<u>Lab 3 : APPIOT</u> <u>Advance Message Queuing Protocol (AMQP) and</u> <u>Extensible Messaging and Presence Protocol (XMPP)</u>

AMQP :

After launching proton_receiver.py I obtained this from wireshark :

an	nqp				× +
No.	Time	Source	Destination	Protocol	Length Info
	129 34.965006677	127.0.0.1	127.0.0.1	AMOP	74 Protocol-Header 1-0-0
	131 34,967772062	127.0.0.1	127.0.0.1	AMOP	74 Protocol-Header 1-0-0
1	133 34.968351470	127.0.0.1	127.0.0.1	AMOP	118 sasl.mechanisms
1	135 34.968687378	127.0.0.1	127.0.0.1	AMOP	102 sasl.init
1	136 34.969023268	127.0.0.1	127.0.0.1	AMOP	83 sasl.outcome
1	137 34.969295034	127.0.0.1	127.0.0.1	AMOP	310 Protocol-Header 1-0-0 open begin attach flow
1 1	138 34.969440021	127.0.0.1	127.0.0.1	AMOP	74 Protocol-Header 1-0-0
1 1	139 34.971045625	127.0.0.1	127.0.0.1	AMOP	400 open
1. 1	141 34.990030633	127.0.0.1	127.0.0.1	AMOP	102 begin
Ê 1	142 35.002891645	127.0.0.1	127.0.0.1	AMOP	261 attach
Ê l	144 35.003704082	127.0.0.1	127.0.0.1	AMOP	100 flow
				a de la companya de l	

- Protocol-Header 1-0-0: This is the initial handshake packet in an AMQP connection, indicating the start of a new AMQP protocol communication. It's sent to negotiate the protocol version to be used between the client and server.
- sasl.mechanisms: This packet lists the authentication mechanisms that the server supports, as part of the SASL (Simple Authentication and Security Layer) negotiation process.
- sasl.init: This is the client's response to the sasl.mechanisms packet, initiating the authentication process by selecting a mechanism and possibly providing initial authentication data.
- sasl.outcome: This packet indicates the outcome of the SASL authentication process initiated by the sasl.init message. It tells the client whether the authentication was successful or not.
- Protocol-Header 1-0-0: These are additional AMQP protocol header packets that might indicate a re-negotiation of the protocol or a reinitialization of the connection attempt.
- open: This packet signifies the actual opening of an AMQP connection after successful SASL negotiation, indicating that both the client and the server are ready to start the session.
- begin: This packet marks the start of an AMQP session within the established connection, setting up a context in which messaging can occur.
- attach: This packet represents the client's request to attach a link to the session, which is necessary for creating a channel for message transfer.
- flow: This packet is related to flow control within the AMQP session, indicating how many messages the receiver is ready to accept or how many messages the sender is willing to send.

In the captured flow packet, we see the link credit set at 10, indicating that the sender is permitted to send ten messages before awaiting further acknowledgments or flow control updates from the receiver. This mechanism efficiently prevents the receiver from becoming overwhelmed with messages. Additionally, the window size is specified as 65535, which is the count of frames the receiver is prepared to process before sending back pressure signals or acknowledgements. These controls are pivotal in maintaining a stable message flow, ensuring neither sender nor receiver exceeds their processing capacity and preserves the integrity of the message queue.



In the open packet, the maximum frame size is set to 32767, delineating the largest permissible size for a single frame in the session, ensuring compatibility and efficient processing between communicating peers.

stem mit: m.sl	136 34.969023268 127 137 34.969295034 127 138 34.969440021 127 139 34.971045625 127 141 34.990030633 127	.0.0.1 .0.0.1 .0.0.1 .0.0.1 .0.0.1	127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1	AMQP 3: AMQP 3: AMQP 4 AMQP 4 AMOP 1:	33 sasl.outcome 10 Protocol-Header 1-0-0 open begin attach flow 14 Protocol-Header 1-0-0 19 open 22 begin		
6 /b work	142 35.002891645 127 144 35.003704082 127	.0.0.1 .0.0.1	127.0.0.1 127.0.0.1	AMQP 2 AMQP 1	51 atľach 30 flow	2>	(ml -Dlo
work	6					- F	
work work <mark>rked</mark> T202	 Ethernet II, Src: 00:00 Internet Protocol Versi Transmission Control Pr Advanced Message Queuei Length: 334 Doff: 2 	:00_00:00:00 (00:0 on 4, Src: 127.0.0 otocol, Src Port: ng Protocol	0:00:00:00:00), Dst: .1, Dst: 127.0.0.1 5672, Dst Port: 5340	00:00:00_00: 02, Seq: 86, 4	00:00 (00:00:00:00:00:00) .ck: 289, Len: 334	Î NQF	þ
obj ject obj elta	Type: AMQP (0) Channel: 0 Performative: open (~ Arguments Container-Id: rabl	16) pit@networkedss-Vir	tualBox				
100% <mark>rked</mark> T202	Max-Frame-Size: 3: Channel-Max: 3276 Idle-Timeout: 600 ▶ Properties (map o	2768 7 90 F 6 elements)				1PF	2
obj ject	0000 00 00 00 00 00 00 00 0010 01 82 1d fb 40 00 46	00 00 00 00 00 00 06 1d 79 7f 00 00	3 00 45 00 9 01 7f 00@.@	Е. . у		-	

In the packet sasl.mechanisms, we observe a SASL packet detailing the authentication mechanisms available for securing the AMQP communication. The sasl.mechanisms listed include ANONYMOUS, PLAIN, and AMQPLAIN, indicating different levels of authentication offered by the server. The presence of these mechanisms is the first step in ensuring secured communication, with the understanding that mechanisms like PLAIN should be used over a secured layer like TLS to encrypt credentials in transit. This packet is pivotal in demonstrating the initiation of secure communication within the AMQP protocol.

