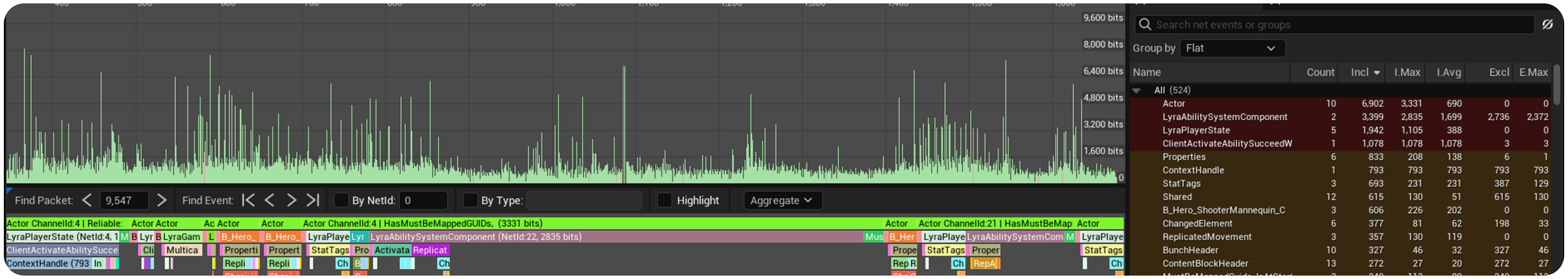
Networking Insights

Overview of Networking Insights, the network performance profiling tool



Group by Flat

Name	Count	Incl	I.Max	I.Avg	Excl	E.Max
All (524)						
Actor	10	6,902	3,331	690	0	0
LyraAbilitySystemComponent	2	3,399	2,835	1,699	2,736	2,372
LyraPlayerState	5	1,942	1,105	388	0	0
ClientActivateAbilitySucceedW	1	1,078	1,078	1,078	3	3
Properties	6	833	208	138	6	1
ContextHandle	1	793	793	793	793	793
StatTags	3	693	231	231	387	129
Shared	12	615	130	51	615	130
B_Hero_ShooterMannequin_C	3	606	226	202	0	0
ChangedElement	6	377	81	62	198	33
ReplicatedMovement	3	357	130	119	0	0
BunchHeader	10	327	46	32	327	46
ContentBlockHeader	13	272	27	20	272	27

Networking Insights is a telemetry capture and analysis tool you can use to analyze, optimize, and debug network traffic. Networking Insights is part of the **Unreal Insights** suite of tools.

Users can record trace information to visualize network behavior with the following features:

- **Packet Overview** panel visualizes packet timelines (and sizes) being transmitted or received during a game.
- **Packet Content** panel reveals packet's content, such as replicated objects, properties, and remote function calls.

- **Connection Selections** panel controls information display based on game instance, connection, and connection mode.
- **Net Stats** panel shows trace events for selected packets, including statistics about the total, maximum, and average exclusive (or inclusive) packet sizes.
- **Net Stats Counters** panel displays the stats counters recorded for the replication system.

Setup

In order to use Networking Insights, you must first ensure that Unreal Insights is running.

Using Networking Insights

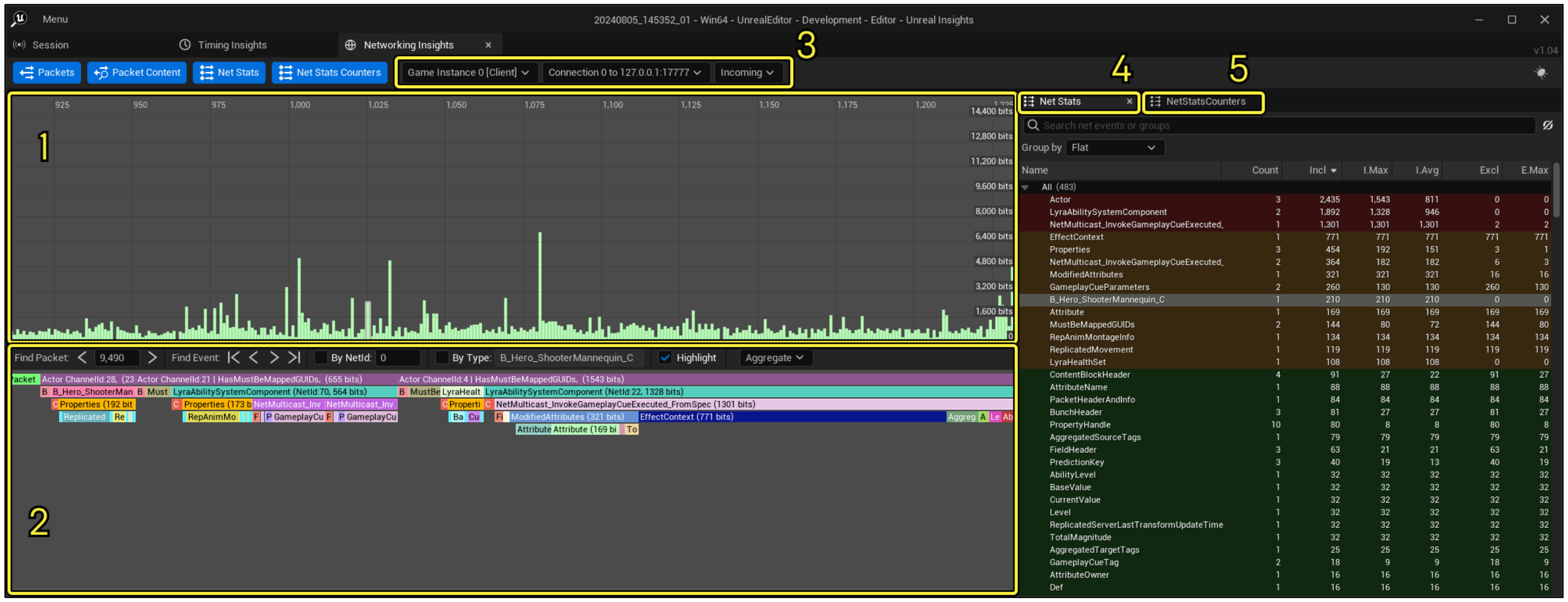
To analyze data start your client game instances with the following command-line option:

```
-trace=net -NetTrace=<VERBOSITY> [-tracehost=localhost]
```

Copy full snippet

These command-line options specify different settings for Networking Insights. The `-trace` command-line option specifies which categories Unreal Insights should trace. Specifying `-trace=net` tells Unreal Insights to start Networking Insights and trace the networking category. The `-NetTrace` command-line option specifies the verbosity of Networking Insights recordings. Set verbosity to a value of greater than zero. For example, if a game instance starts with `-NetTrace=1`, Unreal Insights collects and reports network trace data for analysis by opening the session in Unreal Insights. The editor runs its own data store for trace data, which means that you need to specify the tracehost in order to collect network trace data while running in the editor. To specify the tracehost, use: `-NetTrace=1 -trace=net -tracehost=localhost`

