```
In [1]: from code.DQN.train import DQN_WaterNetwork,DEFAULT_ACTION_ZONE ,REWARD_FUNC
        from pathlib import Path
        from code.DQN.config import BASE_MODEL_PATH
        from IPython.display import Image
```

Validation

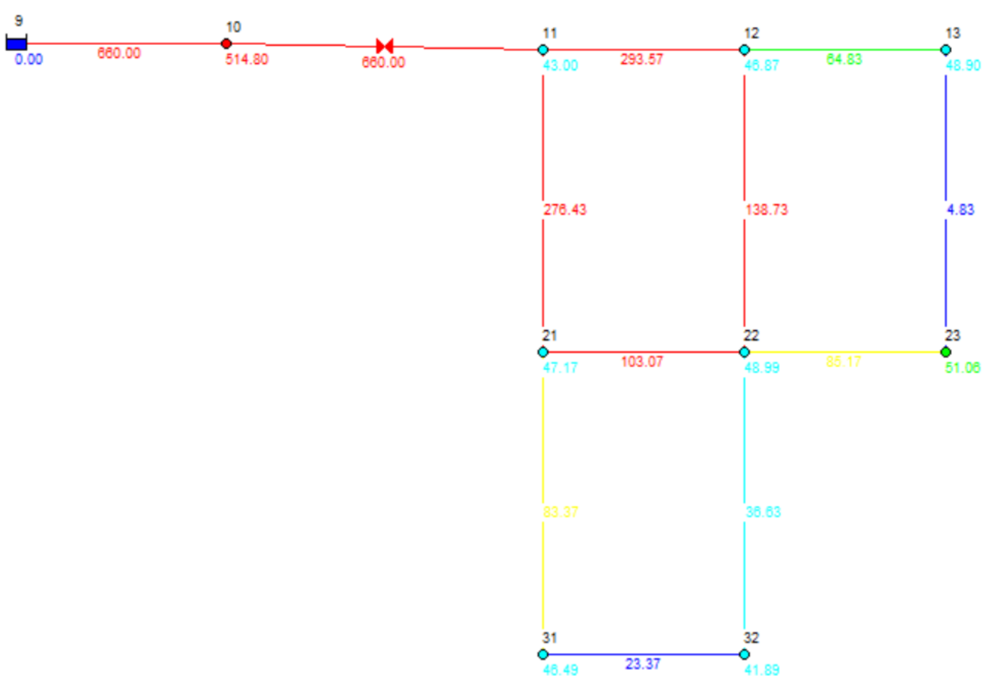
In this section, we have trained a model on an uncertain environment with a high rate of random node-broken. To validate the results we have provided some EPANET networks which we know what the best valve setting to control the network. Now we want to show that the model works properly after fewer iterations can be trained and choose the best actions.

NET 43

The best valve setting in this network is 43, but our action is one of (10, 20,30,40,50 ,..., 240). The model should find a closet setting.