	amqp					\boxtimes	•
No		Time	Source	Destination	Protocol	Length Info	
	129	34.965006677	127.0.0.1	127.0.0.1	AMQP	74 Protocol-Header 1-0-0	
	131	34.967772062	127.0.0.1	127.0.0.1	AMQP	74 Protocol-Header 1-0-0	
	133	34.968351470	127.0.0.1	127.0.0.1	AMQP	118 sasl.mechanisms	
	135	34.968687378	127.0.0.1	127.0.0.1	AMQP	102 sasl.init	
	136	34.969023268	127.0.0.1	127.0.0.1	AMQP	83 sasl.outcome	
	137	34.969295034	127.0.0.1	127.0.0.1	AMQP	310 Protocol-Header 1-0-0 open begin attach flow	
	138	34.969440021	127.0.0.1	127.0.0.1	AMQP	/4 Protocol-Header 1-0-0	
	139	34.9/1045625	127.0.0.1	127.0.0.1	AMQP	400 open	
1	141	34.990030633	127.0.0.1	127.0.0.1	AMOP	102 begin	
	142	35.002891045	127.0.0.1	127.0.0.1	AMOD	201 ALLACH	
	144	33.003104002	127.0.0.1	127.0.0.1	Anyr	100 1100	
4							
4	Frame	133: 118 byte	s on wire (944	bits). 118 bytes captur	red (944 bits) on interface lo, id 0	
•	Frame Etherr	133: 118 byte net II. Src: 0	s on wire (944 0:00:00 00:00:	bits), 118 bytes captu 00 (00:00:00:00:00:00),	red (944 bits Dst: 00:00:00) on interface lo, id 0 0 00:00:00 (00:00:00:00:00)	
4	Frame Etherr Interr	133: 118 byte net II, Src: G net Protocol V	s on wire (944 0:00:00_00:00: ersion 4, Src:	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.0	red (944 bits Dst: 00:00:00) on interface lo, id 0 0_00:00:00 (00:00:00:00:00:00)	
4 + + +	Frame Etherr Interr Transm	133: 118 byte met II, Src: G met Protocol V mission Contro	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.0 c Port: 5672, Dst Port:	red (944 bits Dst: 00:00:00 0.1 53402, Seq: 9) on interface lo, id 0 0_00:00:00 (00:00:00:00:00:00) 9, Ack: 9, Len: 52	
4 + + + +	Frame Etherr Interr Transm Advance	133: 118 byte het II, Src: G het Protocol V hission Contro ced Message Qu	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr eueing Protoco	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.0 c Port: 5672, Dst Port: 1	red (944 bits Dst: 00:00:00 0.1 53402, Seq: 9) on interface lo, id 0 0_00:00:00 (00:00:00:00:00) 9, A ck: 9, Len: 52	
4 + + + +	Frame Etherr Interr Transm Advanc Len	133: 118 byte het II, Src: G het Protocol V dission Contro hed Message Qu gth: 52	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr eueing Protoco	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.(c Port: 5672, Dst Port: 1	red (944 bits Dst: 00:00:00 0.1 53402, Seq: 9) on interface lo, id 0 0_00:00:00 (00:00:00:00:00) 9, Ack: 9, Len: 52	
4 * * * *	Frame Etherr Interr Transm Advand Len Dof	133: 118 byte let II, Src: 0 let Protocol V lission Contro led Message Qu gth: 52 f: 2	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr eueing Protoco	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.0 c Port: 5672, Dst Port: 1	red (944 bits Dst: 00:00:00 0.1 53402, Seq: 9) on interface lo, id 0 0_00:00:00 (00:00:00:00:00:00) 9, Ack: 9, Len: 52	
4 + + + +	Frame Ethern Intern Transm Advand Len Dof Typ	133: 118 byte het II, Src: 6 het Protocol V hission Contro red Message Qu gth: 52 f: 2 e: SASL (1)	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr eueing Protoco	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.0 c Port: 5672, Dst Port: 1	red (944 bits Dst: 00:00:00 0.1 53402, Seq: 9) on interface 10, id 0 0_00:00:00 (00:00:00:00:00) 9, Ack: 9, Len: 52	
4 + + + +	Frame Ethern Intern Transm Advanc Len Dof Typ Cha	133: 118 byte het II, Src: 0 het Protocol V bission Contro ced Message Qu gth: 52 f: 2 e: SASL (1) nnel: 0	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr eueing Protoco	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.(c Port: 5672, Dst Port: 1	red (944 bits Dst: 00:00:00 0.1 53402, Seq: 9) on interface lo, id 0 0_00:00:00 (00:00:00:00:00) 9, Ack: 9, Len: 52	
4	Frame Etherr Interr Transm Advanc Len Dof Typ Cha SAS	133: 118 byte let II, Src: 0 let Protocol V ilssion Contro ced Message Qu gth: 52 f: 2 e: SASL (1) nnel: 0 L Method: sas	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr eueing Protoco 1.mechanisms (1	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.0 c Port: 5672, Dst Port: 1	red (944 bits Dst: 00:00:04 9.1 53402, Seq: 9) on interface lo, id 0 0_00:00:00 (00:00:00:00:00) 9, Ack: 9, Len: 52	
• • • • •	Frame Etherr Interr Transm Advand Len Dof Typ Cha SAS T	133: 118 byte let II, Src: 0 let Protocol V ission Contro ed Message Qu gth: 52 f: 2 e: SASL (1) nnel: 0 L Method: sas uments	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr eueing Protoco 1.mechanisms (4	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.0 c Port: 5672, Dst Port: 1	red (944 bits Dst: 00:00:00 0.1 53402, Seq: 9) on interface 10, id 0 0_00:00:00 (00:00:00:00:00) 9, Ack: 9, Len: 52	
4	Frame Etherr Interr Transm Advanc Len Dof Typ Cha SAS - Arg	133: 118 byte let II, Src: 6 let Protocol V bission Contro red Message Qu gth: 52 f: 2 e: SASL (1) nnel: 0 L Method: sas uments Mechanisms (a	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr eueing Protoco 1.mechanisms (f rray of 3 eleme	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.(c Port: 5672, Dst Port: 1 64) ents)	red (944 bits Dst: 00:00:00 9.1 53402, Seq: 9) on interface lo, id 0 0_00:00:00 (00:00:00:00:00) 9, Ack: 9, Len: 52	
1 > > > > > > > > > >	Frame Etherr Interr Transm Advanc Len Dof Typ Cha SAS Arg	133: 118 byte let II, Src: 6 let Protocol V dission Contro ed Message Qu gth: 52 f: 2 e: SASL (1) nnel: 0 L. Method: sas uments Mechanisms (a Mechanisms	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr eueing Protoco 1.mechanisms (1 rray of 3 eleme (sym32): ANONY	bits), 118 bytes captur 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.0 C Port: 5672, Dst Port: 1 64) ents) MOUS	red (944 bits Dst: 00:00:00 0.1 53402, Seq: 9) on interface lo, id 0 0_00:00:00 (00:00:00:00:00) 9, Ack: 9, Len: 52	
• • • • • •	Frame Etherr Interr Transm Advanc Len Dof Typ Cha SAS - Arg	133: 118 byte let II, Src: 6 let Protocol V ission Contro ed Message Qu gth: 52 f: 2 e: SASL (1) nnel: 0 L Method: sas uments Mechanisms (a Mechanisms	s on wire (944 0:00:00_00:00: ersion 4, Src: 1 Protocol, Sr eueing Protoco 1.mechanisms (/ rray of 3 eleme (sym32): ANONY (sym32): ANONY	bits), 118 bytes captum 00 (00:00:00:00:00:00), 127.0.0.1, Dst: 127.0.(c Port: 5672, Dst Port: 1 64) ents) MOUS AIN	red (944 bits Dst: 00:00:00).1 53402, Seq: 9) on interface 10, id 0 0_00:00:00 (00:00:00:00:00) 9, Ack: 9, Len: 52	

Running the proton_sender.py script initiated a sequence of AMQP operations captured in Wireshark, which are key to the AMQP messaging process. The transfer performative observed in the captures is fundamental for the delivery of messages from the sender to the receiver. It includes the frame header and message header, which are essential for routing and delivering the message correctly. The transfer performative appearing twice indicates that both the sending of the message and the subsequent acknowledgment or disposition are being captured. This process underlines the Quality of Service at the protocol level, ensuring message integrity and reliable delivery.