Networking Insights Window


Menu
20240805_145352_01 - Win64 - UnrealEditor - Development - Editor - Unreal Insights
Session
Timing Insights
Networking Insights
v1.04
Packets
Packet Content
Net Stats
Net Stats Counters
Game Instance 0 [Client]
Connection 0 to 127.0.0.1:17777
Incoming
1
2
3
4
5
14,400 bits
12,800 bits
11,200 bits
9,600 bits
8,000 bits
6,400 bits
4,800 bits
3,200 bits
1,600 bits
Find Packet: 9,490
Find Event:
By NetId: 0
By Type: B_Hero_ShooterMannequin_C
Highlight
Aggregate
Actor ChannelId:28, (23 Actor ChannelId:21 | HasMustBeMappedGUIDs, (655 bits)
Actor ChannelId:4 | HasMustBeMappedGUIDs, (1543 bits)
LyraAbilitySystemComponent (NetId:70, 564 bits)
LyraAbilitySystemComponent (NetId:22, 1328 bits)
NetMulticast_InvokeGameplayCueExecuted_FromSpec (1301 bits)
ModifiedAttributes (321 bits)
EffectContext (771 bits)
Net Stats
NetStatsCounters
Search net events or groups
Group by Flat

Name	Count	Incl	I.Max	I.Avg	Excl	E.Max
All (483)						
Actor	3	2,435	1,543	811	0	0
LyraAbilitySystemComponent	2	1,892	1,328	946	0	0
NetMulticast_InvokeGameplayCueExecuted_	1	1,301	1,301	1,301	2	2
EffectContext	1	771	771	771	771	771
Properties	3	454	192	151	3	1
NetMulticast_InvokeGameplayCueExecuted_	2	364	182	182	6	3
ModifiedAttributes	1	321	321	321	16	16
GameplayCueParameters	2	260	130	130	260	130
B_Hero_ShooterMannequin_C	1	210	210	210	0	0
Attribute	1	169	169	169	169	169
MustBeMappedGUIDs	2	144	80	72	144	80
RepAnimMontageInfo	1	134	134	134	134	134
ReplicatedMovement	1	119	119	119	119	119
LyraHealthSet	1	108	108	108	0	0
ContentBlockHeader	4	91	27	22	91	27
AttributeName	1	88	88	88	88	88
PacketHeaderAndInfo	1	84	84	84	84	84
BunchHeader	3	81	27	27	81	27
PropertyHandle	10	80	8	8	80	8
AggregatedSourceTags	1	79	79	79	79	79
FieldHeader	3	63	21	21	63	21
PredictionKey	3	40	19	13	40	19
AbilityLevel	1	32	32	32	32	32
BaseValue	1	32	32	32	32	32
CurrentValue	1	32	32	32	32	32
Level	1	32	32	32	32	32
ReplicatedServerLastTransformUpdateTime	1	32	32	32	32	32
TotalMagnitude	1	32	32	32	32	32
AggregatedTargetTags	1	25	25	25	25	25
GameplayCueTag	2	18	9	9	18	9
AttributeOwner	1	16	16	16	16	16
Def	1	16	16	16	16	16

The Networking Insights window features multiple panels providing various functionality.

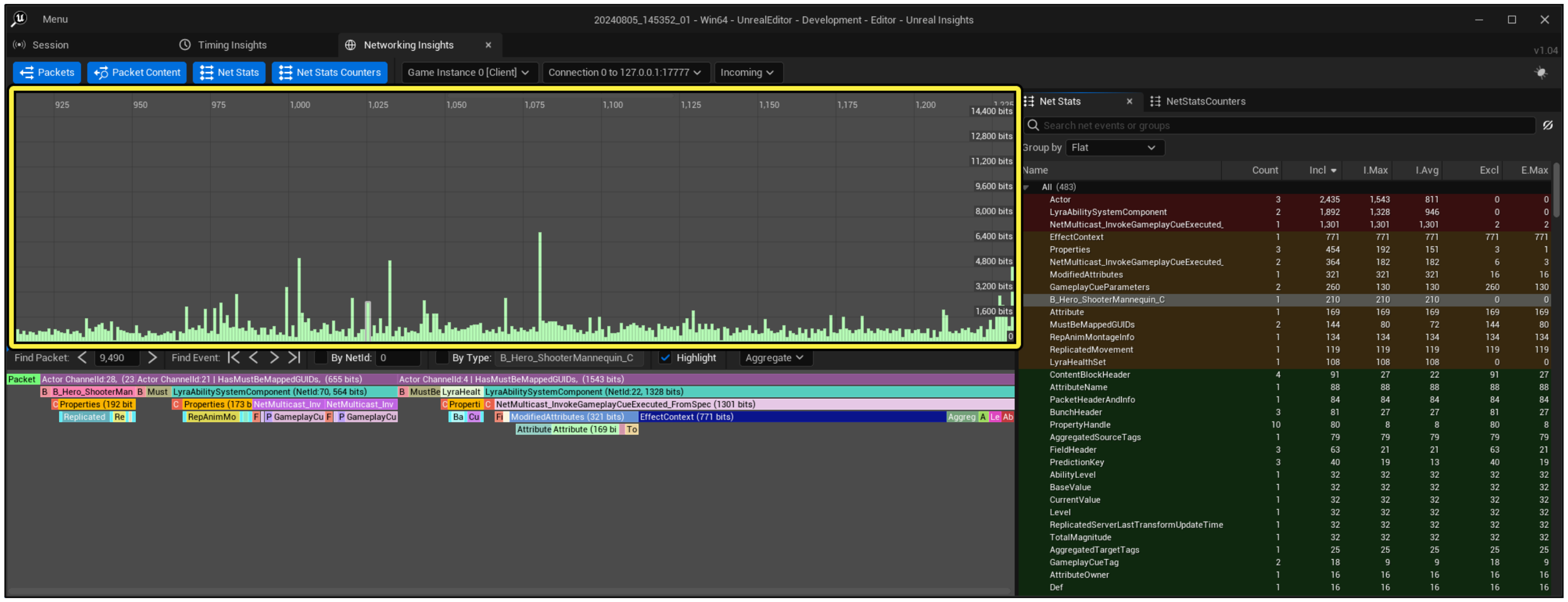
The Networking Insights window is where you'll find networking session data, including:

1. **Packet Overview** panel visualizes packet timelines (and sizes) being transmitted or received during a game.
2. **Packet Content** panel reveals packet's content, such as replicated objects, properties, and remote function calls.
3. **Connection Selections** panel controls information display based on game instance, connection, and connection mode.
4. **Net Stats** panel shows trace events for selected packets, including statistics about the total, maximum, and average exclusive (or inclusive) packet sizes.

5. **Net Stats Counters** panel tracks stats associated to a particular frame or packet.

For additional commands, tips, and information on other Unreal Insights windows , refer to the Reference documentation.

Packet Overview Panel



Name	Count	Incl	I.Max	I.Avg	Excl	E.Max
All (483)						
Actor	3	2,435	1,543	811	0	0
LyraAbilitySystemComponent	2	1,892	1,328	946	0	0
NetMulticast_InvokeGameplayCueExecuted_	1	1,301	1,301	1,301	2	2
EffectContext	1	771	771	771	771	771
Properties	3	454	192	151	3	1
NetMulticast_InvokeGameplayCueExecuted_	2	364	182	182	6	3
ModifiedAttributes	1	321	321	321	16	16
GameplayCueParameters	2	260	130	130	260	130
B_Hero_ShooterMannequin_C	1	210	210	210	0	0
Attribute	1	169	169	169	169	169
MustBeMappedGUIDs	2	144	80	72	144	80
RepAnimMontageInfo	1	134	134	134	134	134
ReplicatedMovement	1	119	119	119	119	119
LyraHealthSet	1	108	108	108	0	0
ContentBlockHeader	4	91	27	22	91	27
AttributeName	1	88	88	88	88	88
PacketHeaderAndInfo	1	84	84	84	84	84
BunchHeader	3	81	27	27	81	27
PropertyHandle	10	80	8	8	80	8
AggregatedSourceTags	1	79	79	79	79	79
FieldHeader	3	63	21	21	63	21
PredictionKey	3	40	19	13	40	19
AbilityLevel	1	32	32	32	32	32
BaseValue	1	32	32	32	32	32
CurrentValue	1	32	32	32	32	32
Level	1	32	32	32	32	32
ReplicatedServerLastTransformUpdateTime	1	32	32	32	32	32
TotalMagnitude	1	32	32	32	32	32
AggregatedTargetTags	1	25	25	25	25	25
GameplayCueTag	2	18	9	9	18	9
AttributeOwner	1	16	16	16	16	16
Def	1	16	16	16	16	16

The Packet Overview panel displays a bar graph that shows the size (in bits) for every packet received or transmitted. This is useful for identifying large packets that need further investigation. Left-click the bar to select a packet.