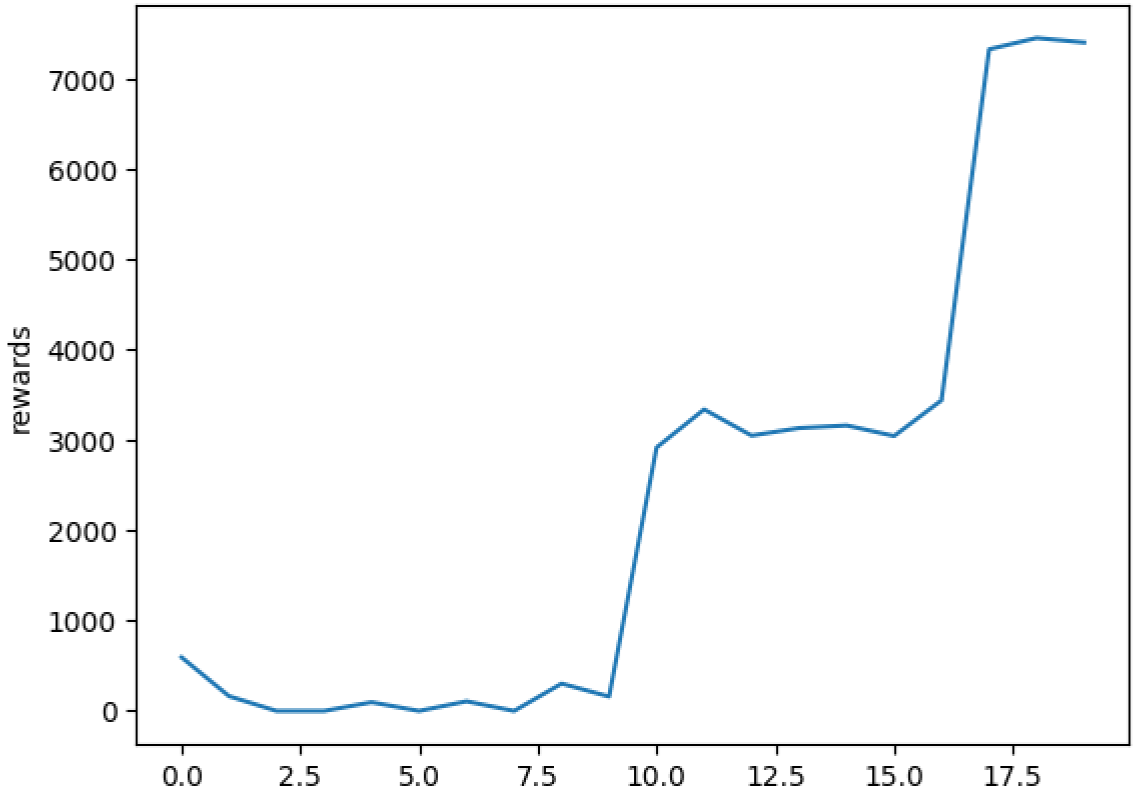
```
In [17]: Image(filename='./images/net_40.PNG')
```

Out[17]:




```
In [ ]: network_name="simple_nets/net_43"
        action_zone =DEFAULT_ACTION_ZONE
        dqn_water = DQN_WaterNetwork(network_name,action_zone,iterations=20,epsilon_
        dqn_water.load_model(BASE_MODEL_PATH.joinpath("simple_net_randomness"))
        dqn_water.train()
```

In [6]:
```
chosen_actions = [dqn_water.env.actions_index[x]  for i , x in dqn_water.ste
dqn_water.plot_rewards(show=True)
print("Last chosen action after train is ",[x * dqn_water.env.PSI_UNIT for x
```



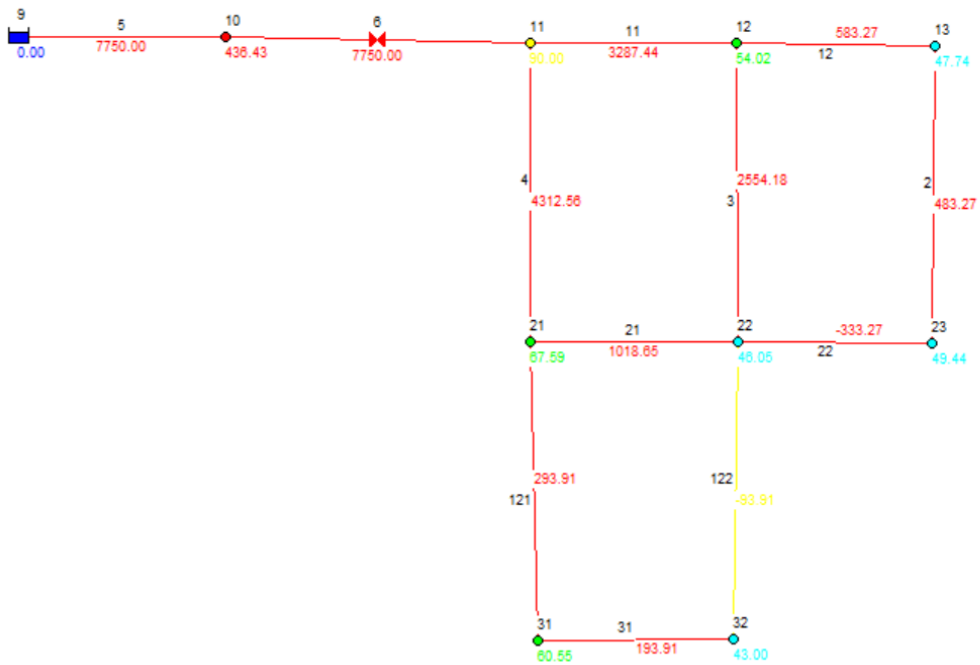

```
Last chosen action after train is  [43.511321402166]
```