ne Source Destination Protocol Length Info 000757474 127.0.0.1 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1.0-0 0007694095 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1.0-0 001002872 127.0.0.1 127.0.0.1 AMQP 108 sasl.mechanisms 001419288 127.0.0.1 127.0.0.1 AMQP 83 sasl.utice 001705599 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 open begin attach 001705598 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 002388741 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 002388741 127.0.0.1 127.0.0.1 AMQP 102 begin 002417660 127.0.0.1 AMQP 102 begin 102 002417660 127.0.0.1 AMQP 102 begin 102 002417660 127.0.0.1 AMQP 12 transfer 102 002147990 127.0.0.1 AMQP 12 close 102 <th>amqp</th> <th></th> <th></th> <th></th> <th></th> <th>+</th>	amqp					+
0000517474 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 0000064095 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 001002872 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 00117715 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 00117715 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 00117715 127.0.0.1 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 00114288 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 000000000000000000000000000000000000	ne	Source	Destination	Protocol	Lengt ¹ Info	
0000640955 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 001002872 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 001002872 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 00147715 127.0.0.1 127.0.0.1 AMOP 183 sal.mechanisms 001412834 001705802 127.0.0.1 127.0.0.1 AMOP 83 sal.outcome 00161.4 001785081 127.0.0.1 127.0.0.1 AMOP 74 Protocol-Header 1.0.0 00161.4 002386741 127.0.0.1 127.0.0.1 AMOP 74 Protocol-Header 1.0.0 00161.4 002386741 127.0.0.1 127.0.0.1 AMOP 192 begin 0024001671 000 0055178061 127.0.0.1 127.0.0.1 AMOP 194 disposition 0055178061 127.0.0.1 127.0.0.1 0000 000117490 114 disposition 005174901 127.0.0.1 127.0.0.1 AMOP 18 disposition 0065174901 001 127.0.0.1 127.0.0.1 AMOP 10 close 1 1 1 1 127.0.0.1	000757474	127.0.0.1	127.0.0.1	AMOP	74 Protocol-Header 1-0-0	
001002972 127.0.0.1 127.0.0.1 AMQP 118 sasl.mechanisms 001419288 127.0.0.1 127.0.0.1 AMQP 102 sasl.init 001419288 127.0.0.1 127.0.0.1 AMQP 102 sasl.init 00170715 127.0.0.1 127.0.0.1 AMQP 28 sasl.outcome 001795989 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 002388741 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 00346314 127.0.0.1 127.0.0.1 AMQP 102 begin 0044001674 127.0.0.1 127.0.0.1 AMQP 216 attach flow 005515502 127.0.0.1 127.0.0.1 AMQP 102 begin 00515522 127.0.0.1 127.0.0.1 AMQP 10 begostinon 00515522 <td>000964905</td> <td>127.0.0.1</td> <td>127.0.0.1</td> <td>AMOP</td> <td>74 Protocol-Header 1-0-0</td> <td></td>	000964905	127.0.0.1	127.0.0.1	AMOP	74 Protocol-Header 1-0-0	
001419288 127.0.0.1 127.0.0.1 AMQP 102 sasl.init 001477715 127.0.0.1 127.0.0.1 AMQP 83 sasl.outcome 001795889 127.0.0.1 127.0.0.1 AMQP 78 Protocol-Header 1-0-0 00 00238741 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 00 00238741 127.0.0.1 127.0.0.1 AMQP 140 Pop 162 begin 004001674 127.0.0.1 127.0.0.1 AMQP 192 begin 00 004001674 127.0.0.1 127.0.0.1 AMQP 190 bigsition 00 005515624 127.0.0.1 127.0.0.1 AMQP 144 transfer 00 005417806 127.0.0.1 127.0.0.1 AMQP 89 disposition 00 005147490 127.0.0.1 127.0.0.1 AMQP 89 disposition 00 005147240 127.0.0.1 127.0.0.1 AMQP 89 disposition 00 006616152 127.0.0.1 127.0.0.1 AMQP 80 disposition 00 006616152 127.0.0.1 AMQP 80 disposition	001002872	127.0.0.1	127.0.0.1	AMOP	118 sasl.mechanisms	
00147715 127.0.0.1 127.0.0.1 AMQP 278 Protocol-Header 1-0-0 open begin attach 001795989 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 002388741 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 002388741 127.0.0.1 127.0.0.1 AMQP 400 open 003464314 127.0.0.1 127.0.0.1 AMQP 102 begin 004001674 127.0.0.1 127.0.0.1 AMQP 102 begin 004001674 127.0.0.1 127.0.0.1 AMQP 216 attach flow 005515024 127.0.0.1 127.0.0.1 AMQP 80 disposition 005515024 127.0.0.1 127.0.0.1 AMQP 80 disposition 00551524 127.0.0.1 127.0.0.1 AMQP 80 disposition 0055152245 127.0.0.1 127.0.0.1 AMQP 81 close 005149256 127.0.0.1 127.0.0.1 AMQP 78 close 005149256 127.0.0.1 127.0.0.1 AMQP 78 close 005149256 127.0.0.1 127.0.0.1 AMQP 78 close 00514926 127.0.0.1 127.0.0.1 AMQP 78 close 00516 00 00 00 00 00 00 00 00 00 00 00 00 00	001419288	127.0.0.1	127.0.0.1	AMOP	102 sasl init	
001795980 127.0.0.1 127.0.0.1 127.0.0.1 AMQP 278 Protocol-Header 1-0-0 open begin attach 001424284 127.0.0.1 127.0.0.1 AMQP 74 Protocol-Header 1-0-0 00238874 127.0.0.1 127.0.0.1 AMQP 400 open 002388741 127.0.0.1 127.0.0.1 AMQP 400 open 002388741 127.0.0.1 127.0.0.1 AMQP 120 begin 004001671 127.0.0.1 127.0.0.1 AMQP 120 begin 005515924 127.0.0.1 127.0.0.1 AMQP 120 begin 005515926 127.0.0.1 127.0.0.1 AMQP 120 begin 005515926 127.0.0.1 127.0.0.1 AMQP 180 disposition 005515926 127.0.0.1 127.0.0.1 AMQP 180 close 005515927 127.0.0.1 127.0.0.1 AMQP 120 close 005117900 127.0.0.1 127.0.0.1 AMQP 120 close 005115921 127.0.0.1 127.0.0.1 AMQP 120 close 006511592 127.0.0.1 127.0.0.1 AMQP 120 close	001477715	127.0.0.1	127.0.0.1	AMOP	83 sasl outcome	
001842684 127.0.0.1 127.0.0.1 AMQp 74 Protocol-Header 1.0.0 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.	001795989	127 0 0 1	127 0 0 1	AMOP	278 Protocol-Header 1-0-0 onen begin attach	
0002388741 127.0.0.1 127.0.0.1 ANOP 400 open 003484314 127.0.0.1 127.0.0.1 ANOP 120 begin 004001674 127.0.0.1 127.0.0.1 ANOP 216 attach flow 00515524 127.0.0.1 127.0.0.1 ANOP 122 transfer 005417806 127.0.0.1 127.0.0.1 ANOP 144 transfer 006816525 127.0.0.1 127.0.0.1 ANOP 184 transfer 006117490 127.0.0.1 127.0.0.1 ANOP 184 transfer 006117490 127.0.0.1 127.0.0.1 ANOP 18 tclose 00681652 127.0.0.1 127.0.0.1 ANOP 80 disposition 00681652 127.0.0.1 127.0.0.1 ANOP 81 close 4 Frame 20: 144 bytes on wire (1152 bits), 144 bytes captured (1152 bits) on interface 10, 1d 0 > Ethernet II, Src: 00:00:00_00:00:00:00:00:00:00:00:00:00:0	001842684	127 0 0 1	127 0 0 1	ΔΜΟΡ	74 Protocol-Header 1-0-0	
000346431 127.0.0.1 127.0.0.1 ANOP 102 begin 0004601674 127.0.0.1 127.0.0.1 ANOP 216 attach flow 0005155024 127.0.0.1 127.0.0.1 ANOP 216 attach flow 005155024 127.0.0.1 127.0.0.1 ANOP 90 disposition 005451506 127.0.0.1 127.0.0.1 ANOP 89 disposition 005452566 127.0.0.1 127.0.0.1 ANOP 89 disposition 005452566 127.0.0.1 127.0.0.1 ANOP 89 disposition 006914095 127.0.0.1 127.0.0.1 ANOP 80 close 006914095 127.0.0.1 127.0.0.1 ANOP 81 close 4 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 5 127.0.0.1 127.0.0.1 127.0.0.1	002388741	127 0 0 1	127 0 0 1	AMOP	400 open	
0004001014 127.0.0.1 127.0.0.1 ANQP 216 attach flow 0005417800 127.0.0.1 127.0.0.1 ANQP 216 attach flow 0005417800 127.0.0.1 127.0.0.1 ANQP 89 disposition 0005417800 127.0.0.1 127.0.0.1 ANQP 89 disposition 000617490 127.0.0.1 127.0.0.1 ANQP 89 disposition 000617490 127.0.0.1 127.0.0.1 ANQP 80 disposition 006816050 127.0.0.1 127.0.0.1 ANQP 81 close 1 Frame 20: 144 bytes on wire (1152 bits), 144 bytes captured (1152 bits) on interface 10, 1d 0 + 1 Internet Protocol, Src Port: 57.0.1, Dst: 127.0.0.1 + + <tr< td=""><td>002000741</td><td>127 0 0 1</td><td>127 0 0 1</td><td>AMOD</td><td>102 begin</td><td></td></tr<>	002000741	127 0 0 1	127 0 0 1	AMOD	102 begin	
000101011 1101011 1100011 1100011 0005115024 127.0.0.1 127.0.0.1 AMQP 124 transfer 0005115024 127.0.0.1 127.0.0.1 AMQP 194 transfer 005115024 127.0.0.1 127.0.0.1 AMQP 89 disposition 005115024 127.0.0.1 127.0.0.1 AMQP 89 disposition 0056117490 127.0.0.1 127.0.0.1 AMQP 89 disposition 0066117490 127.0.0.1 127.0.0.1 AMQP 80 disposition 00661152 127.0.0.1 127.0.0.1 AMQP 81 close 4 Frame 20: 144 bytes on wire (1152 bits), 144 bytes captured (1152 bits) on interface 10, id 0 * * Ethernet II, Src: 00:00:00:00:00:00:00:00:00:00:00:00:00	004001674	127.0.0.1	127.0.0.1	AMOR	216 attach flow	
000541786 127.0.0.1 127.0.0.1 AWQr 142 transfer 00541786 127.0.0.1 127.0.0.1 AWQP 184 transfer 00514780 127.0.0.1 127.0.0.1 AWQP 78 close 006117490 127.0.0.1 127.0.0.1 AWQP 78 close 006541786 127.0.0.1 127.0.0.1 AWQP 78 close 005041000 000:00:00:00:00:00:00:00:00:00.00 000:00:00:00:00:00:00:00 000:00:00:00:00:00:00:00 1 Transmission Control Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 Frame Frame 1 * Transmission Control Protocol, Src Port: 5672, Dst Port: 53402, Seq: 1, Ack: 1, Len: 78 Advanced Message Queueing Protocol * Advanced Message Queueing Protocol ************************************	005155024	127.0.0.1	127.0.0.1	AMOR	122 transfer	