Hovering over a packet displays its information. This information includes:

- Packet Index
- Sequence Number
- Content Size in bits

- Total Size in bytes, with unused bits
- A timestamp for when the packet was sent or received
- The packet's current status (delivered or undelivered)
- The connection state in which the packet was received
- Engine frame number of when the packet was sent or received

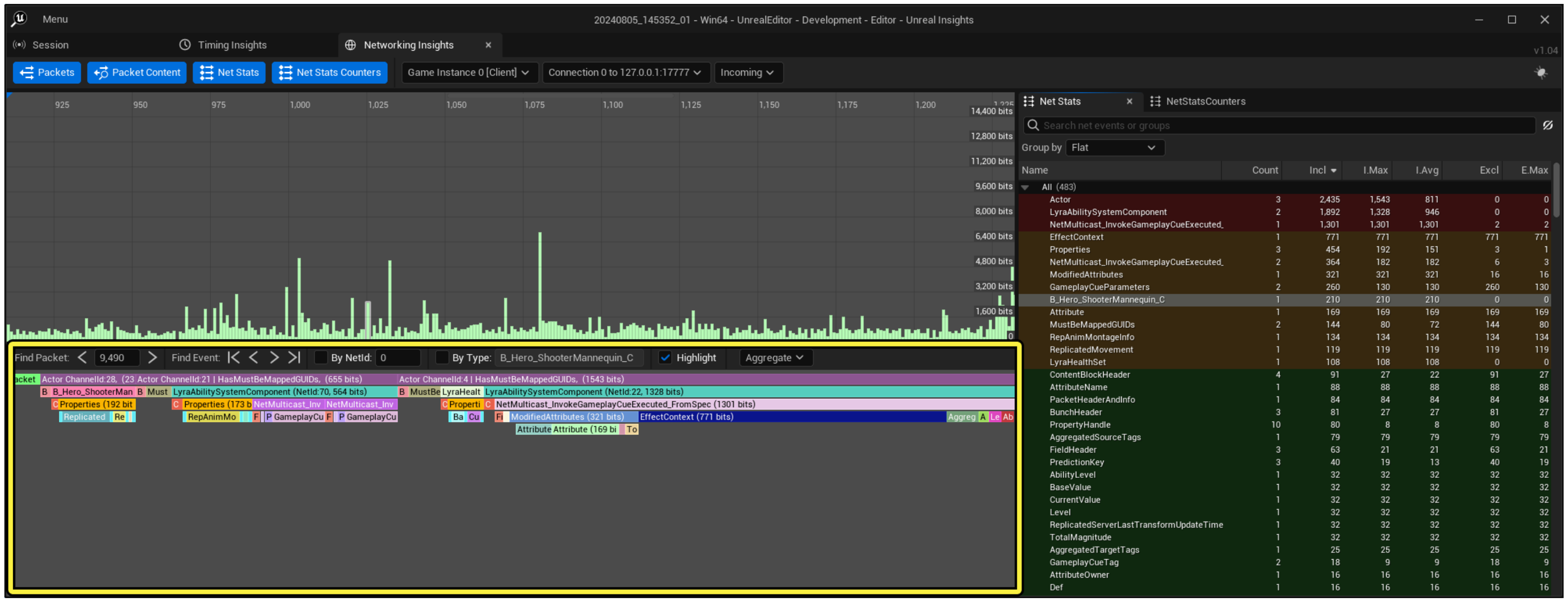
The reported size of each packet is before compression, so the actual data on the wire may be smaller than what the report shows.

Dropped packets are displayed in red. To select multiple packets, click to select an inital packet, then hold the **Shift** key and click other packets. You can highlight an event in the Packet View, to help you see how many bits that specific event is using in the packet. This makes it possible to quickly see how much bandwidth an event is using relative to other data.

You can choose to highlight either:

- The aggregated bitcount for all instances of the event; or
- The maximum bitcount used by a single event.

Packet Content Panel



Name	Count	Incl	I.Max	I.Avg	Excl	E.Max
All (483)						
Actor	3	2,435	1,543	811	0	0
LyraAbilitySystemComponent	2	1,892	1,328	946	0	0
NetMulticast_InvokeGameplayCueExecuted_	1	1,301	1,301	1,301	2	2
EffectContext	1	771	771	771	771	771
Properties	3	454	192	151	3	1
NetMulticast_InvokeGameplayCueExecuted_	2	364	182	182	6	3
ModifiedAttributes	1	321	321	321	16	16
GameplayCueParameters	2	260	130	130	260	130
B_Hero_ShooterMannequin_C	1	210	210	210	0	0
Attribute	1	169	169	169	169	169
MustBeMappedGUIDs	2	144	80	72	144	80
RepAnimMontageInfo	1	134	134	134	134	134
ReplicatedMovement	1	119	119	119	119	119
LyraHealthSet	1	108	108	108	0	0
ContentBlockHeader	4	91	27	22	91	27
AttributeName	1	88	88	88	88	88
PacketHeaderAndInfo	1	84	84	84	84	84
BunchHeader	3	81	27	27	81	27
PropertyHandle	10	80	8	8	80	8
AggregatedSourceTags	1	79	79	79	79	79
FieldHeader	3	63	21	21	63	21
PredictionKey	3	40	19	13	40	19
AbilityLevel	1	32	32	32	32	32
BaseValue	1	32	32	32	32	32
CurrentValue	1	32	32	32	32	32
Level	1	32	32	32	32	32
ReplicatedServerLastTransformUpdateTime	1	32	32	32	32	32
TotalMagnitude	1	32	32	32	32	32
AggregatedTargetTags	1	25	25	25	25	25
GameplayCueTag	2	18	9	9	18	9
AttributeOwner	1	16	16	16	16	16
Def	1	16	16	16	16	16

The packet content panel displaying bunches of data contained in a packet. This includes information about replicated objects and properties.

The Packet Content panel displays the actual content of the currently selected packet. The packet's content displays as a hierarchical event graph starting with bit zero of the packet on the left side. This is the main tool to investigate the data contents in each packet down to the property level.

Packets consist of multiple levels of information. Level 0 is at the top of the Packet Content panel and displays the bunches contained in a packet. Each bunch is named after the channel it belongs to. If the reported bunch has its debug name set when reported, the `debugname` is used instead of the `channelname` to provide additional context. In the example in the image above, this consists of: `PacketHeader`, `Actor ChannelId:28`, and `Actor ChannelId:4`.

The second level typically displays replicated objects contained in the bunch. If the event has an assigned NetId (NetGUID), the ID is displayed also. Traced events display `NetId:NN` until the event is assigned a NetId. In the example in the image above, one of the replicated objects is `LyraAbilitySystemComponent` with `NetId:22`.

Events further down the event hierarchy display replicated properties and RPCs, including how many bits they wrote. In the example image above, one of these RPCs is `NetMulticast_InvokeGameplayCueExecuted_FromSpec` consisting of

`1301` bits.