As we see, we could train model to find the best valve setting which is 43

NET 90

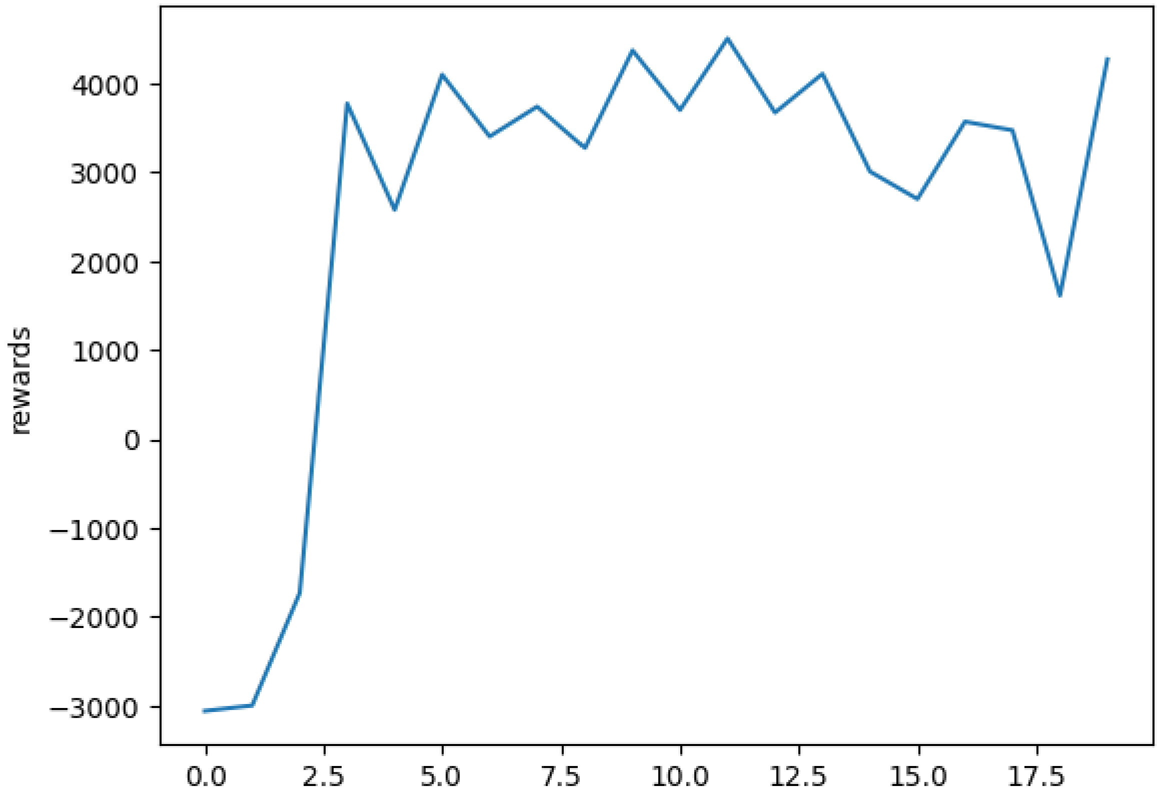
In [18]:
```
Image(filename='./images/net_90.PNG')
```

Out[18]:



In [ ]:
```
network_name="simple_nets/net_90"
action_zone =DEFAULT_ACTION_ZONE
dqn_water = DQN_WaterNetwork(network_name,action_zone,iterations=20,epsilon_
dqn_water.load_model(BASE_MODEL_PATH.joinpath("simple_net_randomness"))
dqn_water.train()
```

In [16]:
```
chosen_actions = [dqn_water.env.actions_index[x]  for i , x in dqn_water.ste
dqn_water.plot_rewards(show=True)
print("Last chosen action after train is ",[x * dqn_water.env.PSI_UNIT for x
```