G005452565 127.0.0.1 127.0.0.1 AWQP 144 fransfer 006117490 127.0.0.1 127.0.0.1 AWQP 89 disposition 0060117490 127.0.0.1 127.0.0.1 AMQP 89 disposition 0060117490 127.0.0.1 127.0.0.1 AMQP 89 disposition 00601202 127.0.0.1 127.0.0.1 AMQP 81 close 006014055 127.0.0.1 127.0.0.1 AMQP 81 close 4 Frame 20: 144 bytes on wire (1152 bits), 144 bytes captured (1152 bits) on interface lo, id 0 91 close 4 Frame 20: 147.0.0.1, Src: 127.0.0.1, Dst: 127.0.0.1 91 close 4 Frame 20: 144 bytes on wire (1152 bits), 144 bytes captured (1152 bits) on interface lo, id 0 9 Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00:00:00:00:00:00:00:00:	005113024	127 0 0 1	127 0 0 1	AMOR	80 disposition	
000012000 11101011 11101011 11101011 0006117490 127.0.0.1 127.0.0.1 AMQP 89 disposition 0006117490 127.0.0.1 127.0.0.1 AMQP 78 close 0006117490 127.0.0.1 127.0.0.1 AMQP 81 close 4	005452566	127.0.0.1	127.0.0.1	AMOR	1/4 transfor	
0006861652 127.0.0.1 127.0.0.1 AMQP 78 close 006861652 127.0.0.1 127.0.0.1 AMQP 81 close 006816055 127.0.0.1 127.0.0.1 AMQP 81 close 1 1 127.0.0.1 AMQP 81 close 1 1 + Frame 20: 144 bytes on wire (1152 bits), 144 bytes captured (1152 bits) on interface 10, id 0 + + > Fthernet II, Src: 00:00:00.00:00:00:00:00:00:00:00:00:00:0	006117400	127.0.0.1	127.0.0.1	AMOR	89 disposition	
0000010102 127.0.0.1 127.0.0.1 AMQP 10 Close 4 000010102 127.0.0.1 127.0.0.1 AMQP 81 close 4 • Frame 20: 144 bytes on wire (1152 bits), 144 bytes captured (1152 bits) on interface 10, 1d 0 • • Ethernet II, Src: 00:00:00_00:00:00(00:00:00:00), Dst: 00:00:00_00:00:00:00:00:00:00:00:00:00:0	000117490	127.0.0.1	127.0.0.1	AMOD	79 close	
0000 <	1.11.11.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.			ALCOND. THE		
<pre> Frame 20: 144 bytes on wire (1152 bits), 144 bytes captured (1152 bits) on interface lo, id 0 Ethernet II, Src: 00:00:00_00:00:00:00:00:00:00, Dst: 00:00:00:00:00:00:00:00:00:00:00:00:00</pre>	000801052	127.0.0.1	127 0 0 1	AMOD	91 close	
<pre>> Frame 20: 144 bytes on wire (1152 bits), 144 bytes captured (1152 bits) on interface 10, 10 0 Ethernet II, Src: 00:00:00 (00:00:00:00:00), bst: 00:00 (00:00:00:00:00:00:00:00:00:00) > Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 > Transmission Control Protocol, Src Port: 5672, Dst Port: 53402, Seq: 1, Ack: 1, Len: 78 > Advanced Message Queueing Protocol</pre>	006914095	127.0.0.1	127.0.0.1	AMOP	81 close	
0000 <	<pre>000801052 006914095 Frame 2 Etherne Interne</pre>	127.0.0.1 127.0.0.1 0: 144 bytes on w t II, Src: 00:00: t Protocol Versio	127.0.0.1 ire (1152 bits), 144 b 00_00:00:00 (00:00:00: n 4. Src: 127.0.0.1. D	AMOP ovtes capture 00:00:00), D ost: 127.0.0.	d (1152 bits) on interface lo, id 0 st: 00:00:00_00:00:00 (00:00:00:00:00:00) 1	4
0020 00 01 16 28 d0 9a 4d 53 96 ec 9d 7f 10 6f 80 18 (··MS ····0·· 0030 02 00 fe 76 00 00 01 01 08 0a 4b e8 05 24 4b e7 ···(··MS ····0·· 0040 cd 1a 00 00 06 4e 02 00 00 00 05 31 4c 00 16 0b ···(··MS ····S·· 0056 43 52 0a a0 88 00 00 00 00 00 00 43 42 42 CR ·····CBB	<pre>006801652 006914095 Frame 2 Etherne Interne Transmi Advance</pre>	127.0.0.1 127.0.0.1 0: 144 bytes on w I II, Src: 00:00: t Protocol Versio ssion Control Pro d Message Queuein	127.0.0.1 ire (1152 bits), 144 b 00_00:000:00 (00:00:00: n 4, Src: 127.0.0.1, D tocol, Src Port: 5672, g Protocol	AMOP vytes capture 00:00:00), D vst: 127.0.0. Dst Port: 5	81 close 81 close d (1152 bits) on interface lo, id 0 st: 00:00:00_00:00:00 (00:00:00:00:00:00) 1 3402, Seq: 1, Ack: 1, Len: 78	
0038 02 00 fe 76 00 00 01 01 08 0a 4b e8 05 24 4b e7 ····································	0006914095 () Frame 2) Etherne) Interne) Transmi) Advance	127.0.0.1 0: 144 bytes on w II, Src: 00:00: t Protocol Versio ssion Control Pro d Message Queuein	127.0.0.1 ire (1152 bits), 144 b 00_00:00:00 (00:00:00: n 4, Src: 127.0.0.1, D tocol, Src: Port: 5672, g Protocol 00 00 00 00 00 00 08 00 4 06 12 26 7f 00 00 1	AMOP vytes capture 00:00:00), D vst: 127.0.0. Dst Port: 5 45 00	81 close 81 close d (1152 bits) on interface lo, id 0 st: 00:00:00_00:00:00 (00:00:00:00:00) 1 3402, Seq: 1, Ack: 1, Len: 78	
0040 cd 1a 00 00 04 02 00 00 00 53 14 c0 16 0b ·····N····S···· 0050 43 52 0a a0 08 00 00 00 00 04 42 42 CR······CBB	0000014095 () Frame 2 Etherne) Interne) Transmi) Advance 0000 00 6 0010 00 8 0020 00 6	127.0.0.1 0: 144 bytes on w II, Src: 00:00: t Protocol Versio ssion Control Pro d Message Queuein 00 00 00 00 00 00 00 22 1e 4e 40 00 40 31 16 28 d0 9a 4d	127.0.0.1 ire (1152 bits), 144 b 00_00:00:00 (00:00:00: n 4, Src: 127.0.0.1, D tocol, Src Port: 5672, g Protocol 00 00 00 00 00 08 00 4 06 1e 26 7f 00 00 01 5 53 96 ec 9d 7f 10 6f 8	AMOP vytes capture 00:00:00), D vst: 127.0.0. Dst Port: 5 45 00	81 close 81 close d (1152 bits) on interface lo, id 0 st: 00:00:00_00:00:00 (00:00:00:00:00:00) 1 3402, Seq: 1, Ack: 1, Len: 78 0 -	
0050 43 52 0a a0 08 00 00 00 00 00 00 00 00 b 43 42 42 CR····· CBB	0006014095 () Frame 2 Etherne Interne Transmi Advance	127.0.0.1 127.0.0.1 0: 144 bytes on wt II, Src: 00:00: t Protocol Versio ssion Control Pro d Message Queuein 00 00 00 00 00 00 00 22 1e 4e 40 00 40 01 16 28 d0 9a 4d 06 Fe 76 00 00 00 00 00	127.0.0.1 ire (1152 bits), 144 b 00-00:00:00 (00:00:00: n 4, Src: 127.0.0.1, D tocol, Src Port: 5672, g Protocol 00 00 00 00 00 08 00 4 06 1e 26 7f 00 00 01 53 96 ec 9d 7f 10 6f 3 53 96 a 4b e8 05 24 4	AMOP sytes capture 00:00:00; D st: 127.0.0. Dst Port: 5 45 00	81 close d (1152 bits) on interface lo, id 0 ss: 00:00:00_00:00:00 (00:00:00:00:00:00) 1 3402, Seq: 1, Ack: 1, Len: 78	
	0006014095 4 Frame 2 Etherne Interne Transmi Advance	127.0.0.1 127.0.0.1 0: 144 bytes on w I I, Src: 00:00: t Protocol Versio ssion Control Pro d Message Queuein 00 00 00 00 00 00 00 00 12 1e 4e 40 00 40 01 16 28 d0 9a 4d 00 fe 76 00 00 14 a 00 00 06 4e 02	127.0.0.1 ire (1152 bits), 144 b 00_00:00:00 (00:00:00: n 4, Src: 127.0.0.1, D tocol, Src Port: 5672, g Protocol 00 00 00 00 00 08 00 4 06 1e 26 7f 00 00 01 53 96 ec 9d 7f 10 6f 4 01 08 0a 4b e8 05 24 4 00 00 00 00 53 14 c0 :	AMOP ytes capture 00:00:00), D st: 127.0.0. Dst Port: 5 45 00 76 00	81 close 81 close d (1152 bits) on interface lo, id 0 st: 00:00:00_00:00:00 (00:00:00:00:00) 1 3402, Seq: 1, Ack: 1, Len: 78	
0060 40 40 42 42 42 00 53 70 c0 06 05 42 40 40 41 40 @@BBB Sp ··· B@@A@	0006914095 () Frame 2 Etherne) Interne) Interne) Transmi) Advance 0010 00 6 0010 00 6 0000 00 6 0000 000 000 000 000 0000000000	127.0.0.1 127.0.0.1 0: 144 bytes on w t II, Src: 00:00: t Protocol Versio ssion Control Pro d Message Queuein 00 00 00 00 00 00 00 12 1e 4e 40 00 40 12 1e 4e 40 09 40 16 28 d0 9a 4d 16 76 00 00 01 1a 00 00 00 4e 02 20 aa 0 08 00 00	127.0.0.1 ire (1152 bits), 144 b 00_00:00:00 (00:00:00: n 4, Src: 127.0.0.1, D tocol, Src Port: 5672, g Protocol 00_00_00_00_00_00_00 01_08_0a_4b_e8_05_24 00_00_00_05_314_c0 00_00_00_00_00_04_32	AMOP vytes capture 00:00:00), D vst: 127.0.0. Dst Port: 5 45 00 77 00 80 18 18(4b e7 	81 close 81 close d (1152 bits) on interface lo, id 0 st: 00:00:00_00:00:00 (00:00:00:00:00:00) 1 3402, Seq: 1, Ack: 1, Len: 78 00 - & MS MS	
0070 00 53 73 45 00 53 77 a1 17 53 65 71 75 65 6e 63 SSE-Sw Sequenc	0006914095 (006914095 Frame 2 Etherne Interne Transmi Advance 0020 00 0 0020 00 0 0020 00 0 0020 00 0 0030 02 0 0040 cd 1 0054 43 2 0056 43 4 0056 43 4 0056 43 4 0056 43 4 0056 43 4 0056 43 4 0056 4 00	127.0.0.1 127.0.0.1 0: 144 bytes on w II, Src: 00:00: t Protocol Versio ssion Control Pro d Message Queuein 0 00 00 00 00 00 00 21 e 4e 40 00 40 11 fo 28 d0 9a 4d 00 fe 76 00 00 01 1a 00 00 00 4e 02 52 0a a0 08 00 00 42 42 42 00 53	127.0.0.1 ire (1152 bits), 144 b 00_00:00:00 (00:00:00: n 4, Src: 127.0.0.1, D tocol, Src: Port: 5672, g Protocol 00 00 00 00 00 00 00 00 10 80 a be 80 524 00 00 00 00 00 53 14 c0 00 00 00 00 00 53 14 c0 00 00 00 00 00 bd 33 00 c0 06 05 42 40 40	AMOP bytes capture 00:00:00), D st: 127.0.0. Dst Port: 5 45 00	81 close 81 close d (1152 bits) on interface lo, id 0 st: 00:00:00_00:00:00:00:00:00) 1 3402, Seq: 1, Ack: 1, Len: 78	