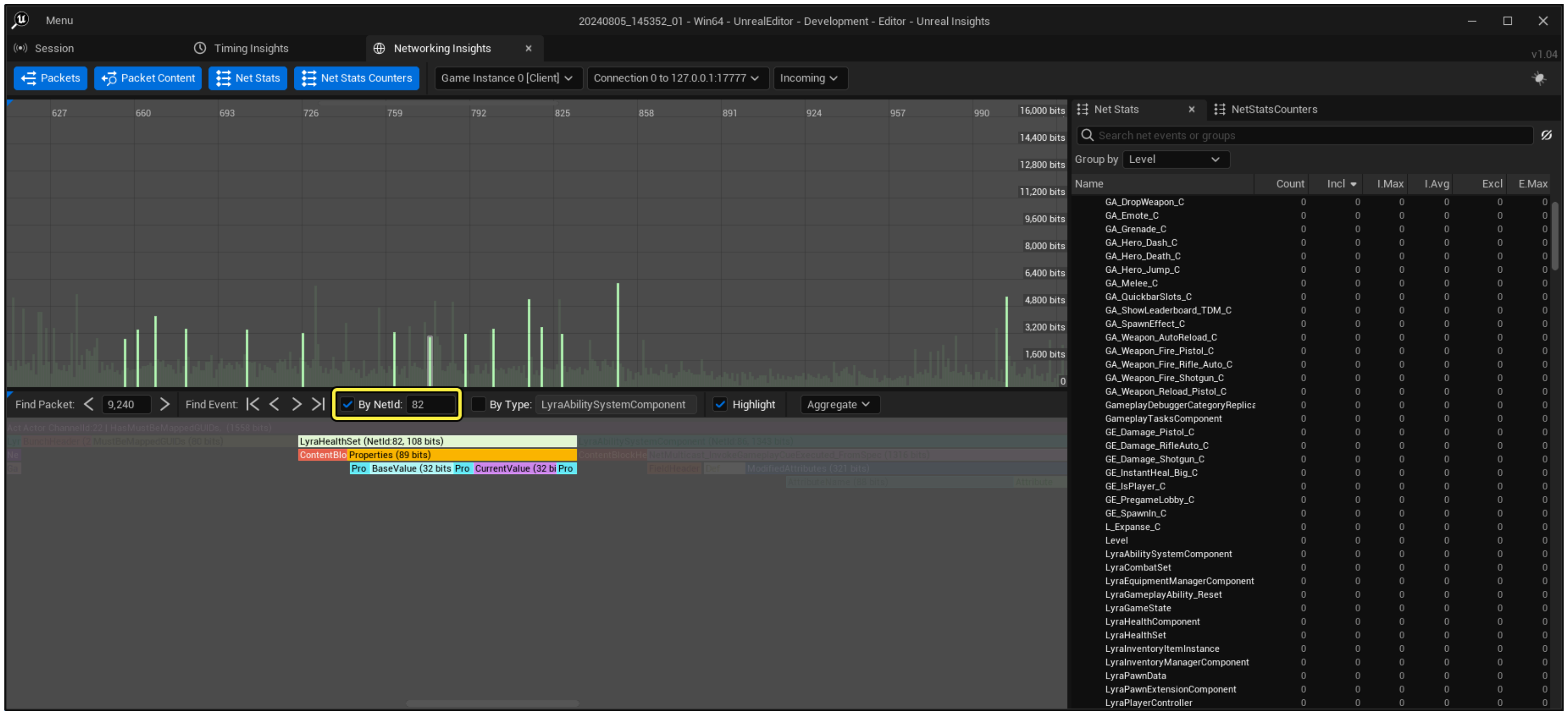
Hovering over an event displays all of the event's information.

The panel features controls to find a packet, or to find events belonging to a specific NetId.

Filtering

You can filter packet contents by NetId and event type in combination. This is useful to highlight specific events for a specific instance.

To filter by `NetId`, check the box next to **By NetId:** in the packet contents panel, then enter the `NetId` you want to filter by in the text box.

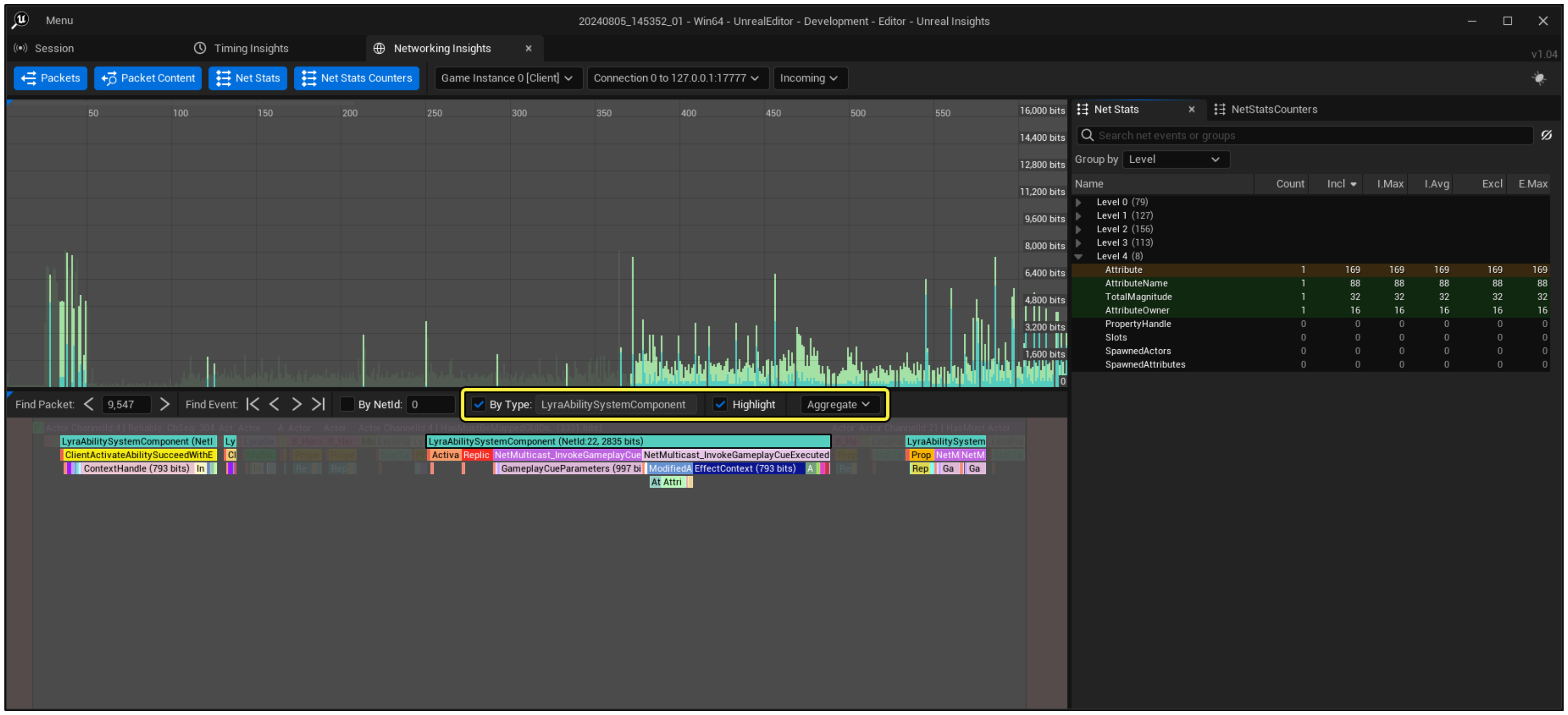


Find Packet: 9,240 Find Event: [✓ By NetId: 82] By Type: LyraAbilitySystemComponent [✓ Highlight] Aggregate