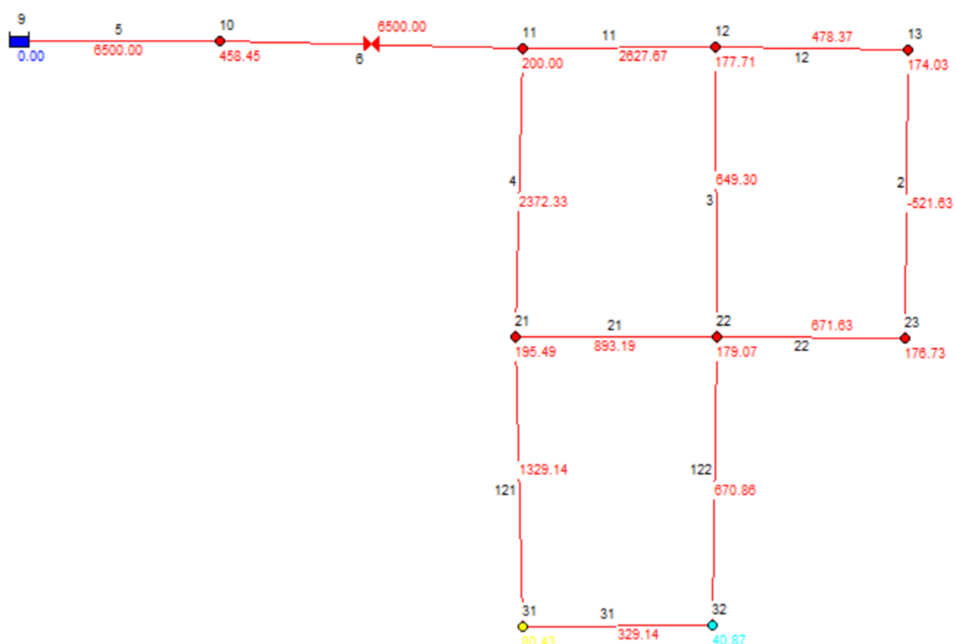



```
Last chosen action after train is  [87.022642804332]
```

NET 200

In [19]:
```python
Image(filename='./images/net_200.PNG')
```

Out[19]:

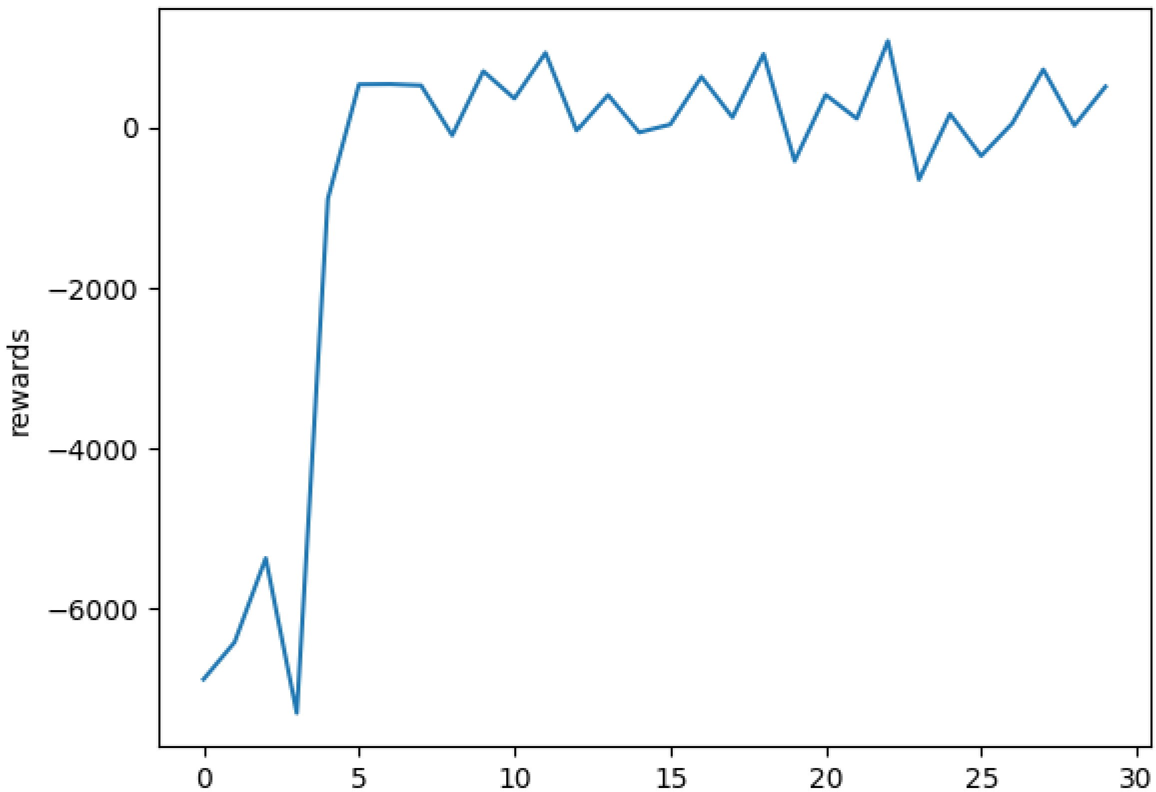


In this network we have high pressure at most nodes instead all nodes have minimum pressure. We have provided a reward function that tries to handle decreasing average pressure among all nodes and preparing minium pressure for all nodes which is 40.

So the best answer in this network is 200 which is found manually with EPANET tool.

In [ ]:
```
network_name="simple_nets/net_200"
action_zone =DEFAULT_ACTION_ZONE
dqn_water = DQN_WaterNetwork(network_name,action_zone,iterations=30,epsilon_
dqn_water.load_model(BASE_MODEL_PATH.joinpath("simple_net_randomness"))
dqn_water.train()
```

In [12]:
```
chosen_actions = [dqn_water.env.actions_index[x]  for i , x in dqn_water.ste
dqn_water.plot_rewards(show=True)
print("Last chosen action after train is ",[x * dqn_water.env.PSI_UNIT for x
```



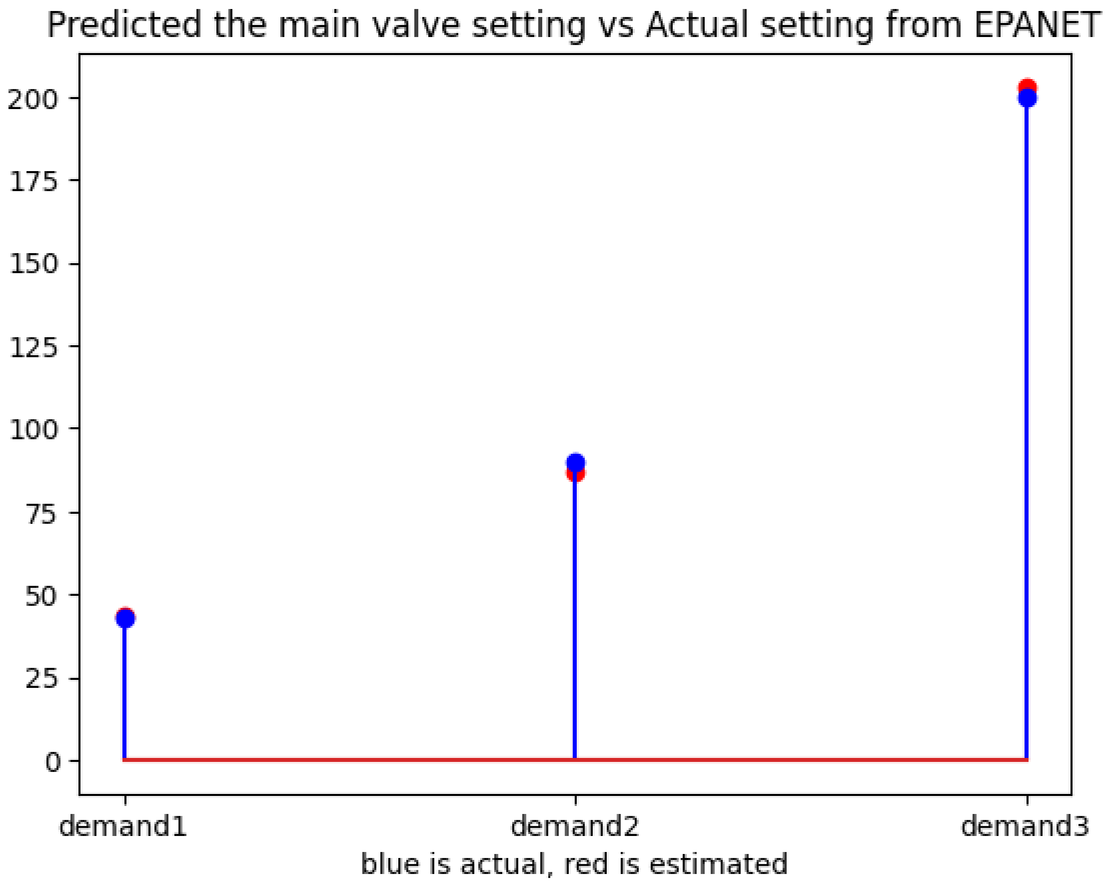

```
Last chosen action after train is  [203.052833210108]
```