The disposition packet captured in Wireshark confirms that messages have been processed by the receiver with the Settled field set to True. This indicates a Quality of Service level of "At least once," ensuring that each message is acknowledged after processing, with no further action required for these messages.

004001674 005155024 005417806 005452566 006117490	127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1	127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1	AMQP AMQP AMQP AMQP AMQP AMQP	102 Jogin 216 attach flow 122 transfer 89 disposition 89 disposition	
.006861652	127.0.0.1	127.0.0.1	AMQP	78 close	
.006914095	127.0.0.1	127.0.0.1	AMQP	81 close	•
4					•
- Advanced	d Message Queuein	g Protocol			
Clengt Doff: Type: Chann Perfo → Argum Ro Fi La Se Ac	11: 23 2 2 MAQP (0) hel: 0 ormative: disposit hents he: receiver rst: 0 st: 0 st: 0 ttled: True cepted (list of 0	tion (21) elements)			*
0000 00 0	0 00 00 00 00 00 0	00 00 00 00 00 08 00 4	5 00	+ x + x + x + E +	
0010 00 4	b 24 92 40 00 40	06 18 19 7f 00 00 01 7	f00 ∙K\$∙@•@•	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
0020 00 0	1 16 28 81 5e 37	a2 68 47 fc 81 51 f9 80	0 18 · · · (· ^7 ·	hG··Q···	
0030 02 0	0 fe 3f 00 00 01	01 08 0a 4b e8 05 24 4I	be8 ···?····	K\$K-	
0040 05 2	3 00 00 00 17 02	00 00 00 00 53 15 c0 0	a 06 -#	· · · S · · · ·	
0050 41 4	3 43 41 00 53 24	45 40	ACCA S\$E	0	