Group by Level

Name	Count	Incl	I.Max	I.Avg	Excl	E.Max
GA_DropWeapon_C	0	0	0	0	0	0
GA_Emote_C	0	0	0	0	0	0
GA_Grenade_C	0	0	0	0	0	0
GA_Hero_Dash_C	0	0	0	0	0	0
GA_Hero_Death_C	0	0	0	0	0	0
GA_Hero_Jump_C	0	0	0	0	0	0
GA_Melee_C	0	0	0	0	0	0
GA_QuickbarSlots_C	0	0	0	0	0	0
GA_ShowLeaderboard_TDM_C	0	0	0	0	0	0
GA_SpawnEffect_C	0	0	0	0	0	0
GA_Weapon_AutoReload_C	0	0	0	0	0	0
GA_Weapon_Fire_Pistol_C	0	0	0	0	0	0
GA_Weapon_Fire_Rifle_Auto_C	0	0	0	0	0	0
GA_Weapon_Fire_Shotgun_C	0	0	0	0	0	0
GA_Weapon_Reload_Pistol_C	0	0	0	0	0	0
GameplayDebuggerCategoryReplica	0	0	0	0	0	0
GameplayTasksComponent	0	0	0	0	0	0
GE_Damage_Pistol_C	0	0	0	0	0	0
GE_Damage_RifleAuto_C	0	0	0	0	0	0
GE_Damage_Shotgun_C	0	0	0	0	0	0
GE_InstantHeal_Big_C	0	0	0	0	0	0
GE_IsPlayer_C	0	0	0	0	0	0
GE_PregameLobby_C	0	0	0	0	0	0
GE_SpawnIn_C	0	0	0	0	0	0
L_Expanse_C	0	0	0	0	0	0
Level	0	0	0	0	0	0
LyraAbilitySystemComponent	0	0	0	0	0	0
LyraCombatSet	0	0	0	0	0	0
LyraEquipmentManagerComponent	0	0	0	0	0	0
LyraGameplayAbility_Reset	0	0	0	0	0	0
LyraGameState	0	0	0	0	0	0
LyraHealthComponent	0	0	0	0	0	0
LyraHealthSet	0	0	0	0	0	0
LyraInventoryItemInstance	0	0	0	0	0	0
LyraInventoryManagerComponent	0	0	0	0	0	0
LyraPawnData	0	0	0	0	0	0
LyraPawnExtensionComponent	0	0	0	0	0	0
LyraPlayerController	0	0	0	0	0	0

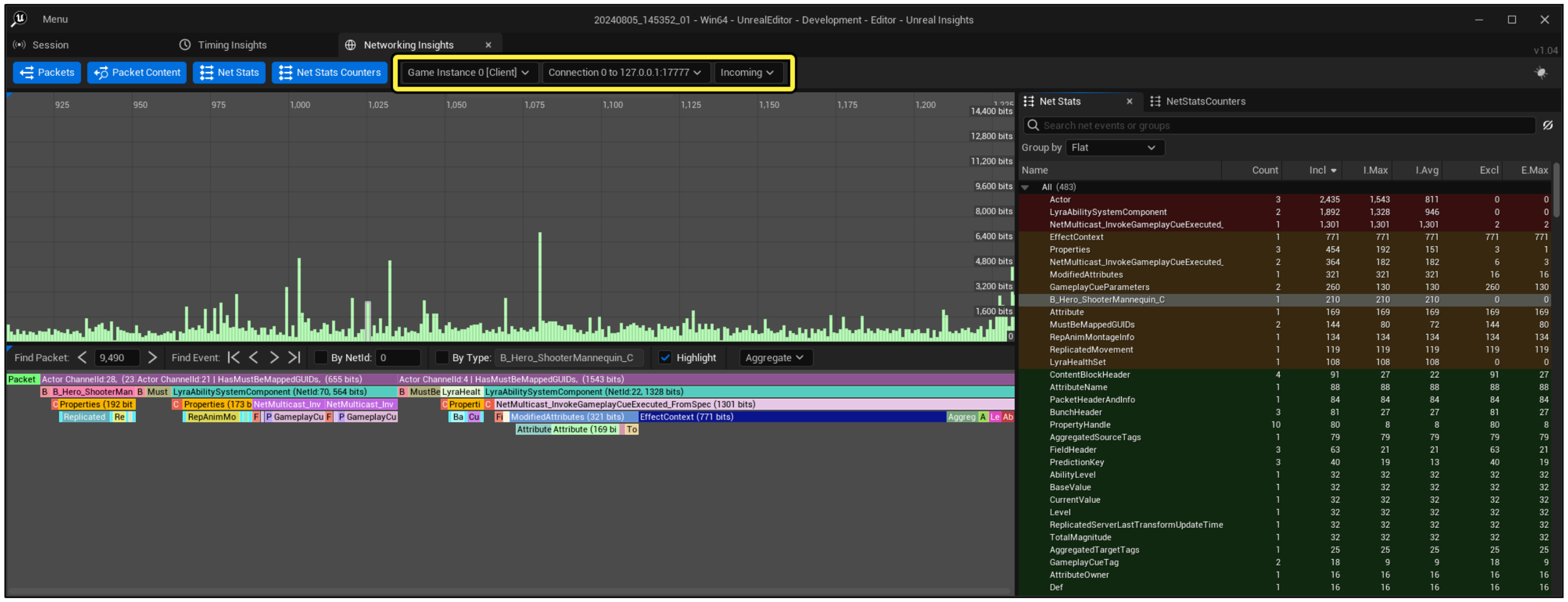
To filter by packet type, check the box next to **By Type:** in the packet contents panel, then select a packet type within the packet contents panel. The text box next to the filter **By Type:** box will populate with the type of packet content you chose. You can further filter by selecting **Highlight**, which highlights any packets that match your filter criteria.


Menu
20240805_145352_01 - Win64 - UnrealEditor - Development - Editor - Unreal Insights
Session
Timing Insights
Networking Insights
v1.04
Packets
Packet Content
Net Stats
Net Stats Counters
Game Instance 0 [Client]
Connection 0 to 127.0.0.1:17777
Incoming
Find Packet: 9,547
Find Event:
By NetId: 0
By Type: LyraAbilitySystemComponent
Highlight
Aggregate
LyraAbilitySystemComponent (NetId:22, 2835 bits)
NetMulticast_InvokeGameplayCueExecuted
GameplayCueParameters (997 bi
EffectContext (793 bits)
ContextHandle (793 bits)
ClientActivateAbilitySucceedWithE
Net Stats
NetStatsCounters
Search net events or groups
Group by Level

Name	Count	Incl	I.Max	I.Avg	Excl	E.Max
Level 0 (79)						
Level 1 (127)						
Level 2 (156)						
Level 3 (113)						
Level 4 (8)						
Attribute	1	169	169	169	169	169
AttributeName	1	88	88	88	88	88
TotalMagnitude	1	32	32	32	32	32
AttributeOwner	1	16	16	16	16	16
PropertyHandle	0	0	0	0	0	0
Slots	0	0	0	0	0	0
SpawnedActors	0	0	0	0	0	0
SpawnedAttributes	0	0	0	0	0	0

You can navigate the Packet Content Panel and Packet Overview Panel using your keyboard. The up and down arrow keys will move between levels. The left and right arrow keys will move between events on the same level. If you hold the CTRL or CMD key and press the left or right arrow keys, you will move between packets.

