of users. The actual maximum number of users is a minimum of this value and of a implementation defined group size limit.
- 32-bit group signature, reserved for future use.

Table 1: Group Flags and Reserved Flag Ranges

Flag Name	Flag Value	Description
`LOCK`	0b000001	If set the group can't be joined
`NOSEND`	0b000010	If set members can't use U2R_SEND
`NOBROD`	0b000100	If set members can't use U2R_BROD
`NOP2P`	0b001000	If set U2R_SEND can only send data to host
`BINARY`	0b010000	Used to mark groups using binary data
`LORESERVED`	0x000007E0	Reserved for future use
`LOCLIENT`	0x0000F800	Reserved for client-defined semantics
`HIRESERVED`	0x00FF0000	Reserved for future use
`HICLIENT`	0xFF000000	Reserved for client-defined semantics

When a host leaves their group, the group is disbanded (all members return to the connected state) and the group is closed (removed). The maximum number of concurrently "open" groups is implementation defined. When a new group is created the following actions are taken by the relay (in relation to the group):

1. A new `GID` number is assigned.
2. The flags and password attributes are set to 0x00000000.
3. The host's `GID` number is updated to the `GID` number of the new group and the host's `UID` is stored in the group.

Packets

URP packets are send between the relay and clients over a TCP byte stream. Packets are send without padding. Each packet starts with a 1-byte header that contains the packet ID. Packet's length, structure, and contents depend on that packet type.

In the following tables in the Structure column are shown the fields that make the body of the packet, incoming data is assumed to be little ending and in multiples of 8-bit bytes. The syntax "`T[L] N`" indicates an array named "`N`" of type "`T`" and length defined by the value of field "`L`".

Table 2: User to Relay Packets

Packet Name	Identifier	Structure	Description
`U2R_MAKE`	0x80		Create user group
`U2R_JOIN`	0x81	u32 gid, u32 password	Join user group
`U2R_QUIT`	0x82		Leave user group
`U2R_BROD`	0x83	u32 uid, u32 length, u8[length] data	Broadcast message
`U2R_SEND`	0x84	u32 uid, u32 length, u8[length] data	Send message
`U2R_SETS`	0x85	u8 notify, u32 key, u32 value	Set setting
`U2R_GETS`	0x86	u32 key	Get setting
`U2R_KICK`	0x87	u32 uid	Kick user from group

Table 3: Relay to User Packets

Packet Name	Identifier	Structure	Description
`R2U_WELC`	0x00	u16 ver, u16 rev, u32 uid, u8[64] brand	Send to connecting users
`R2U_TEXT`	0x01	u32 uid, u32 length, u8[length] data	Transmits message
`R2U_MADE`	0x02	u8 code, u32 gid	Group joined/created
`R2U_JOIN`	0x03	u32 uid	Notifies host that user joined
`R2U_LEFT`	0x04	u32 uid	Notifies host that user left
`R2U_STAT`	0x05	u8 status	Notifies user of status update
`R2U_VALS`	0x06	u32 key, u32 value	Transmits key-value pair

Notes:

- The size and placement of the two first fields of the `R2U_WELC` packet (ver and rev) are guarantied to never change across URP versions, so that the protocol's version can be reliably checked.
- If the structure column was left empty that indicates that the packet has no data after the header.
- All "User to Relay" packets have the most significant bit of the identifier set to 1.

U2R_MAKE

User to relay packet, allowed for clients with the CONNECTED role, otherwise ignored. Used by clients for creating new groups and becoming their host. Has no attributes. The relay responds to this packet with the R2U_MADE and R2U_STAT packet. If an error occurs during the handling of the U2R_MAKE packet the R2U_MADE packet is still send but with a non-successful status, refer to the R2U_MADE documentation.

U2R_JOIN

User to relay packet, allowed for clients with the CONNECTED role, otherwise ignored. Used by clients for joining already existing groups as a member. Has two 32-bit attributes, GID of the group the client want to join, and the 32-bit group password. If a group of the given GID and password exists and the group in question doesn't have the LOCK flag set, then the user is added to the member list of that group, and notified of that fact by the R2U_MADE and R2U_STAT packets. If an error occurs during the handling of the U2R_JOIN packet (invalid GID, invalid password, group locked, or any other) the R2U_MADE packet is still send but with a non-successful status, refer to the R2U_MADE documentation. If a user does join the group, the group's host user is notified of that fact with the R2U_JOIN packet.

U2R_QUIT

User to relay packet, allowed for clients with the MEMBER and HOST role, otherwise ignored. Used by clients for leaving a user group. Has no attributes. The relay responds to this packet with the R2U_STAT packet. If an error occurs during the handling of the U2R_QUIT packet then nothing is send back to the client. If a member user does leave the group the group's host is notified of that fact with the R2U_LEFT packet. If the user who issued the U2R_QUIT packet was the host of the group then the group is disbanded and all members who were in that group at the time are issued the R2U_STAT packet.

U2R_BROD

User to relay packet, allowed for clients with the MEMBER and HOST role, otherwise ignored. Used by clients for broadcasting a message to all members of a group. Has two 32-bit attributes, UID of the excluded user (the user to whom the broadcast won't be send) this is typically either set to 0 in order to broadcast to everyone or to the sender's UID, after this comes the length of the message in bytes and the message itself. If an error occurs during the handling of the U2R_BROD packet or the group the sender is in has the NOBROD flag set then the packet is ignored. After receiving the packet the relay sends a R2U_TEXT packet with a copy of the broadcast message to all group members except for the excluded one.