In [ ]:

In [ ]:

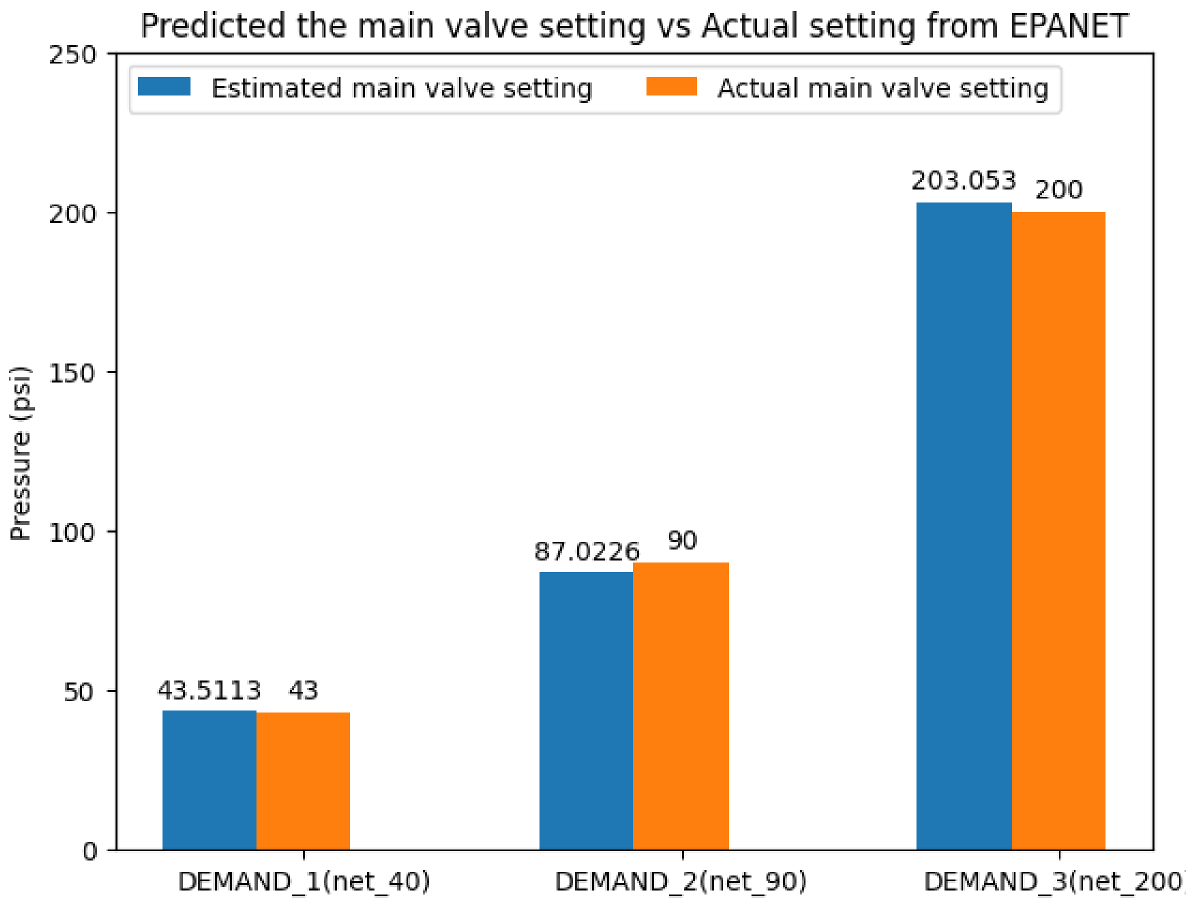
In [36]:
```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0.1, 2 * np.pi, 41)
origin_y = [1,2,3]
x = ['demand1','demand2','demand3']
plt.stem(x, [43.511321402166,87.022642804332,203.052833210108], linefmt='rec
plt.stem(x, [43,90,200],linefmt='blue')
plt.title("Predicted the main valve setting vs Actual setting from EPANET")
```

```
plt.xlabel("blue is actual, red is estimated")
plt.show()
```