Connection Selection Panel



Name	Count	Incl	I.Max	I.Avg	Excl	E.Max
All (483)						
Actor	3	2,435	1,543	811	0	0
LyraAbilitySystemComponent	2	1,892	1,328	946	0	0
NetMulticast_InvokeGameplayCueExecuted_	1	1,301	1,301	1,301	2	2
EffectContext	1	771	771	771	771	771
Properties	3	454	192	151	3	1
NetMulticast_InvokeGameplayCueExecuted_	2	364	182	182	6	3
ModifiedAttributes	1	321	321	321	16	16
GameplayCueParameters	2	260	130	130	260	130
B_Hero_ShooterMannequin_C	1	210	210	210	0	0
Attribute	1	169	169	169	169	169
MustBeMappedGUIDs	2	144	80	72	144	80
RepAnimMontageInfo	1	134	134	134	134	134
ReplicatedMovement	1	119	119	119	119	119
LyraHealthSet	1	108	108	108	0	0
ContentBlockHeader	4	91	27	22	91	27
AttributeName	1	88	88	88	88	88
PacketHeaderAndInfo	1	84	84	84	84	84
BunchHeader	3	81	27	27	81	27
PropertyHandle	10	80	8	8	80	8
AggregatedSourceTags	1	79	79	79	79	79
FieldHeader	3	63	21	21	63	21
PredictionKey	3	40	19	13	40	19
AbilityLevel	1	32	32	32	32	32
BaseValue	1	32	32	32	32	32
CurrentValue	1	32	32	32	32	32
Level	1	32	32	32	32	32
ReplicatedServerLastTransformUpdateTime	1	32	32	32	32	32
TotalMagnitude	1	32	32	32	32	32
AggregatedTargetTags	1	25	25	25	25	25
GameplayCueTag	2	18	9	9	18	9
AttributeOwner	1	16	16	16	16	16
Def	1	16	16	16	16	16

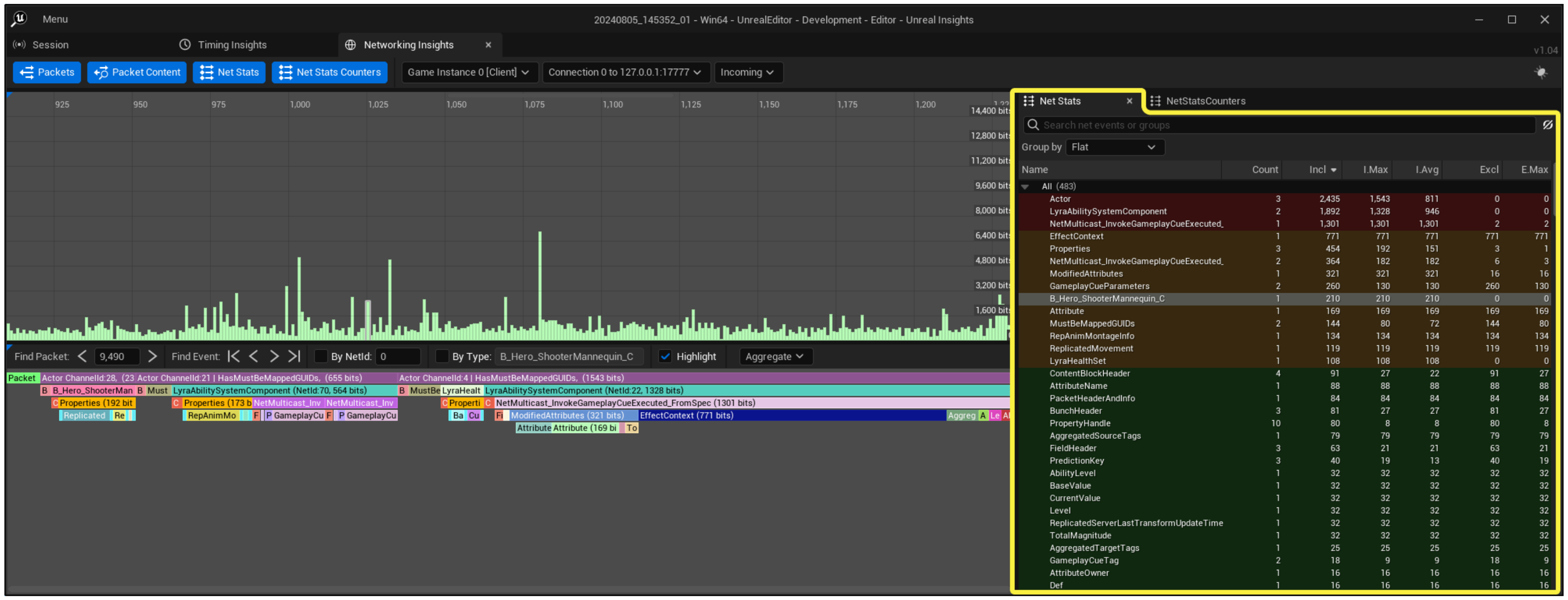
The connection dropdown lists located at the top of the Packet Overview panel. This example shows Player 0's connection in a client game instance to a localhost server instance with incoming data selected.

The top of the Packet Overview panel has controls that let users select the data to display.

Three dropdown lists are available in the Connection Selection panel:

Dropdown List	Description
Game Instance	Displays the created Game Instance for each unique NetDriver observed during a recording session. For example, when using PIE, there are separate instances for the server and each client.
Connection	This dropdown selects a specific connection, displaying every observed connection during the selected Game Instance's session.
Connection Mode	This dropdown controls whether to visualize incoming our outgoing data.

Net Stats Panel



Net Stats | NetStatsCounters

Group by: Flat

Name	Count	Incl	I.Max	I.Avg	Excl	E.Max
All (483)						
Actor	3	2,435	1,543	811	0	0
LyraAbilitySystemComponent	2	1,892	1,328	946	0	0
NetMulticast_InvokeGameplayCueExecuted_	1	1,301	1,301	1,301	2	2
EffectContext	1	771	771	771	771	771
Properties	3	454	192	151	3	1
NetMulticast_InvokeGameplayCueExecuted_	2	364	182	182	6	3
ModifiedAttributes	1	321	321	321	16	16
GameplayCueParameters	2	260	130	130	260	130
B_Hero_ShooterMannequin_C	1	210	210	210	0	0
Attribute	1	169	169	169	169	169
MustBeMappedGUIDs	2	144	80	72	144	80
RepAnimMontageInfo	1	134	134	134	134	134
ReplicatedMovement	1	119	119	119	119	119
LyraHealthSet	1	108	108	108	0	0
ContentBlockHeader	4	91	27	22	91	27
AttributeName	1	88	88	88	88	88
PacketHeaderAndInfo	1	84	84	84	84	84
BunchHeader	3	81	27	27	81	27
PropertyHandle	10	80	8	8	80	8
AggregatedSourceTags	1	79	79	79	79	79
FieldHeader	3	63	21	21	63	21
PredictionKey	3	40	19	13	40	19
AbilityLevel	1	32	32	32	32	32
BaseValue	1	32	32	32	32	32
CurrentValue	1	32	32	32	32	32
Level	1	32	32	32	32	32
ReplicatedServerLastTransformUpdateTime	1	32	32	32	32	32
TotalMagnitude	1	32	32	32	32	32
AggregatedTargetTags	1	25	25	25	25	25
GameplayCueTag	2	18	9	9	18	9
AttributeOwner	1	16	16	16	16	16
Def	1	16	16	16	16	16

The Net Stats panel with events grouped by level.

The Net Stats panel lists all trace events for the selected packet range in the Packet Overview panel.