U2R_SEND

User to relay packet, allowed for clients with the MEMBER and HOST role, otherwise ignored. Used by clients for sending a message to a specific member of a group. Has two 32-bit attributes, UID of the recipient user, and the length of the message in bytes, after that the message itself. If an error occurs during the handling of the U2R_SEND packet or the group the sender is in has the NOSEND flag set or the NOP2P flag set and the recipient UID wasn't of the group's host then the packet is ignored. After receiving the packet the relay sends a R2U_TEXT packet with a copy of the broadcast message to the recipient group member.

U2R_SETS

User to relay packet, allowed for clients of all roles. Used by users to modify certain relay states. Has one 8-bit attribute and two 32-bit attributes, 8-bit attribute is a bit field that controls the way the server will respond to this packet, 32 bit state key which is the symbolic name of a state to write to, and a 32-bit value to be written to that state. Each state key can impose it's own restrictions on the senders role independent of the packet itself, if those requirements are not met, or an error occurs the packet should be ignored. The NOTIFY control flag is only relevant if modifying a state that is stored in the group, ignored otherwise. All states whose key name starts with "GROUP" are group states. After receiving the packet the relay sends a R2U_VALS packet with a value of the requested state to all users specified by the control bit field.

Key Name	Key Value	Description	Allowed Roles
GROUP_PASS	0x01	Set group password	HOST
GROUP_FLAGS	0x02	Set group flags	HOST
GROUP_MAX	0x03	Set group member limit	HOST

Flag Name	Flag Value	Description
RESPOND	0b00000001	Send new value to sender
NOTIFY	0b00000010	Send new value to group members
RESERVED	0b11111100	Reserved for future use

U2R_GETS

User to relay packet, allowed for clients of all roles. Used by users to query certain relay states. Has one 32-bit attributes, a state key which is the symbolic name of a state to query. Each state key can impose it's own restrictions on the senders role independent of the packet itself, if those requirements are not met, or an error occurs the packet should be ignored. After receiving the packet the relay sends a R2U_VALS packet with a value of the requested state.

Key Name	Key Value	Description	Allowed Roles
GROUP_PASS	0x01	Get group password	MEMBER, HOST
GROUP_FLAGS	0x02	Get group flags	MEMBER, HOST
GROUP_MAX	0x03	Get group member limit	MEMBER, HOST

U2R_KICK

User to relay packet, allowed for clients with the HOST role, otherwise ignored. Used by hosts for forcefully disconnecting a specific user from a group. Has one 32-bit attribute, the UID of the offending user. If the UID isn't of a group member, or an error occurs, the packet is ignored. From the outside a group host using the U2R_KICK is no different from the user quitting by it's own using U2R_QUIT, same R2U_STAT packet is send back from the server to the kicked user, and R2U_LEFT to the group host. If the UID of the user to kick is equal to the UID of the group host the packet should be handled the same way as if the host sent a U2R_QUIT packet.

R2U_WELC

Relay to user packet. Send by relays to newly connected users to notify them of their UID, protocol version and relay brand. First two attributes are the 16-bit *version* and *revision*, those values specify the version of the URP used by the relay. URP clients should allow connections only with specifically supported *versions* while allowing any equal or larger *revision*, for example, client that supports URP versions *1.0*, *11.5* and *111.5* should be able to connect with relays implementing URP *1.12*, *11.6*, *111.5* but not with *2.0*, *11.0*, or *112.0*. Next attribute is a 32-bit UID of the user, and a 64-byte long string designating a implementation defined relay brand, this string can be shorter than the provided 64 bytes if a null byte (\0) is inserted into it before the end. Note that clients needs to be able to handle non-null terminated strings (strings of length equal exactly 64). If an error occurs during processing of this packet the client should terminate the connection with the relay.

R2U_TEXT

Relay to user packet. A message packet used by relays to relay the message send using the U2R_SEND packet or U2R_BROD packet to it's destination. First two attributes are the 32-bit sender UID and message length in bytes. Then follows the message itself of exactly the given length.

R2U_MADE

Relay to user packet. Used as a response to the U2R_MAKE and U2R_JOIN packets. Has two attributes, an 8-bit status code described in the table below, and a GID of the joined or created group.

Status Name	Status Value	Description
STAT_MADE	0x01	The group was successfully created
STAT_JOIN	0x02	The group was successfully joined
STAT_PASS	0x11	Invalid group authentication provided
STAT_LOCK	0x12	Joining is disabled for requested group
STAT_FULL	0x13	The group is full, unable to join
STAT_LIMIT	0x14	Too many currently open groups
STAT_FAIL	0x15	Other error occurred

R2U_JOIN

Relay to user packet. Send to the group host when a user joins their group. This packet has one 32-bit argument – the `UID` of new member.

R2U_LEFT

Relay to user packet. Send to the group host when a member leaves their group. This packet has one 32-bit argument – the `UID` of member that left the group.

R2U_STAT

Relay to user packet. Send to a user when their role (`CONNECTED`, `MEMBER`, `HOST`) changes. This packet has one 8-bit argument that contains the identifier of the new role.

R2U_VALS

Relay to user packet. Used in response to the `U2R_GETS` and `U2R_SETS` packets. This packet has two 32-bit attributes, first is the settings key, seconds is the setting value. For more information refer to the specification of `U2R_GETS` and `U2R_SETS` packets. If an error occurs during processing of this packet the client should ignore it.