In order to send more message I put self.total equals to 20 and I obtained this with wireshark:

	-		Dentirelier	Buckeyel	the second second
10.	Time	Source	Destination	Protocol	Lengtr Info
	4 0.000827575	127.0.0.1	127.0.0.1	AMQP	74 Protocol-Header 1-0-0
	6 0.001051378	127.0.0.1	127.0.0.1	AMQP	74 Protocol-Header 1-0-0
	7 0.001090272	127.0.0.1	127.0.0.1	AMQP	118 sasl.mechanisms
	10 0.001483234	127.0.0.1	127.0.0.1	AMQP	102 sasl.init
	11 0.001541034	127.0.0.1	127.0.0.1	AMQP	83 sasl.outcome
	12 0.001867990	127.0.0.1	127.0.0.1	AMQP	278 Protocol-Header 1-0-0 open begin attach
	13 0.001917647	127.0.0.1	127.0.0.1	AMOP	74 Protocol-Header 1-0-0
	14 0.002436743	127.0.0.1	127.0.0.1	AMOP	400 open
	17 0.005075309	127.0.0.1	127.0.0.1	AMOP	102 begin
	19 0.005639343	127.0.0.1	127.0.0.1	AMOP	216 attach flow
	21 0.008655014	127.0.0.1	127.0.0.1	AMOP	1226 transfer transfer transfer transfer transfer transfer transfer.
	22 0.009576309	127.0.0.1	127.0.0.1	AMOP	90 disposition
	23 0.009633277	127.0.0.1	127.0.0.1	AMOP	91 disposition
	24 0.009650102	127.0.0.1	127.0.0.1	AMOP	144 transfer
1	26 0,009691138	127.0.0.1	127.0.0.1	AMOP	91 disposition
	27 0.009707503	127.0.0.1	127.0.0.1	AMOP	144 transfer
	29 0.009735586	127.0.0.1	127.0.0.1	AMOP	144 transfer
	31 0.009760882	127.0.0.1	127.0.0.1	AMOP	144 transfer
	33 0 009785434	127 0 0 1	127 0 0 1	AMOP	144 transfer
	35 0 000808019	127 0 0 1	127 0 0 1	AMOP	144 transfer
	37 0 000840320	127 0 0 1	127.0.0.1	AMOD	1// transfer
	30 0 0000000023	127.0.0.1	127.0.0.1	AMOD	1// transfer
	41 0 000041704	127.0.0.1	127.0.0.1	AMOD	144 transfor
	41 0.009941794	127.0.0.1	127.0.0.1	AMOD	144 transfor
	45 0.009977202	127.0.0.1	127.0.0.1	AMOD	124 flow disposition
	43 0.01233/0//	127.0.0.1	127.0.0.1	AMOD	102 flou
	4/ 0.012//4//2	127.0.0.1	127.0.0.1	AMOD	103 110W 777 transfer transfer transfer transfer transfer transfer transfer
	50 0.013/88409	127.0.0.1	127.0.0.1	AMQP	101 Eranster transfer transfer transfer transfer transfer transfer.
	52 0.025096052	127.0.0.1	127.0.0.1	AMOP	124 Flow disposition
	53 0.025370825	127.0.0.1	127.0.0.1	AMQP	103 TIOW
	54 0.025578000	127.0.0.1	127.0.0.1	AMQP	145 transfer
	56 0.026474857	127.0.0.1	127.0.0.1	AMQP	78 CLOSE
	57 0.026535134	127.0.0.1	127.0.0.1	AMQP	81 CLOSE
	58 0.026860470	127.0.0.1	127.0.0.1	AMQP	123 flow disposition
	59 0.026976813	127.0.0.1	127.0.0.1	AMQP	102 flow
	60 0.027398951	127.0.0.1	127.0.0.1	AMQP	100 flow
Ta	vt Editor 7502610	127.0.0.1	127.0.0.1	AMOP	102 flow

After increasing the number of messages sent by the proton_sender.py script, we can see several transfer frames indicating message transmissions and disposition frames indicating message acknowledgments.

Initially, the server transfers ten messages, as evidenced by the ten consecutive transfer packets observed in the capture. This number corresponds to the link credit of 10 specified in a preceding flow packet, which dictates the sender can dispatch ten messages before requiring additional credit. Following the initial burst, the sender pauses, awaiting replenishment of link credit from the receiver, as indicated in a subsequent flow packet where the link credit has decreased to 9. The sender resumes transmissions, sending the next nine messages, then halts again, dispatching the final message once the link credit is restored. Eventually, after a certain time interval, the link credit

is reset to its initial count of 10, signaling the sender to proceed with the message flow. This cycle effectively demonstrates the AMQP's flow control mechanism, ensuring balanced message delivery without overloading the receiver.