In addition to aggregating data based on a selected range, users can sort the list in ascending or descending order by the values in any active column. Right-click anywhere in the list to change the sort order, or to activate or deactivate columns. The following columns are available:

- Type
- Level
- Instance Count
- Total Inclusive Size (bits)
- Max Inclusive Size (bits)
- Total Exclusive Size
- Max Exclusive Size

To filter events by name, use the "search net events or groups" field. Users can group according to these event types:

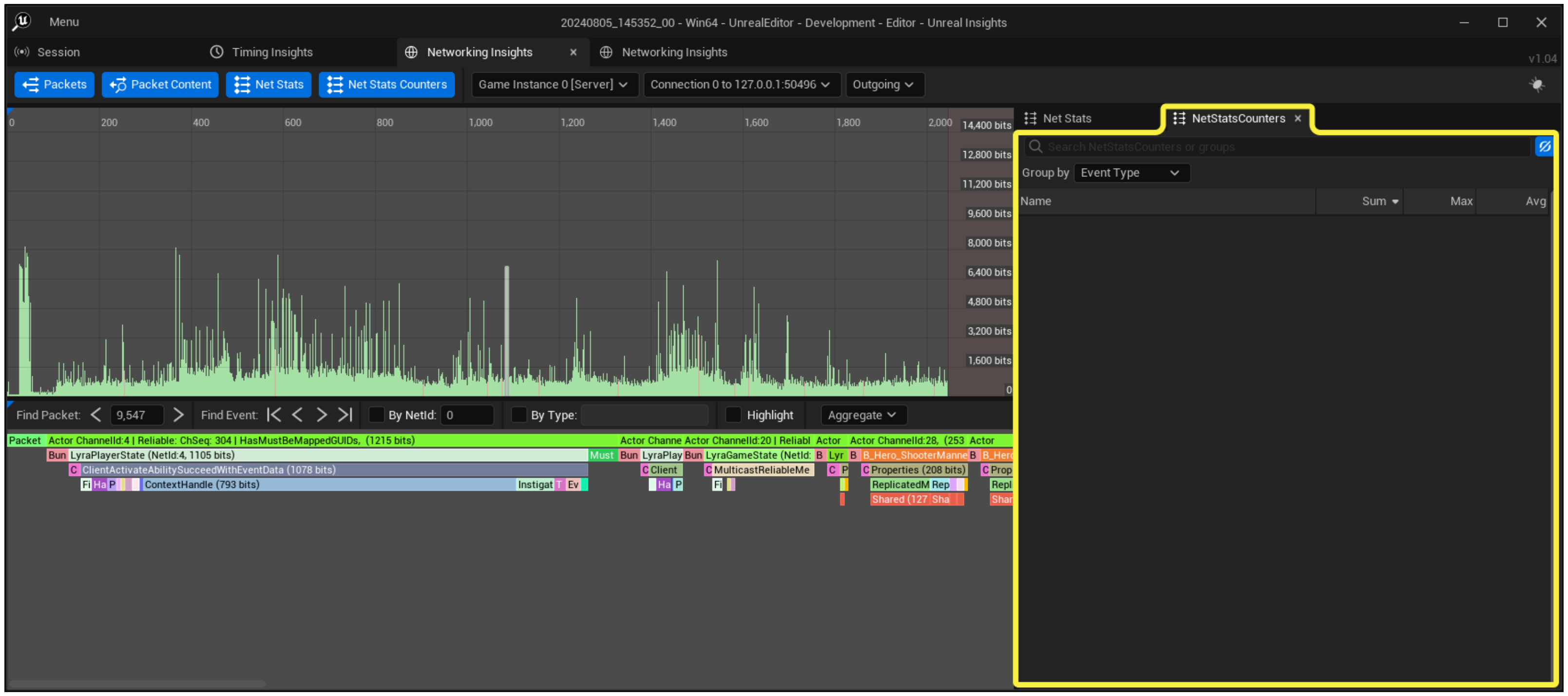
Group Name	Description
Level	Creates one group for each level.
Flat	Creates a single group. Includes all Net Events.
Name	Creates one group for one letter

Grouping by Level helps users understand their data better because the same name can appear on different levels. The following table describes these levels:

Level	Description
0	Typically channel information that displays all bunches using the channel name.
1	Replicated Object (Actor), most events at this level show the serialized object.
2	Reports events originating from property replication or from RPCs.
3	Reports events when serializing properties.
4	Reports additional events when serializing properties, array content, and more.

The Net Stats panel features a **!0** toggle to filter out event types that have no data in the selected range.

Net Stats Counters



The **Net Stats Counters** panel is where you can view stats counters related to the replication system for the selected packet or range of packets.

There are two different categories of stats:

- **Frame Stats**: Stats counters for the entire frame. This is typically used for reporting stat counters related to networking, for work that is shared between all connections. For example, how many objects are currently being updated on this frame.

- **Packet Stats:** Stats counters related to each specific packet. For example, how many objects were scheduled to be serialized, and how many actually did fit into the packet.

To add information to this panel, use the macros `UE_NET_TRACE_FRAME_STATSCOUNTER` and `UE_NET_TRACE_PACKET_STATSCOUNTER` in your C++ code. Data traced with either of these macros will be displayed in the Net Stats Counters panel.

Net Stats Counter Macros

To use the trace stats counter for information associated with the current frame, add `UE_NET_TRACE_FRAME_STATSCOUNTER` to your code. This macro is used as follows:

```
UE_NET_TRACE_FRAME_STATSCOUNTER(<GAME_INSTANCE_ID>, <STAT_NAME>, <STAT_VALUE>, <NET_TRACE_VERBOSITY>)
```

Copy full snippet

To use the trace stats counter for information associated with the current packet, add `UE_NET_TRACE_PACKET_STATSCOUNTER` to your code. This macro is used as follows:

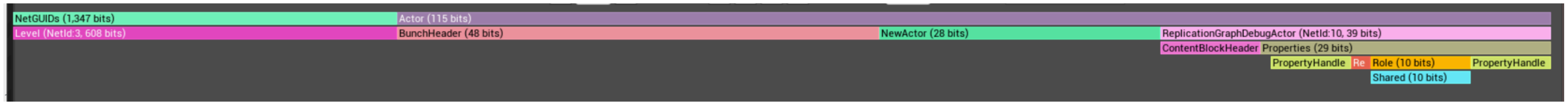
```
UE_NET_TRACE_FRAME_STATSCOUNTER(<GAME_INSTANCE_ID>, <CONNECTION_ID>, <STAT_NAME>, <STAT_VALUE>, <NET_TR
```

Copy full snippet

Example Data

To better understand the data that Networking Insights visualizes, we offer the following examples:

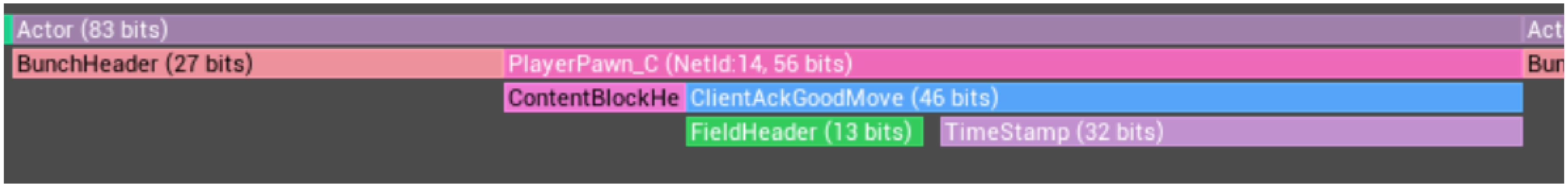
Replicated Objects



This is the initial replication of a new Actor--ReplicationGraphDebugActor, NetGUID 10.

