NTP is abbreviation for Network Time Protocol which is used for clock synchronization of various devices on the net. There are three typical implementations of NTP in network infrastructure: a) no implementation b) useful implementation c) vital implementation.

The first option is an indicator that network is in poor condition. It has no real owner or owner isn't a network professional, etc.

The second option is the most common case for serious corporate network. The owner cares about event logging (at least) and event correlation in different parts of network. Complex debugging, security incident investigation requires the "right time" to be set. But still network functioning in general or service availability isn't tied with NTP.

And we have the third option. Vital dependency is when your network can't function without reliable NTP infrastructure. If your devices have wrong time that means no service for end-user. That's bad, isn't it? :)

I can name at least four technologies which comes to mind when we talk about NTP vital dependency:

1. IPSec/IKE protocol with RSA-sig authentication. RSA-sig auth uses PKI and digital certificates which have "valid from/to" fields. If time is wrong then certificate is wrong then no IPSec tunnel then no service for end-user. Right?
2. Moreover. I suppose any technology which uses certificates have to have the right time. The 802.1x standard (PEAP, EAP-TLS sounds familiar?) which is widely used for WI-FI or wired user authentication is also depends on NTP.
3. It was a surprise for me but when our NS-team was engaged in Layer 2 Security implementation(Dynamic ARP Inspection, DHCP snooping, IP source Guard) for some customer we've spend several hours debugging Cat3750 because we had missed one small note from Config Guide . It says "...the switch writes binding changes to the binding file only when the switch system clock is synchronized with NTP". Period.
4. And of course we have to use NTP if we make time-based ACLs with references to the absolute time. Although I've never used it

Well, I suppose now is the right time for configs :). By the way, if you are not aware yet - I like to do things in a secure way :))

IOS config: NTP server (broadcast and unicast)

```
interface FastEthernet0/0
 description Broadcast distribution
 ip address 192.168.100.1 255.255.255.0
 ntp broadcast key 1
interface Serial0/0
 ip address 192.168.13.1 255.255.255.252
!
access-list 2 permit 127.127.7.1  !Router itself (Not sure)
access-list 2 permit 192.168.100.10  !Broadcast client
access-list 2 permit 192.168.13.10  !Unicast client
!
ntp authentication-key 1 md5 cisco
ntp authenticate
ntp trusted-key 1
ntp access-group serve-only 2
ntp master 2
```

IOS config: NTP unicast client

```
interface Serial0/0
 ip address 192.168.13.10 255.255.255.0
!
access-list 2 permit 192.168.13.1
access-list 2 deny any log
!
ntp authentication-key 1 md5 cisco
ntp authenticate
ntp trusted-key 1
ntp access-group peer 2
ntp server 192.168.13.1 key 1
```

IOS config: NTP broadcast client

```
interface FastEthernet0/0
 ip address 192.168.100.10 255.255.255.0
 ntp broadcast client
!
access-list 2 permit 192.168.100.1
access-list 2 deny any log
!
ntp authentication-key 1 md5 cisco
ntp authenticate
```

```
ntp trusted-key 1
ntp access-group peer 2
```

Links:
Best Practices White Paper
NTP command reference