	$\begin{array}{c} 24 \ e, 0.009650128\\ 27 \ e, 0.009707503\\ 29 \ e, 0.00737586\\ 31 \ e, 0.00737586\\ 32 \ e, 0.007376434\\ 35 \ e, 0.008866622\\ 33 \ e, 0.008866622\\ 41 \ e, 0.00941794\\ 43 \ e, 0.00941794\\ 43 \ e, 0.00941784\\ 43 \ e, 0.00941784\\ 55 \ e, 0.12737477\\ 56 \ e, 0.61277477\\ 56 \ e, 0.61277477\\ 56 \ e, 0.612774672\\ 56 \ e, 0.612774572\\ 56 \ e, 0.612774572\\ 56 \ e, 0.612774572\\ 56 \ e, 0.612774572\\ 56 \ e, 0.625371846\\ 56 \ e, 0.625371845\\ 56 \ e, 0.625371845\\ 56 \ e, 0.625371845\\ 56 \ e, 0.625371845\\ 56 \ e, 0.626977852\\ 57 \ e, 0.2659511\\ 60 \ e, 0.27592611\\ 61 \ e, 0.27592611\\ 61 \ e, 0.27592612\\ \end{array}$	$\begin{array}{c} 127.0,0.1\\ 127.$	$\begin{array}{c} 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 0\\ 127, 0, 0, 1\\ 127, 0, 0, 0\\ 127, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0\\ 12$	AMQP AMQP AMQP AMQP AMQP AMQP AMQP AMQP	9144 transfer 91 disposition 91 disposition 91 44 transfer 91 100 disposition 91 100 disposition 91 100 tisposition 91 100 disposition 91 100 disposition	
≁ Ad	Type: AMQP (0) Channel: 0 Performative: flc Arguments Next-Incoming-Windo Next-Outgoing-Windo Handle: 0 Delivery-Count Link-Credit: 9 Drain: False Vanced Message Que Length: 22	w (19) Id: 52 w: 2147483647 Id: 0 w: 2147483647 : 52 eueing Protocol				
	Doff: 2 Type: AMQP (0) Channel: 0 Dofferentive, die	manitian /94)				
-	1 v. veroeccout 2 0.099576399 2 0.099576399 2 0.09958182 2 0.09955112 2 0.09955112 2 0.099761583 2 0.099761583 3 0.099761882 3 0.099761842 3 0.099761842 3 0.099761842 3 0.0997761832 3 0.0997761842 3 0.0997761842 3 0.0997761842 3 0.0997761842 4 0.09941794 4 0.09941794 4 0.2237677 5 0.22570625 5 0.22570625 5 0.22570625 5 0.22570625 5 0.22570625 5 0.22570625 5 0.22570625 5 0.22570625 5 0.22570625 5 0.225707625 5 0.2	$\begin{array}{c} 127, 0, 0, 1\\ 127, 0, 0, 0, 1\\ 127, 0, 0, 1\\ 127, 0, 0, 0\\ 127, 0, 0, 1\\ 127, 0, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0, 0\\ 127, 0\\ 127, 0, 0\\ 127, 0\\ 0, 0\\ 127, 0\\ 0, 0\\ 127, 0\\ 0\\ 127, 0\\ 0, 0\\ 127, 0\\ 0\\ 0\\ 0\\ 0\\ 127, 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ $	$\begin{array}{c} 127, 0, 0, .1\\$	амбь чибь чибь чибь чибь чибь чибь чибь чи	1220 transfer transfer transfer transfer transfer transfer 90 disposition 11 disposition 13 disposition 144 transfer 144 transfer 145 transfer transfer transfer transfer transfer transfe 124 flow disposition 103 flow 145 transfer 145 transfer 146 transfer 147 transfer transfer transfer transfer transfer transfe 124 flow disposition 135 transfer 136 transfer 136 transfer 137 transfer 136 transfer 137 transfer 136 transfer 137 transfer 138 transfer 139 flow 139 transfer 130 transfer 130 transfer 130 transfer 130 transfer 130 transfer 130 transfer 130 transfer 130 transfer 131 transfer 132 flow disposition	
$\overline{\mathbf{O}}$	60 0.027398951 61 0.027502610	127.0.0.1 127.0.0.1 127.0.0.1	127.0.0.1 127.0.0.1 127.0.0.1	AMQP AMQP AMQP	100 flow 100 flow	
<	Internet Protocol > Transmission Cont Advanced Message Length: 34 Doff: 2 Type: AMQP (0) Channel: 0 Performative: Argumet: 0 Newformative: Newf	Version 4, Src: 127.0. rol Protocol, Src Port: Queueing Protocol flow (19) ng-Id: 62 ndow: 2147483647 ng-Id: 0 ndow: 2147483647 ig-Id: 0 : 10 8	0.1, Dst: 127.0.0.1 53402, Dst Port: 567	72, Seq: 17	174, Ack: 1681, Len: 34	
	0010 00 00 00 00 00 0010 00 56 e6 ba 40 0020 00 01 d0 9a 16	00 00 00 00 00 00 00 40 06 55 e5 7f 00 28 9d 7f 13 89 4d 53	00 00 45 00 00 01 7f 00 V 0 0 a8 68 80 18 (U MS h		

XMPP:

The first step is to update the code echo_bot.py and sender.py

echo_bot.py:

In the revised version of the EchoBot, significant improvements have been implemented to enhance its performance and scalability in handling multiple user sessions concurrently. By integrating asyncio, the bot now operates on an asynchronous event loop, allowing it to manage different user interactions more efficiently and without blocking. The addition of threading enables separate instances for each user, such as Alice and Bob, to run simultaneously, thus maintaining session independence while sharing the same execution environment. The message handling mechanism has also been refined to only respond to messages that contain actual text, preventing the bot from sending replies to empty or irrelevant messages. Furthermore, the bot now supports additional XMPP protocols through the implementation of the xep_0092 plugin for software versioning, enhancing its interoperability and adherence to standard communication protocols. These enhancements not only make the bot more robust and practical for real-world applications but also improve its capability to be scaled for larger deployments.

```
6 class EchoBot(ClientXMPP):
7
8
      def __init__(self, jid, password):
 9
           super().__init__(jid, password)
           self.add_event_handler("session_start", self.session_start)
10
           self.add_event_handler("message", self.message)
11
           self.add_event_handler("iq", self.iq)
12
13
           # Save the plugin 'xep 0092' for the software version
14
15
           self.register plugin('xep 0092')
16
           # Configuration of the system informations
17
           self['xep 0092'].software name = 'My Echo Bot'
18
19
           self['xep 0092'].version = '1.0'
           self['xep 0092'].os = 'Custom 05'
20
21
      def session start(self, event):
22
           self.send presence()
23
24
           self.get roster()
25
26
      def message(self, msg):
           if msg['type'] in ('chat', 'normal') and msg['body']:
27
28
               body=msg['body']
29
               print(body)
30
               msg.reply(f"Thanks for sending\n{body}").send()
31
32
      def iq(self, iq):
           if iq['type'] == 'get':
    if iq['query'].xmlns == 'jabber:iq:version':
33
34
                   iq.reply().send()
35
```

```
21
38 def start_bot(jid, password):
39
      loop = asyncio.new_event_loop()
40
      asyncio.set_event_loop(loop)
41
      xmpp = EchoBot(jid, password)
42
43
      xmpp.connect()
      loop.run_until_complete(xmpp.process(forever=False))
44
45
46 if _____ == '____main___':
47
      # Replace the following with Alice's and Bob's real JIDs and passwords
48
49
      alice_jid = "alice@networkedss-virtualbox'
      alice_password = "alice"
50
      bob_jid = "bob@networkedss-virtualbox"
51
      bob_password = "bob'
52
      logging.basicConfig(level=logging.DEBUG, format='%(levelname)-8s %(message)s')
53
54
      # Start Alice's bot in a separate thread
55
56
      alice_thread = threading.Thread(target=start_bot, args=(alice_jid, alice_password))
57
      alice_thread.start()
58
      # Start Bob's bot in a separate thread
59
60
      bob thread = threading.Thread(target=start bot, args=(bob jid, bob password))
61
      bob_thread.start()
62
      # Wait for both threads to complete
63
      alice thread.join()
64
65
      bob_thread.join()
```

Sender.py:

in order to send a message between the both account I used this command:

python sender.py -j alice@networkedss-virtualbox -p alice -t bob@networkedss-virtualbox -m "Hello"

wireshark:

прр			
Time	Source	Destination	Protocol Length Info
15 0.069138157	127.0.0.1	127.0.1.1	XMPP/X 207 STREAM > networkedss-virtualbox
17 0.070509998	127.0.1.1	127.0.0.1	XMPP/X 1055 STREAM < networkedss-virtualbox
19 0.071013755	127.0.0.1	127.0.1.1	XMPP/X 207 STREAM > networkedss-virtualbox
21 0.072032689	127.0.1.1	127.0.0.1	XMPP/X 1055 STREAM < networkedss-virtualbox
23 0.073156469	127.0.0.1	127.0.1.1	XMPP/X 184 AUTH
24 0.074212492	127.0.1.1	127.0.0.1	XMPP/X 258 CHALLENGE
25 0.076777420	127.0.0.1	127.0.1.1	XMPP/X 180 AUTH
26 0.104152442	127.0.1.1	127.0.0.1	XMPP/X 258 CHALLENGE
28 0.147829071	127.0.0.1	127.0.1.1	XMPP/X 252 RESPONSE
29 0.148749038	127.0.1.1	127.0.0.1	XMPP/X 166 SUCCESS
32 0.165550449	127.0.0.1	127.0.1.1	XMPP/X 207 STREAM > networkedss-virtualbox
33 0.166237678	127.0.1.1	127.0.0.1	XMPP/X 898 STREAM < networkedss-virtualbox
34 0.167283227	127.0.0.1	127.0.1.1	XMPP/X 173 IQ(set) BIND
35 0.167541943	127.0.1.1	127.0.0.1	XMPP/X 322 IQ(result) BIND

The network packet trace indicates an XMPP session where an initial STREAM is established between the client and the server, designated by local IP addresses. This is followed by a two-step authentication process: a CHALLENGE is issued by the server, to which the client responds with an AUTH message containing the credentials. The server replies with another CHALLENGE, and the

client sends a RESPONSE. The successful authentication is acknowledged with a SUCCESS message. Afterward, an IQ(set) BIND stanza is transmitted by the client to negotiate a unique resource for the session, and the server confirms this binding with an IQ(result) BIND, thus establishing a secure and identified XMPP session.