Before the bunch containing the new actor, there is a NetGUID bunch containing all of the exported references from the new Actor, including NetGUIDS.

RPCs

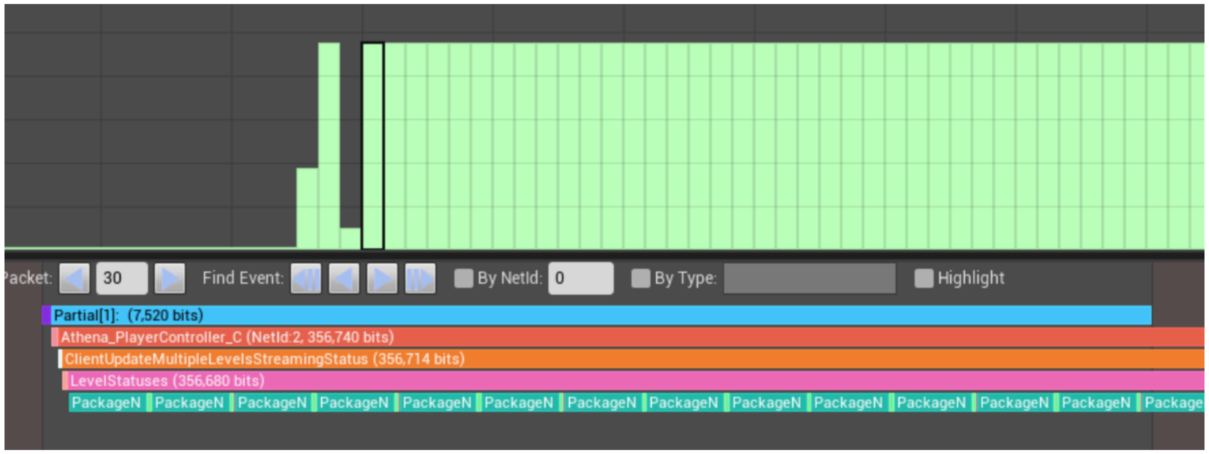


This is an RPC named ClientAckGoodMove that targets PlayerPawn_C with NetGUID 14.

GUIDs must be mapped.

Split Bunches

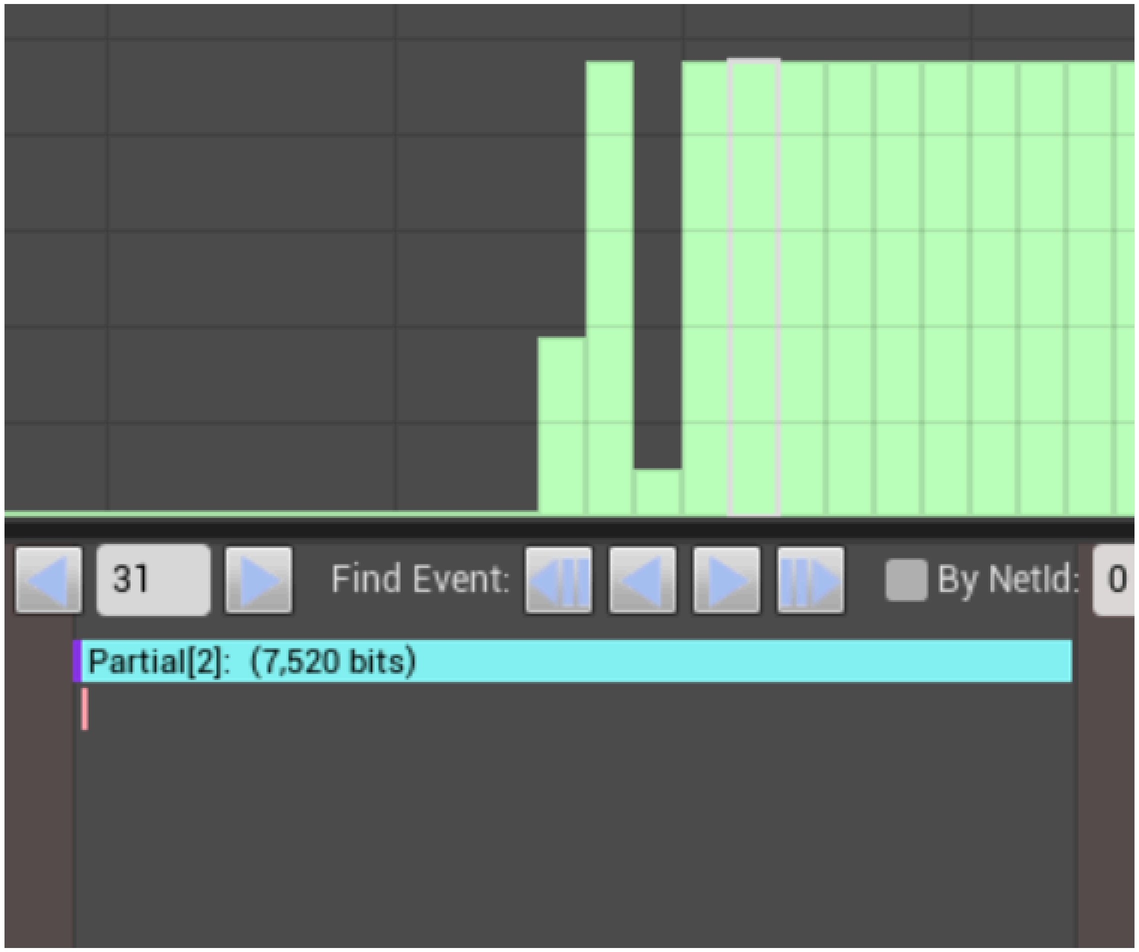
Event data generated from transmitting a split bunch report with the first part--when receiving a split bunch, parts are reported first, and when all parts are received, events are reported with the final part.



Partial[1]: (7,520 bits)
Athena_PlayerController_C (NetId:2, 356,740 bits)
ClientUpdateMultipleLevelsStreamingStatus (356,714 bits)
LevelStatuses (356,680 bits)
PackageN PackageN PackageN PackageN PackageN PackageN PackageN PackageN PackageN PackageN PackageN PackageN PackageN Package

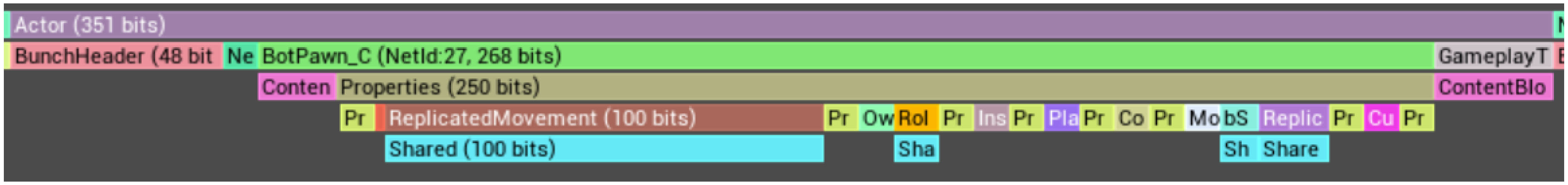
This is a split outgoing bunch. Note that the events in the bunch are larger than the Bunch.

The next few packets look like the following:



Incoming split bunches exhibit the opposite pattern, where events report with the last partial bunch.

Sub-Objects



This is a SubObject example--note that the second Actor after BotPawn_C shares the same bunch.