In [43]:
```
import matplotlib.pyplot as plt
import numpy as np

species = ("DEMAND_1(net_40)", "DEMAND_2(net_90)", "DEMAND_3(net_200)")
penguin_means = {
    'Estimated main valve setting': [43.511321402166,87.022642804332,203.052
    'Actual main valve setting': [43,90,200],

}

x = np.arange(len(species))  # the label locations
width = 0.25  # the width of the bars
multiplier = 0

fig, ax = plt.subplots(layout='constrained')

for attribute, measurement in penguin_means.items():
    offset = width * multiplier
    rects = ax.bar(x + offset, measurement, width, label=attribute)
    ax.bar_label(rects, padding=3)
    multiplier += 1

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Pressure (psi)')
ax.set_title('Predicted the main valve setting vs Actual setting from EPANET
```

```
ax.set_xticks(x + width, species)
ax.legend(loc='upper left', ncols=3)
ax.set_ylim(0, 250)

plt.show()
```



Additonal information about the demand patterns

Demand_40

In [16]:
```
Image(filename='./images/net_40_juncs.png')
```

Out[16]:

```
[JUNCTIONS]
;ID              	Elev        	Demand      	Pattern         
 10              	710         	0           	1               	;
 11              	710         	150         	1               	;
 12              	700         	150         	1               	;
 13              	695         	100         	1               	;
 21              	700         	150         	1               	;
 22              	695         	200         	1               	;
 23              	690         	150         	1               	;
 31              	700         	100         	1               	;
 32              	710         	100         	1               	;
```

In [11]:

```python
Image(filename='./images/demand_40.png')
```

Out[11]:

```
[PATTERNS]
;ID              	Multipliers
;Demand Pattern
 1               	1.0         	1.2         	2           	1.6         	1.4         	1.2
 1               	1.0         	0.8         	0.6         	0.4         	0.6         	0.8
```

Demand_90

In [12]:

```python
Image(filename='./images/net_90_juncs.png')
```

Out[12]:

```
[JUNCTIONS]
;ID              	Elev        	Demand      	Pattern         
 10              	710         	0           	                	;
 11              	710         	150         	                	;
 12              	700         	150         	                	;
 13              	695         	100         	                	;
 21              	700         	150         	3               	;
 22              	695         	200         	3               	;
 23              	690         	150         	                	;
 31              	700         	100         	                	;
 32              	710         	100         	                	;
```

In [13]:

```python
Image(filename='./images/demand_90.png')
```

Out[13]:

```
[PATTERNS]
;ID                 Multipliers
;Broken pattern
 2                  10               10               10               10               10               10
 2                  10               10               10               10               10               10
 2                  10               10               10               10               10               10
 2                  10               10               10               10               10               10
;
 3                  20               20               20               20               20               20
 3                  20               20               20               20               20               20
 3                  20               20               20               20               20               20
 3                  20               20               20               20               20               20
```

Demand_200

In [14]:

```python
Image(filename='./images/net_200_juncs.png')
```

Out[14]:

```
[JUNCTIONS]
;ID                 Elev            Demand          Pattern
 10                 710             0                                   ;
 11                 710             150             2                   ;
 12                 700             150             2                   ;
 13                 695             100             2                   ;
 21                 700             150             1                   ;
 22                 695             200             1                   ;
 23                 690             150             1                   ;
 31                 700             100             2                   ;
 32                 710             100             2                   ;
```

In [15]:

```python
Image(filename='./images/demand_200.png')
```

Out[15]:

```
[PATTERNS]
;ID                 Multipliers
;Broken pattern
 2                  10               10               10               10               10               10
 2                  10               10               10               10               10               10
 2                  10               10               10               10               10               10
 2                  10               10               10               10               10               10
```