36 0.169109395	127.0.0.1	127.0.1.1	XMPP/X	130 PRESENCE
37 0.169284662	127.0.1.1	127.0.0.1	XMPP/X	273 PRESENCE < alice@networkedss-virtualbox/e704b9df-f84c-45b7-9d
38 0.170121997	127.0.0.1	127.0.1.1	XMPP/X	165 IQ(get) QUERY(jabber:iq:roster)
39 0.170243148	127.0.1.1	127.0.0.1	XMPP/X	246 IQ(result) QUERY(jabber:iq:roster)
40 0.177051678	127.0.0.1	127.0.1.1	XMPP/X	252 RESPONSE
41 0.177739372	127.0.1.1	127.0.0.1	XMPP/X	166 SUCCESS
43 0.179571561	127.0.0.1	127.0.1.1	XMPP/X	207 STREAM > networkedss-virtualbox
44 0.180145634	127.0.1.1	127.0.0.1	XMPP/X	898 STREAM < networkedss-virtualbox
45 0.186142039	127.0.0.1	127.0.1.1	XMPP/X	173 IQ(set) BIND
46 0.186643949	127.0.1.1	127.0.0.1	XMPP/X	320 IQ(result) BIND
47 0.188489181	127.0.0.1	127.0.1.1	XMPP/X	130 PRESENCE
48 0.189366743	127.0.1.1	127.0.0.1	XMPP/X	269 PRESENCE < bob@networkedss-virtualbox/06127af5-837d-4584-908b
49 0.190108583	127.0.0.1	127.0.1.1	XMPP/X	165 IQ(get) QUERY(jabber:iq:roster)
50 0.190255034	127.0.1.1	127.0.0.1	XMPP/X	241 IQ(result) QUERY(jabber:iq:roster)
60 2.825619419	127.0.0.1	127.0.1.1	XMPP/X	207 STREAM > networkedss-virtualbox
62 2.826803217	127.0.1.1	127.0.0.1	XMPP/X	1055 STREAM < networkedss-virtualbox
64 2.827520613	127.0.0.1	127.0.1.1	XMPP/X	184 AUTH
65 2.828509951	127.0.1.1	127.0.0.1	XMPP/X	262 CHALLENGE
66 2.848098803	127.0.0.1	127.0.1.1	XMPP/X	252 RESPONSE
67 2.848942263	127.0.1.1	127.0.0.1	XMPP/X	166 SUCCESS
68 2.849268529	127.0.0.1	127.0.1.1	XMPP/X	207 STREAM > networkedss-virtualbox
69 2.849837108	127.0.1.1	127.0.0.1	XMPP/X	898 STREAM < networkedss-virtualbox
70 2.850347779	127.0.0.1	127.0.1.1	XMPP/X	173 IQ(set) BIND
71 2.851044808	127.0.1.1	127.0.0.1	XMPP/X	322 IQ(result) BIND

Continuing from the initial XMPP session setup, the packet trace captures presence broadcasts and roster queries. After the STREAM and BIND processes are successfully completed, the client sends a PRESENCE stanza, signaling its online status to the server and other users. Subsequently, an IQ(get) stanza requesting the user's roster (jabber:iq:roster) is observed, to which the server responds with an IQ(result) containing the requested roster information. This exchange is essential for the client to retrieve its contact list. Finally, a similar sequence of authentication (AUTH, CHALLENGE, RESPONSE, SUCCESS) and resource binding (BIND) is repeated, likely for another user, as indicated by the presence stanza with a different full JID in the FROM field, suggesting the establishment of a separate, concurrent XMPP session for a second user on the same local network.

71 2.851044808	127.0.1.1	127.0.0.1	XMPP/X	322 IQ(result) BIND
72 2.852425956	127.0.0.1	127.0.1.1	XMPP/X	130 PRESENCE
73 2.852541921	127.0.1.1	127.0.0.1	XMPP/X	273 PRESENCE < alice@networkedss-virtualbox/e704b9df-f84c-45b7-9d
74 2.852632177	127.0.1.1	127.0.0.1	XMPP/X	273 PRESENCE < alice@networkedss-virtualbox/1cd3cd2d-a908-4c77-80
75 2.853124888	127.0.1.1	127.0.0.1	XMPP/X	273 PRESENCE < alice@networkedss-virtualbox/1cd3cd2d-a908-4c77-80
77 2.855583139	127.0.0.1	127.0.1.1	XMPP/X	165 IQ(get) QUERY(jabber:iq:roster)
78 2.855721909	127.0.1.1	127.0.0.1	XMPP/X	246 IQ(result) QUERY(jabber:iq:roster)
79 2.856589285	127.0.0.1	127.0.1.1	XMPP/X	199 MESSAGE > bob@networkedss-virtualbox
80 2.856928990	127.0.1.1	127.0.0.1	XMPP/X	272 MESSAGE < alice@networkedss-virtualbox/1cd3cd2d-a908-4c77-807
82 2.857754299	127.0.0.1	127.0.1.1	XMPP/X	257 MESSAGE > alice@networkedss-virtualbox/1cd3cd2d-a908-4c77-807
83 2.858018939	127.0.1.1	127.0.0.1	XMPP/X	328 MESSAGE < bob@networkedss-virtualbox/06127af5-837d-4584-908b
84 2.858263741	127.0.0.1	127.0.1.1	XMPP/X	82 STREAM END
85 2.858360007	127.0.1.1	127.0.0.1	XMPP/X	82 STREAM END
87 2.858870361	127.0.1.1	127.0.0.1	XMPP/X	240 PRESENCE < alice@networkedss-virtualbox/1cd3cd2d-a908-4c77-80
96 5.168126292	127.0.1.1	127.0.0.1	XMPP/X	234 IQ(get) QUERY(jabber:iq:version) < networkedss-virtualbox
98 5.169092203	127.0.0.1	127.0.1.1	XMPP/X	<pre>235 IQ(result) QUERY(jabber:iq:version) > networkedss-virtualbox</pre>
99 5.192927099	127.0.1.1	127.0.0.1	XMPP/X	<pre>232 IQ(get) QUERY(jabber:iq:version) < networkedss-virtualbox</pre>
100 5.195373993	127.0.0.1	127.0.1.1	XMPP/X	235 IQ(result) QUERY(jabber:iq:version) > networkedss-virtualbox

The follow-up in the XMPP packet trace shows further interaction post-authentication. Multiple PRESENCE stanzas indicate the user is announcing their availability to the network. Additionally, there are direct MESSAGE stanzas being exchanged between users, indicative of actual chat communication taking place. This part of the session demonstrates the users actively engaging in sending and receiving messages, which is the core function of XMPP.

Subsequently, the STREAM END stanzas suggest the closing of XMPP streams, signaling that the users or the server are terminating their sessions. In between these messages, IQ(get) queries for the jabber:iq:version are noted, which are requests for information about the XMPP client version being used; the server responds with IQ(result) providing the requested information. This exchange is

typically used for compatibility and diagnostic purposes. Overall, this segment of the trace captures the active usage phase of an XMPP session, including messaging and session termination.